MATEMATIKA

Experimental Design and Data Analysis Using R

Akhmad Dakhlan

R Experimental Design and Data Analysis Using R

R is now widely acknowledged as a scientific skill and increasingly more applied in many scientific area because of its powerful and flexible and also can be freely downloaded and installed in many platforms (Windows, MacOSX and Linux). This open source licence along with a relatively simple scripting syntax has promoted diverse and rapid evolution and contribution, make the popularity of R as a teaching and research tool continues to accelerate.

This book discusses theory and application of experimental design, especially in animal and agricultural science, and shows how R can be friendly used as a tool in data analysis although the use of excel or hand calculation is not ruled out. I believe that this book is useful to student and researcher in learning process, to help them in applying appropriate experimental designs and statistical methods using software R.

Akhmad Dakhlan was born on August 10, 1969 in Sumenep, East Java, Indonesia. He finished elementary to high school in Sumenep and started bachelor degree in Animal Science in Universitas Mataram in 1986 and finished in 1990. He finished his master degree in Animal Breeding and Genetics in Universitas Gadjah Mada Yogyakarta in 1994 and since 1995 he started his career as a lecturer in Universitas Lampung until present. His doctoral degree in Animal Breeding and Genetics in School of Environmental and Rural Science, The University of New England, Armidale, NSW Australia was done from 2013 to 2017. During his candidature he was introduced with R environment and other software training using big data. His passion in statistics and experimental design emerged since his bachelor degree study. This is the first book he began writing.

ISBN: 978-602-5966-18-7

MATEMATIKA

HALAMAN PENGESAHAN

1. Judul Buku : Experimental Design and Data Analysis Using R
2. Penulis : Ir. Akhmad Dakhlan, M.P., Ph.D.
3. Instansi : Jurusan Peternakan Fakultas Pertanian Universitas Lampung
4. Jenis Publikasi : Buku ber-ISBN
5. Nomor ISBN : 978-602-5966-18-7
6. Penerbit : Graha Ilmu
7. Tahun Terbit : Juli 2019
8. Alamat Penerbit : Ruko Jambusari No. 7A Yogyakarta 55283
   Telp.+62-274-889398

Bandar Lampung, 4 Maret 2020
Mengetahui
Penulis,

Ir. Akhmad Dakhlan, M.P., Ph.D.
NIP. 19690810 199512 1 001

Menyetujui,

Dekan Fakultas Pertanian
Universitas Lampung

Prof. Dr. Ir. Irwan Sukri Banuwa, M.Si.
NIP. 19611020 198603 1 002

Ketua LP3M Universitas Lampung

Prof. Dr. Ahmad Saudi Samosir, S.T., M.T.
NIP. 19710415 199803 1 005

PREFACE

First of all, author wishes to thank to God, The Almighty, with the finished of the first draft of this book. The intension of writing this book is to serve students and researchers, especially in animal and agricultural sciences, to help them in applying appropriate experimental designs and statistical methods using software R.

The first part of this book presents the very basic of R introduction and basic principles of experimental design in order the readers be able to follow subsequent applications. In every chapter the readers will be introduced with a brief theoretical background, and then enriched with examples, mostly from animal and agricultural sciences which can be solved using excel or calculator and then followed by R example solution so that the readers can compare the results using calculation technique and software R.

The first chapter of this book tries to introduce the readers how to get started using R, including the website where to get free software and install R. The second chapter provides readers with terminology in experimental design followed by the next chapter discussing the simplest experimental design: Completely Randomized Design (CRD). Chapter 4 describes multiple comparison, including LSD, Tukey, Duncan, SNK, Dunnett, Scheffe, Boferroni, and orthogonal comparison and contrast. Assumption for ANOVA and data transformation is discussed in chapter 5 and 6.

Chapters 7 to 14 focus on specific experimental designs and their analyses, including randomized complete block design, Latin square design, crossover designs, factorials, nested designs, split plots and strip plot design, analysis of covariance, and repeated measures design. Examples with sample R script are provided for each topic. The chapter 15 covers the special topic of analysis of numerical treatment levels including orthogonal polynomial contrast. The final chapter discusses linear and nonlinear regression with common nonlinear model used in agriculture.

Author would like to express many gratitude to everyone who helped author produce this book. Author extends a special acknowledgement to Professor Bambang Setiyadi, Ph.D. for his assistance with editing to publish this book.

Bandar Lampung, March 2019

Akhmad Dakhlan

TABLE OF CONTENT

I. Getting Started with R

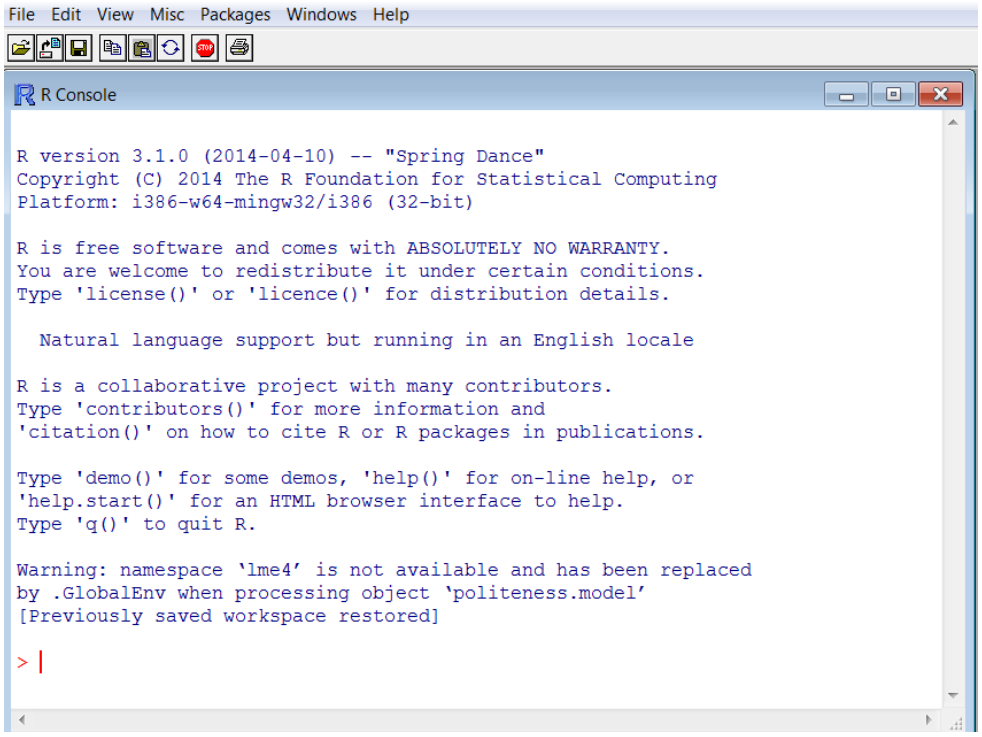
1.1 Introduction

R is an elegant and comprehensive statistical and graphic programming language. Why do many people switch to using R? R is free software, it can be run on various platforms such as Windows, Unix and MacOS, the program is regularly updated, and it has artistic graphic capacity.

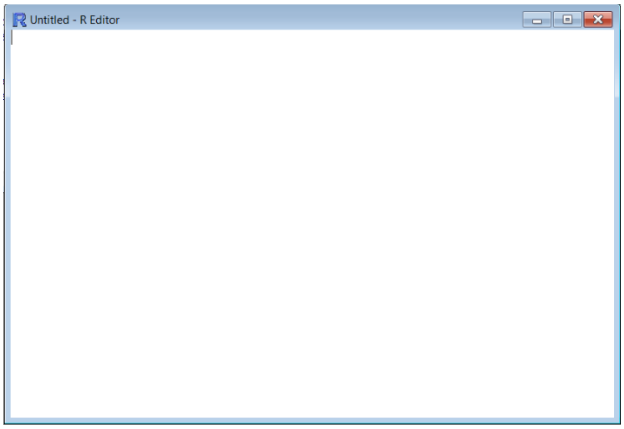
R software can be downloaded for free and installed easily through one of the closest sites on the **Comprehensive R Archive Network (CRAN) mirror**, for example, https://cran.r-project.org/ or in Indonesia: https://repo.bppt.go.id/cran/ (BPPT). When R is installed, there is a help system to ask various things in R. For example:

```
>help.start()      #common help
>help(t.test)     #help on t.test, or
>??t.test     #the same thing with help(t.test)
>help(anova)    #help on anova
```

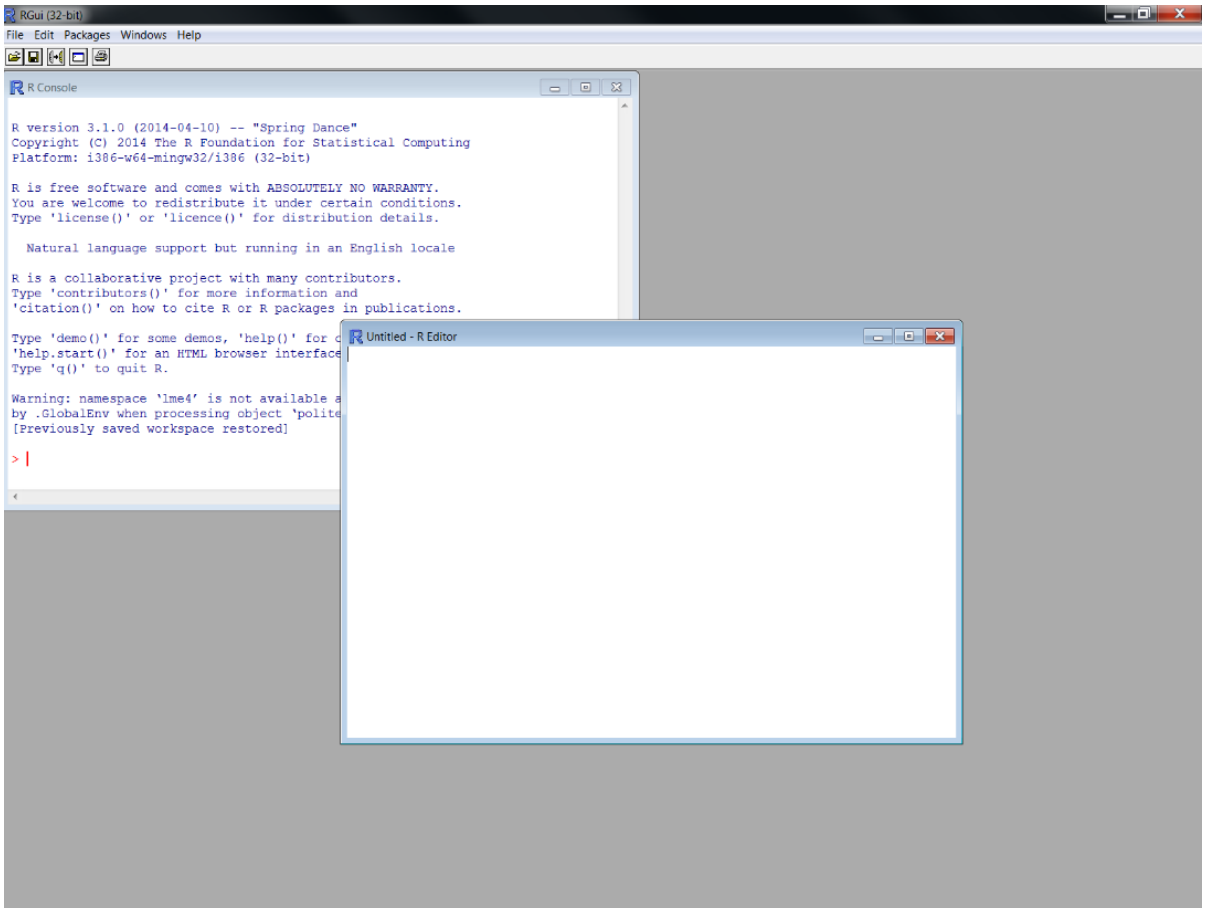
To start R, please double-click the R symbol on your computer desktop, then the R Console will appear, which is where we start working, as shown below.



On the file menu, click then select Change directory, so that we can confirm in the folder where we will work and save the data in. Then click the file menu again and select New script, then the R editor will appear blank as below.



Thus, the console and R editor will appear on your computer monitor like this:



This R editor is a place where we write R scripts that we can save and can reopen when we need them later. Actually we can directly write the R script on the console and press enter to execute it, but we should write R scripts in the R editor so we can manage and save the R file we want, so we can reopen the R file when we need it. To execute the script that we wrote in the R editor, click control r (Ctrl r) simultaneously on each line of the script we write.

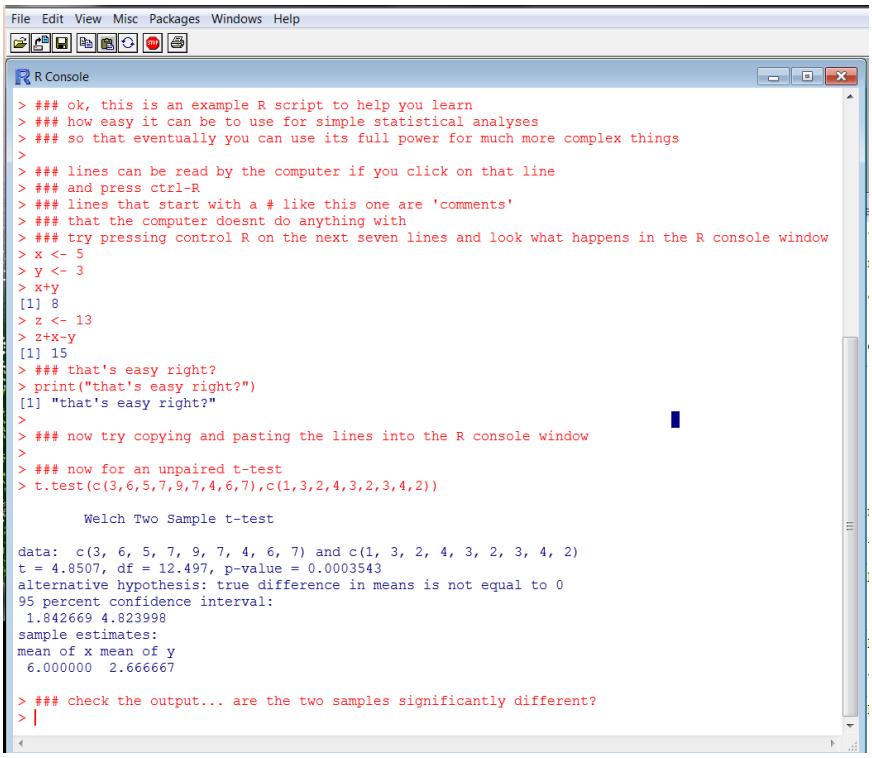
1.2 Getting Started Using R

To begin with, now try writing the script below in your R editor. Then click control r (Ctrl r) on every line.

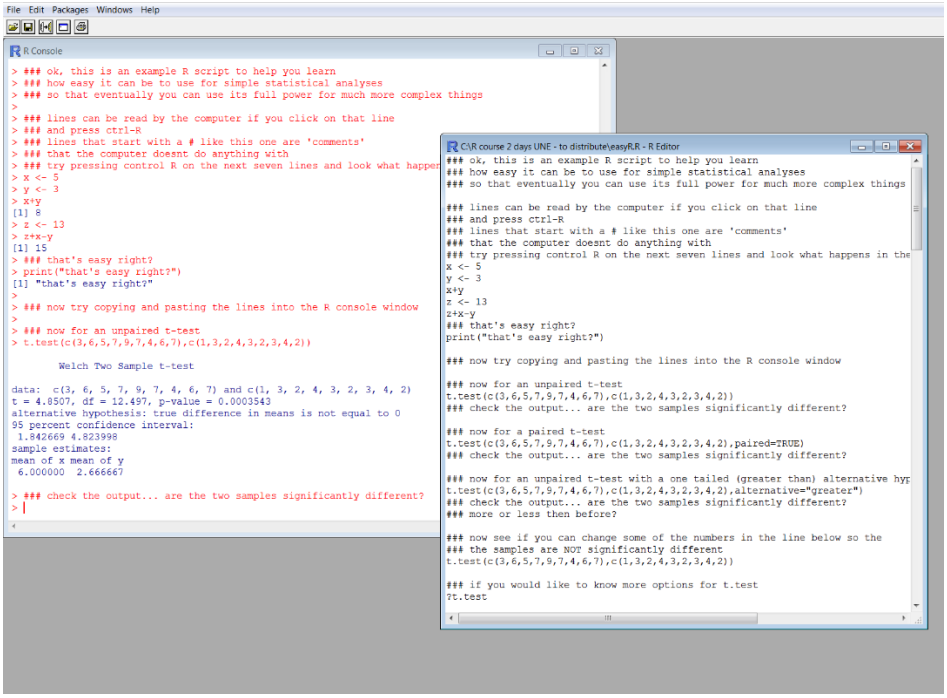
```
### This is an example of R scripts to help you
   learn
### how easy it can be to use for simple statistical
   analyses
### so that finally we can use its full power for
   much more complex things

### lines can be read by the computer if we click on
   that line
### and press ctrl-R
### lines that start with a # like this one are
   'comments'
### that the computer doesn't do anything with
### Try pressing control R on the next seven lines
   and look what happens in the R console window
x <- 6
y <- 2
x+y
z <- 15
z+x-y
### that's easy right?
print("that's easy right?")

### now try pressing lines below to compare mean of
   two data of different population using t.test

### now for an unpaired t-test
t.test(c(3,6,5,7,9,7,4,6,7),c(1,3,2,4,3,2,3,4,2))
### check the output... are the two samples
   significantly different?

### now for a paired t-test
```

```
t.test(c(3,6,5,7,9,7,4,6,7),c(1,3,2,4,3,2,3,4,2),pai
   red=TRUE)
### check the output... are the two samples
   significantly different?

### now for an unpaired t-test with a one tailed
   (greater than) alternative hypothesis
t.test(c(3,6,5,7,9,7,4,6,7),c(1,3,2,4,3,2,3,4,2),alt
   ernative="greater")
### check the output... are the two samples
   significantly different?
### more or less then before?

### if you would like to know more options for
   t.test
?t.test
### don't forget to connect to internet to do that
```

The results will appear in the console below.



Or in full on your monitor it will appear as below.



Now close your R editor by clicking on the cross (X) in the upper right corner, the request will be saved or not. Alternative way to save editor file is by clicking File menu and choose Save Workspace or by clicking the icon of Save Worspace in the upper left side. Save your R editor in the folder or directory you are using now with your file name. Suppose you are working on the R_Project folder, and the file for your script with the name Coba, then when opening the R_Project folder there will be a file with the name Coba.R.

II. Terminology in Experimental Design

2.1 Introduction

Experiments are done based on our questions that we want to find the answers. For example, is growth of broiler affected by the addition of prebiotic in its ration, and how much prebiotic the best to broiler growth? For these questions we need to design an experiment carefully to answer the questions. In this case, we need some DOC (day old chick) of broiler which are homogeneous, we need different level of prebiotic in rations, and we need cages to place groups of birds. So the component of this experiment consisted of measurement unit (DOC), experimental units (cage), factor (addition of prebiotic that influence the broiler growth), treatment (different level of prebiotic), replication (some cages with the same level of prebiotic), responses or outcome (the growth of broiler), randomization (we do not chose certain DOC to be placed in a cage or in other words, we just place DOC randomly in a cage), control or standard/baseline treatment (no addition of prebiotic, base ration), controlled (other environment factor that influence on broiler growth is controlled, only the effect of prebiotic that we want to know and investigate on broiler growth), and experimental error (the same experimental unit with the same treatment give different outcome).

Other example, we want to know the effect of fertilizer addition on rice production. In this case we need some plots of land to plant the rice. Experimental unit in this case is different field plots, the measurement units might be a subset of the rice plants on the field plot, fertilizer is factor, the treatment is level of fertilizer. replication (some plots with the same level of fertilizer), responses or outcome (rice production), randomization (we do not chose certain rice plant to be placed in a plot or in other words, we just place rice plants randomly in a plot), control or standard/baseline treatment (no addition of fertilizer), controlled (other environment factor that influence on rice production is controlled, only the effect of fertilizer that we want to know and investigate on rice production).

2.2 Terminology

Based on the example above, experimental unit is the material of experiment which can be applied or assigned, at random, to a treatment. Potential examples of experimental units might be plots of land, individual animals, and populations. A

treatment is methods or various ways which are applied to experimental units. Experimental units which receive the same treatment is called a treatment group. Experimental units which is applied without treatment is called control group or standard treatment. A factor is combination of treatments and controls, and the different treatments/controls are called the levels of the factor.

There are three basic principle in experimental design including replication which means the experiment has to be carried out on several units in order to measure the sampling error. The second principle is randomization where the units have to be assigned randomly to treatments. Furthermore, the treatments should be comparable, the units should be similar in structure meaning that animals are of almost in the same age and live in a similar environment. The third principle is local control, blocking that stratifies the units into groups with similar (homogeneous) characteristics such as age, sex, and other factors affecting the outcome or responses is called local control.

The following script is an example to make randomization in R.

```
> data <- data.frame(label=letters[1:8],number=11:18)
> data
> data <- data[sample(1:nrow(data)), ]
> data
```

III. COMPLETELY RANDOMIZED DESIGN (CRD)

3.1 Balanced CRD

The simplest experimental design is a completely randomized design or just called CRD. CRD is appropriate if the experimental unit and the environments of the experiment are homogeneous, and there is only one factor with levels under the study.

For example, a research is conducted to investigate the effect of prebiotic addition in ration on broiler performance (body weight gain). Factor in this research is prebiotic addition with four treatments applied to broiler chicken, those are base ration (T1), T1 plus 0.2% prebiotic addition (T2), T1 plus 0.4% prebiotic addition (T3), and T1 plus 0.6% prebiotic addition (T4). The treatments are replicated four times. Hypothesis for this design is $H_0 : \mu1 = \mu2 = \mu3 = \mu4$ or $H_0 : \tau1 = \tau2 = \tau3 = \tau4$, while the alternative hypothesis is H_1: at least one of the means are different from the others.

First thing to do is doing randomization. In this case, there are 4x5 = 20 experimental units. Give numbers 1 to 20 to a group of chickens that will be used as experimental units, and then randomize the layout of the experiment as below.

```
> randomize <- data.frame(label=rep(c(letters[1:4]),
  each=5), number=1:20)
> randomize
   label number
1      a      1
2      a      2
3      a      3
4      a      4
5      a      5
6      b      6
7      b      7
8      b      8
9      b      9
10     b     10
11     c     11
12     c     12
13     c     13
14     c     14
15     c     15
16     d     16
17     d     17
18     d     18
19     d     19
```

```
20     d     20
> randomize <- randomize[sample(1:nrow(randomize)), ]


> randomize
   label number
19     d     19
4      a      4
10     b     10
13     c     13
14     c     14
1      a      1
12     c     12
20     d     20
9      b      9
17     d     17
7      b      7
15     c     15
16     d     16
11     c     11
18     d     18
3      a      3
8      b      8
6      b      6
2      a      2
5      a      5
>
```

So the first experimental unit is filled by treatment d, the second experimental unit is filled by treatment a, and soon until twenty experimental units. Linear model for this CRD is

$$Yij = \mu + \tau i + \varepsilon ij. \qquad i= 1, ..., t;\ j = 1, ..., n$$

where:

yij = observation *j* in treatment *i*
μ = the overall mean
τi = the fixed effect of treatment *i*
εij = random error

If the number replication is the same, the sample means of the data in the ith level of the treatment factor can be formulated with

$$\bar{y}\text{i.} = \frac{1}{ri}\sum_{j=1}^{ri} yij$$

The grand mean can be formulated with

$$\bar{y}.. = \frac{1}{t}\sum_{i=1}^{t} \bar{y}i. = \frac{1}{n}\sum_{i=1}^{t}\sum_{j=1}^{ri} yij\,.$$

where n = $\sum ri$.

Total variance in CRD is variance of treatment and variance of residual and can be written as follow.

$$(\text{yij} - \bar{\text{y}}..) = (\text{yi}. - \bar{\text{y}}..) + (\text{yij} - \bar{\text{y}}\text{i}.)$$

Sum squares of the above equation can be formulated as below.

$$\text{SST} = \text{SSt} + \text{SSE}$$

where SST is sum square total, SSt is sum square treatment and SSE is sum square error.

$$\text{SST} = \sum_{i=1}^{t}\sum_{j=1}^{ri}(yij - \bar{y}..)^2$$

$$\text{SSt} = \sum_{i=1}^{t}\sum_{j=1}^{ri}(\bar{y}i. - \bar{y}..)^2$$

$$\text{SSE} = \sum_{i=1}^{t}\sum_{j=1}^{ri}(yij - yi.)^2\,.$$

Degree of freedom of total (dfT) of SST = N – 1 = tr – 1 = 20 -1 = 19

Degree of freedom of treatment (dft) of SSt = t – 1 = 4 – 1 = 3

Degree of freedom of error (dfe) of SSE =t(r – 1) = N – t = 20 – 4 = 16

Mean square treatment (MSt) = SSt/dft

Mean square error (MSE) = SSE/dfe

Finally, theoretical table of ANOVA (analysis of variance) can be describe as follows.

Source of Variation	Degree of freedom	Sum square (SS)	Mean square (MS)	Fstatistic
Treatment	t – 1	SSt	SSt/(t – 1)	MSt/MSE
Error	N – t	SSE	SSE/(N – t)	
Total	N – 1	SST		

Significance of the test is by comparing Fsatistic from Ftable, where Ftable in R can be scripted as qf(p, dft, dfe) or qf(0.95, 3, 16) if the alpha = 0.05.

```
> qf(0.95, 3, 16)
[1] 3.238872
>
```

Data of body weight gain of broiler of 4 weeks of age treated with four different ration (T1, T2, T3, and T4) are presented in table below.

Replication	Treatments			
	T1	T2	T3	T4
1	0.7651	1.3113	1.452	1.6298
2	1.0150	1.3034	1.8463	1.5055
3	1.2759	1.5975	1.2639	1.7790
4	0.9837	0.6453	1.3987	1.4540
5	0.8557	1.1484	1.1541	1.4434

Solution 1 : using excel

Treatment	BodyWeightGain (BWG)	GroupAverage
T1	0.7651	
T1	1.015	
T1	1.2759	
T1	0.9837	
T1	0.8557	0.9791
T2	1.3113	
T2	1.3034	
T2	1.5975	
T2	0.6453	
T2	1.1484	1.2012
T3	1.452	
T3	1.8463	
T3	1.2639	
T3	1.3987	
T3	1.1541	1.4230
T4	1.6298	
T4	1.5055	
T4	1.779	
T4	1.454	
T4	1.4434	1.5623
GrandAverage	1.2914	

Treat	Xij	BWG	Xi average	(Xij–Xi av)	(Xij–Xi av)^2	X average	(Xi av–X av)	(Xi av–X av)^2	(Xij – X av)	(Xij – X av)^2
T1	X11	0.7651	0.9791	-0.214	0.045796	1.2914	-0.3123	0.09753129	-0.5263	0.27699
T1	X12	1.015	0.9791	0.0359	0.00128881	1.2914	-0.3123	0.09753129	-0.2764	0.0764
T1	X13	1.2759	0.9791	0.2968	0.08809024	1.2914	-0.3123	0.09753129	-0.0155	0.00024
T1	X14	0.9837	0.9791	0.0046	2.116E-05	1.2914	-0.3123	0.09753129	-0.3077	0.09468
T1	X21	0.8557	0.9791	-0.1234	0.01522756	1.2914	-0.3123	0.09753129	-0.4357	0.18983
T2	X22	1.3113	1.2012	0.1101	0.01212201	1.2914	-0.0902	0.00813604	0.0199	0.0004
T2	X23	1.3034	1.2012	0.1022	0.01044484	1.2914	-0.0902	0.00813604	0.012	0.00014
T2	X24	1.5975	1.2012	0.3963	0.15705369	1.2914	-0.0902	0.00813604	0.3061	0.0937
T2	X31	0.6453	1.2012	-0.5559	0.30902481	1.2914	-0.0902	0.00813604	-0.6461	0.41745
T2	X32	1.1484	1.2012	-0.0528	0.00278784	1.2914	-0.0902	0.00813604	-0.143	0.02045
T3	X33	1.452	1.423	0.029	0.000841	1.2914	0.1316	0.01731856	0.1606	0.02579
T3	X34	1.8463	1.423	0.4233	0.17918289	1.2914	0.1316	0.01731856	0.5549	0.30791
T3	X35	1.2639	1.423	-0.1591	0.02531281	1.2914	0.1316	0.01731856	-0.0275	0.00076
T3	X36	1.3987	1.423	-0.0243	0.00059049	1.2914	0.1316	0.01731856	0.1073	0.01151
T3	X37	1.1541	1.423	-0.2689	0.07230721	1.2914	0.1316	0.01731856	-0.1373	0.01885
T4	X41	1.6298	1.5623	0.0675	0.00455625	1.2914	0.2709	0.07338681	0.3384	0.11451
T4	X42	1.5055	1.5623	-0.0568	0.00322624	1.2914	0.2709	0.07338681	0.2141	0.04584
T4	X43	1.779	1.5623	0.2167	0.04695889	1.2914	0.2709	0.07338681	0.4876	0.23775
T4	X44	1.454	1.5623	-0.1083	0.01172889	1.2914	0.2709	0.07338681	0.1626	0.02644
T4	X45	1.4434	1.5623	-0.1189	0.01413721	1.2914	0.2709	0.07338681	0.152	0.0231
Sum					**SSE=1.0006**		**0**	**SSt=0.9819**	**0**	**SST=1.9827**

Solution 2 : manually, but using R

```
> data=read.csv("crd1.csv", header=TRUE)
> head(data)
  Treatment BodyWeightGain
1        T1          0.7651
2        T1          1.0150
3        T1          1.2759
4        T1          0.9837
5        T1          0.8557
6        T2          1.3113
> tail(data)
   Treatment BodyWeightGain
15        T3           1.154
16        T4           1.630
17        T4           1.505
18        T4           1.779
19        T4           1.454
20        T4           1.443
> GrandMean=mean(data$BodyWeightGain)
```

```
> GrandMean
[1] 1.291
> SST=sum((data$BodyWeightGain-GrandMean)^2)
> SST
[1] 1.983
> SSt=5*((mean(data[1:5,2])-GrandMean)^2+
  (mean(data[6:10,2])-GrandMean)^2+
  (mean(data[11:15,2])-GrandMean)^2+
  (mean(data[16:20,2])-GrandMean)^2)
> SSt
[1] 0.9821
> SSE=SST-SSt
> SSE
[1] 1.001
```

Solution 3 : short cut computation manually, but using R

Correction Factor (CF) = $(\Sigma Y..)^2/t.r$

```
> CF=(sum(data[,2]))^2/(4*5)
> CF
[1] 33.35
```

$SST = \Sigma Yij^2 - CF$

```
> SST=sum(data[,2]^2) - CF
> SST
[1] 1.983
```

$SSt = \Sigma(Yi.)^2/r - CF$

```
> SSt=(((sum(data[1:5,2]))^2+(sum(data[6:10,2]))^2+
 (sum(data[11:15,2]))^2+(sum(data[16:20,2]))^2)/5)-CF
> SSt
[1] 0.9821
```

$SSE = SST - SSE$

```
> SSE=SST-SSt
> SSE
[1] 1.001
```

Mean square for each variation can be calculated as below.

```
> MSt=SSt/3
> MSt
[1] 0.3274
> MSE=SSE/16
> MSE
[1] 0.06254
```

```
> Fstatistic=MSt/MSE
> Fstatistic
[1] 5.235
> qf(0.95,3,16) ##alpha=0.05
[1] 3.239
> qf(0.99,3,16) ##alpha=0.01
[1] 5.292
>
```

Based on solution 1, solution 2, and solution 3, ANOVA table can be describe as follows:

Table. ANOVA

Source of Variation	Degree of freedom	Sum square (SS)	Mean square (MS)	Fstatistic
Treatment	4 – 1	0.9821	0.3274	5.235*
Error	20 – 4	1.001	0.06254	
Total	20 – 1	1.983		

Alpha 0.05 = 3.239; alpha 0.01 = 5.292

Solution 4 : using R

```
> data=read.csv("crd1.csv", header=TRUE)
> head(data)
  Treatment BodyWeightGain
1        T1         0.7651
2        T1         1.0150
3        T1         1.2759
4        T1         0.9837
5        T1         0.8557
6        T2         1.3113
> data ## all data are displayed
   Treatment BodyWeightGain
1         T1         0.7651
2         T1         1.0150
3         T1         1.2759
4         T1         0.9837
5         T1         0.8557
6         T2         1.3113
7         T2         1.3034
8         T2         1.5975
9         T2         0.6453
10        T2         1.1484
11        T3         1.4521
12        T3         1.8463
13        T3         1.2639
14        T3         1.3987
15        T3         1.1541
16        T4         1.6298
```

```
17        T4         1.5055
18        T4         1.7790
19        T4         1.4540
20        T4         1.4434
> modelCRD=aov(BodyWeightGain~Treatment, data=data)
> summary(modelCRD)

            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    3  0.982   0.327    5.23   0.01 *
Residuals   16  1.001   0.063
---
Signif.codes:0‘***’0.001‘**’0.01‘*’ 0.05‘.’ 0.1‘ ’ 1
```

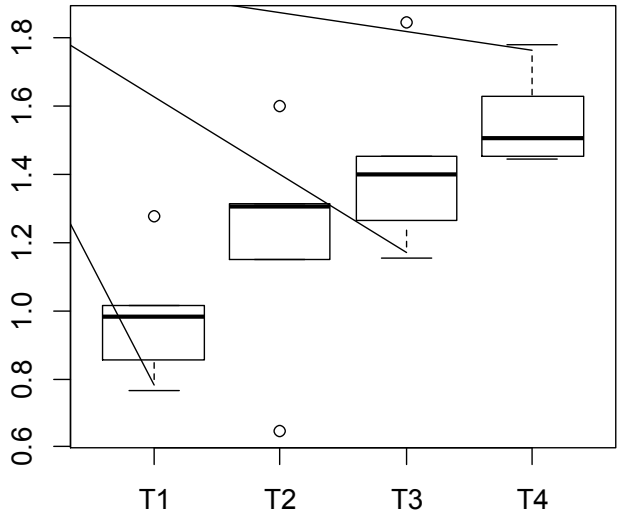
The data above is written in csv file in excel with file name crd1.csv which is saved in a folder where we work in. Actually we can directly write data and script together in R console or in R editor, as follows.

```
> Treatment2 <- rep(c("T1","T2","T3","T4"), each=5)
> Treatment2
 [1] "T1" "T1" "T1" "T1" "T1" "T2" "T2" "T2" "T2"
  "T2" "T3" "T3" "T3" "T3" "T3"
[16] "T4" "T4" "T4" "T4" "T4"
> BodyWeightGain2 <-
  c(0.7651,1.0150,1.2759,0.9837,0.8557,1.3113,
+
  1.3034,1.5975,0.6453,1.1484,1.4521,1.8463,1.2639,1.
  3987,1.1541,
+    1.6298,1.5055,1.7790,1.4540,1.4434)
> BodyWeightGain2
 [1] 0.7651 1.0150 1.2759 0.9837 0.8557 1.3113 1.3034
  1.5975 0.6453 1.1484
[11] 1.4521 1.8463 1.2639 1.3987 1.1541 1.6298 1.5055
  1.7790 1.4540 1.4434
> dat=aov(BodyWeightGain2~Treatment2)
> summary(dat)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment2   3  0.982   0.327    5.23   0.01 *
Residuals   16  1.001   0.063
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

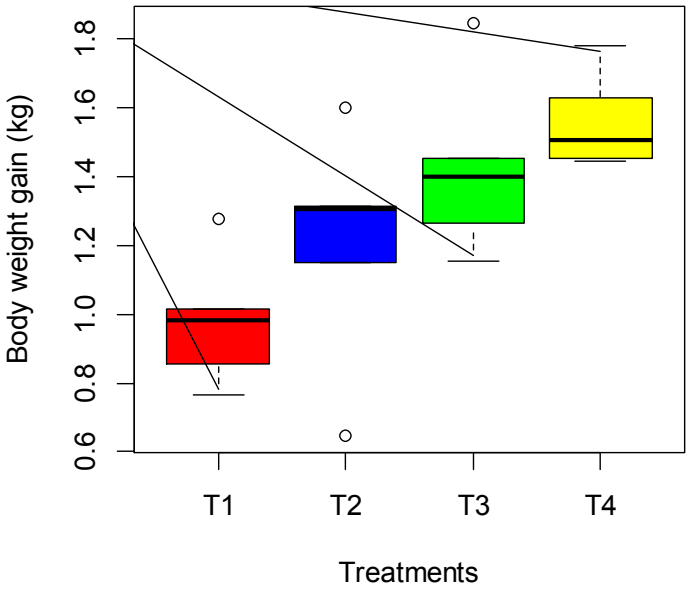
Based on ANOVA table, it can be concluded that different treatment or addition of prebiotic in ration affected broiler performance (body weight gain). Which treatments are differed will be discussed in the next chapter (Multiple

Comparison). However, to check the difference effect of treatments visually, we can use boxplot, as below.

```
> boxplot(BodyWeightGain~Treatment, data=data)
> boxplot(BodyWeightGain~Treatment, col=c("red",
+    "blue","green","yellow"),xlab="Treatments",
+     ylab="Body weight gain (kg)",data=data)
>
```



Or



To check the reliability of the experiment, we can see the coefficient of variation (CV). The degree of precision with which the treatments are compared is influenced by the CV, the higher the CV value the lower the reliability of the experiment. If an experimental results has a CV value more than 30 % meaning that the experiment is to be viewed with caution. Coefficient of variation can be formulated as:

$$CV = \frac{\sqrt{MSE}}{\bar{Y}} \; x \; 100\%$$

where MSE is mean square error and $\bar{Y}$ is overall mean or grand mean. The above example we got MSE is 0.063 and grand mean is 1.291405.

```
> MSE=0.063
> ybar=mean(data$BodyWeightGain)
> ybar
[1] 1.291405
> CV=(sqrt(MSE)/ybar)*100
> CV
[1] 19.43604
>
```

3.2 Unbalanced CRD

If the data is unbalance for which the replication for each treatment is not the same, ANOVA can still be done. For example, in the previous research example, data for both T3 and T4 consist of 4 and 3 data, respectively, as in table below.

Replication	Treatments			
	T1	T2	T3	T4
1	0.7651	1.3113	1.452	1.6298
2	1.0150	1.3034	1.8463	1.5055
3	1.2759	1.5975	1.2639	1.7790
4	0.9837	0.6453	1.3987	-
5	0.8557	1.1484	-	-

In R:

```
> data=read.csv("crd2.csv", header=TRUE)
> data
   Treatment BodyWeightGain
1         T1      0.7650848
2         T1      1.0149890
```

```
3          T1      1.2758507
4          T1      0.9837027
5          T1      0.8557105
6          T2      1.3113046
7          T2      1.3034288
8          T2      1.5974956
9          T2      0.6453479
10         T2      1.1484184
11         T3      1.4521396
12         T3      1.8462775
13         T3      1.2638854
14         T3      1.3987220
15         T4      1.6298013
16         T4      1.5054732
17         T4      1.7789886

> modelCRD2=aov(BodyWeightGain~Treatment, data=data)
> summary(modelCRD2)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    3 1.0453  0.3484   5.224 0.0138 *
Residuals   13 0.8671  0.0667
---
Signif.codes:0‘***’ 0.001‘**’0.01‘*’ 0.05‘.’ 0.1‘ ’ 1
```

Based on ANOVA table, treatments affected broiler performance ($P<0.05$).

IV. MULTIPLE COMPARISON

4.1 Introduction

F-test in ANOVA table tells us if there is a significant difference among groups or treatments. If the F-test is significant (H_0 is rejected), the question is between which pairs of treatments differed significantly from one another.

There are some procedures for pair-wise comparisons of means, for example, the Least Significance Difference (LSD), Tukey (honestly significant difference, HSD), Duncan's Multiple Range Test (DMRT), Student-Newman-Keuls (SNK), Dunnett, Scheffe, and Bonferroni test. Different researchers have offered some guidelines for choosing which test more appropriate, but actually there is no set rule for making decision to use a test. The following multiple test is example to be explained.

4.2 Least Significance Difference (LSD)

This procedure aims to test or compare the least difference between a pair of treatment means weather significant or not. If the difference of the two treatment means is greater than the LSD then this pair of treatment differ significantly. The advantage of the LSD, it has a low level of type II error and will most likely detect a difference if a difference really exists. The disadvantage of this test, it has a high level of type I error. Formula of LSD can be calculated as follows:

$$LSD_{12} = t_{\alpha/2,dfe}\sqrt{MSE\left(\frac{1}{n1}+\frac{1}{n2}\right)}$$

where $t_{\alpha/2}$ is t table (qt(p, df), dfe is degree of freedom for error, MSE is mean square error, and n1 and n1 is replication or number of data of treatment 1 and 2, respectively. For example, data in chapter III can be used for treatment comparison, as below.

```
> data=read.csv("crd1.csv", header=TRUE)
> modelCRD=aov(BodyWeightGain~Treatment, data=data)
> summary(modelCRD)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    3 0.9821  0.3274   5.235 0.0104 *
Residuals   16 1.0006  0.0625
---
Signif.codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘ ’ 1
```

Based on ANOVA table above the MSE is 0.0625, so for example we want to compare between treatment 1 (T1) with mean of 0.9790 and treatment 4 (T4) with mean 1.5623. The LSD can be calculated as follows:

```
> alpha=0.05
> qt(1-alpha/2,16)
[1] 2.119905
> t=qt(1-alpha/2,16)
> t
[1] 2.119905
> MSE=0.0625
> LSD=t*sqrt((MSE*((1/5)+(1/5))))
> LSD
[1] 0.3351865
> T4=1.5623
> T1=0.9791
> T4-T1
[1] 0.5832
```

Based on the calculation above, the different between T1 and T4 is 0.5832 and LSD is 0.3352, meaning that the difference between T1 and T4 is greater than LSD. It can be concluded that the two treatment (T1 and T4) are different ($P<0.05$). By using R package agricolae, the LSD procedure can be done like below.

```
> library(agricolae)
> LSD.test(modelCRD,"Treatment",alpha=0.05,console=T)

Study: modelCRD ~ "Treatment"

LSD t Test for BodyWeightGain

Mean Square Error:  0.06253821

Treatment,  means and individual ( 95 %) CI

   BodyWeightGain       std r       LCL      UCL       Min      Max
T1      0.9790676 0.1939057 5 0.7419825 1.216153 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.9641140 1.438284 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.1859425 1.660113 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.3252419 1.799412 1.4433824 1.778989

Alpha: 0.05 ; DF Error: 16
Critical Value of t: 2.119905

least Significant Difference: 0.3352889

Treatments with the same letter are not significantly
  different.
```

```
   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275     ab
T2      1.2011990     bc
T1      0.9790676      c
```

4.3 Tukey Test (HSD)

Tukey test is also known as the honestly significant difference (HSD). The advantage of this test, it has fewer incorrect conclusions of μ1 ≠ μ.. (type I errors) compared to the LSD, but the disadvantage of this test, there will be more incorrect μ1 = μ.. conclusions (type II errors). Tukey test is calculated from:

$$HSD_{12} = q_{\alpha(t,\text{dfe})}\sqrt{MSE/r}$$

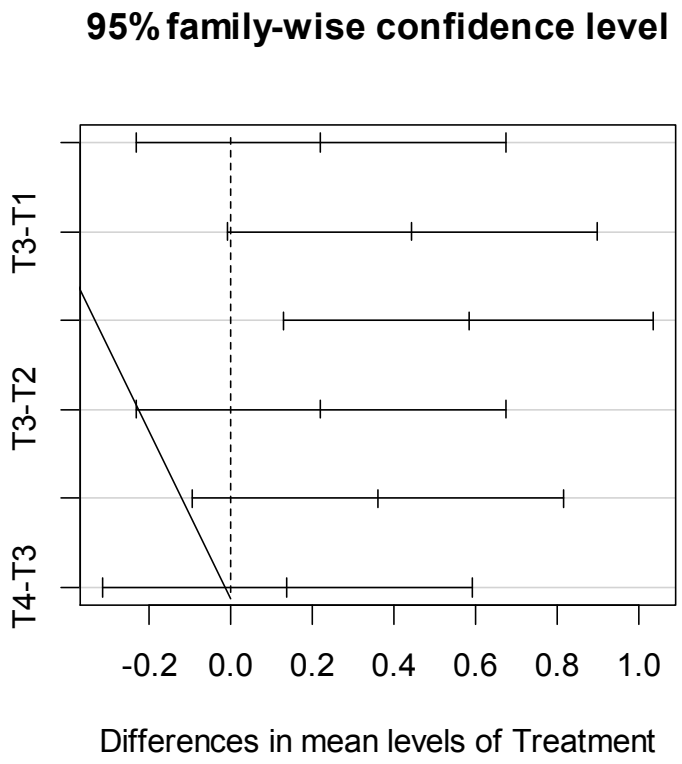
where q_α is chi-square table based on significance α and number of treatment (t) and degree of freedom for error (dfe) or in R (qtukey(p = 0.95, nmeans = 4, df = 16)), MSE is mean square error, and r is number of replication. For example, data in chapter III can be used for treatment comparison, as below.

```
> MSE=0.0625
> r=5
> alpha=0.05
> q=qtukey(p = 0.95, nmeans = 4, df = 16)
> q
[1] 4.046093
> HSD=q*sqrt(MSE/r)
> HSD
[1] 0.452367
> T1=0.9791
> T4=1.5623
> T4-T1
[1] 0.5832
>
```

Based on the calculation above, the different between T1 and T4 is 0.5832 and HSD is 0.4524, meaning that the difference between T1 and T4 is greater than HSD. It can be concluded that the two treatment (T1 and T4) are different (P<0.05). By using R package agricolae, the LSD procedure can be done like below.

```
> TukeyHSD(modelCRD,conf.level=0.05)
  Tukey multiple comparisons of means
    5% family-wise confidence level
```

```
Fit: aov(formula=BodyWeightGain~Treatment,data=data)

$`Treatment`
           diff        lwr       upr     p adj
T2-T1 0.2221315 0.13799198 0.3062710 0.5145986
T3-T1 0.4439600 0.35982045 0.5280995 0.0554362
T4-T1 0.5832594 0.49911993 0.6673989 0.0096706
T3-T2 0.2218285 0.13768896 0.3059680 0.5157092
T4-T2 0.3611279 0.27698844 0.4452675 0.1436508
T4-T3 0.1392995 0.05515997 0.2234390 0.8147416

>plot(TukeyHSD(modelCRD), conf.level=.95)
```



If using R package agricolae, the HSD procedure can be done like below.

```
> library(agricolae)
>HSD.test(modelCRD,"Treatment",alpha=0.05,console=T)

Study: modelCRD ~ "Treatment"

HSD Test for BodyWeightGain

Mean Square Error:  0.06253821

Treatment,  means
```

```
   BodyWeightGain       std r       Min      Max
T1      0.9790676 0.1939057 5 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.4433824 1.778989

Alpha: 0.05 ; DF Error: 16
Critical Value of Studentized Range: 4.046093

Minimun Significant Difference: 0.4525052

Treatments with the same letter are not significantly
  different.

   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275     ab
T2      1.2011990     ab
T1      0.9790676      b
```

4.4 Duncan's Multiple Range Test (DMRT)

DMRT compare between the range of a subset of the sample means and a calculated least significant range (LSR). This LSR increases with the number of sample means in the subset. If the range of the subset is greater than the LSR then the two means or treatments differ significantly according to desired significance level. Because of this sequential test, so the subset with the largest range should be compared first, followed by smaller subsets. The LSR can be computed as follows.

$$\text{LSR} = K_r\sqrt{MSE/r}$$

Where K_r is obtained from Duncan's table of significant ranges for a given α with df for experimental error (dfe). As previous example the MSE = 0.0625, r is 5 and Kr with alpha 0.05 and dfe 16 are

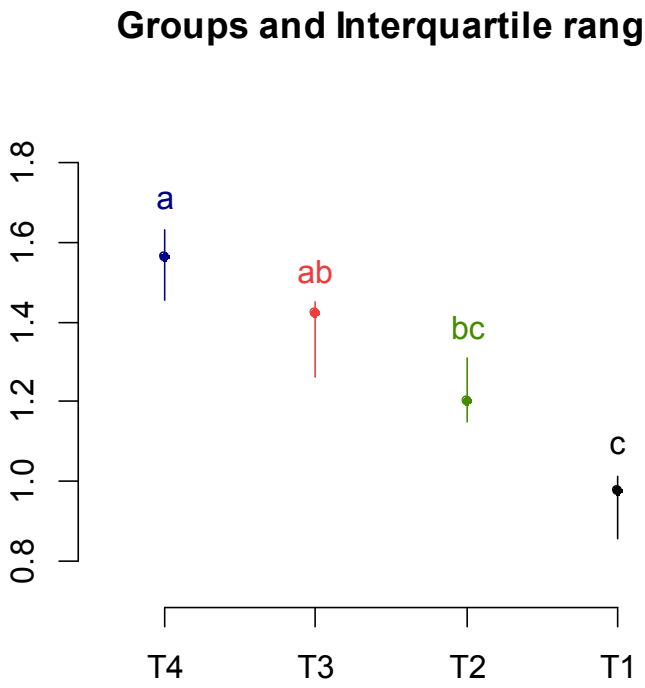
```
> MSE=0.0625
> r=5
> Kr=c(2.998, 3.144, 3.235)
> LSR=Kr*sqrt(MSE/r)
> LSR
[1] 0.3351866 0.3515099 0.3616840
> sequence=aggregate(BodyWeightGain~Treatment,
  data=data, mean)
```

```
> sequence
  Treatment BodyWeightGain
1        T1      0.9790676
2        T2      1.2011990
3        T3      1.4230275
4        T4      1.5623270
> sort(sequence$BodyWeightGain)
[1] 0.9790676 1.2011990 1.4230275 1.5623270
>
```

So the comparison of T1 vs T2, T2 vs T3 and T3 vs T4 should be compared to LSR 2 (0.3351866); comparison of T1 vs T3 and T2 vs T4 should be compared to LSR 3 (0.3515099), and comparison of T1 vs T4 should be compared to LSR 4 (0.3616840). For example, the difference between T1 and T4 (T1 vs T4) is 0.5832594 which is greater than LSR 4 (0.3616840), meaning that treatment T1 and T4 is different (P<0.05). By using agricolae package the DMRT can be done like below.

```
> library(agricolae)
> modelCRD=aov(BodyWeightGain~Treatment, data=data)
> out=duncan.test(modelCRD, "Treatment", alpha=0.05,
  console=T)
> out

Study: modelCRD ~ "Treatment"

Duncan's new multiple range test
for BodyWeightGain

Mean Square Error:  0.06253821

Treatment,  means

   BodyWeightGain       std r       Min      Max
T1      0.9790676 0.1939057 5 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.4433824 1.778989

Alpha: 0.05 ; DF Error: 16

Critical Range
        2         3         4
0.3352889 0.3515952 0.3617883
```

```
Means with the same letter are not significantly
  different.

   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275     ab
T2      1.2011990     bc
T1      0.9790676      c
> plot(out,variation="IQR")
```



4.5 Student-Newman Keuls (SNK)

Like DMRT, Student-Newman Keuls test is step down procedure where the difference between the largest and the smallest means are compared first and if there is significant different then continue to the next set of treatment pairs (the second largest vs the smallest or the second smallest vs the largest), or stop if the pair is not significant. The test is continued until founding a non-significant pair comparison of means.

The SNK is based on the studentized range distribution. The SNK can be computed as follows.

$$\text{SNK} = \frac{(\bar{y}1 - \bar{y}2)}{\sqrt{\frac{MSE}{2}\left(\frac{1}{r1} + \frac{1}{r2}\right)}}$$

where $\bar{y}1$ is mean of treatment 1 (T1) and $\bar{y}2$ is mean of treatment 2 (T2), MSE is mean square error, r1 and r2 is number of replication for treatment 1 and 2, respectively. For example, we want to compare between T1 (the smallest) and T4 (the largest). As previous example, the MSE = 0.0625, r1 and r4 is 5 each and T1 mean is 0.9791, while T4 mean is 1.5623. SNK can be calculated as follows.

```
> MSE=0.0625
> r1=5
> r4=5
> T1=0.9791
> T4=1.5623
> T2=1.2012
> T3=1.4230
> q=qtukey(p = 0.95, nmeans = 2:4, df = 16)
> q
[1] 2.997999 3.649139 4.046093
> SNK=(T4-T1)/(sqrt((MSE/2)*((1/r4)+(1/r1))))
> SNK
[1] 5.216299
> SNK=(T2-T1)/(sqrt((MSE/2)*((1/r4)+(1/r1))))
> SNK
[1] 1.986523
> SNK=(T3-T1)/(sqrt((MSE/2)*((1/r4)+(1/r1))))
> SNK
[1] 3.970362
> SNK=(T3-T2)/(sqrt((MSE/2)*((1/r4)+(1/r1))))
> SNK
[1] 1.98384
> SNK=(T4-T2)/(sqrt((MSE/2)*((1/r4)+(1/r1))))
> SNK
[1] 3.229777
>
```

Based on computation above it can be concluded that T4 and T1 is different with SNK (5.216299) which is greater than q (4.046093); T2 and T1 is not different with SNK (1.986523) which is not greater than q (2.997999); and soon, so that the overall comparison resulted in like below.

T1 T2 T3 T4

By using agricolae package, SNK test can be done like below.

```
> SNK.test(modelCRD,"Treatment",alpha=0.05,console=T)

Study: modelCRD ~ "Treatment"

Student Newman Keuls Test
for BodyWeightGain

Mean Square Error:  0.06253821

Treatment,  means

   BodyWeightGain       std r       Min      Max
T1      0.9790676 0.1939057 5 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.4433824 1.778989

Alpha: 0.05 ; DF Error: 16

Critical Range
        2         3         4
0.3352889 0.4081108 0.4525052

Means with the same letter are not significantly
different.

   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275      a
T2      1.2011990     ab
T1      0.9790676      b
```

The same thing with manual procedure, but in this agricolae package, critical range is for mean different. For example, mean different between T1 and T2 is 0.2221 which is not different from critical range 0.3352889, but mean different between T4 and T1 is 0.5832 which is different from critical range 0.4525052, and soon. Basically the result is the same weather using procedure manually or using agricolae package.

4.6 Dunnett's Test

Sometimes we are only interested in the comparison between controls and other treatments. For example, comparing a local variety of rice with several new varieties. In this case we can use the Dunnet test. In the Dunnet test only one comparative value

is needed to compare the controls with other treatments. The Dunnet test is similar to LSD, but the t-value used is not the student-t used in the LSD test, Dunnet test uses a different t table, called Dunnet table (http://sciences.ucf.edu/biology/d4lab/wp-content/uploads/sites/139/2016/11/Dunnetts-table.pdf).

$$Dunnet = t_{\alpha/2(dft,dfe)}\sqrt{\frac{MSE}{r}}$$

For example, using previous data with dft = 3 and dfe = 16, dunnet table is 2.59. Considering T1 as control, Dunnet test can be computed as below.

```
> MSE=0.0625
> r=5
> t=2.59 ##alpha=0.05, dft = 3 and dfe = 16
> T1=0.9791
> T4=1.5623
> T2=1.2012
> T3=1.4230
> Dunnet=t*sqrt(MSE/r)
> Dunnet
[1] 0.2895708
> T2-T1
[1] 0.2221
> T3-T1
[1] 0.4439
> T4-T1
[1] 0.5832
>
```

Based on Dunnet calculation, Dunnet test is `0.2895708`, T1 and T2 (`0.2221`) is not different (P>0.05), T1 and T3 (`0.4439`) is different (P<0.05) and T1 and T4 (`0.5832`) is different (P<0.05). By using "DescTools" package Dunnet test can be done as below.

```
> library(DescTools)
> DunnettTest(BodyWeightGain ~ Treatment, data = data)

  Dunnett's test for comparing several treatments with
  a control :
    95% family-wise confidence level
```

```
$`T1`
           diff       lwr.ci    upr.ci   pval
T2-T1 0.2221315 -0.18792148 0.6321845 0.3856
T3-T1 0.4439600  0.03390699 0.8540129 0.0325 *
T4-T1 0.5832594  0.17320647 0.9933124 0.0054 **

---
Signif.codes: 0'***'0.001'**'0.01'*'0.05'.'0.1' ' 1
```

4.7 Scheffe Test

This test aims to protect against a Type I error when all possible complex and simple comparisons are made. Scheffe test is used to make unplanned comparisons, rather than pre-planned comparisons. This test uses a different critical value (or at least it makes an adjustment to the critical value of F). The advantage of this test is flexibility to test any comparisons that appear interesting, but it has very low statistical power.

Scheffe test (ST) formula can be written as below.

$$\mathrm{ST} = \sqrt{(k-1) f_{value}\, MSE(1/r1 \; + \; 1/r2)}$$

Where k-1 is dft (degree of freedom between treatment), f_{value} is from ANOVA, MSE is mean square error r1 and r2 is replication or number of data of treatment 1 and 2, respectively. From previous example, ST can be computed as below.

```
> data=read.csv("crd1.csv", header=TRUE)
> modelCRD=aov(BodyWeightGain~Treatment, data=data)
> summary(modelCRD)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    3 0.9821  0.3274   5.235 0.0104 *
Residuals   16 1.0006  0.0625
---
Signif.codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘ ’ 1

> dft=3
> MSE=0.0625
> fvalue=qf(0.95,3,16)
> r1=5
> r2=5
> T1=0.9791
> T4=1.5623
> T2=1.2012
> T3=1.4230
> ST=sqrt(dft*fvalue*MSE*((1/r1)+(1/r2)))
> ST
[1] 0.4928644
```

```
> T2-T1
[1] 0.2221
> T3-T1
[1] 0.4439
> T4-T1
[1] 0.5832
> T3-T2
[1] 0.2218
> T4-T3
[1] 0.1393
> T4-T2
[1] 0.3611

> library(agricolae)
> scheffe.test(modelCRD,"Treatment",alpha=0.05,
  console=T)

Study: modelCRD ~ "Treatment"

Scheffe Test for BodyWeightGain

Mean Square Error  : 0.06253821

Treatment,  means

   BodyWeightGain       std r       Min      Max
T1      0.9790676 0.1939057 5 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.4433824 1.778989

Alpha: 0.05 ; DF Error: 16
Critical Value of F: 3.238872

Minimum Significant Difference: 0.4930151

Means with the same letter are not significantly
different.

   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275     ab
T2      1.2011990     ab
T1      0.9790676      b
>
```

4.8 Boferroni Test

Bonferroni test is conservative test that protects from Type I Error and prevent data from incorrectly appearing to be statistically significant by lowering the alpha value. Bonferroni test or Bonferroni correction considers p-value for each test is equal to alpha divided by the number of tests. The disadvantage of Bonferroni test is too conservative and may fail to catch some significant findings and vulnerable to Type II errors.

For example, in previous example the α for LSD is 0.05 and the number of comparison is 6 (T1 vs T2, T1 Vs T3, T1 vs T4, T2 vs T3, T2 vs T4, and T3 vs T4), then critical value for Bonferroni correction will be 0.05/6 = 0.008333333. By using LSD test as the same as previous test, Bonferroni correction or adjustment will be like below.

```
> LSD.test(modelCRD,"Treatment",alpha=0.05, console=TRUE)

Study: modelCRD ~ "Treatment"

LSD t Test for BodyWeightGain

Mean Square Error:  0.06253821

Treatment,  means and individual ( 95 %) CI

   BodyWeightGain       std r       LCL      UCL       Min      Max
T1      0.9790676 0.1939057 5 0.7419825 1.216153 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.9641140 1.438284 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.1859425 1.660113 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.3252419 1.799412 1.4433824 1.778989

Alpha: 0.05 ; DF Error: 16
Critical Value of t: 2.119905

least Significant Difference: 0.3352889

Treatments with the same letter are not significantly different.

   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275     ab
T2      1.2011990     bc
T1      0.9790676      c
> LSD.test(modelCRD,"Treatment",alpha=0.05, p.adj="bonferroni",
   console=T)

Study: modelCRD ~ "Treatment"

LSD t Test for BodyWeightGain
P value adjustment method: bonferroni

Mean Square Error:  0.06253821
```

```
Treatment,  means and individual ( 95 %) CI

   BodyWeightGain       std r       LCL      UCL       Min      Max
T1      0.9790676 0.1939057 5 0.7419825 1.216153 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.9641140 1.438284 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.1859425 1.660113 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.3252419 1.799412 1.4433824 1.778989

Alpha: 0.05 ; DF Error: 16
Critical Value of t: 3.008334

Minimum Significant Difference: 0.4758047

Treatments with the same letter are not significantly different.

   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275     ab
T2      1.2011990     ab
T1      0.9790676      b
```

If we do LSD test with alpha 0.008333333 will result in the same thing as Bonferroni correction.

```
> LSD.test(modelCRD,"Treatment",alpha=0.008333333, console=TRUE)

Study: modelCRD ~ "Treatment"

LSD t Test for BodyWeightGain

Mean Square Error:  0.06253821

Treatment,  means and individual ( 99.16667 %) CI

   BodyWeightGain       std r       LCL      UCL       Min      Max
T1      0.9790676 0.1939057 5 0.6426228 1.315512 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.8647543 1.537644 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.0865828 1.759472 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.2258823 1.898772 1.4433824 1.778989

Alpha: 0.008333333 ; DF Error: 16
Critical Value of t: 3.008334

least Significant Difference: 0.4758047

Treatments with the same letter are not significantly different.

   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275     ab
T2      1.2011990     ab
T1      0.9790676      b
>
```

4.9 Orthogonal Comparison and Contrast

Orthogonal contrast is a linear combination of variables whose total coefficients is zero which allow comparison of different treatments. For example, the first treatment will be compared with treatment 2,3,4, and treatment 2 will be compared

with treatment 3,4. and so on depending on the predetermined hypothesis. This mean test can be used for the planned comparison of the treatments. In previous example, for instance, we want to compare control (T1) versus prebiotic addition (T2, T3, T4) treatment, T2 versus T3 and T4, and T3 versus T4, as describe below.

```
> modelCRD <- aov( BodyWeightGain ~ Treatment, data = data )
> summary(modelCRD)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    3 0.9821  0.3274   5.235 0.0104 *
Residuals   16 1.0006  0.0625
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> comp1 <- c(3, -1, -1, -1) # T1 or control vs. T2, T3, and T4
> comp2 <- c(0, 2, -1, -1) # T2 vs. T3 and T4
> comp3 <- c(0, 0, 1, -1) # T3 vs. T4
> comparison <- cbind(comp1,comp2,comp3) ##combine the three comparison
> # tell R that the matrix to provide the contrasts that we want
> contrasts(data$Treatment) <- comparison
> modelCRD.new <- aov(BodyWeightGain ~ Treatment, data = data)
> summary(modelCRD.new)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    3 0.9821  0.3274   5.235 0.0104 *
Residuals   16 1.0006  0.0625
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary.aov(modelCRD.new, split=list(Treatment=list("T1 or control vs.
   prebiotic addition"=1, "T2 vs T3 & T4"=2, "T3 vs T4"=3)))
                                                 Df Sum Sq Mean Sq F value
Treatment                                         3 0.9821  0.3274   5.235
  Treatment: T1 or control vs. prebiotic addition 1 0.6504  0.6504  10.399
  Treatment: T2 vs T3 & T4                        1 0.2832  0.2832   4.528
  Treatment: T3 vs T4                             1 0.0485  0.0485   0.776
Residuals                                        16 1.0006  0.0625
                                                  Pr(>F)
Treatment                                        0.01042 *
  Treatment: T1 or control vs. prebiotic addition 0.00529 **
  Treatment: T2 vs T3 & T4                       0.04923 *
  Treatment: T3 vs T4                            0.39150
Residuals
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary.lm(modelCRD.new) ##or use this script, the same thing

Call:
aov(formula = BodyWeightGain ~ Treatment, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.55585 -0.12005 -0.00984  0.10420  0.42325

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.29141    0.05592  23.094 1.03e-13 ***
Treatmentcomp1 -0.10411    0.03228  -3.225  0.00529 **
Treatmentcomp2 -0.09716    0.04566  -2.128  0.04923 *
Treatmentcomp3 -0.06965    0.07908  -0.881  0.39150
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2501 on 16 degrees of freedom
Multiple R-squared:  0.4953,    Adjusted R-squared:  0.4007
F-statistic: 5.235 on 3 and 16 DF,  p-value: 0.01042

>
```

Based on the comparison above, control differ from prebiotic addition, T2 differ from T3 and T4.

V. ANOVA ASSUMPTION

5.1 Introduction

Before doing analysis of variance there are several assumptions that should be fulfilled. ANOVA test can be applied only when the observations are obtained independently and randomly from the population, the experimental errors are normally distributed, and these normal populations have a common variance. What if we analyze data that actually does not meet the assumptions of variance analysis? If that happens, then the conclusions taken will not describe the actual situation and even misleading. Thus, before conducting a variance analysis, it is suggested to first check whether the data has met the basic assumptions of variance analysis or not. Violation to one of these assumptions will affect to bias conclusion of the research.

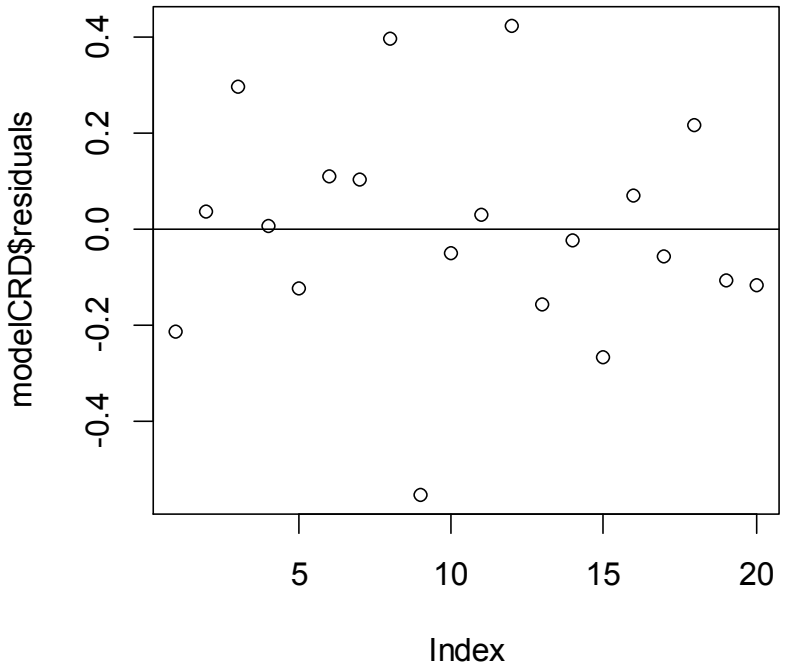
5.2 Independency

The sample we use should be selected randomly and independently. The residual value and data for each observation of the experimental unit must be free from each other, both in the treatment itself (within group) or between treatments (between groups). If this condition is not met, it will be difficult to detect any real differences that might exist. Independency test can use Durbin Watson Test, for example, by using previous example (chapter IV), the independency of the data can be detected as following.

```
> durbinWatsonTest(modelCRD)
 lag Autocorrelation D-W Statistic p-value
   1      -0.2398849       2.41987   0.776
 Alternative hypothesis: rho != 0
```

We can see from p-value (0.776) indicated that the data is independent and not auto correlate. The simplest graphical way to check for independence is by plotting the residuals, like below. If the points are symmetrically distributed around a horizontal line with a roughly constant variance meaning that the data is independent.

```
> plot(modelCRD$residuals)
> abline(h = 0)
```

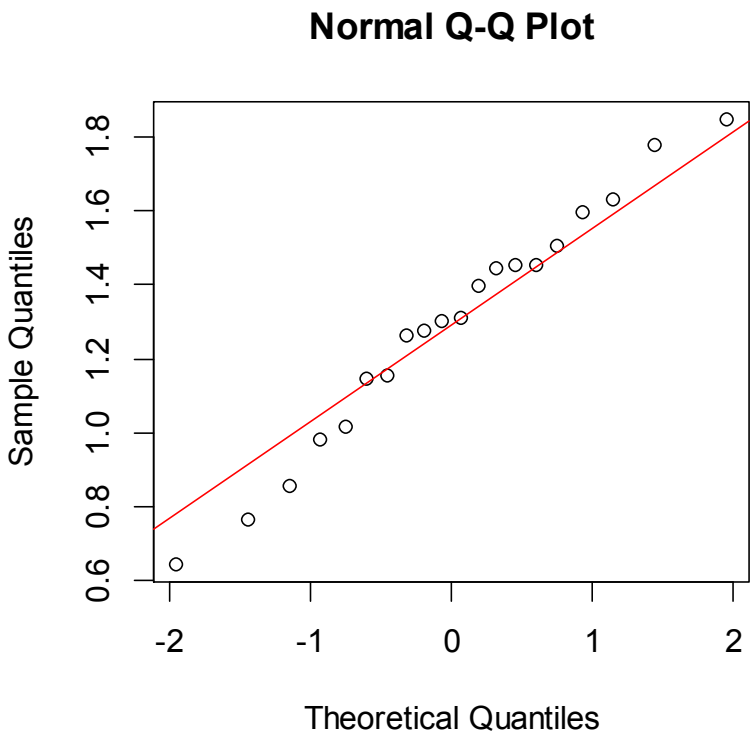


5.3 Normality

Normality means the residual value (εij) in each treatment (group) associated with the Yi observation value and this residual value should be normally distributed. If the residual value is normally distributed, then the Yi value will be normally distributed. If the sample size are the same and variance of each treatment are homogeneous, then the ANOVA test is very strong against this assumption, and even the impact of abnormalities is not too serious. However, if the abnormality is accompanied by heterogeneous variance, the problem can be serious on research conclusion taken. If the data size is large, normality assumption can be relaxed, but if the data size is very small then normality is very important.

Normality assumption can be seen visually using qqplot, like below.

```
> qqnorm(data$BodyWeightGain)
> qqline(data$BodyWeightGain,col="red")
```



The closer the spot of data to the red line the better meaning data are normally distributed. To make sure that data is normally distributed, we can use Shapiro.test like below.

```
> shapiro.test(modelCRD$residuals)

        Shapiro-Wilk normality test

data:  modelCRD$residuals
W = 0.9669, p-value = 0.6885

>
```

Based on Shapiro.test it can be seen that p-value is 0.6885 which is greater than 0.05, meaning not significant (P>0.05). This result indicated that data is normally distributed.

5.4 Homogeneity of Variance

Another assumption underlying the analysis of variance is homogeneity of the variance or it is called assumption of homoscedasticity. Homoscedasticity means that the variance of residual values is constant. The assumption of homogeneity requires that the residual distribution for each treatment or group must have the same variance.

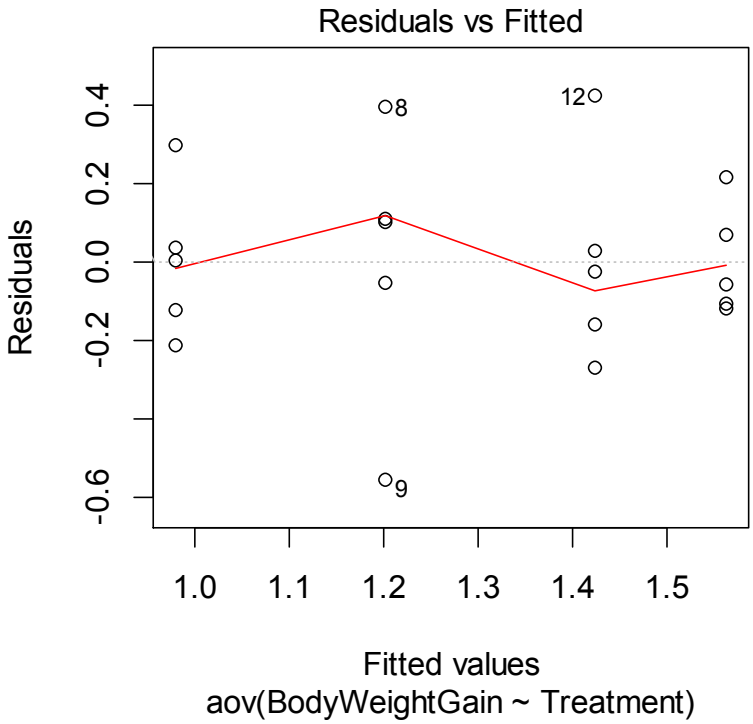
In practice, this means that the value of Yij at each level of the independent variable varies around the mean value. Testing for equal variances between treatments is levenTest for one-way ANOVA or barlett.test, like below. Bartlett's test can be used to test homogeneity of variances in k samples that can be more than two. While leveneTest is more robust than bartlett.test when the distributions of the data are not normal, and fligner.test is another test for homogeneity of variances which is the most robust test.

```
> library(car)
Loading required package: carData
> leveneTest(BodyWeightGain~Treatment, data=data)

Levene's Test for Homogeneity of Variance (center =
  median)
      Df F value Pr(>F)
group  3   0.417 0.7432
      16

> ##or
> bartlett.test(BodyWeightGain~Treatment, data=data)

        Bartlett test of homogeneity of variances

data:  BodyWeightGain by Treatment
Bartlett's K-squared = 3.1145, df = 3, p-value = 0.3743

>##or
> fligner.test(BodyWeightGain~Treatment, data=data)

  Fligner-Killeen test of homogeneity of variances

data:  BodyWeightGain by Treatment
Fligner-Killeen:med chi-squared = 1.2826, df = 3, p-
  value = 0.7333
>
```

Based on levenTest with p-value 0.7432, barlett.test with p-value 0.3743, and fligner.test with p-value 0.7333, all of them are greater than 0.05, meaning that residual variance of each treatment are homogeneous.

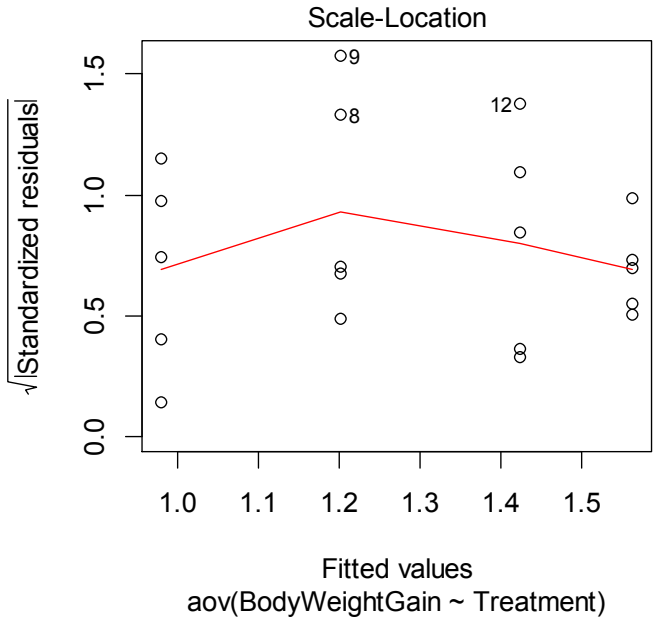
Or we can see the homogeneity of residual by plotting them, like below.

```
> plot(modelCRD, 1)
```



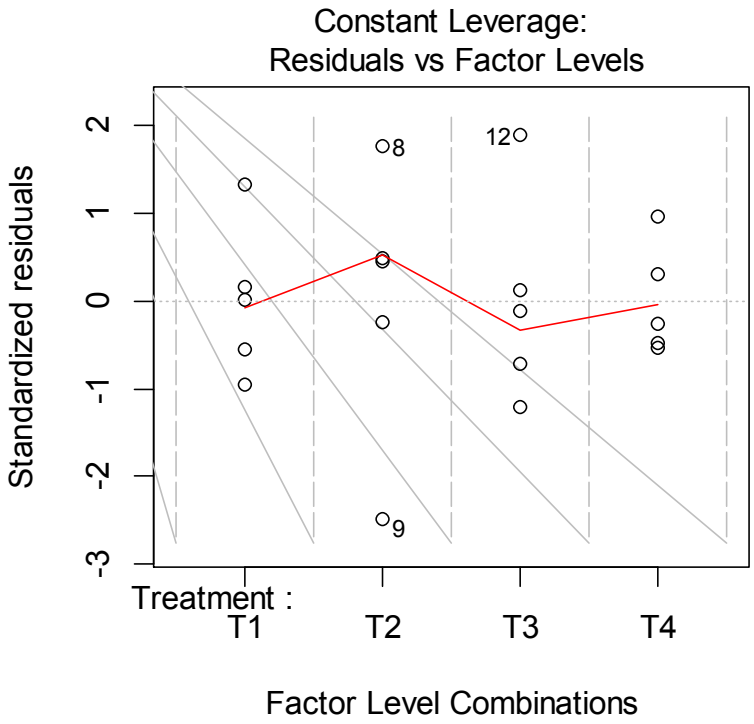
This plot shows the pattern of residuals, ideally the residuals should show similar scatter for each condition of treatments. It can be seen that there is a similarity of residuals with the larger fitted values. This is called homoscedasticity meaning that variance in the response equal across groups or treatments.

```
> plot(modelCRD, 3)
```



This is like the first plot but now to specifically test if the residuals increase with the fitted values. The plot shows that residuals does not increase with the fitted value, meaning residual variances are homogeneous among the treatments.

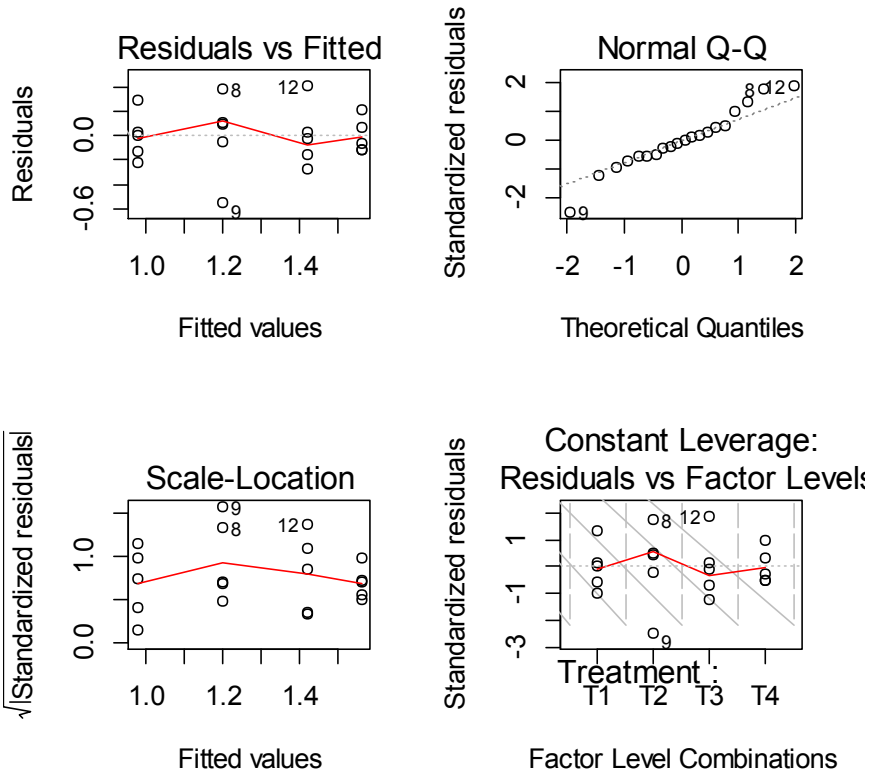
```
> plot(modelCRD, 5)
```



This plot shows which levels of the treatment are best fitted, T4 is best fitted.

Checking all assumption can use script like below.

```
> check <- par(mfrow=c(2,2),cex=.8)
> plot(modelCRD)
> par(check)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
Treatment :
T1 T2 T3 T4
Factor Level Combinations

VI. DATA TRANSFORMATION

6.1 Introduction

Transformation is an effort carried out with the main goal of changing the scale of measurement of original data into another form so that the new data can meet the assumptions underlying the variance analysis. In other words, data transformation is needed when the data violates ANOVA assumption in order to achieve the assumption so that the conclusions taken describe the actual situation and not misleading. Data transformation usually deals with normalizing or scaling data and handling skewness.

6.2 Data Transformation

Data transformation can be a form of natural logarithm, common logarithm, square root, cube root, reciprocal, reciprocal square root, sine, arcsine, power of 3, etc. Which one is appropriate depending on the data condition. In general, data transformation is to make a variable linear. Therefore, various transformations can be tried and tested for linearity using tests for normality, as well as visual displays, Q-plots, etc.

Other test that can be used to check our data is looking at the skewness of the data. If the value of skewness lies above +1 or below -1, data is highly skewed (need transformation), between +0.5 to -0.5 is moderately skewed (need transformation), and if the value is 0, then the data is symmetric (no need transformation) (Vadali, 2017).

Skewness test for previous example is like below.

```
> data=read.csv("crd1.csv", header=TRUE)
> library(e1071)
> checkData<-skewness(data$BodyWeightGain)
> checkData
[1] -0.2621765
>
```

Based on the test above it can be concluded that the data is not skewness meaning relatively symmetric or normally distributed.

6.3 Examples of Data Transformation

Data like growth rates usually use exponential and log transforms, and this type of transformation is appropriate particularly if the variance increases with the mean. If

a log transform does not normalize our data we could try a reciprocal (1/x) transformation. This is often used for enzyme reaction rate data. For count data, for example, blood cells on a haemocytometer or woodlice in a garden, square root transformation is often used. While arcsine transformation is useful for data like percentage, ages and proportions.

Tabachnick and Fidell (2007) and Howell (2007) suggested the following guidelines to transform data (see table).

Data condition	Suggested data transformation
Moderately positive skewness	Square root newX = sqrt(X)
Substantially positive skewness	Logarithmic (Log 10) newX = log10(X)
Substantially positive skewness (with zero value	Logarithmic (Log 10) newX = log10(X + C)
Moderately negative skewness	Square root newX = sqrt(K – X)
Substantially negative skewness	Logarithmic (Log 10) newX = log10(K – X)

C = a constant added to each score so that the smallest score is 1.
K = a constant from which each score is subtracted so that the smallest score is 1; usually equal to the largest score + 1.

Below is an example to make data transformation. There are three feed treatment, A is conventional feed, B and C new introducing feed. The three ration are given to turkey for two month trial (0-60 days of age). Body weight at 60 days of age is presented in table below. Is there any different body weight of turkey treated with the different feed?

Table. Body weight (60 days) of turkey fed 3 different ration

Turkey	Ration		
	A	B	C
1	2	5	3
2	3	6	5
3	2	5	4
4	2	4	10

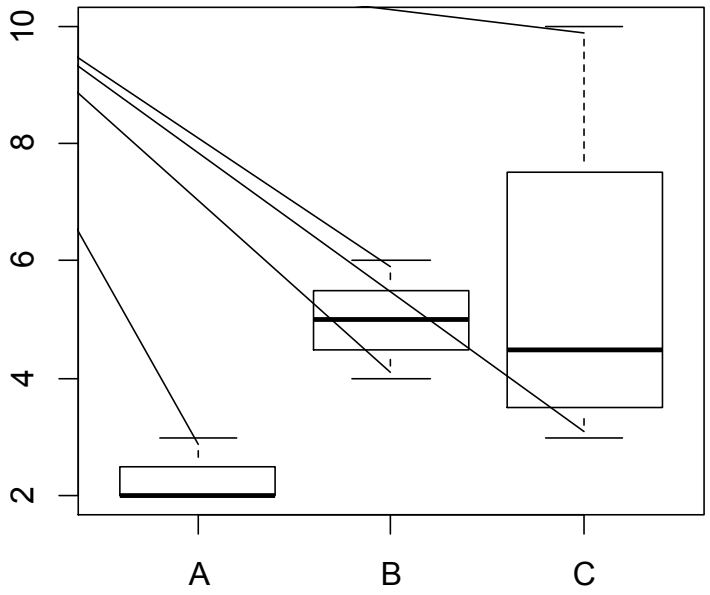
Before checking the assumption, we can do ANOVA to see the result look like. The data of body weight of turkey can be arranged and read like below.

```
> data=read.csv("bodyWeight.csv", header=T)
> data
   Treatment BodyWeight
1          A          2
2          A          3
3          A          2
4          A          2
5          B          5
6          B          6
7          B          5
8          B          4
9          C          3
10         C          5
11         C          4
12         C         10

> fit=aov(BodyWeight~Treatment, data=data)
> summary(fit)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    2  24.50  12.250   3.472 0.0763 .
Residuals    9  31.75   3.528
---
Signif.codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’ 0.1 ‘ ’ 1


> library(agricolae)
> LSD.test(fit, "Treatment", alpha=0.05, console=T)

Study: fit ~ "Treatment"

LSD t Test for BodyWeight

Mean Square Error:  3.527778

Treatment,  means and individual ( 95 %) CI

  BodyWeight       std r       LCL      UCL Min Max
A       2.25 0.5000000 4 0.1255653 4.374435   2   3
B       5.00 0.8164966 4 2.8755653 7.124435   4   6
C       5.50 3.1091264 4 3.3755653 7.624435   3  10

Alpha: 0.05 ; DF Error: 9
Critical Value of t: 2.262157

least Significant Difference: 3.004404
```

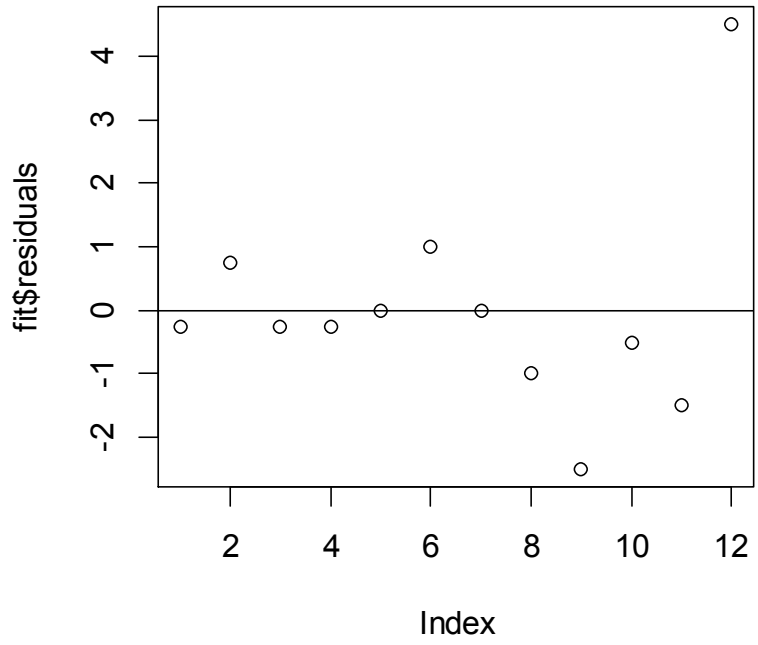
```
Treatments with the same letter are not significantly
  different.

  BodyWeight groups
C       5.50      a
B       5.00     ab
A       2.25      b
> range(data$BodyWeight)
[1]  2 10
>
```

The range is quite far (2 and 10) and mean of treatment A and B or C is quite different, but the result of ANOVA is not significant (P>0.05). Thus, there is something that need to be checked.

```
> #homogeneity variance
> bartlett.test(BodyWeight~Treatment, data=data)

        Bartlett test of homogeneity of variances

data:  BodyWeight by Treatment
Bartlett's K-squared = 8.6359, df = 2, p-value =
  0.01333

> #normality
> shapiro.test(fit$residuals)

        Shapiro-Wilk normality test

data:  fit$residuals
W = 0.84396, p-value = 0.03095

>
>
> library(e1071)
> checkData<-skewness(data$BodyWeight)
> checkData
[1] 1.496461
>

> #Independency
> boxplot(BodyWeight~Treatment, data=data)
```



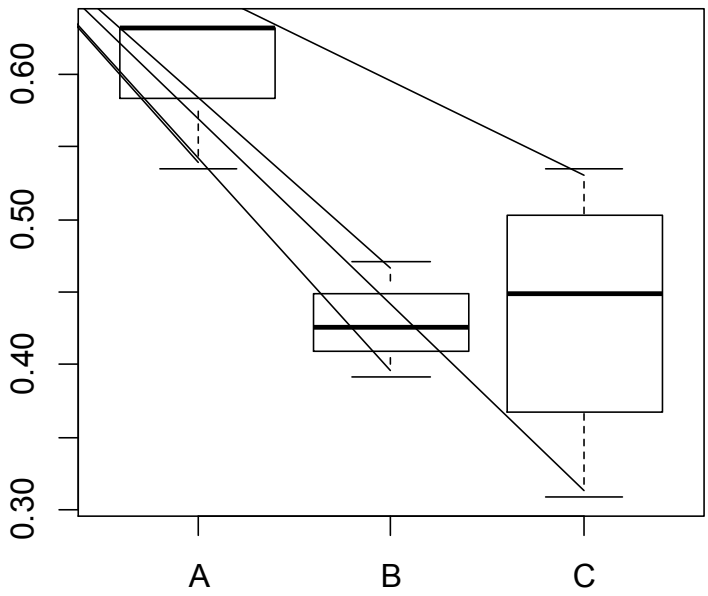

```
> durbinWatsonTest(fit)
 lag Autocorrelation D-W Statistic p-value
   1     -0.08070866      1.521654    0.11
 Alternative hypothesis: rho != 0
> plot(fit$residuals)
> abline(h=0)
```



Based on homogeneity test (0.0133), normality test (0.03095) , skewness test (1.496461, highly skewed), and residual plot (negative skewed), it can be concluded that the data violated the ANOVA assumption, although independency test showed that the data is independent (0.11, P>0.05). Thus, data transformation is needed.

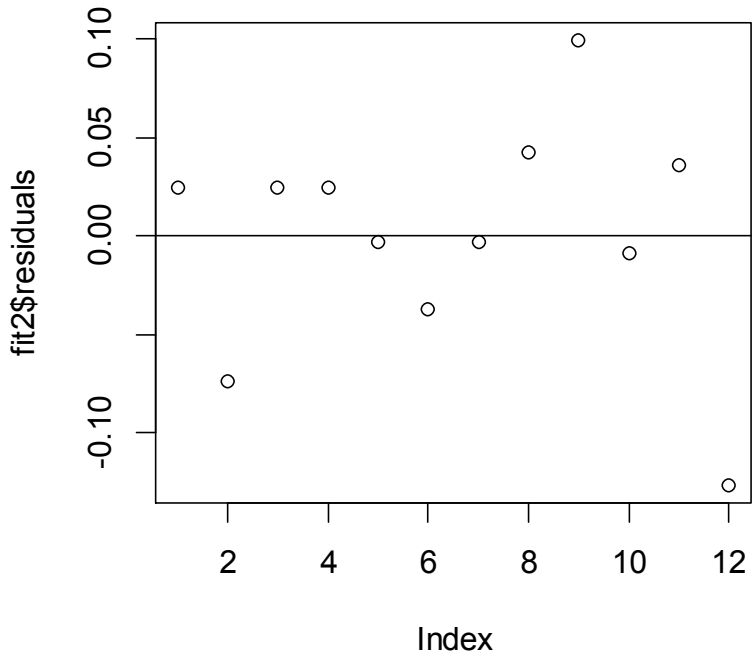
a. Reciprocal square root transformation

```
> data$BWtrans=1/sqrt(data$BodyWeight+0.5)
> checkData<-skewness(data$BWtrans)
> checkData
[1] 0.08445666
> bartlett.test(BWtrans~Treatment, data=data)

        Bartlett test of homogeneity of variances

data:  BWtrans by Treatment
Bartlett's K-squared = 3.0234, df = 2, p-value = 0.2205

> boxplot(BWtrans~Treatment, data=data)
```




```
> fit2=aov(BWtrans~Treatment, data=data)
> summary(fit2)
            Df  Sum Sq Mean Sq F value  Pr(>F)
```

```
Treatment    2 0.08249 0.04125   9.863 0.00539 **
Residuals    9 0.03764 0.00418
---
Signif.codes:0‘***’0.001‘**’0.01‘*’0.05‘.’0.1 ‘ ’ 1
> LSD.test(fit2, "Treatment", alpha=0.05, console=T)

Study: fit2 ~ "Treatment"

LSD t Test for BWtrans

Mean Square Error:  0.004181736

Treatment,  means and individual ( 95 %) CI

    BWtrans        std r       LCL       UCL       Min       Max
A 0.6079723 0.04896652 4 0.5348296 0.6811150 0.5345225 0.6324555
B 0.4291099 0.03247289 4 0.3559672 0.5022526 0.3922323 0.4714045
C 0.4352338 0.09535722 4 0.3620911 0.5083765 0.3086067 0.5345225

Alpha: 0.05 ; DF Error: 9
Critical Value of t: 2.262157

least Significant Difference: 0.1034394

Treatments with the same letter are not significantly
  different.

    BWtrans groups
A 0.6079723      a
C 0.4352338      b
B 0.4291099      b
> require(car)
> durbinWatsonTest(fit2)
 lag Autocorrelation D-W Statistic p-value
   1      -0.1210421      1.800113   0.284
 Alternative hypothesis: rho != 0

> shapiro.test(fit2$residuals)

        Shapiro-Wilk normality test

data:  fit2$residuals
W = 0.93843, p-value = 0.478

> plot(fit2$residuals)
```

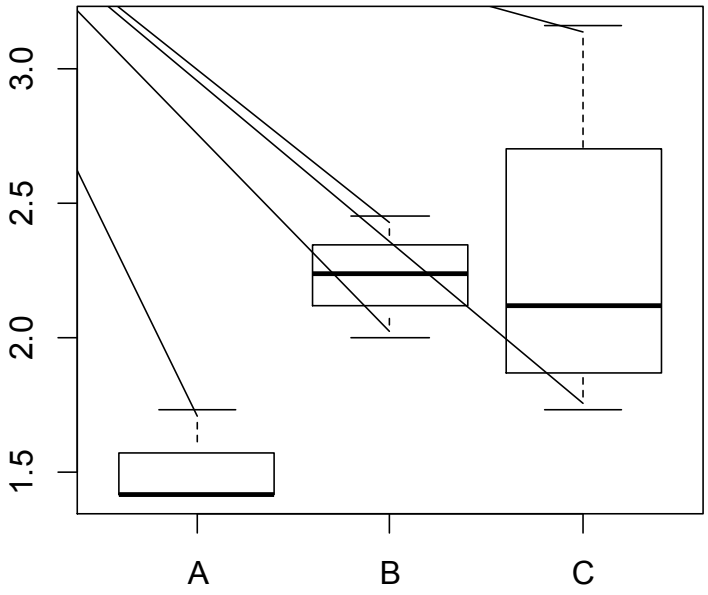


Although the data transformed to reciprocal square root meet all the ANOVA assumption (independent, homogeneous variance, and normally distributed), but based on LSD.test it is not plausible because the smallest mean of original data change to the largest mean after transformation.

b. Square root transformation

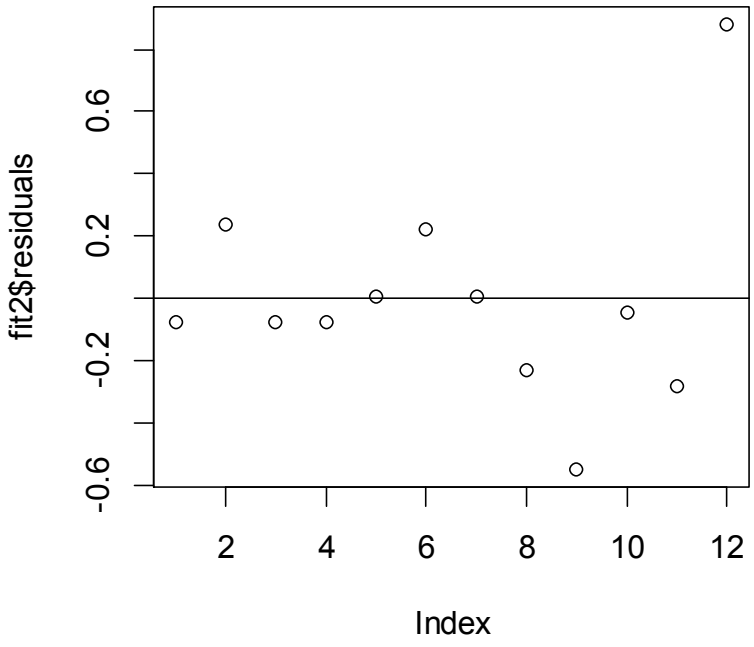
```
> data$BWtrans=sqrt(data$BodyWeight)
> data
   Treatment BodyWeight  BWtrans
1          A          2 1.414214
2          A          3 1.732051
3          A          2 1.414214
4          A          2 1.414214
5          B          5 2.236068
6          B          6 2.449490
7          B          5 2.236068
8          B          4 2.000000
9          C          3 1.732051
10         C          5 2.236068
11         C          4 2.000000
12         C         10 3.162278
> checkData<-skewness(data$BWtrans)
> checkData
[1] 0.6422686
> bartlett.test(BWtrans~Treatment, data=data)
```

```
        Bartlett test of homogeneity of variances

data:  BWtrans by Treatment
Bartlett's K-squared = 6.001, df = 2, p-value = 0.04976
> boxplot(BWtrans~Treatment, data=data)
```




```
> fit2=aov(BWtrans~Treatment, data=data)
> summary(fit2)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    2  1.557  0.7786   5.246 0.0309 *
Residuals    9  1.336  0.1484
---
Signif. codes:0‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘ ’ 1
> LSD.test(fit2, "Treatment", alpha=0.05, console=T)

Study: fit2 ~ "Treatment"

LSD t Test for BWtrans

Mean Square Error:  0.148431

Treatment,  means and individual ( 95 %) CI

   BWtrans       std r      LCL      UCL      Min      Max
A 1.493673 0.1589186 4 1.057905 1.929441 1.414214 1.732051
```

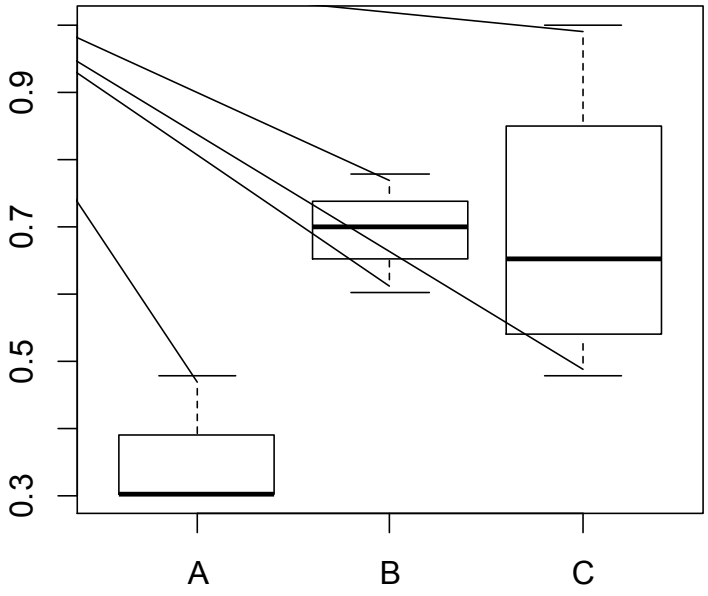
```
B 2.230406 0.1836198 4 1.794639 2.666174 2.000000 2.449490
C 2.282599 0.6215478 4 1.846831 2.718367 1.732051 3.162278

Alpha: 0.05 ; DF Error: 9
Critical Value of t: 2.262157

least Significant Difference: 0.6162687

Treatments with the same letter are not significantly
  different.

   BWtrans groups
C 2.282599      a
B 2.230406      a
A 1.493673      b
> require(car)
> durbinWatsonTest(fit2)
 lag Autocorrelation D-W Statistic p-value
   1     -0.08520361      1.586411   0.178
 Alternative hypothesis: rho != 0
> shapiro.test(fit2$residuals)

        Shapiro-Wilk normality test

data:  fit2$residuals
W = 0.87967, p-value = 0.08679

> plot(fit2$residuals)
```



The result indicated that after transformation into square root the data is still skew and not normally distributed, variance is not homogeneous, although the data is independent and ANOVA is significant with fair mean comparison.

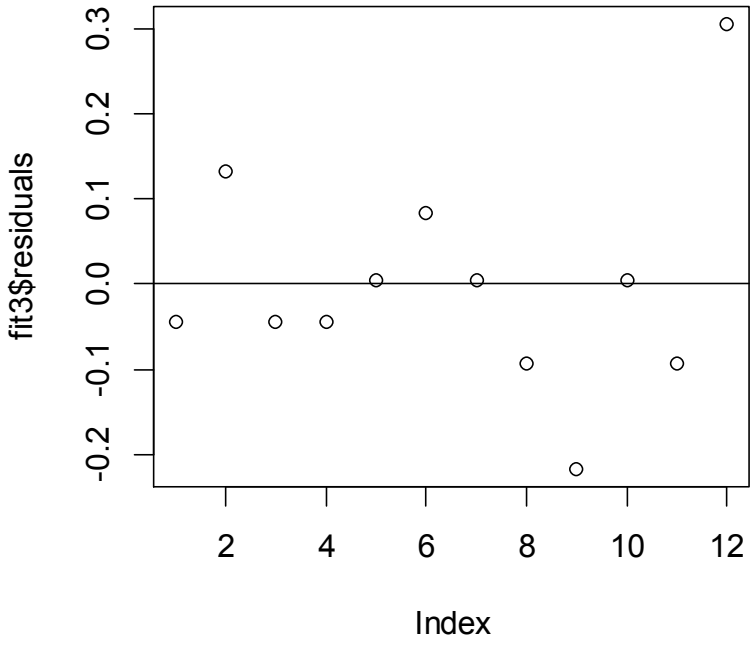
c. Log transformation

```
> data$BWtrans=log10(data$BodyWeight)
> checkData<-skewness(data$BWtrans)
> checkData
[1] 0.2448218
> bartlett.test(BWtrans~Treatment, data=data)

        Bartlett test of homogeneity of variances

data:  BWtrans by Treatment
Bartlett's K-squared = 3.9602, df = 2, p-value = 0.1381

>
> boxplot(BWtrans~Treatment, data=data)
```




```
> fit3=aov(BWtrans~Treatment, data=data)
> summary(fit3)
            Df Sum Sq Mean Sq F value Pr(>F)
Treatment    2 0.3257 0.16285   7.797 0.0108 *
Residuals    9 0.1880 0.02089
---
Signif.codes:0‘***’0.001‘**’0.01‘*’0.05‘.’0.1 ‘ ’ 1
```

```
> LSD.test(fit3, "Treatment", alpha=0.05, console=T)

Study: fit3 ~ "Treatment"

LSD t Test for BWtrans

Mean Square Error:  0.02088752

Treatment,  means and individual ( 95 %) CI

    BWtrans        std r       LCL       UCL       Min       Max
A 0.3450528 0.08804563 4 0.1815835 0.5085221 0.3010300 0.4771213
B 0.6945378 0.07207090 4 0.5310685 0.8580071 0.6020600 0.7781513
C 0.6945378 0.22297152 4 0.5310685 0.8580071 0.4771213 1.0000000

Alpha: 0.05 ; DF Error: 9
Critical Value of t: 2.262157

least Significant Difference: 0.2311805

Treatments with the same letter are not significantly
  different.

    BWtrans groups
B 0.6945378      a
C 0.6945378      a
A 0.3450528      b
> require(car)
> durbinWatsonTest(fit3)


 lag Autocorrelation D-W Statistic p-value
   1      -0.1014409      1.696225   0.218
 Alternative hypothesis: rho != 0


> shapiro.test(fit3$residuals)

        Shapiro-Wilk normality test

data:  fit3$residuals
W = 0.91999, p-value = 0.2858

> plot(fit3$residuals)
```



Finally, by using log transformation the data meet all ANOVA assumption and the ANOVA result is significant with reasonable mean comparison.

VII. RANDIMIZED COMPLETE BLOCK DESIGN (RCBD)

7.1 Introduction

Randomized Complete Block Designs (RCBD) is a standard design for agricultural experiments where factor levels are randomly applied to separate experimental units within each block. Block is not factor that we want to investigate, but block is only a way to reduce error variation which is caused by not homogeneity of the background of the experimental unit. In this design the different background of the experimental unit is grouped into several groups where within the groups the experimental unit is homogeneous. Treatment or factor levels then is applied to experimental unit within each block. Randomization for the RCBD is done only to experimental units within each block, while in CRD randomization is done to all experimental units.

For example, a research is conducted to investigate the effect of prebiotic addition in ration on broiler performance (body weight gain). There are 4 treatments applied to broiler chicken, those are base ration (T1), T1 plus 0.2% prebiotic addition (T2), T1 plus 0.4% prebiotic addition (T3), and T1 plus 0.6% prebiotic addition (T4), and in this experiment there are five broiler strains, those are S1, S2, S3, S4, and S5. Here we suspect that different strain of broiler has different effect on body weight gain, so we consider to localize the effect of strain by separating each strain as blocks. Hypothesis for this design is H_0 : $\mu1 = \mu2 = \mu3 = \mu4 = \mu5$ or H_0 : $\tau1 = \tau2 = \tau3 = \tau4 = \tau5$; and H_1: at least one of the means are different from the others.

7.2 Linear Model and randomization in RCBD

Linear model for the RCBD is

$$Yij = \mu + \tau i + \beta j + \varepsilon ij. \qquad i = 1,...,t; \quad j = 1,...,r$$

where:

yij = an observation in treatment i and block j
μ = the overall mean
τi = the effect of treatment i
βj = the fixed effect of block j
εij = random error
t = the number of treatments; r = the number of blocks

Total sum of squares of RCBD can be sum square of block, treatment and residual, as below.

$$\text{SST} = \text{SSt} + \text{SSb} + \text{SSE}$$

where

$$\text{SST} = \sum_i \sum_j (yij - \bar{y}..)^2$$

$$\text{SSt} = \sum_i \sum_j (\bar{y}i. - \bar{y}..)^2$$

$$\text{SSb} = \sum_i \sum_j (\bar{y}.j - \bar{y}..)^2$$

$$\text{SSE} = \sum_i \sum_j (yij - \bar{y}i. - \bar{y}.j + \bar{y}..)^2 = SST - SSt - SSb$$

Sums of squares above can be calculated using computation below.

$$\text{CF} = \frac{\left(\sum_i \sum_j yij\right)^2}{t.r}$$

$$\text{SST} = \sum_i \sum_j yij^2 - CF$$

$$\text{SSt} = \sum_i \frac{\left(\sum_j yij\right)^2}{r} - CF$$

$$\text{SSb} = \sum_j \frac{(\sum_i yij)^2}{t} - CF$$

$$SSE = SST - SSt - SSb$$

The corresponding degrees of freedom of SST = SSt + SSb + SSE are:

$$(tr - 1) = (t - 1) + (r - 1) + (t - 1)(r - 1),$$

then mean square can be calculated as below.

MSt = SSt/dft = SSt/(t-1)

MSb = SSb/dfb =SSb/(r-1)

MSE = SSE/dfe = SSE/(t-1)(r-1)

F staistic = MSt/MSE compared to F table with (t-1) and (t-1)(r-1) degrees of freedom for critical value. For an α level of significance H_0 is rejected if *F statistic* > $F_{\alpha,(t-1),(t-1)(r-1)}$.

ANOVA table can be describe as below.

Source of variation	df	SS	MS=SS/df	F
Treatment	t-1	SSt	MSt	MSt/MSE
Block	r-1	SSb	MSb	MSb/MSE
Residual	(t-1)(r-1)	SSE	MSE	
Total	tr-1	SST		

Randomization for 4 treatments and 5 blocks, which is 20 experimental units can be done like below.

```
> sample(1:4,size=4,replace=FALSE)
[1] 2 1 4 3
> sample(1:4,size=4,replace=FALSE)
[1] 4 3 1 2
> sample(1:4,size=4,replace=FALSE)
[1] 2 1 4 3
> sample(1:4,size=4,replace=FALSE)
[1] 3 4 2 1
> sample(1:4,size=4,replace=FALSE)
[1] 4 2 3 1
>
```

So the first experimental unit in Strain 1 is filled by treatment T2, the second experimental unit in Strain 1 is filled by treatment T1, and soon until twenty experimental units, as below.

Table. Randomization

Strain	Treatments			
1	T2	T1	T4	T3
2	T4	T3	T1	T2
3	T2	T1	T4	T3
4	T3	T4	T2	T1
5	T4	T2	T3	T1

After getting research data for easier analysis we arrange table like below.

Table. Data arrangement for data analysis

Strain	Treatments			
	T1	T2	T3	T4
1	T1	T2	T3	T4
2	T1	T2	T3	T4
3	T1	T2	T3	T4
4	T1	T2	T3	T4
5	T1	T2	T3	T4

7.3 Example of RCBD

Example 1. The result of the effect of prebiotic addition in ration on broiler performance (body weight gain) treated with four different ration applied to five broiler strains is presented in table below.

Table. Body weight gain of broiler treated with different prebiotic addition in ration

Strain	Treatments				Mean strain	Total strain
	T1	T2	T3	T4		
1	0.765	1.311	1.452	1.630	1.290	5.158
2	1.015	1.303	1.846	1.505	1.418	5.670
3	1.276	1.597	1.264	1.779	1.479	5.916
4	0.984	0.645	1.399	1.454	1.120	4.482
5	0.856	1.148	1.154	1.443	1.150	4.602
Mean treatment	0.979	1.201	1.423	1.562	Total = 25.828	
Total treatment	4.895	6.006	7.115	7.812		

Computation for ANOVA is like below.

$$\text{CF} = \frac{\left(\sum_i \sum_j yij\right)^2}{t.r} = \frac{25.828^2}{4.5} = 33.355$$

$$\text{SST} = \sum_i \sum_j yij^2 \; - CF = (0.765^2 + \ldots + 1.443^2) - \text{CF} = 1.983$$

$$\text{SSt} = \sum_i \frac{\left(\sum_j yij\right)^2}{r} - CF = \frac{(4.895^2 + \ldots + 7.812^2)}{5} - \text{CF} = 0.982$$

$$\text{SSb} = \sum_j \frac{(\sum_i yij)^2}{t} - CF = \frac{(5.158^2 + \ldots + 4.602^2)}{4} - \text{CF} = 0.401$$

$$SSE = SST - SSt - SSb = 1.983 - 0.982 - 0.621 = 0.600$$

dfT = t.r-1=4*5-1 = 19

dft = t-1 = 4-1 = 3

dfb = r-1 = 5-1 = 4

dfe = (t-1)(r-1) = 3*4 = 12

Manually using R:

```
> CF=(sum(data[,3])^2)/(4*5)
> CF
[1] 33.35455
> SST=(sum(data[,3]^2))-CF
> SST
[1] 1.982686
> SSt=(((sum(data[1:5,3])^2)+(sum(data[6:10,3])^2)+
+ (sum(data[11:15,3])^2)+(sum(data[16:20,3])^2)
+ )/5)-CF
> SSt
[1] 0.982075
> newdata <- data[order(data$Strain),]
> newdata
   Treatment Strain BodyWeightGain
1         T1     S1      0.7650848
6         T2     S1      1.3113046
11        T3     S1      1.4521396
16        T4     S1      1.6298013
2         T1     S2      1.0149890
7         T2     S2      1.3034288
12        T3     S2      1.8462775
17        T4     S2      1.5054732
3         T1     S3      1.2758507
8         T2     S3      1.5974956
13        T3     S3      1.2638854
18        T4     S3      1.7789886
4         T1     S4      0.9837027
9         T2     S4      0.6453479
14        T3     S4      1.3987220
19        T4     S4      1.4539894
5         T1     S5      0.8557105
10        T2     S5      1.1484184
15        T3     S5      1.1541131
20        T4     S5      1.4433824
>
> SSb=(((sum(newdata[1:4,3])^2)+(sum(newdata[5:8,3]
```

```
+ )^2)+(sum(newdata[9:12,3])^2)+(sum(newdata[13:16,
+ 3])^2)+(sum(newdata[17:20,3])^2))/4)-CF
> SSb
[1] 0.4009439
> SSE=SST-SSt-SSb
> SSE
[1] 0.5996673
> MSt=SSt/3
> MSt
[1] 0.3273583
> MSb=SSb/4
> MSb
[1] 0.100236
> MSE=SSE/(3*4)
> MSE
[1] 0.04997228
>
> Fstatistic_t=MSt/MSE
> Fstatistic_t
[1] 6.550799
> Fstatistic_b=MSb/MSE
> Fstatistic_b
[1] 2.005832
>
> qf(0.95,3,12)
[1] 3.490295
> qf(0.99,3,12)
[1] 5.952545
> qf(0.95,4,12)
[1] 3.259167
> qf(0.99,4,12)
[1] 5.411951
>
```

Based on computation above, ANOVA table can be arranged as below

Table. ANOVA

Source of variance	df	SS	MS	F statistic
SSt	3	0.982	0.327	6.551**
SSb	4	0.401	0.100	$2.006^{n.s}$
SSE	12	0.600	0.050	
SST	19	1.983		

In R

```
> data=read.csv("rcbd1.csv", header=T)
> data
   Treatment Strain BodyWeightGain
```

```
1          T1     S1      0.7650848
2          T1     S2      1.0149890
3          T1     S3      1.2758507
4          T1     S4      0.9837027
5          T1     S5      0.8557105
6          T2     S1      1.3113046
7          T2     S2      1.3034288
8          T2     S3      1.5974956
9          T2     S4      0.6453479
10         T2     S5      1.1484184
11         T3     S1      1.4521396
12         T3     S2      1.8462775
13         T3     S3      1.2638854
14         T3     S4      1.3987220
15         T3     S5      1.1541131
16         T4     S1      1.6298013
17         T4     S2      1.5054732
18         T4     S3      1.7789886
19         T4     S4      1.4539894
20         T4     S5      1.4433824
> modelRCBD=aov(BodyWeightGain~Treatment+Strain,
+ data=data)
> summary(modelRCBD)
            Df Sum Sq Mean Sq F value  Pr(>F)
Treatment    3 0.9821  0.3274   6.551 0.00715 **
Strain       4 0.4009  0.1002   2.006 0.15772
Residuals   12 0.5997  0.0500
---
Signif.codes:0‘***’0.001‘**’0.01‘*’0.05 ‘.’ 0.1 ‘ ’ 1
> library(agricolae)
> duncan.test(modelRCBD, "Treatment", alpha=0.05,
+ console=T)

Study: modelRCBD ~ "Treatment"

Duncan's new multiple range test
for BodyWeightGain

Mean Square Error:  0.04997228

Treatment,  means

   BodyWeightGain       std r       Min      Max
T1      0.9790676 0.1939057 5 0.7650848 1.275851
T2      1.2011990 0.3504930 5 0.6453479 1.597496
T3      1.4230275 0.2637328 5 1.1541131 1.846277
T4      1.5623270 0.1419615 5 1.4433824 1.778989

Alpha: 0.05 ; DF Error: 12
```

```
Critical Range
        2         3         4
0.3080452 0.3224349 0.3311535

Means with the same letter are not significantly
  different.

   BodyWeightGain groups
T4      1.5623270      a
T3      1.4230275     ab
T2      1.2011990     bc
T1      0.9790676      c
>
```

If we use ExpDes package we will get assumption, ANOVA and further test together, like below.

```
> library(ExpDes)

Attaching package: 'ExpDes'

The following objects are masked from 'package:agricolae':

    lastC, order.group, tapply.stat

The following object is masked from 'package:stats':

    ccf

Warning message:
package 'ExpDes' was built under R version 3.5.2
> rbd(data$Treatment, data$Strain, data$BodyWeightGain,
+  quali = TRUE, mcomp='duncan', hvar='oneillmathews',
+  sigT = 0.05, sigF = 0.05)
------------------------------------------------------------------
Analysis of Variance Table
------------------------------------------------------------------
           DF      SS      MS     Fc   Pr>Fc
Treatament  3 0.98208 0.32736 6.5508 0.00715
Block       4 0.40094 0.10024 2.0058 0.15772
Residuals  12 0.59967 0.04997
Total      19 1.98269
------------------------------------------------------------------
```

```
CV = 17.31 %

------------------------------------------------------------------
Shapiro-Wilk normality test
p-value:  0.3119083
According to Shapiro-Wilk normality test at 5% of significance,
residuals can be considered normal.
------------------------------------------------------------------

------------------------------------------------------------------
Homogeneity of variances test
p-value:  0.4910947
According to the test of oneillmathews at 5% of significance,
the variances can be considered homocedastic.
------------------------------------------------------------------

Duncan's test
------------------------------------------------------------------
Groups  Treatments  Means
a        T4          1.562327
ab       T3          1.423028
 bc      T2          1.201199
  c      T1          0.9790676
------------------------------------------------------------------
```

Example 2. The following RCBD example is hypothetical data about the effect of three types of ration given to two breeds of sheep during pregnancy on birth weights of their lambs. Breed or type of sheep here is as a group or block so it uses a randomized complete block design (RCBD).

Table. Birth weight of two breed of lambs treated with three different types of ration during dam pregnancy.

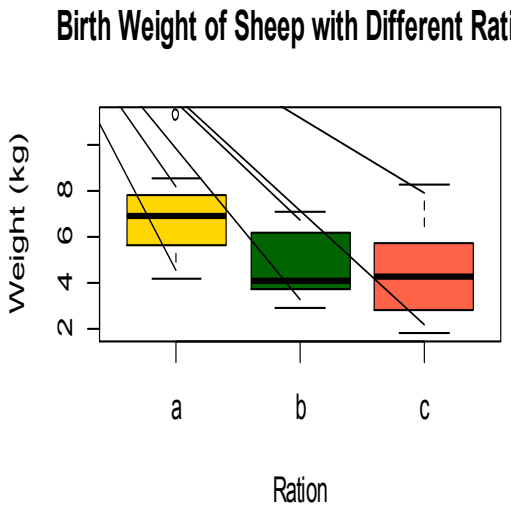
Obervation	Ration					
	1		2		3	
	Merino	Dorset	Merino	Dorset	Merino	Dorset
1	5.667	6.998	3.989	5.054	2.850	5.654
2	6.819	7.106	3.640	7.044	1.898	6.707
3	4.179	7.760	3.899	3.724	1.878	4.505
4	6.038	11.199	4.177	6.828	3.990	8.226
5	4.784	8.488	2.949	6.180	3.527	5.751

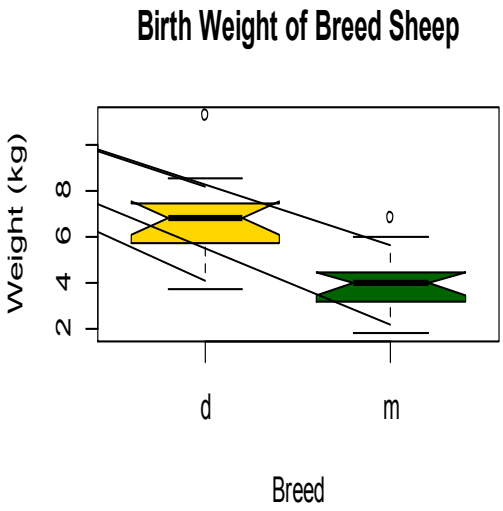
First we write the data directly in the R editor as below.

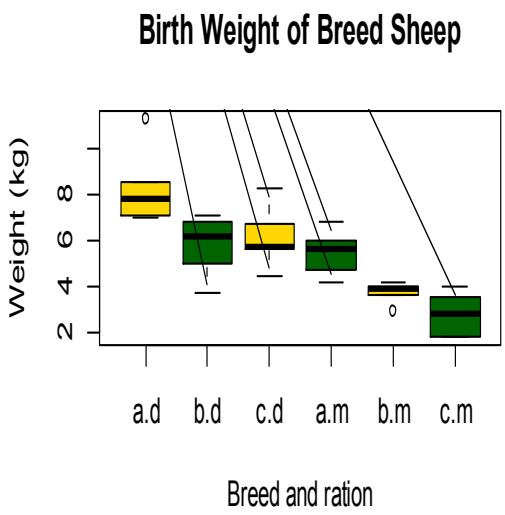
```
> ration <- c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,
+  2,2,3,3,3,3,3,3,3,3,3,3,3)
> breed <- c(1,1,1,1,1,2,2,2,2,2,1,1,1,1,1,2,2,2,
+  2,2,1,1,1,1,1,2,2,2,2,2)
> bw <- c(5.667,6.819,4.179,6.038,4.784,6.998,7.106,
+  7.760,11.199,8.488,3.989,3.640,3.899,4.177,
+  2.949,5.054,7.044,3.724,6.828,6.180,2.850,
+  1.898,1.878,3.990,3.527,5.654,6.710,4.505,
+   8.226,5.751)
>
> ration <- as.factor(ration)
> breed <- as.factor(breed)
> data <- data.frame(ration, breed, bw)
> fit <- aov(bw~ration+breed)
> anova(fit)

Analysis of Variance Table

Response: bw
          Df Sum Sq Mean Sq F value    Pr(>F)
ration     2 34.972  17.486  11.906 0.0002134 ***
breed      1 55.878  55.878  38.047 1.598e-06 ***
Residuals 26 38.185   1.469
---Signif.codes:0‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘’1

> TukeyHSD(fit,"ration")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = bw ~ ration + breed)

$`ration`
       diff       lwr        upr     p adj
2-1 -2.1554 -3.502128 -0.8086716 0.0013950
3-1 -2.4049 -3.751628 -1.0581716 0.0004228
3-2 -0.2495 -1.596228  1.0972284 0.8902374

> #Or by reading excel file
> data <- read.csv('rcbd.csv', header=T)
> data
   ration breed        bw
1       a     m  5.667092
2       a     m  6.818530
3       a     m  4.179033
4       a     m  6.038148
5       a     m  4.784378
```

```
6       a     d  6.998002
7       a     d  7.106330
8       a     d  7.759698
9       a     d 11.199271
10      a     d  8.487953
11      b     m  3.988762
12      b     m  3.639712
13      b     m  3.899011
14      b     m  4.176846
15      b     m  2.949012
16      b     d  5.053727
17      b     d  7.043697
18      b     d  3.724377
19      b     d  6.828300
20      b     d  6.180444
21      c     m  2.849605
22      c     m  1.898496
23      c     m  1.878447
24      c     m  3.989679
25      c     m  3.527487
26      c     d  5.654328
27      c     d  6.706697
28      c     d  4.504877
29      c     d  8.225741
30      c     d  5.751020
> fit2 <- aov(bw ~ ration + breed, data=data)
> anova(fit2)

Analysis of Variance Table

Response: bw
          Df Sum Sq Mean Sq F value    Pr(>F)
ration     2 34.978  17.489  11.912 0.0002128 ***
breed      1 55.870  55.870  38.054 1.595e-06 ***
Residuals 26 38.173   1.468
---Signif.codes:0‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘ ’1

> TukeyHSD(fit2, "ration")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = bw ~ ration + breed, data = data)

$`ration`
          diff       lwr        upr     p adj
b-a -2.1554547 -3.501981 -0.8089285 0.0013925
c-a -2.4052060 -3.751732 -1.0586798 0.0004215
c-b -0.2497513 -1.596277  1.0967748 0.8899988
```

```
> par(mfrow=c(2,2))
> boxplot(bw~ration, notch=FALSE,col=c("gold",
+ "darkgreen","tomato"),
+ main="Birth Weight of Sheep with Different Ration",
+ xlab="Ration", ylab="Weight (kg)", data=data)
>
> boxplot(bw~breed, data=data, notch=TRUE,
+   col=c("gold","darkgreen"),
+  main="Birth Weight of Breed Sheep", xlab="Breed",
+   ylab="Weight (kg)")
>

> boxplot(bw~ration*breed, data=data, notch=FALSE,
+   col=c("gold","darkgreen"),
+   main="Birth Weight of Breed Sheep",
+   xlab="Breed and ration", ylab="Weight (kg)")
>
> boxplot(bw~ration*breed, range = 1.5, width = NULL,
+   varwidth = FALSE, notch = FALSE, outline = TRUE,
+   names, plot = TRUE, border = par("fg"),
+  col = c("turquoise","tomato","orange"), log = "",
+   pars = list(boxwex = 0.8, staplewex = 0.5,
+   outwex = 0.5), horizontal = FALSE, add = FALSE,
+   at = NULL, xlab="Breed and ration",
+   ylab="Weight (kg)",
+   main="Birth Weight of Breed Sheep", data=data)
>
```







The results as shown above indicated that the ration affected the birth weights of lambs ($P<0.05$), and also that we have correctly classified breeds as blocks ($P<0.05$) meaning that using RCBD has been already correct or appropriate . The results of further tests showed that only rations 2 and 3 which was not significantly different, while the others (rations 1 and 2, and rations 1 and 3) were significantly different ($P<0.05$).

Tukey test above resulted in no notation yet, even though it could actually be made manually. Therefore, we use the Agricolae package as below to find out the notation directly.

```
> library(agricolae)
> HSD.test(fit2, "ration",alpha=0.05, console=TRUE)

Study: fit2 ~ "ration"

HSD Test for bw

Mean Square Error:  1.468196

ration,  means

        bw      std  r      Min       Max
a 6.903844 1.998484 10 4.179033 11.199271
b 4.748389 1.448338 10 2.949012  7.043697
c 4.498638 2.087491 10 1.878447  8.225741

Alpha: 0.05 ; DF Error: 26
Critical Value of Studentized Range: 3.514171

Minimun Significant Difference: 1.346526

Treatments with the same letter are not significantly
  different.

        Bw      groups
a  6.903844         a
b  4.748389         b
c  4.498638         b
>
```

The conclusion is that different rations (A, B and C) affect the birth weight of lambs. Judging from Tukey's advanced test, it turned out that ration A was significantly different (P <0.05) from rations B and C, but the ration B and C were not significantly different (P> 0.05) in influencing the birth weight of the lamb.

We can use ExpDes package as the following.

```
> library(ExpDes)
> rbd(data$ration, data$breed, data$bw, quali = TRUE, mcomp='tukey',
+ hvar='oneillmathews', sigT = 0.05, sigF = 0.05)
------------------------------------------------------------------------
Analysis of Variance Table
------------------------------------------------------------------------
           DF      SS     MS     Fc     Pr>Fc
Treatament  2  34.978 17.489 11.912 2.128e-04
Block       1  55.870 55.870 38.054 1.595e-06
Residuals  26  38.173  1.468
Total      29 129.021
```

```
------------------------------------------------------------------------
CV = 22.51 %

------------------------------------------------------------------------
Shapiro-Wilk normality test
p-value:  0.4088082
According to Shapiro-Wilk normality test at 5% of significance, residuals can
be considered normal.
------------------------------------------------------------------------

------------------------------------------------------------------------
Homogeneity of variances test
p-value:  1
According to the test of oneillmathews at 5% of significance, the variances can
be considered homocedastic.
------------------------------------------------------------------------

Tukey's test
------------------------------------------------------------------------
Groups Treatments Means
a        a       6.903844
 b       b       4.748389
 b       c       4.498638
------------------------------------------------------------------------
>
```

7.4 Randomized Complete Block Design with Two or More Experimental Units per Treatment and Block

In previous RCBD there is only one experimental unit per treatment x block combination. For repeated block and treatment in RCBD mean that there will be more than one experimental unit per treatment x block combination. For example, consider we have five blocks, four treatments, and ten animals per block, that is, two animals per block x treatment combination. In this design treatments are randomly allocated to 5x2 experimental units in each block. Each treatment is assigned to 2 experimental units within each block, like below.

```
> sample(rep(1:4,size=4,each=2),replace=FALSE)
[1] 3 2 4 4 1 2 3 1
> sample(rep(1:4,size=4,each=2),replace=FALSE)
[1] 1 4 1 3 3 2 2 4
> sample(rep(1:4,size=4,each=2),replace=FALSE)
[1] 1 3 4 3 2 2 1 4
> sample(rep(1:4,size=4,each=2),replace=FALSE)
```

```
[1] 2 4 1 4 2 1 3 3
> sample(rep(1:4,size=4,each=2),replace=FALSE)
[1] 1 2 3 4 3 4 1 2
>
```

	Strain or block				
	1	2	3	4	5
No. of animal & treatment	No.1(T3)	No.9(T1)	No.17(T1)	No.25(T2)	No.33(T1)
	No.2(T2)	No.10(T4)	No.18(T3)	No.26(T4)	No.34(T2)
	No.3(T4)	No.11(T1)	No.19(T4)	No.27(T1)	No.35(T3)
	No.4(T4)	No.12(T3)	No.20(T3)	No.28(T4)	No.36(T4)
	No.5(T1)	No.13(T3)	No.21(T2)	No.29(T2)	No.37(T3)
	No.6(T2)	No.14(T2)	No.22(T2)	No.30(T1)	No.38(T4)
	No.7(T3)	No.15(T2)	No.23(T1)	No.31(T3)	No.39(T1)
	No.8(T1)	No.16(T4)	No.24(T4)	No.32(T3)	No.40(T2)

For companion in ANOVA the table can be arranged as follows.

	Strain or block				
	1	2	3	4	5
T1	y111	y121	y131	y141	y151
	y112	y122	y132	y142	y152
T2	y211	y221	y231	y241	y251
	y212	y222	y232	y242	y252
T3	y311	y321	y331	y341	y351
	y312	y322	y332	y342	y352
T4	y411	y421	y431	y441	y451
	y412	y422	y432	y442	y452

The linear model for this design is:

$$yijk = \mu + \tau i + \beta j + \tau\beta ij + \varepsilon ijk \qquad i = 1,...,t; \quad j = 1,...,b; \quad k = 1,\ldots,n$$

where:

$yijk$ = observation k in treatment i and block j

μ = the overall mean

τi = the effect of treatment i

βj = the effect of block j

$\tau\beta ij$ = the interaction effect of treatment i and block j.

εijk = random error

t = number of treatments
b = number of blocks

n = number of observations in each treatment *x* block combination.

Total sum square variation for this design will be as follows.

$$\text{SST} = \text{SSt} + \text{SSb} + \text{SSt} * \text{SSb} + \text{SSE},$$

with corresponding degrees of freedom as follows.

$$(tbn - 1) = (t - 1) + (b - 1) + (t - 1)(b - 1) + tb(n - 1)$$

Where:

$$\text{SST} = \sum_{i=1}^{t}\sum_{j=1}^{b}\sum_{i=1}^{n}(yijk - \bar{y}...)^2$$

$$\text{SSt} = \sum_{i=1}^{t}\sum_{j=1}^{b}\sum_{i=1}^{n}(\bar{y}i.. - \bar{y}...)^2$$

$$\text{SSb} = \sum_{i=1}^{t}\sum_{j=1}^{b}\sum_{i=1}^{n}(\bar{y}.j. - \bar{y}...)^2$$

$$\text{SStb} = n\sum_{i=1}^{t}\sum_{i=1}^{b}(\bar{y}ij. - \bar{y}...)^2 - SSt - SSb$$

$$\text{SSE} = \sum_{i=1}^{t}\sum_{j=1}^{b}\sum_{i=1}^{n}(yijk - \bar{y}ij.)^2$$

Sum squares above can be computed as below

$$\text{CF} = \frac{\left(\sum_i \sum_j \sum_k yijk\right)^2}{t.r.n}$$

$$\text{SST} = \sum_i \sum_j \sum_k yijk^2 - CF$$

$$\text{SSt} = \sum_i \frac{\left(\sum_j \sum_k yijk\right)^2}{n.r} - CF$$

$$\text{SSb} = \sum_j \frac{(\sum_i \sum_k yijk)^2}{n.t} - CF$$

$$\text{SStb} = \sum_i \sum_j \frac{(\sum_k yijk)^2}{n} - SSt - SSb - CF$$

$$SSE = SST - SSt - SSb - SStb$$

Mean squares (MS) of each variation can be calculated as below.

MSt = SSt/dft = SSt/(t-1)

MSb = SSb/dfb = SSb/(r-1)

MStb = SStb/dftb = SStb/(t-1)(r-1)

MSE = SSE/dfe = SSE/(tr(n-1))

Example 1. The result of the effect of prebiotic addition in ration on broiler performance (body weight gain) treated with four different ration applied to five broiler strains is presented in table below.

Strain	Treatments				Mean Strain	Total Strain
	T1	T2	T3	T4		
1	0.765	1.101	1.252	1.630	1.189	9.515
	0.876	1.112	1.224	1.555		
2	1.015	1.303	1.446	1.505	1.341	10.728
	1.124	1.264	1.416	1.655		
3	1.276	1.597	1.464	1.779	1.512	12.097
	1.213	1.444	1.544	1.780		
4	1.284	1.345	1.599	1.854	1.547	12.373
	1.322	1.422	1.658	1.889		
5	1.456	1.648	1.954	1.943	1.776	14.208
	1.534	1.736	1.956	1.981		
Mean treatment	1.186	1.397	1.551	1.757		
Total treatment	11.865	13.972	15.513	17.571		58.921

Sum squares above can be computed as below

$$\text{CF} = \frac{\left(\sum_i \sum_j \sum_k yijk\right)^2}{t.r.n} = \frac{58.923^2}{4.5.2} = 86.792$$

$$\text{SST} = \sum_i \sum_j \sum_k yijk^2 - CF = 0.765^2 + \ldots + 1.981^2 - CF$$

$$= 3.520$$

$$\text{SSt} = \sum_i \frac{\left(\sum_j \sum_k yijk\right)^2}{n.r} - CF = \frac{11.864^2 + \ldots + 17.572^2}{2.5} - CF$$

$$= 1.747$$

$$\text{SSb} = \sum_j \frac{(\sum_i \sum_k yijk)^2}{n.t} - CF = \frac{9.515^2 + \ldots + 14.209^2}{2.4} - CF$$

$$= 1.573$$

$$\text{SStb} = \sum_i \sum_j \frac{(\sum_k yijk)^2}{n} - SSt - SSb - CF$$

$$= \frac{(0.765 + 0.876)^2 + \ldots + (1.943 + 1.981)^2}{2} - SSt$$

$$- SSb - CF = 0.142$$

$$SSE = SST - SSt - SSb - SStb = 0.058$$

MSt = SSt/dft =1.747/3 = 0.582

MSb = SSb/dfb = 1.573/4 = 0.393

MStb = SStb/dftb = 0.142/12 = 0.012

MSE = SSE/dfe = 0.058/20 = 0.003

Fstatistic_t = MSt/MSE = 0.582/0.003 = 199.419

Fstatistic_b = MSb/MSE = 0.393/0.003 = 134.691

F table for alpha = 0.05: Treatment (dft, dfe), block (dfb, dfe) and interaction between treatment and block (dftb, dfe) can be computed using R as follows.

```
> qf(0.95, 3, 20)
[1] 3.098391
> qf(0.95, 4, 20)
[1] 2.866081
> qf(0.95, 12, 20)
[1] 2.277581
>
```

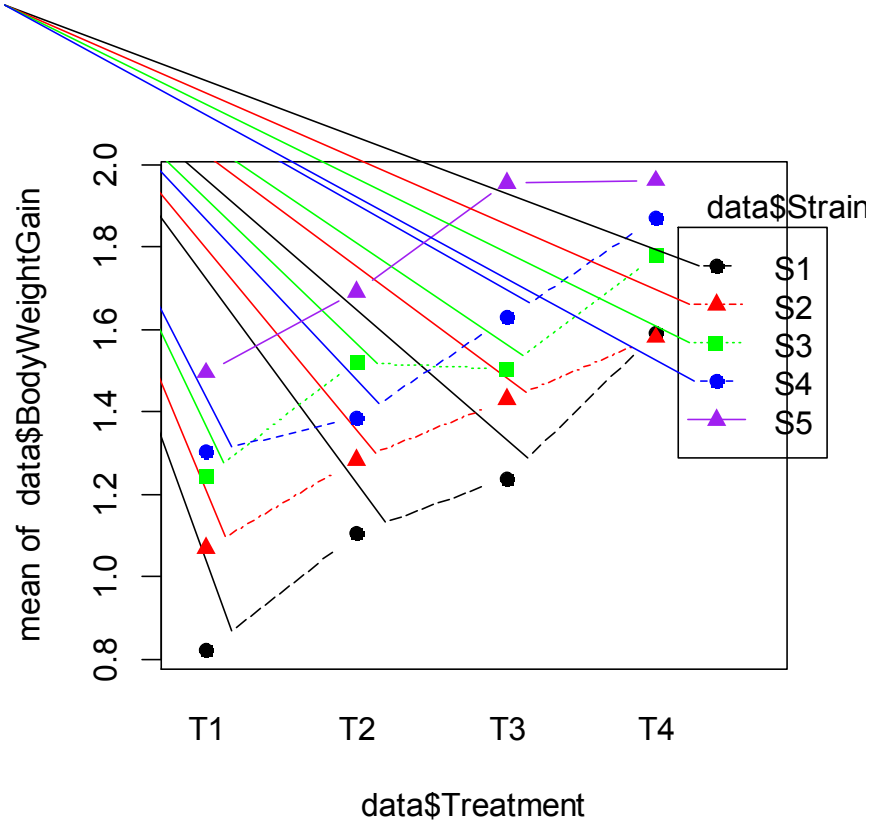
In R

```
> data=read.csv("rcbd3.csv", header=T)
> head(data)
  Treatment Strain BodyWeightGain
1        T1     S1           0.765
2        T1     S1           0.876
```

```
3         T1     S2           1.015
4         T1     S2           1.124
5         T1     S3           1.276
6         T1     S3           1.213
> tail(data)
   Treatment Strain BodyWeightGain
35        T4     S3           1.779
36        T4     S3           1.780
37        T4     S4           1.854
38        T4     S4           1.889
39        T4     S5           1.943
40        T4     S5           1.981
>
> fit=aov(BodyWeightGain~Treatment*Strain, data=data)
> summary(fit)
               Df Sum Sq Mean Sq F value   Pr(>F)
Treatment       3 1.7467  0.5822 199.419 4.57e-15 ***
Strain          4 1.5730  0.3933 134.691 3.67e-14 ***
Treatment:Strain 12 0.1416  0.0118   4.042  0.00292 **
Residuals      20 0.0584  0.0029
---
Signif.codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’ 0.1 ‘ ’ 1
>
```

Based on ANOVA above, it can be concluded that treatment, block or strain and interaction between treatment and block significantly ($P<0.05$) affected body weight gain of broiler. In addition that there is increasing of body weight gain with treatment and strain of broiler, as describe by figure below.

```
> interaction.plot(x.factor = data$Treatment,
+      trace.factor = data$Strain,
+      response     = data$BodyWeightGain,
+      fun = mean,
+      type="b",
+      col=c("black","red","green","blue","purple"),
+      pch=c(19, 17, 15),
+      fixed=TRUE,
+      leg.bty = "o")
>
```


mean of data$BodyWeightGain
0.8
1.0
1.2
1.4
1.6
1.8
2.0
T1
T2
T3
T4
data$Treatment
data$Strain
S1
S2
S3
S4
S5

VIII. LATIN SQUARE DESIGN

8.1 Introduction

Latin square design is experimental design that control two sources of error variation simultaneously related to rows and columns, it is also known as double blocking design. In this design number of rows and columns are the same as the number of treatment levels. In this design *n x n* table is filled with *n* different symbols in such a way that each symbol occurs exactly once in each row and exactly once in each column. It is also assumed that there is no interaction between rows and columns and the treatment under study. For example, a research is conducted to investigate the effect of milk replacer on growth rate of calf of beef cattle. There are two other factors influencing the growth rate, but they are not interesting for researcher to be investigated, so the effect of these two factors are localized by blocking them. These two factors are parity and birth weight. They grouped parity and birth weight into homogeneous blocks so that within each block the experimental units are homogeneous.

8.2 Lay-out and Randomization

Randomization for Latin square design can be first randomly permute the columns, then randomly permute the rows, and finally assign the treatments to the Latin letters in a random way. Example of randomization for Latin square is described as follows.

Latin square 3 x 3

B	C	A
A	B	C
C	A	B

C	B	A
B	A	C
A	C	B

Latin square 4 x 4

C	A	D	B
D	C	B	A
B	D	A	C
A	B	C	D

B	C	A	D
D	B	C	A
A	D	B	C
C	A	D	B

Take for example the first Latin square 4x4 to make randomization.

C	A	D	B
D	C	B	A
B	D	A	C
A	B	C	D

```
> column=sample(1:4,size=4,replace=FALSE)
> column
[1] 3 4 1 2
> row=sample(1:4,size=4,replace=FALSE)
> row
[1] 2 3 1 4
>
```

Based on column randomization, for column 3 is converted to column 1, column 4 is converted to column 2, column 1 is converted to column 3 and column 2 is converted to column 4, as like below.

D	B	C	A
B	A	D	C
A	C	B	D
C	D	A	B

Based on row randomization, row 2 convert to row 1, row 3 convert to row 2, row 1 convert to row 3 and row 4 remain in row 4, as like below.

B	A	D	C
A	C	B	D
D	B	C	A
C	D	A	B

Then the final structure will be like below.

T2	T1	T4	T3
T1	T3	T2	T4
T4	T2	T3	T1
T3	T4	T1	T2

For example a research is conducted to investigate the effect of four milk replacer on growth rate of calf of beef cattle. There are four parity and four group of birth weight: 1(25-28), 2(29-32), 3(33-36), 4(37-40) kg. Structure of the data is like below.

Row (Parity)	Column (group of birth weight)			
	1	2	3	4
1	T2	T1	T4	T3
2	T1	T3	T2	T4
3	T4	T2	T3	T1
4	T3	T4	T1	T2

For easier analysis, after getting research data we can tabulate the data like table below.

Row (Parity)	Column (group of birth weight)			
	1	2	3	4
1	y11(T2)	y12(T1)	y13(T4)	y14(T3)
2	y21(T1)	y22(T3)	y23(T2)	y24(T4)
3	y31(T4)	y32(T2)	y33(T3)	y34(T1)
4	y41(T3)	y42(T4)	y43(T1)	y44(T2)

where y11(T2) is observation in row 1, column 1 and T2; y21(T1) is observation in row 2, column 1 and T1; and soon. The linear model for Latin square is:

$$yij(k) = \mu + Ri + Cj + \tau(k) + \varepsilon ij(k) \qquad i,j,k = 1,...,r$$

where:

$yij(k)$ = observation in row i col j and treatment (k)

μ = the overall mean

Ri = the effect of row i

Cj = the effect of column j

$\tau(k)$ = the fixed effect of treatment k

$\varepsilon ij(k)$ = random error

r = the number of treatments, rows and columns

Sum of squares total is sum of squares of columns, rows, treatments and residual:

$$\text{SST} = \text{SSR} + \text{SSC} + \text{SSt} + \text{SSE}$$

The Degrees of freedom of corresponding sum square above are:

$$r^2 - 1 = (r-1) + (r-1) + (r-1) + (r-1)(r-2)$$

The sums of squares above can be formulated as below.

where

$$\text{SST} = \sum_i \sum_j (yij(k) - \bar{y}..)^2$$

$$\text{SSR} = \text{r} \sum_i (\bar{y}i. - \bar{y}..)^2$$

$$\text{SSC} = \text{r} \sum_j (\bar{y}.j - \bar{y}..)^2$$

$$\text{SSt} = \text{r} \sum_k (\bar{y}k - \bar{y}..)^2$$

$$\text{SSE} = \sum_i \sum_j (\bar{y}ij - \bar{y}i. - \bar{y}.j - \bar{y}k + 2\bar{y}..)^2$$

Sums of squares above can be calculated using computation below.

$$\text{CF} = \frac{\left(\sum_i \sum_j yijk\right)^2}{r^2}$$

$$\text{SST} = \sum_i \sum_j yijk^2 - CF$$

$$\text{SSR} = \sum_i \frac{\left(\sum_j yijk\right)^2}{r} - CF$$

$$\text{SSC} = \sum_j \frac{(\sum_i yijk)^2}{r} - CF$$

$$\text{SSt} = \sum_k \frac{\left(\sum_i \sum_j yijk\right)^2}{r} - CF$$

$$SSE = SST - SSR - SSC - SSt$$

Mean square (MS) can be calculated as below.

MSt = SSt/dft = SSt/(r-1)

MSR = SSR/dfr =SSR/(r-1)

MSC = SSC/dfc = SSC/(r-1)

MSE = SSE/dfe = SSE/(r-1)(r-2)

The null and alternative hypotheses are:

H_0: $\tau 1 = \tau 2 = \ldots = \tau r$, treatment effects are the same

H_1: $\tau i \neq \tau r$, at least the effects of one pair of treatment is different

F statistic = MSt/MSE compared to F table with (r-1) and (r-1)(r-2) degrees of freedom for critical value. For an α level of significance H_0 is rejected if *F statistic* > $F_{\alpha,(r-1),(r-1)(r-2)}$.

The results ANOVA can be summarized in table below.

Source of variation	df	SS	MS=SS/df	F
Row	r-1	SSR	MSR	MSR/MSE
Column	r-1	SSC	MSC	MSC/MSE
Treatment	r-1	SSt	MSt	MSt/MSE
Residual	(r-1)(r-2)	SSE	MSE	
Total	r^2-1	SST		

8.3 Example of Latin Square Design

For example a research is conducted to investigate the effect of four milk replacer on growth rate of calf (2-4 month of age) of beef cattle. There are four parity and four group of birth weight: 1(25-28), 2(29-32), 3(33-36), 4(37-40) kg. Structure of the data is like below.

Table. Average growth rate (g) from 2-4 month of age given four different milk replacer with different parity and birth weight group.

Row (Parity)	Column (group of birth weight)				Total Row
	1	2	3	4	
1	222(T2)	198(T1)	243(T4)	234(T3)	897
2	200(T1)	232(T3)	234(T2)	248(T4)	914
3	238(T4)	232(T2)	233(T3)	220(T1)	923
4	241(T3)	242(T4)	220(T1)	244(T2)	947
Total Column	901	904	930	946	
Total Treatment	T1=838	T2=932	T3=940	T4=971	Total=3681

$$\text{CF} = \frac{\left(\sum_i \sum_j yijk\right)^2}{r^2} = \frac{(3681)^2}{4^2} = 846{,}860.0625$$

$$\text{SST} = \sum_i \sum_j yijk^2 \; - CF = (222^2 + \ldots + 244^2) - CF$$

$$= 3254.9375$$

$$\text{SSR} = \sum_i \frac{\left(\sum_j yijk\right)^2}{r} \; - \; CF = \frac{(897^2 + \ldots + 947^2)}{4} - CF$$

$$= 325.6875$$

$$\text{SSC} = \sum_j \frac{(\sum_i yijk)^2}{r} \; - \; CF = \frac{(901^2 + \ldots + 946^2)}{4} - CF$$

$$= 348.1875$$

$$\text{SSt} = \sum_k \frac{\left(\sum_i \sum_j yijk\right)^2}{r} \; - \; CF = \frac{(838^2 + \ldots + 971^2)}{4} - CF$$

$$= 2{,}467.1875$$

$$SSE = SST \; - \; SSR \; - \; SSC - \; SSt = 113.875$$

Mean square (MS) can be calculated as below.

MSt = SSt/dft = SSt/(r-1) = 2,467.1875/3 = 822.396

MSR = SSR/dfr =SSR/(r-1) = 325.6875/3 = 108.563

MSC = SSC/dfc = SSC/(r-1) = 348.1875/3 = 116.0625

MSE = SSE/dfe = SSE/(r-1)(r-2) = 113.875/6 = 18.979

Fstatistic_t = MSt/MSE = 822.396/18.979 = 43.332

Fstatistic_R = MSt/MSE = 108.563/18.979 = 5.720

Fstatistic_C = MSt/MSE = 116.0625/18.979 = 6.115

```
> qf(0.95,3,6)
[1] 4.757063
>
```

Source of variation	df	SS	MS=SS/df	F
Row	3	325.6875	108.563	5.720*
Column	3	348.1875	116.0625	6.115*
Treatment	3	2467.1875	822.396	43.332**
Residual	6	113.875	18.979	
Total	15	3254.9375		

In R:

```
> data=read.csv("latin.csv",header=T)
> data
   Treatment Row Column GrowthRate
1         T2   1      1         222
2         T1   2      1         200
3         T4   3      1         238
4         T3   4      1         241
5         T1   1      2         198
6         T3   2      2         232
7         T2   3      2         232
8         T4   4      2         242
9         T4   1      3         243
10        T2   2      3         234
11        T3   3      3         233
12        T1   4      3         220
13        T3   1      4         234
14        T4   2      4         248
15        T1   3      4         220
16        T2   4      4         244
> str(data)
'data.frame':   16 obs. of  4 variables:
 $ Treatment : Factor w/ 4 levels "T1","T2","T3",..: 2 1 4 3  ...
 $ Row       : int  1 2 3 4 1 2 3 4 1 2 ...
 $ Column    : int  1 1 1 1 2 2 2 2 3 3 ...
 $ GrowthRate: int  222 200 238 241 198 232 232 242 243 234 ...
> data$Row=as.factor(data$Row)
> data$Column=as.factor(data$Column)
        >
   modelLatin=aov(GrowthRate~Row+Column+Treatment,
   data=data)
> summary(modelLatin)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)
Row          3  325.7   108.6   5.720 0.034123 *
Column       3  348.2   116.1   6.115 0.029551 *
Treatment    3 2467.2   822.4  43.332 0.000185 ***
Residuals    6  113.9    19.0
---
Signif.codes: 0 ‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘ ’1
> library(agricolae)
> duncan.test(modelLatin, "Treatment", alpha=0.05,
  console=T)

Study: modelLatin ~ "Treatment"

Duncan's new multiple range test
for GrowthRate

Mean Square Error:  18.97917

Treatment,  means

   GrowthRate       std r Min Max
T1     209.50 12.151817 4 198 220
T2     233.00  9.018500 4 222 244
T3     235.00  4.082483 4 232 241
T4     242.75  4.112988 4 238 248

Alpha: 0.05 ; DF Error: 6

Critical Range
       2        3        4
7.537752 7.812304 7.948306

Means with the same letter are not significantly
different.

   GrowthRate groups
T4     242.75      a
T3     235.00      b
T2     233.00      b
T1     209.50      c
>
```

Or we can use ExpDes package, as follows.

```
> library(ExpDes)
> latsd(data$Treatment, data$Row, data$Column, data$GrowthRate,
+     quali = TRUE, mcomp = "duncan", sigT = 0.05, sigF = 0.05)
------------------------------------------------------------------
Analysis of Variance Table
------------------------------------------------------------------
            DF     SS     MS     Fc    Pr>Fc
Treatament   3 2467.2 822.40 43.332 0.000185
Row          3  325.7 108.56  5.720 0.034123
Column       3  348.2 116.06  6.115 0.029551
Residuals    6  113.9  18.98
Total       15 3254.9
------------------------------------------------------------------
CV = 1.89 %

------------------------------------------------------------------
Shapiro-Wilk normality test
p-value:  0.7721845
    According to Shapiro-Wilk normality test at 5% of significance,
residuals can be considered normal.
------------------------------------------------------------------

Duncan's test
------------------------------------------------------------------
Groups  Treatments  Means
a         T4          242.75
 b        T3          235
 b        T2          233
  c       T1          209.5
------------------------------------------------------------------
>
```

IX. CROSSOVER DESIGN

9.1 Simple Crossover Design

Crossover design or also known as change-over design is experimental design where two or more treatments applied to the same subject (usually animal) in different periods sequentially. In this design, measurement of each animal is more than once, and each measurement is correspond to a different treatment with random order of the treatment. An animal here is used as a block and usually called a subject. For example, an experiment is conducted to test three different treatments (T1, T2 and T3) on milk production using six cows. Lay out of the experiment can be seen as below.

Period	cow 1	cow 2	cow 3	cow 4	cow 5	cow 6
1	T2	T1	T2	T1	T3	T3
2	T1	T3	T3	T2	T1	T2
3	T3	T2	T1	T3	T2	T1

If subjects is considered as blocks, the model is similar to a randomized block design model, with the subject effect defined as random:

$$yij = \mu + \tau i + Sj + \varepsilon ij \; i = 1,...,t; \;\; j = 1,...,n;$$

where:

yij = observation on subject (cow) j in treatment i

τi = the fixed effect of treatment i

Sj = the random effect of subject (cow) j

εij = random error

t = number of treatments

n = number of subjects (cows)

Randomization for crossover design can be assigned randomly to treatment order in every subject with period. Example of randomization for crossover design is described as follows.

```
> cow1=sample(1:3,size=3,replace=FALSE)
> cow1
[1] 2 1 3
```

```
> cow2=sample(1:3,size=3,replace=FALSE)
> cow2
[1] 1 3 2
>
```

Based on treatment order randomization, the order of treatment for cow 1 is T2 in period 1, T1 in period 2 and T3 in period 3; the order for cow 2 is T1 in period 1, T3 in period 2 and T2 in period 3; and soon shown in table above.

Total sum of squares is sums of squares between subjects and within subjects:

$$SST = SSs + SSws$$

Sum of squares within subjects is sums of treatment sum of squares and residual sum of squares:

$$SSws = SSt + SSE$$

Thus, the total sum of squares is:

$$SST = SSs + SSt + SSE$$

with corresponding degrees of freedom:

$$(tn - 1) = (n - 1) + (t - 1) + (n - 1)(t - 1)$$

The sums of squares above can be formulated as below.

$$\text{SST} = \sum_i \sum_j (yij - \bar{y}..)^2$$

$$\text{SSs} = \sum_i \sum_j (\bar{y}.j - \bar{y}..)^2$$

$$\text{SSt} = \sum_i \sum_j (\bar{y}i. - \bar{y}..)^2$$

$$\text{SSws} = \sum_i \sum_j (\bar{y}ij - \bar{y}.j)^2$$

$$\text{SSE} = \sum_i \sum_j (yij - \bar{y}i. - \bar{y}.j + \bar{y}..)^2$$

Mean square (MS) can be calculated as below.

MSt = SSt/dft = SSt/(t-1)

MSs = SSs/dfs =SSs/(n-1)

MSws = SSws/dfws = SSws/n(t-1)

MSE = SSE/dfe = SSE/(t-1)(n-1)

The null and alternative hypotheses are:

H_0: $\tau 1 = \tau 2 = ... = \tau r$, treatment effects are the same

H_1: $\tau i \neq \tau r$, at least the effects of one pair of treatment is different

F statistic = MSt/MSE compared to F table with (t-1) and (t-1)(n-1) degrees of freedom for critical value. For an α level of significance H_0 is rejected if F *statistic* > $F_{\alpha,(t-1),(t-1)(n-1)}$.

The results ANOVA can be summarized in table below.

Source of variation	df	SS	MS=SS/df	F
Between subject	s-1	SSs	MSs	
Within subject	n(t-1)	SSws	MSws	
Treatment	t-1	SSt	MSt	MSt/MSE
Residual	(t-1)(n-1)	SSE	MSE	

For example, an experiment is conducted to test three different treatments (T1, T2 and T3) on milk production using six cows. Measurement is taken in month two, three and four during lactation. Milk production per month of the six cows for three month (2^{nd} , 3^{rd} and 4^{th} month) can be seen as below.

Table. Milk production (kg) per month of six cows during 2^{nd}, 3^{rd} and 4^{th} month of lactation.

Period	cow 1	cow 2	cow 3	cow 4	cow 5	cow 6
1	T2(680)	T1(600)	T2(670)	T1(620)	T3(700)	T3(680)
2	T1(650)	T3(700)	T3(710)	T2(680)	T1(640)	T2(700)
3	T3(730)	T2(700)	T1(700)	T3(740)	T2(710)	T1(680)

ANOVA for the above table is like RCBD or Latin square analysis, where cow and period as blocks.

In R:

```
> data = read.csv("crossover4.csv", header=T)
> data
   Cow Period Treatment MilkProduction
1    1      1        T2             680
2    1      2        T1             650
3    1      3        T3             730
4    2      1        T1             600
5    2      2        T3             700
6    2      3        T2             700
7    3      1        T2             670
8    3      2        T3             710
```

```
9    3      3        T1            700
10   4      1        T1            620
11   4      2        T2            680
12   4      3        T3            740
13   5      1        T3            700
14   5      2        T1            640
15   5      3        T2            710
16   6      1        T3            680
17   6      2        T2            700
18   6      3        T1            680
> str(data)
'data.frame':   18 obs. of  4 variables:
 $ Cow           : int  1 1 1 2 2 2 3 3 3 4 ...
 $ Period        : int  1 2 3 1 2 3 1 2 3 1 ...
 $ Treatment     : Factor w/ 3 levels "T1","T2","T3":
  2 1 3 1 3 2 2 3 1 1 ...
 $ MilkProduction: int  680 650 730 600 700 700 670
  710 700 620 ...
> data$Cow=as.factor(data$Cow)
> data$Period=as.factor(data$Period)
> data$Treatment=as.factor(data$Treatment)
> str(data)
'data.frame':   18 obs. of  4 variables:
 $ Cow         : Factor w/ 6 levels "1","2","3","4",..:
  1 1 1 2 2 2 3 3 3 4 ...
 $ Period         : Factor w/ 3 levels "1","2","3": 1
  2 3 1 2 3 1 2 3 1 ...
 $ Treatment     : Factor w/ 3 levels "T1","T2","T3":
  2 1 3 1 3 2 2 3 1 1 ...
 $ MilkProduction: int  680 650 730 600 700 700 670
  710 700 620 ...
> xover=aov(MilkProduction~Cow+Treatment, data=data)
> summary(xover)
            Df Sum Sq Mean Sq F value Pr(>F)
Cow          5   1228     246   0.265 0.9220
Treatment    2  11878    5939   6.417 0.0161 *
Residuals   10   9256     926
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’
  0.1 ‘ ’ 1
> library(agricolae)
>    duncan.test(xover,    "Treatment",    alpha=0.05,
  console=T)

Study: xover ~ "Treatment"

Duncan's new multiple range test
for MilkProduction
```

```
Mean Square Error:  925.5556

Treatment,  means

   MilkProduction      std r Min Max
T1       648.3333 37.10346 6 600 700
T2       690.0000 15.49193 6 670 710
T3       710.0000 21.90890 6 680 740

Alpha: 0.05 ; DF Error: 10

Critical Range
       2        3
39.13658 40.89737

Means with the same letter are not significantly
  different.

   MilkProduction groups
T3       710.0000      a
T2       690.0000      a
T1       648.3333      b
>
```

Based on ANOVA result it can be concluded that different milk replacer affected the growth rate of calf with T3 is the largest effect. We can use ExpDes package for this case, as below.

```
> library(ExpDes)
> rbd(data$Treatment, data$Cow, data$MilkProduction, quali = TRUE,
+   mcomp='duncan',hvar='oneillmathews', sigT = 0.05, sigF = 0.05)
------------------------------------------------------------------------
Analysis of Variance Table
------------------------------------------------------------------------
           DF      SS     MS     Fc   Pr>Fc
Treatament  2 11877.8 5938.9 6.4166 0.01611
Block       5  1227.8  245.6 0.2653 0.92200
Residuals  10  9255.6  925.6
Total      17 22361.1
------------------------------------------------------------------------
CV = 4.46 %

------------------------------------------------------------------------
Shapiro-Wilk normality test
p-value:  0.4500391
```

```
     According to Shapiro-Wilk normality test at 5% of significance, residuals
can be considered normal.
     ------------------------------------------------------------------------


     ------------------------------------------------------------------------
     Homogeneity of variances test
     p-value:  0.767789
     According to the test of oneillmathews at 5% of significance, the
variances can be considered homocedastic.
     ------------------------------------------------------------------------


     Duncan's test
     ------------------------------------------------------------------------
     Groups  Treatments  Means
     a        T3          710
     a        T2          690
      b       T1          648.3333
     ------------------------------------------------------------------------
     >
```

9.2 Crossover Design with Periods and Sequences Effects

The next example is crossover design using Latin square to investigate four different drug (A, B, C and D) on cortisol level of women. This experiment used eight women for two round. Each of the first four women are exposed to a different drug with randomly assigned order, then a time (three days) is allowed to pass and the observation is recorded. Then a washout period (three days) passes to eliminate the effects of the first drug, and each of the woman are treated with a second different drug in the second time period. This is repeated until the Latin square is complete. The experiment is performed using two rounds, where the first round is completed using the first four women and the second round is completed using the remaining women.

Table. Cortisol level (micrograms per deciliter (ug/dl)) of women exposed to four different drug

Sequence/round	Periode	Women			
		1	2	3	4
1	1	C(13)	A(8.6)	D(11)	B(9)
	2	D(11.4)	C(13.5)	B(9.4)	A(8.9)
	3	B(9.6)	D(11.6)	A(9)	C(13.8)
	4	A(8.8)	B(10)	C(14)	D(12)

		Women			
		5	6	7	8
2	1	B(9)	C(13)	A(8)	D(11.4)
	2	D(11.3)	B(9.2)	C(13.5)	A(8.4)
	3	A(8.8)	D(11.4)	B(9.4)	C(13.8)
	4	C(13.8)	A(9)	D(11.5)	B(9.8)

In R:

```
> data = read.csv("crossover5.csv", header=T)
> data
   Sequence Period Women Drug Cortisol
1         1      1     1    C     13.0
2         1      2     1    D     11.4
3         1      3     1    B      9.6
4         1      4     1    A      8.8
5         1      1     2    A      8.6
6         1      2     2    C     13.5
7         1      3     2    D     11.6
8         1      4     2    B     10.0
9         1      1     3    D     11.0
10        1      2     3    B      9.4
11        1      3     3    A      9.0
12        1      4     3    C     14.0
13        1      1     4    B      9.0
14        1      2     4    A      8.9
15        1      3     4    C     13.8
16        1      4     4    D     12.0
17        2      1     5    B      9.0
18        2      2     5    D     11.3
19        2      3     5    A      8.8
20        2      4     5    C     13.8
21        2      1     6    C     13.0
22        2      2     6    B      9.2
23        2      3     6    D     11.4
24        2      4     6    A      9.0
25        2      1     7    A      8.0
26        2      2     7    C     13.5
27        2      3     7    B      9.4
28        2      4     7    D     11.5
29        2      1     8    D     11.4
30        2      2     8    A      8.4
31        2      3     8    C     13.8
32        2      4     8    B      9.8
> str(data)
'data.frame':   32 obs. of  5 variables:
 $ Sequence: int  1 1 1 1 1 1 1 1 1 1 ...
 $ Period  : int  1 2 3 4 1 2 3 4 1 2 ...
 $ Women   : int  1 1 1 1 2 2 2 2 3 3 ...
```

```
 $ Drug    : Factor w/ 4 levels "A","B","C","D": 3 4 2 1 1 3 4 2 4 2 ...
 $ Cortisol: num  13 11.4 9.6 8.8 8.6 13.5 11.6 10 11 9.4 ...
> data$Sequence=as.factor(data$Sequence)
> data$Period=as.factor(data$Period)
> data$Women=as.factor(data$Women)
> str(data)
'data.frame':   32 obs. of  5 variables:
 $ Sequence: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ Period  : Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 4 1 2 ...
 $ Women   : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3
  ...
 $ Drug    : Factor w/ 4 levels "A","B","C","D": 3 4 2 1 1 3 4 2 4 2 ...
 $ Cortisol: num  13 11.4 9.6 8.8 8.6 13.5 11.6 10 11 9.4 ...
> xover=aov(Cortisol~Sequence+Period+Women+Drug, data=data)
> summary(xover)
            Df Sum Sq Mean Sq  F value   Pr(>F)
Sequence     1   0.17    0.17    6.677   0.0187 *
Period       3   2.42    0.81   32.529 1.76e-07 ***
Women        6   0.28    0.05    1.864   0.1428
Drug         3 114.69   38.23 1544.226  < 2e-16 ***
Residuals   18   0.45    0.02
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> library(agricolae)
> LSD.test(xover, "Drug", alpha=0.05, console=T)

Study: xover ~ "Drug"

LSD t Test for Cortisol

Mean Square Error:  0.02475694

Drug,  means and individual ( 95 %) CI

  Cortisol       std r       LCL       UCL Min Max
A   8.6875 0.3440826 8  8.570627  8.804373   8   9
B   9.4250 0.3615443 8  9.308127  9.541873   9  10
C  13.5500 0.3779645 8 13.433127 13.666873  13  14
D  11.4500 0.2828427 8 11.333127 11.566873  11  12

Alpha: 0.05 ; DF Error: 18
Critical Value of t: 2.100922

least Significant Difference: 0.1652831

Treatments with the same letter are not significantly
  different.

  Cortisol groups
C  13.5500      a
D  11.4500      b
B   9.4250      c
A   8.6875      d
>
```

Based on ANOVA result it can be concluded that different drug affected the cortisol level of women with drug C had the highest affect.

X. FACTORIAL DESIGN

10.1 Introduction

Factorial design is experimental design where two or more sets of treatments with their levels are analysed at the same time. This design is actually an extension of single factor ANOVA designs with addition of other factors, so that treatment level combination, which is called interaction, between the two or more factors are generated.

There are main factor effect and simple (interaction) effect in this design. If the interaction effect is significant, all combinations of factor levels are tested. However, if there is no interaction effect, the main factor effect should be focused and tested in the experiment. Randomization in this design is that all combinations of factors are randomly applied to experimental units.

10.2 Simple Factorial Design (2x2)

Suppose there are two factors *A* and *B* in an experiment with *a* levels of factor *A* and *b* levels of factor *B* and *n* is the number of experimental units for each *A* x *B* combination. Linear model with two factors A and B is like below.

$$yijk = \mu + Ai + Bj + (AB)ij + \varepsilon ijk \qquad i = 1,\ldots,a;\ j = 1,\ldots,b;\ k = 1,\ldots,n$$

where:

yijk = observation *k* in level *i* of factor *A* and level *j* of factor *B*

μ = the overall mean

Ai = the effect of level *i* of factor *A*

Bj = the effect of level *j* of factor *B*

(*AB*)*ij* = the interaction effect of level *i* of factor *A* with level *j* of factor *B*

εijk = random error

a = number of levels of factor *A*

b = number of levels of factor *B*

n = number of observations for each *A x B* combination.

Factorial 2 x 2 is the simplest factorial experimental design which mean there are two factors with 2 levels for each factor. Factorial 3 x 2 is factorial experimental design with two factors where the first factor consists of three levels and the second

factor consists of two levels, and soon. Combination of factorial 2 x 2 can be seen at table below.

Table. Factorial 2 x 2 with factor A and factor B consist of two levels each

Factor B	Factor A	
	A1	A2
B1	A1B1	A2B1
B2	A1B2	A2B2

Possible combination of the two factors above with n replication can be describe as table below.

A1		A2	
B1	B2	B1	B2
y111	y121	y211	y221
y112	y122	y212	y222
…	…	…	…
y11n	y12n	y21n	y22n

Note: *yijk* denotes measurement *k* of level *i* of factor *A* and level *j* of factor *B*.

Sum square of factorial 2 x 2 with factor A and factor B is

$$\text{SST} = \text{SSA} + \text{SSB} + \text{SSAB} + \text{SSE}$$

The corresponding degrees of freedom is

$$(abn\text{-}1) = (a\text{-}1) + (b\text{-}1) + (a\text{-}1)(b\text{-}1) + ab(n\text{-}1)$$

The sums of squares above can be formulated as below.

where

$$\text{SST} = \sum_i \sum_j \sum_k (yijk - \bar{y}...)^2$$

$$\text{SSA} = \sum_i \sum_j \sum_k (\bar{y}i.. - \bar{y}...)^2$$

$$\text{SSB} = \sum_i \sum_j \sum_k (\bar{y}.j. - \bar{y}...)^2$$

$$\text{SSAB} = \text{n} \sum_i \sum_j (\bar{y}ij. - \bar{y}...)^2 - SSA - SSB$$

$$\text{SSE} = \sum_i \sum_j \sum_k (\bar{y}ijk \; - \; \bar{y}ij.)^2$$

Sums of squares above can be calculated using computation below.

$$\text{CF} = \frac{\left(\sum_i \sum_j \sum_k yijk\right)^2}{abn}$$

$$\text{SST} = \sum_i \sum_j \sum_k yijk^2 \; - CF$$

$$\text{SSA} = \sum_i \frac{\left(\sum_j \sum_k yijk\right)^2}{bn} \; - \; CF$$

$$\text{SSB} = \sum_j \frac{(\sum_i \sum_k yijk)^2}{an} \; - \; CF$$

$$\text{SSAB} = \sum_i \sum_j \frac{(\sum_k yijk)^2}{n} \; - \; SSA - SSB - CF$$

$$SSE = SST \; - \; SSA \; - \; SSB - \; SSAB$$

Mean square (MS) can be calculated as below.

MSA = SSt/dfa = SSA/(a-1)

MSB = SSR/dfb =SSB/(b-1)

MSAB = SSAB/dfab = SSAB/(a-1)(b-1)

MSE = SSE/dfe = SSE/ab(n-1)

There are three null and alternative hypotheses, those are:

For factor A: H_0: $\tau 1 = \tau 2 = ... = \tau i$, treatment effects are the same

H_1: $\tau i \neq \tau a$, at least the effects of one pair of treatment is different

For factor B: H_0: $\tau 1 = \tau 2 = ... = \tau j$, treatment effects are the same

H_1: $\tau i \neq \tau b$, at least the effects of one pair of treatment is different

For factor A: H_0: $\tau 11 = \tau 12 = ... = \tau ij$, treatment effects are the same

H_1: $\tau i \neq \tau ab$, at least the effects of one pair of treatment combination is different

There are three tests for factorial design, F statistic = MSA/MSE compared to F table with (a-1) and ab(n-1) degrees of freedom for critical value of Factor A, F statistic = MSB/MSE compared to F table with (b-1) and ab(n-1) degrees of freedom for critical value of Factor B, and F statistic = MSAB/MSE compared to F table with (a-1)(b-1)

and ab(n-1) degrees of freedom for critical value of interaction A x B. For an α level of significance H_0 is rejected if F *statistic* > $F_{\alpha,(a-1),ab(n-1)}$, F *statistic* > $F_{\alpha,(b-1),ab(n-1)}$, F *statistic* > $F_{\alpha,(a-1)(b-1),ab(n-1)}$, respectively for Factor A, Factor B and interaction A x B. However if there is significant interaction effect we do not need to test further for the main factor (factor A and or factor B), but we need to test further between combination effects.

The results ANOVA can be summarized in table below.

Source of variation	df	SS	MS=SS/df	F
A	a-1	SSA	MSA	MSA/MSE
B	b-1	SSB	MSB	MSB/MSE
AxB	(a-1)(b-1)	SSAB	MSAB	MSAB/MSE
Residual	ab(n-1)	SSE	MSE	
Total	abn-1	SST		

A study wanted to know the interaction between two types of feed (B, basal ratio and C, basal ration plus concentrate) and breed of sheep on birth weights of lambs. Pregnancy ewes are fed with the two different ration for four months before delivering lambs. The research is design using factorial design 2 x 2. Data from the research results (hypothetical) are presented in the following table.

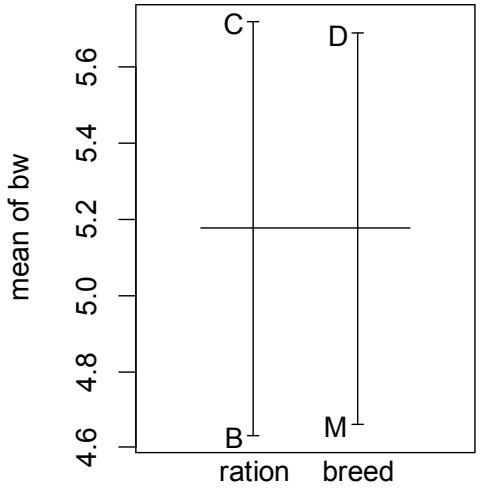
Table. Birth weight of Merino and Dorset lambs whose their dam fed with two different ration.

Observation	Ration			
	B		C	
	Merino	Dorset	Merino	Dorset
1	4.5	5.2	4.8	6.5
2	4.5	5	5.2	6.2
3	3.8	4.7	5.3	6.4
4	4.2	4.8	4.9	6.7
5	4.4	5.2	5	6.2

In R:

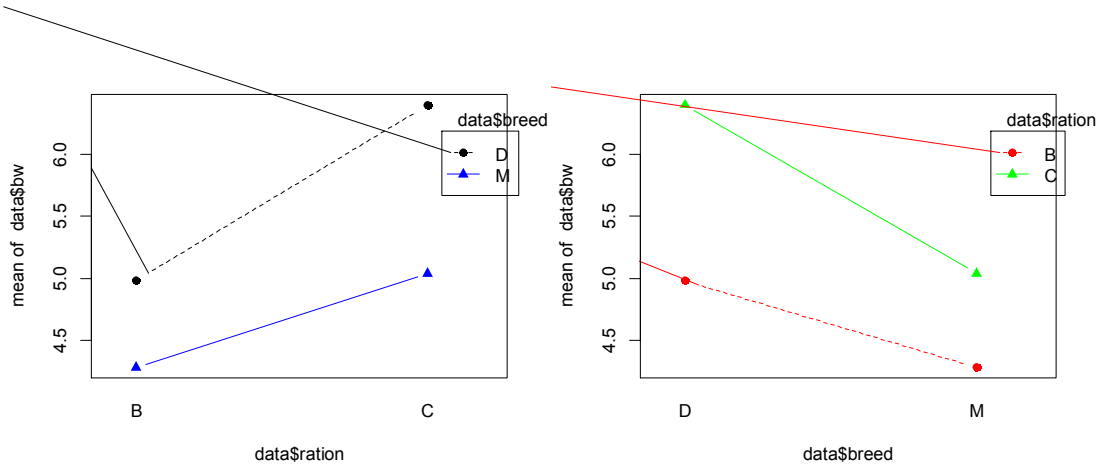
```
> data <- read.csv('factorial1.csv', header=T)
> data
   ration breed  bw
1       B     M 4.5
2       B     M 4.5
3       B     M 3.8
4       B     M 4.2
```

```
5        B      M 4.4
6        B      D 5.2
7        B      D 5.0
8        B      D 4.7
9        B      D 4.8
10       B      D 5.2
11       C      M 4.8
12       C      M 5.2
13       C      M 5.3
14       C      M 4.9
15       C      M 5.0
16       C      D 6.5
17       C      D 6.2
18       C      D 6.4
19       C      D 6.7
20       C      D 6.2
> plot.design(data)
```




```
> fit <- aov(bw ~ ration*breed, data=data)
> summary(fit)
             Df Sum Sq Mean Sq F value   Pr(>F)
ration        1  5.941   5.941 104.678 2.00e-08 ***
breed         1  5.305   5.305  93.471 4.38e-08 ***
ration:breed  1  0.544   0.544   9.595  0.00692 **
Residuals    16  0.908   0.057
---
Signif. codes:0 '***'0.001'**'0.01'*'0.05'.'0.1' ' 1
```

```
> par(mfrow=c(1,2))
> par(mfrow=c(1,2))
> interaction.plot(x.factor = data$ration,
+   trace.factor = data$breed,response = data$bw,
+   fun = mean,type="b", col=c("black","blue"),
+   pch=c(19, 17),fixed=TRUE,leg.bty = "o")
> interaction.plot(x.factor = data$breed,
+   trace.factor = data$ration, response = data$bw,
+   fun = mean,type="b", col=c("red","green"),
+   pch=c(19, 17), fixed=TRUE,leg.bty = "o")
```



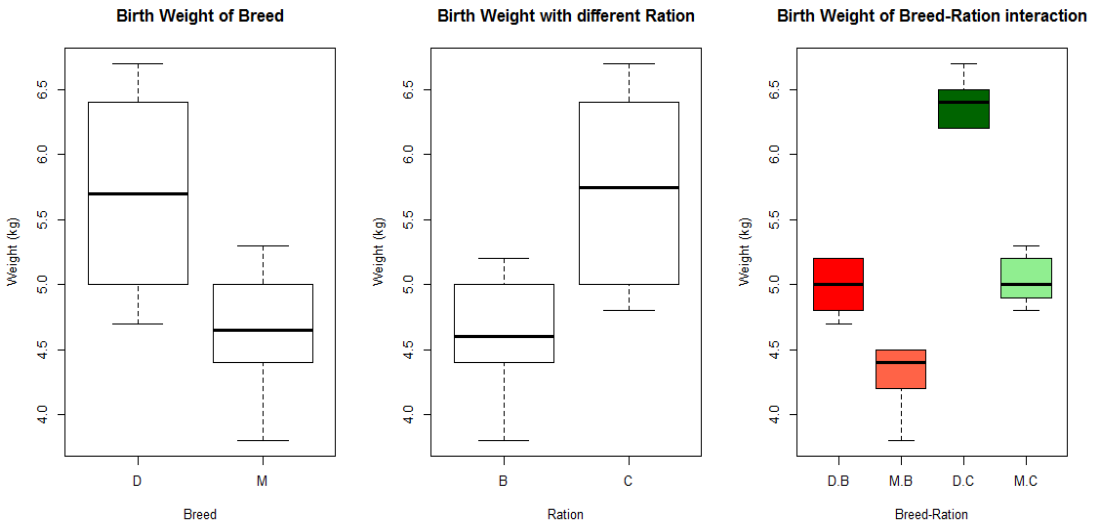

```
> HSD.test(fit, c("ration","breed"), alpha=0.05,
+   console=T)

Study: fit ~ c("ration", "breed")

HSD Test for bw

Mean Square Error:  0.05675

ration:breed,  means

      bw       std r Min Max
B:D 4.98 0.2280351 5 4.7 5.2
B:M 4.28 0.2949576 5 3.8 4.5
C:D 6.40 0.2121320 5 6.2 6.7
C:M 5.04 0.2073644 5 4.8 5.3

Alpha: 0.05 ; DF Error: 16
Critical Value of Studentized Range: 4.046093

Minimun Significant Difference: 0.4310561

Treatments with the same letter are not significantly
different.
```

```
      bw groups
C:D 6.40      a
C:M 5.04      b
B:D 4.98      b
B:M 4.28      c
>
```

Based on ANOVA and HSD test it can be concluded that there is significant interaction between ration and breed on birth weight of lambs with C and D combination (basal ration plus concentrate and Dorset lamb) had the highest effect.

Description of the data graphically can be shown as following boxplot.

```
> par(mfrow=c(1,3))
> boxplot(bw~breed, main="Birth Weight of Breed",
+   xlab="Breed", ylab="Weight (kg)", data=data)
> boxplot(bw~ration, main="Birth Weight with different
  Ration",
+   xlab="Ration", ylab="Weight (kg)", data=data)
>  boxplot(bw~breed*ration,  main="Birth  Weight  of
  Breed-Ration interaction",
+        xlab="Breed-Ration",   ylab="Weight   (kg)",
  col=c("red","tomato",
+   "darkgreen","lightgreen"),data=data)
```




```
> leveneTest(bw~ration*breed, data=data)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  0.0789 0.9705
      16
```

10.3 Factorial Design 3x3

The experiment was conducted to investigate the effect of three factors, namely percentage: protein ration level, methionine supplementation, and lysine supplementation. The experiment was carried out with RCBD with 2 replications. The data recorded is the average body weight gain per day of bulls, as listed in the following table:

Table. Body weight gain of bulls fed ration with different level of protein, methionine and lysin supplementation (kg/day/head)

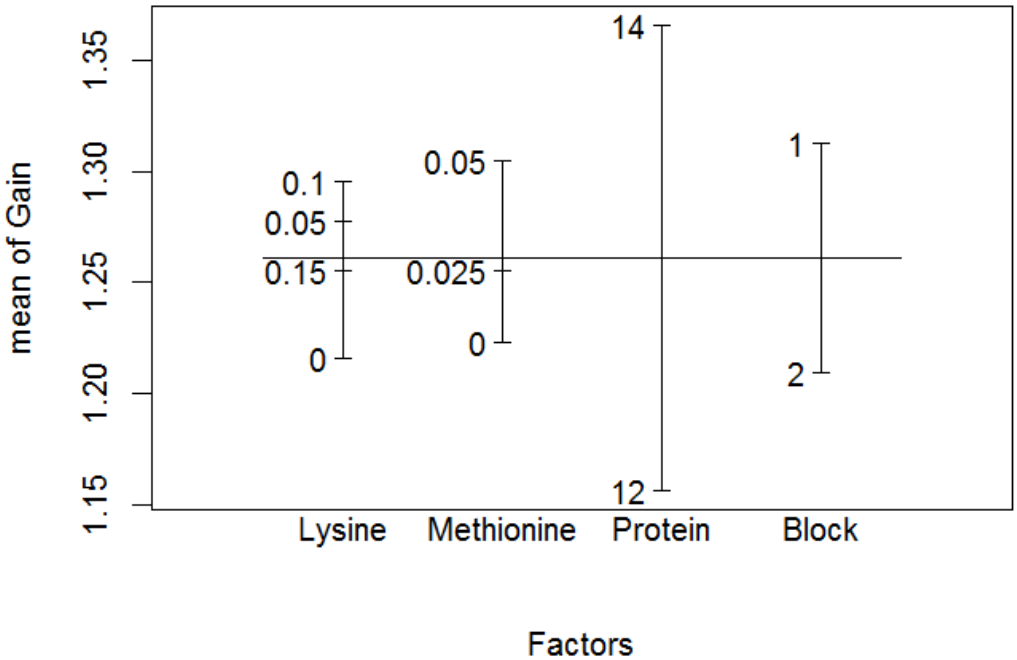
Lysin	Methionine	Protein	Replication		Total treatments
			1	2	
0	0	12	1.11	0.97	2.08
		14	1.52	1.45	2.97
	0.025	12	1.09	0.99	2.08
		14	1.27	1.22	2.49
	0.05	12	0.85	1.21	2.06
		14	1.67	1.24	2.91
0.05	0	12	1.30	1.00	2.30
		14	1.55	1.55	3.10
	0.025	12	1.03	1.21	2.24
		14	1.24	1.34	2.58
	0.05	12	1.12	0.96	2.08
		14	1.76	1.27	3.03
0.10	0	12	1.22	1.13	2.35
		14	1.38	1.08	2.46
	0.025	12	1.34	1.41	2.75
		14	1.40	1.21	2.61
	0.05	12	1.34	1.19	2.53
		14	1.46	1.39	2.85
0.15	0	12	1.19	1.03	2.22
		14	0.80	1.29	2.09
	0.025	12	1.36	1.16	2.52
		14	1.42	1.39	2.81
	0.05	12	1.46	1.07	2.53
		14	1.62	1.27	2.89
Total			31.50	29.03	60.53

In R:

```
> data <- read.csv('factorial2.csv', header=T)
> data
   Lysine Methionine Protein Block Gain
1    0.00      0.000      12     1 1.11
2    0.00      0.000      14     1 1.52
3    0.00      0.025      12     1 1.09
```

```
4    0.00      0.025      14     1 1.27
5    0.00      0.050      12     1 0.85
6    0.00      0.050      14     1 1.67
7    0.05      0.000      12     1 1.30
8    0.05      0.000      14     1 1.55
9    0.05      0.025      12     1 1.03
10   0.05      0.025      14     1 1.24
11   0.05      0.050      12     1 1.12
12   0.05      0.050      14     1 1.76
13   0.10      0.000      12     1 1.22
14   0.10      0.000      14     1 1.38
15   0.10      0.025      12     1 1.34
16   0.10      0.025      14     1 1.40
17   0.10      0.050      12     1 1.34
18   0.10      0.050      14     1 1.46
19   0.15      0.000      12     1 1.19
20   0.15      0.000      14     1 0.80
21   0.15      0.025      12     1 1.36
22   0.15      0.025      14     1 1.42
23   0.15      0.050      12     1 1.46
24   0.15      0.050      14     1 1.62
25   0.00      0.000      12     2 0.97
26   0.00      0.000      14     2 1.45
27   0.00      0.025      12     2 0.99
28   0.00      0.025      14     2 1.22
29   0.00      0.050      12     2 1.21
30   0.00      0.050      14     2 1.24
31   0.05      0.000      12     2 1.00
32   0.05      0.000      14     2 1.55
33   0.05      0.025      12     2 1.21
34   0.05      0.025      14     2 1.34
35   0.05      0.050      12     2 0.96
36   0.05      0.050      14     2 1.27
37   0.10      0.000      12     2 1.13
38   0.10      0.000      14     2 1.08
39   0.10      0.025      12     2 1.41
40   0.10      0.025      14     2 1.21
41   0.10      0.050      12     2 1.19
42   0.10      0.050      14     2 1.39
43   0.15      0.000      12     2 1.03
44   0.15      0.000      14     2 1.29
45   0.15      0.025      12     2 1.16
46   0.15      0.025      14     2 1.39
47   0.15      0.050      12     2 1.07
48   0.15      0.050      14     2 1.27
> str(data)
'data.frame':   48 obs. of  5 variables:
 $ Lysine    : num  0 0 0 0 0 0 0.05 0.05 0.05 0.05
  ...
```

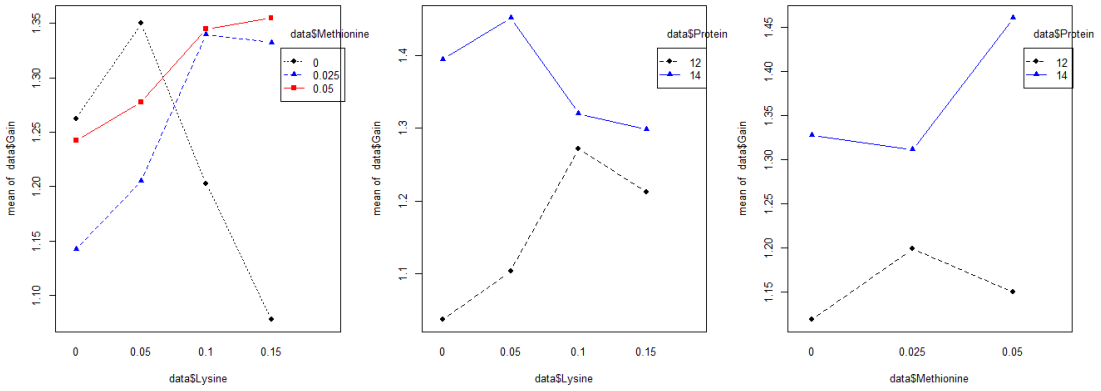
```
 $ Methionine: num  0 0 0.025 0.025 0.05 0.05 0 0 0.025
  0.025 ...
 $ Protein   : int  12 14 12 14 12 14 12 14 12 14 ...
 $ Block     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Gain      : num  1.11 1.52 1.09 1.27 0.85 1.67 1.3
  1.55 1.03 1.24 ...
> data$Lysine=as.factor(data$Lysine)
> data$Methionine=as.factor(data$Methionine)
> data$Protein=as.factor(data$Protein)
> data$Block=as.factor(data$Block)
> str(data)
'data.frame':   48 obs. of  5 variables:
 $ Lysine    : Factor w/ 4 levels "0","0.05","0.1",..:
  1 1 1 1 1 1 2 2 2 2 ...
 $ Methionine: Factor w/ 3 levels "0","0.025","0.05":
  1 1 2 2 3 3 1 1 2 2 ...
 $ Protein   : Factor w/ 2 levels "12","14": 1 2 1 2
  1 2 1 2 1 2 ...
 $ Block     : Factor w/ 2 levels "1","2": 1 1 1 1 1
  1 1 1 1 1 ...
 $ Gain      : num  1.11 1.52 1.09 1.27 0.85 1.67 1.3
  1.55 1.03 1.24 ...
> plot.design(data)
```



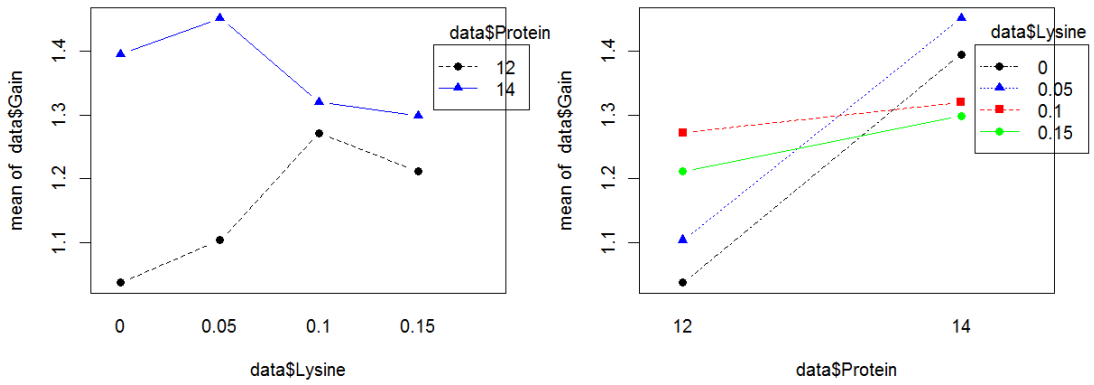

```
> fit <- aov(Gain ~ Lysine*Methionine*Protein +Block,
+   data=data)
> summary(fit)
```

```
                          Df Sum Sq Mean Sq F value   Pr(>F)
Lysine                     3 0.0427  0.0142   0.527 0.668031
Methionine                 2 0.0545  0.0273   1.008 0.380353
Protein                    1 0.5313  0.5313  19.661 0.000191 ***
Block                      1 0.1271  0.1271   4.703 0.040692 *
Lysine:Methionine          6 0.2630  0.0438   1.622 0.186248
Lysine:Protein             3 0.2475  0.0825   3.052 0.048819 *
Methionine:Protein         2 0.0780  0.0390   1.444 0.256655
Lysine:Methionine:Protein  6 0.0696  0.0116   0.429 0.851839
Residuals                 23 0.6215  0.0270
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> par(mfrow=c(1,3))
> interaction.plot(x.factor = data$Lysine,trace.factor =
+   data$Methionine, response = data$Gain,fun = mean,
+   type="b", col=c("black","blue","red","green"),
+   pch=c(19, 17, 15),fixed=TRUE,leg.bty = "o")
> interaction.plot(x.factor = data$Lysine,trace.factor =
+  data$Protein, response = data$Gain,fun = mean,type="b",
+   col=c("black","blue","red","green"),
+   pch=c(19, 17, 15), fixed=TRUE,leg.bty = "o")
> interaction.plot(x.factor = data$Methionine,
+   trace.factor = data$Protein, response = data$Gain,
+   fun = mean,type="b",col=c("black","blue","red",
+   "green"), pch=c(19, 17, 15), fixed=TRUE,leg.bty = "o")
```

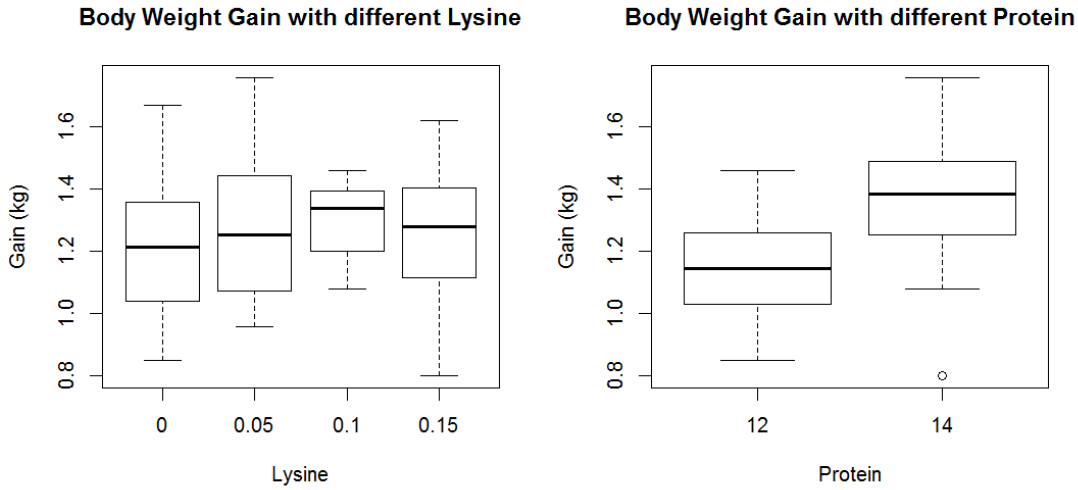



```
> par(mfrow=c(1,2))
> interaction.plot(x.factor = data$Lysine,trace.factor =
+  data$Protein, response = data$Gain,fun = mean,type="b",
+  col=c("black","blue","red","green"), pch=c(19, 17,
+  15),fixed=TRUE,leg.bty = "o")
> interaction.plot(x.factor = data$Protein,trace.factor =
+  data$Lysine, response = data$Gain,fun = mean,type="b",
+ col=c("black","blue","red","green"), pch=c(19, 17, 15),
+  fixed=TRUE,leg.bty = "o")
```

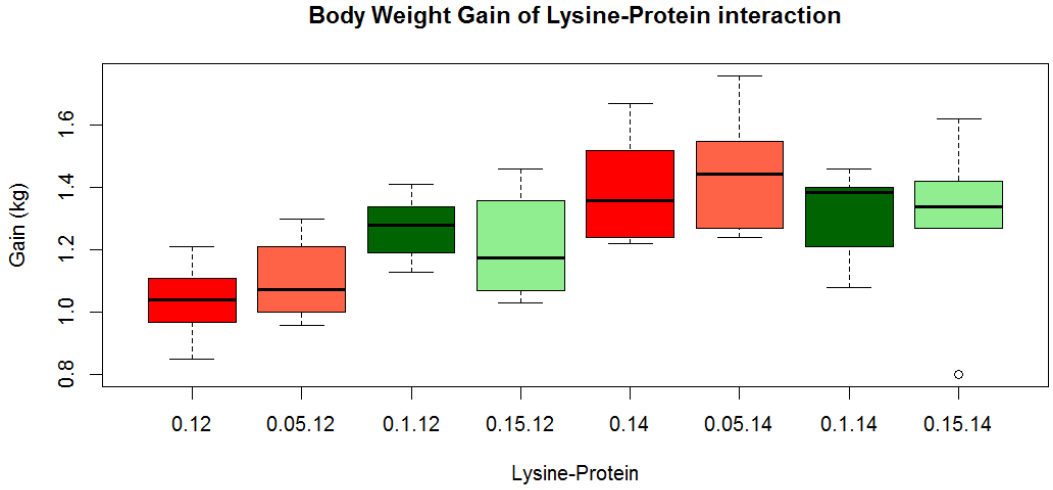



```
> HSD.test(fit, c("Lysine","Protein"), alpha=0.05,
+  console=T)

Study: fit ~ c("Lysine", "Protein")

HSD Test for Gain

Mean Square Error:  0.02702382

Lysine:Protein,  means

             Gain       std r  Min  Max
0.05:12 1.103333 0.1318585 6 0.96 1.30
0.05:14 1.451667 0.2023281 6 1.24 1.76
0.1:12  1.271667 0.1075949 6 1.13 1.41
0.1:14  1.320000 0.1443607 6 1.08 1.46
0.15:12 1.211667 0.1672623 6 1.03 1.46
0.15:14 1.298333 0.2741836 6 0.80 1.62
0:12    1.036667 0.1262801 6 0.85 1.21
0:14    1.395000 0.1814111 6 1.22 1.67

Alpha: 0.05 ; DF Error: 23
Critical Value of Studentized Range: 4.701848

Minimum Significant Difference: 0.3155487

Treatments with the same letter are not significantly
different.

             Gain groups
0.05:14 1.451667      a
0:14    1.395000     ab
0.1:14  1.320000    abc
0.15:14 1.298333    abc
0.1:12  1.271667    abc
0.15:12 1.211667    abc
```

```
0.05:12 1.103333     bc
0:12    1.036667      c

> par(mfrow=c(1,2))
> boxplot(Gain~Lysine, main="Body Weight Gain with
+   different Lysine",xlab="Lysine", ylab="Gain (kg)",
+   data=data)
> boxplot(Gain~Protein, main="Body Weight Gain with
+   different Protein", xlab="Protein", ylab="Gain (kg)",
+   data=data)
```




```
> par(mfrow=c(1,1))
> boxplot(Gain~Lysine*Protein, main="Body Weight Gain of
+   Lysine-Protein interaction", xlab="Lysine-Protein",
+   ylab="Gain (kg)", col=c("red","tomato",
+   "darkgreen","lightgreen"),data=data)
```



Based on ANOVA and HSD test it can be concluded that level protein affected body weight gain of bulls, and this level of protein interact with level of lysine with protein 14% and lysine 0.05% in ration had the highest effect on the bull gain.

XI. SPLIT PLOT DESIGN

11.1 Introduction

Split plot design is experimental design where experimental material is divided into several main units (main plots), and then each of the main units is divided also into several sub units (sub plots). Split plot design is usually used in agricultural research. For example, suppose an experiment is conducted to investigate the effect of three levels of fertilizer and four rice varieties on rice production. This experiment can be designed using large land by dividing the land into three plots for main plot of three levels of fertilizer, therefore, randomization is assigned for the three levels of fertilizer. Each main plot is again divided into four sub plots for four rice varieties, and then this four rice varieties is randomly assigned into the sub plots. Fertilizer level is considered as main plot because it is hard to assign different level of fertilizer into sub plots, while many varieties of rice can be assigned into sub plots easily. Replication can be made according to our design by making several blocks of land because the split plot design can use CRD, RCBD, or Latin square designs, that can be assigned either on main plots or sub plots.

Split plot design can be used if one of the factors needs more experiment material than the second factor. Like in previous example, factor of fertilizer levels need large experimental units, and this factor is applied on the main plots. Whilst rice varieties can be applied or compared on sub plots. Furthermore, plot size and precision of measurement of effects are not the same for both factors. It is very important that the assignment of a particular factor to either the main plot or the sub-plot. Suggestion to choose a specific factor either as main or sub plot can be considered as the following guidelines.

First, if we want factor B is more precise than factor A, assign factor B to the sub-plot and factor A to the main plot. For example, as a plant breeder evaluating five new rice varieties with three levels of fertilization maybe want to have greater precision for varietal comparison than for fertilizer response. In this case, variety should be assigned as the sub-plot factor and fertilizer as the main plot factor. However, as an agronomist maybe assign variety to main plot and fertilizer to sub-plot if he wants greater precision for fertilizer response than variety effect.

Second, if we want to detect the main effect of factor A is expected to be much larger and easier than that of factor B, then factor A can be assigned to the main plot and factor B to the sub-plot. In this case the chance of detecting the difference among levels of factor B which has a smaller effect will increase.

Third, if there is difficulties in the execution of other designs, for example, an experiment to evaluate water management and rice varieties. In this case, water management is desirable to be as the main plot to minimize water movement between adjacent plots and reduce border effects.

11.2 Split Plot Design with Main Plots in a Completely Randomized Design

Suppose factor A with three levels (A1, A2, and A3) is assigned randomly on 12 plots (four replications). Factor B with two levels (B1 and B2) is assigned randomly in each level of factor A in such a way forming a design as below:

A2	A1	A3	A3	A2	A1	A2	A3	A1	A1	A2	A3
B2	B1	B1	B2	B1	B2	B2	B1	B2	B1	B1	B2
B1	B2	B2	B1	B2	B1	B1	B2	B1	B2	B2	B1

The model for the design is:

$$yijk = \mu + Ai + \delta ik + Bj + (AB)ij + \varepsilon ijk \qquad i = 1,...,a; \quad j = 1,...,b; \quad k = 1,...,n$$

where:

$yijk$ = observation k in level i of factor A and level j of factor B

μ = overall mean

Ai = effect of level i of factor A

Bj = effect of level j of factor B

$(AB)ij$ = effect of the ij^{th} interaction of A x B

δik = error a (Ea), the main plot error (the main plots within factor A)

εijk = error b (Eb), the split plot error

$\mu ij = \mu + Ai + Bj + (AB)ij$ = the mean of the $ijth$ A x B interaction

a = number of levels of factor A

b = number of levels of factor B

n = number of replications

Table of ANOVA for the design with three levels of factor *A*, two levels of factor *B* and four replications is presented in the following table.

Source of variation	Degree of freedom	
Factor A	(a – 1) =	3 – 1 = 2
Main plot error (Ea)	a(n – 1) =	3(4 – 1) = 9
Factor B	(b – 1) =	2 – 1 = 1
AxB	(a – 1)(b – 1) =	(3 – 1)(2 – 1) = 2
Split plot error (Eb)	a(b – 1)(n – 1) =	3(2 – 1)(4 – 1) = 9
Total	(abn – 1) =	(3.2.4 – 1) = 23

F statistic for factor A is

$$F = \frac{MSA}{MSEa}$$

F statistic for factor B is

$$F = \frac{MSB}{MSEb}$$

F statistic for the AxB interaction is

$$F = \frac{MSAB}{MSEb}$$

Example for this design, suppose an experiment is conducted to investigate the effect of three levels of fertilizer and two rice varieties on rice production. Fertilizer level is considered as main plot and varieties of rice is assigned into sub plots. Data on rice production is presented in the following table.

Table. Production (ton/ha) of four rice varieties using three levels of fertilizer

Plot	Fertilizer	Variety	Production	Plot	Fertilizer	Variety	Production
1	2	2	6.3	7	3	2	7.8
1	2	1	5.9	7	3	1	7.5
2	3	1	7.0	8	2	2	6.2
2	3	2	7.3	8	2	1	6.0
3	1	1	5.5	9	1	1	5.7
3	1	2	5.7	9	1	2	5.9
4	2	2	6.6	10	2	1	6.2
4	2	1	6.1	10	2	2	6.1
5	1	2	5.9	11	3	2	8.4
5	1	1	5.6	11	3	1	7.9
6	3	2	7.5	12	1	1	5.4
6	3	1	7.2	12	1	2	5.7

In R:

```
> data=read.csv('splitplot11.csv', header=T)
> data
   Plot Block Fertilizer Variety Production
1     1     1          2       2        6.3
2     1     1          2       1        5.9
3     2     1          3       1        7.0
4     2     1          3       2        8.3
5     3     1          1       1        5.5
6     3     1          1       2        5.7
7     4     2          2       2        6.6
8     4     2          2       1        6.1
9     5     2          1       2        5.9
10    5     2          1       1        5.6
11    6     2          3       2        7.5
12    6     2          3       1        7.0
13    7     3          3       2        7.8
14    7     3          3       1        7.0
15    8     3          2       2        6.2
16    8     3          2       1        6.0
17    9     3          1       1        5.7
18    9     3          1       2        5.9
19   10     4          2       1        6.2
20   10     4          2       2        6.1
21   11     4          3       2        8.4
22   11     4          3       1        7.9
23   12     4          1       1        5.4
24   12     4          1       2        5.7
> str(data)
'data.frame':   24 obs. of  5 variables:
 $ Plot      : int  1 1 2 2 3 3 4 4 5 5 ...
 $ Block     : int  1 1 1 1 1 1 2 2 2 2 ...
 $ Fertilizer: int  2 2 3 3 1 1 2 2 1 1 ...
 $ Variety   : int  2 1 1 2 1 2 2 1 2 1 ...
 $ Production: num  6.3 5.9 7 8.3 5.5 5.7 6.6 6.1 5.9 5.6 ...
> data$Plot=as.factor(data$Plot)
> data$Block=as.factor(data$Block)
> data$Fertilizer=as.factor(data$Fertilizer)
> data$Variety=as.factor(data$Variety)
> str(data)
'data.frame':   24 obs. of  5 variables:
 $ Plot      : Factor w/ 12 levels "1","2","3","4",..: 1 1 2 2
3 3 4 4 5 5 ...
 $ Block     : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1
2 2 2 2 ...
 $ Fertilizer: Factor w/ 3 levels "1","2","3": 2 2 3 3 1 1 2 2
1 1 ...
 $ Variety   : Factor w/ 2 levels "1","2": 2 1 1 2 1 2 2 1 2 1
...
 $ Production: num  6.3 5.9 7 8.3 5.5 5.7 6.6 6.1 5.9 5.6 ...
> #splitplot crd
>  modela  <-  aov(Production~Fertilizer*Variety+Error(Plot),
data=data)
```

```
> summary(modela)

Error: Plot
           Df Sum Sq Mean Sq F value   Pr(>F)
Fertilizer  2 16.187   8.094      66 4.19e-06 ***
Residuals   9  1.104   0.123
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Within
                   Df Sum Sq Mean Sq F value   Pr(>F)
Variety             1 1.0838  1.0838  30.127 0.000386 ***
Fertilizer:Variety  2 0.3675  0.1837   5.108 0.032930 *
Residuals           9 0.3238  0.0360
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Based on ANOVA table it can be concluded that fertilizer, rice variety and interaction between fertilizer and rice variety affected on rice production.

11.3 Split Plot Design with Main Plots in a Randomized Completely Block Design

Similar with split plot design with plots in a CRD, but replication is treated as block in split plot design with main plot in a RCBD. Suppose factor A with three levels (A1, A2, and A3) is assigned randomly on 12 plots (four replications). Factor B with two levels (B1 and B2) is assigned randomly in each level of factor A in such a way forming a design as below:

Block I			Block II			Block III			Block IV		
A2	A1	A3	A3	A2	A1	A2	A3	A1	A1	A2	A3
B2	B1	B1	B2	B1	B2	B2	B1	B2	B1	B1	B2
B1	B2	B2	B1	B2	B1	B1	B2	B1	B2	B2	B1

The model for the design is:

$yijk = \mu + \text{Block}k + Ai + \delta ik + Bj + (AB)ij + \varepsilon ijk \qquad i = 1,...,a;\ j = 1,...,b;\ k = 1,...,n$

where:

$yijk$ = observation k in level i of factor A and level j of factor B

μ = overall mean

Block$k$ = effect of the kth of block

Ai = effect of level i of factor A

Bj = effect of level j of factor B

$(AB)ij$ = effect of the ij^{th} interaction of A x B

δik = error a (Ea), the main plot error (the main plots within factor A)

εijk = error b (Eb), the split plot error

$\mu ij = \mu + Ai + Bj + (AB)ij$ = the mean of the $ijth$ A x B interaction

a = number of levels of factor A

b = number of levels of factor B

n = number of replications

Table of ANOVA for the design with three levels of factor A, two levels of factor B and four replications is presented in the following table.

Source of variation	Degree of freedom	
Block	(n – 1) =	4 – 1 = 3
Factor A	(a – 1) =	3 – 1 = 2
Main plot error (Ea)	(n – 1)(a – 1) =	(4 – 1)(3 – 1) = 6
Factor B	(b – 1) =	2 – 1 = 1
AxB	(a – 1)(b – 1) =	(3 – 1)(2 – 1) = 2
Split plot error (Eb)	a(b – 1)(n – 1) =	3(2 – 1)(4 – 1) = 9
Total	(abn – 1) =	(3.2.4 – 1) = 23

F statistic for factor A is

$$F = \frac{MSA}{MSEa}$$

F statistic for factor B is

$$F = \frac{MSB}{MSEb}$$

F statistic for the AxB interaction is

$$F = \frac{MSAB}{MSEb}$$

Data on rice production with replication as block is presented in the following table.

Table. Production (ton/ha) of four rice varieties using three levels of fertilizer

Plot	Block	Fertilizer	Variety	Production	Plot	Block	Fertilizer	Variety	Production
1	1	2	2	6.3	7	3	3	2	7.8
1	1	2	1	5.9	7	3	3	1	7.5
2	1	3	1	7.0	8	3	2	2	6.2
2	1	3	2	7.3	8	3	2	1	6.0
3	1	1	1	5.5	9	3	1	1	5.7
3	1	1	2	5.7	9	3	1	2	5.9
4	2	2	2	6.6	10	4	2	1	6.2
4	2	2	1	6.1	10	4	2	2	6.1
5	2	1	2	5.9	11	4	3	2	8.4
5	2	1	1	5.6	11	4	3	1	7.9
6	2	3	2	7.5	12	4	1	1	5.4
6	2	3	1	7.2	12	4	1	2	5.7

In R:

```
>   modelb   <-   aov(Production~Block+Fertilizer*Variety+
Error(Plot), data=data)
> summary(modelb)

Error: Plot
           Df Sum Sq Mean Sq F value  Pr(>F)
Block       3  0.135   0.045   0.278 0.83979
Fertilizer  2 16.187   8.094  50.107 0.00018 ***
Residuals   6  0.969   0.162
---
Signif.codes:0 ‘***’0.001‘**’0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Within
                  Df Sum Sq Mean Sq F value   Pr(>F)
Variety            1 1.0838  1.0838  30.127 0.000386 ***
Fertilizer:Variety 2 0.3675  0.1837   5.108 0.032930 *
Residuals          9 0.3238  0.0360
---
Signif.codes:0‘***’0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The script below is using package “Agricolae”, the result is the same thing.

```
> library(agricolae)
> attach(data)
> modelb <- sp.plot(Block,Fertilizer,Variety,Production)

ANALYSIS SPLIT PLOT:  Production
Class level information

Fertilizer      :  2 3 1
Variety         :  2 1
Block   :  1 2 3 4
```

```
Number of observations:  24

Analysis of Variance Table

Response: Production
                   Df  Sum Sq Mean Sq F value    Pr(>F)
Block               3  0.1346  0.0449  0.2777 0.8397921
Fertilizer          2 16.1875  8.0938 50.1075 0.0001803 ***
Ea                  6  0.9692  0.1615
Variety             1  1.0838  1.0838 30.1274 0.0003857 ***
Fertilizer:Variety  2  0.3675  0.1837  5.1081 0.0329296 *
Eb                  9  0.3238  0.0360
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

cv(a) = 6.2 %, cv(b) = 2.9 %, Mean = 6.4875

>
```

11.4 Split-split-plot Design

Split-split-plot design is an extension of the split plot design with addition of other factor (third factor). This design has characteristic that there are three plot sizes, namely main plot (the largest plot), sub plot (intermediate plot) and sub-subplot (the smallest plot). In addition, there are three levels of precision, with the main-plot factor having the lowest degree of precision and the sub-subplot factor having the highest degree of precision.

The following example is grain yields of three rice varieties grown under three management practices and five nitrogen levels (Gomez and Gomez, 1984). The experiment is designed in a split-split-plot design with nitrogen as main-plot, management practice as subplot, and variety as sub-subplot factors, with three replications.

Management	Variety								
	V1			V2			V2		
	Rep.I	Rep.II	Rep.III	Rep.I	Rep.II	Rep.III	Rep.I	Rep.II	Rep.III
	N1 (0 kg N/ha)								
M1	3.320	3.864	4.507	6.101	5.122	4.815	5.355	5.536	5.244
M2	3.766	4.311	4.875	5.096	4.873	4.166	7.442	6.462	5.584
M3	4.660	5.915	5.400	6.573	5.495	4.225	7.018	8.020	7.642
	N2 (50 kg N/ha)								
M1	3.188	4.752	4.756	5.595	6780	5.390	6.706	6.546	7.092
M2	3.625	4.809	5.295	6.357	5.925	5.163	8.592	7.646	7.212

M3	5.232	5.170	6.046	7.016	7.442	4.478	8.480	9.942	8.714
	N3 (80 kg N/ha)								
M1	5.468	5.788	4.422	5.442	5.988	6.509	8.452	6.698	8.650
M2	5.759	6.130	5.308	6.398	6.533	6.560	8.662	8.526	8.514
M3	6.215	7.106	6.318	6.953	6.914	7.991	9.112	9.140	9.320
	N4 (110 kg N/ha)								
M1	4.246	4.842	4.863	6.209	6.768	5.779	8.042	7.414	6.902
M2	5.255	5.742	5.345	6.992	7.856	6.164	9.080	9.016	7.778
M3	6.829	5.869	6.011	7.565	7.626	7.362	9.660	8.966	9.128
	N5 (140 kg N/ha)								
M1	3.132	4.375	4.678	6.860	6.894	6.573	9.314	8.508	8.032
M2	5.389	4.315	5.896	6.857	6.974	7.422	9.224	9.680	9.294
M3	5.217	5.389	7.309	7.254	7.812	8.950	10.360	9.896	9.712

In R:

```
> data <- read.csv('splitsplitplot.csv', header=T)
> data
    Management Variety Nitrogen Block  yield
1           M1      V1       N1    R1  3.320
2           M2      V1       N1    R1  3.766
3           M3      V1       N1    R1  4.660
4           M1      V1       N2    R1  3.188
5           M2      V1       N2    R1  3.625
6           M3      V1       N2    R1  5.232
7           M1      V1       N3    R1  5.468
8           M2      V1       N3    R1  5.759
9           M3      V1       N3    R1  6.215
10          M1      V1       N4    R1  4.246
11          M2      V1       N4    R1  5.255
12          M3      V1       N4    R1  6.829
13          M1      V1       N5    R1  3.132
14          M2      V1       N5    R1  5.389
15          M3      V1       N5    R1  5.217
16          M1      V1       N1    R2  3.864
17          M2      V1       N1    R2  4.311
18          M3      V1       N1    R2  5.915
19          M1      V1       N2    R2  4.752
20          M2      V1       N2    R2  4.809
21          M3      V1       N2    R2  5.170
22          M1      V1       N3    R2  5.788
23          M2      V1       N3    R2  6.130
24          M3      V1       N3    R2  7.106
25          M1      V1       N4    R2  4.842
26          M2      V1       N4    R2  5.742
27          M3      V1       N4    R2  5.869
28          M1      V1       N5    R2  4.375
29          M2      V1       N5    R2  4.315
```

30	M3	V1	N5	R2	5.389
31	M1	V1	N1	R3	4.507
32	M2	V1	N1	R3	4.875
33	M3	V1	N1	R3	5.400
34	M1	V1	N2	R3	4.756
35	M2	V1	N2	R3	5.295
36	M3	V1	N2	R3	6.046
37	M1	V1	N3	R3	4.422
38	M2	V1	N3	R3	5.308
39	M3	V1	N3	R3	6.318
40	M1	V1	N4	R3	4.863
41	M2	V1	N4	R3	5.345
42	M3	V1	N4	R3	6.011
43	M1	V1	N5	R3	4.678
44	M2	V1	N5	R3	5.896
45	M3	V1	N5	R3	7.309
46	M1	V2	N1	R1	6.101
47	M2	V2	N1	R1	5.096
48	M3	V2	N1	R1	6.573
49	M1	V2	N2	R1	5.595
50	M2	V2	N2	R1	6.357
51	M3	V2	N2	R1	7.016
52	M1	V2	N3	R1	5.442
53	M2	V2	N3	R1	6.398
54	M3	V2	N3	R1	6.953
55	M1	V2	N4	R1	6.209
56	M2	V2	N4	R1	6.992
57	M3	V2	N4	R1	7.565
58	M1	V2	N5	R1	6.860
59	M2	V2	N5	R1	6.857
60	M3	V2	N5	R1	7.254
61	M1	V2	N1	R2	5.122
62	M2	V2	N1	R2	4.873
63	M3	V2	N1	R2	5.495
64	M1	V2	N2	R2	6.780
65	M2	V2	N2	R2	5.925
66	M3	V2	N2	R2	7.442
67	M1	V2	N3	R2	5.988
68	M2	V2	N3	R2	6.533
69	M3	V2	N3	R2	6.914
70	M1	V2	N4	R2	6.768
71	M2	V2	N4	R2	7.856
72	M3	V2	N4	R2	7.626
73	M1	V2	N5	R2	6.894
74	M2	V2	N5	R2	6.974
75	M3	V2	N5	R2	7.812
76	M1	V2	N1	R3	4.815
77	M2	V2	N1	R3	4.166
78	M3	V2	N1	R3	4.225

79	M1	V2	N2	R3	5.390
80	M2	V2	N2	R3	5.163
81	M3	V2	N2	R3	4.478
82	M1	V2	N3	R3	6.509
83	M2	V2	N3	R3	6.569
84	M3	V2	N3	R3	7.991
85	M1	V2	N4	R3	5.779
86	M2	V2	N4	R3	6.164
87	M3	V2	N4	R3	7.362
88	M1	V2	N5	R3	6.573
89	M2	V2	N5	R3	7.422
90	M3	V2	N5	R3	8.950
91	M1	V3	N1	R1	5.355
92	M2	V3	N1	R1	7.442
93	M3	V3	N1	R1	7.018
94	M1	V3	N2	R1	6.706
95	M2	V3	N2	R1	8.592
96	M3	V3	N2	R1	8.480
97	M1	V3	N3	R1	8.452
98	M2	V3	N3	R1	8.662
99	M3	V3	N3	R1	9.112
100	M1	V3	N4	R1	8.042
101	M2	V3	N4	R1	9.080
102	M3	V3	N4	R1	9.660
103	M1	V3	N5	R1	9.314
104	M2	V3	N5	R1	9.224
105	M3	V3	N5	R1	10.360
106	M1	V3	N1	R2	5.536
107	M2	V3	N1	R2	6.462
108	M3	V3	N1	R2	8.020
109	M1	V3	N2	R2	6.546
110	M2	V3	N2	R2	7.646
111	M3	V3	N2	R2	9.942
112	M1	V3	N3	R2	6.698
113	M2	V3	N3	R2	8.526
114	M3	V3	N3	R2	9.140
115	M1	V3	N4	R2	7.414
116	M2	V3	N4	R2	9.016
117	M3	V3	N4	R2	8.966
118	M1	V3	N5	R2	8.508
119	M2	V3	N5	R2	9.680
120	M3	V3	N5	R2	9.896
121	M1	V3	N1	R3	5.244
122	M2	V3	N1	R3	5.584
123	M3	V3	N1	R3	7.642
124	M1	V3	N2	R3	7.092
125	M2	V3	N2	R3	7.212
126	M3	V3	N2	R3	8.714
127	M1	V3	N3	R3	8.650

```
128          M2       V3        N3     R3  8.514
129          M3       V3        N3     R3  9.320
130          M1       V3        N4     R3  6.902
131          M2       V3        N4     R3  7.778
132          M3       V3        N4     R3  9.128
133          M1       V3        N5     R3  8.032
134          M2       V3        N5     R3  9.294
135          M3       V3        N5     R3  9.712
> str(data)
'data.frame':   135 obs. of  5 variables:
 $ Management: Factor w/ 3 levels "M1","M2","M3": 1 2 3 1 2 3 1 2 3 1 ...
 $ Variety   : Factor w/ 3 levels "V1","V2","V3": 1 1 1 1 1 1 1 1 1 1 ...
 $ Nitrogen  : Factor w/ 5 levels "N1","N2","N3",..: 1 1 1 2 2 2 3 3 3 4 ...
 $ Block     : Factor w/ 3 levels "R1","R2","R3": 1 1 1 1 1 1 1 1 1 1 ...
 $ yield     : num  3.32 3.77 4.66 3.19 3.62 ...
>
> #splitplot rcbd
>   fit   <-   aov(yield   ~      Block   +   Nitrogen*Management*   Variety   +
Error(Block/Nitrogen/Management),data=data)
> summary(fit)

Error: Block
      Df Sum Sq Mean Sq
Block  2  0.732   0.366

Error: Block:Nitrogen
          Df Sum Sq Mean Sq F value   Pr(>F)
Nitrogen   4  61.64  15.410    27.7 9.73e-05 ***
Residuals  8   4.45   0.556
---
Signif.codes:0‘***’0.001‘**’0.01‘*’0.05‘.’0.1 ‘ ’ 1

Error: Block:Nitrogen:Management
                 Df Sum Sq Mean Sq F value  Pr(>F)
Management        2  42.94  21.468  81.996 2.3e-10 ***
Nitrogen:Management 8   1.10   0.138   0.527   0.823
Residuals        20   5.24   0.262
---
Signif.codes:0‘***’0.001‘**’0.01‘*’0.05‘.’0.1 ‘ ’ 1

Error: Within
                          Df Sum Sq Mean Sq F value  Pr(>F)
Variety                    2 206.01  103.01 207.867 < 2e-16 ***
Nitrogen:Variety           8  14.14    1.77   3.568 0.00192 **
Management:Variety         4   3.85    0.96   1.943 0.11490
Nitrogen:Management:Variety 16   3.70    0.23   0.467 0.95376
Residuals                  60  29.73    0.50
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Using Agricolae package resulted in the same thing.

```
> library(agricolae)
> attach(data)
The following objects are masked from data (pos = 3):

    Block, Variety

> modelb <- ssp.plot(Block,Nitrogen,Management,Variety,yield)

ANALYSIS SPLIT-SPLIT PLOT:  yield
Class level information

Nitrogen        :  N1 N2 N3 N4 N5
Management      :  M1 M2 M3
Variety         :  V1 V2 V3
Block   :  R1 R2 R3
```

```
Number of observations:  135

Analysis of Variance Table

Response: yield
                            Df  Sum Sq Mean Sq  F value    Pr(>F)
Block                        2   0.732   0.366   0.6578  0.543910
Nitrogen                     4  61.641  15.410  27.6953 9.734e-05 ***
Ea                           8   4.451   0.556
Management                   2  42.936  21.468  81.9965 2.303e-10 ***
Nitrogen:Management          8   1.103   0.138   0.5266  0.822648
Eb                          20   5.236   0.262
Variety                      2 206.013 103.007 207.8667 < 2.2e-16 ***
Variety:Nitrogen             8  14.145   1.768   3.5679  0.001916 **
Variety:Management           4   3.852   0.963   1.9432  0.114899
Variety:Nitrogen:Management 16   3.699   0.231   0.4666  0.953759
Ec                          60  29.732   0.496
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

cv(a) = 11.4 %, cv(b) = 7.8 %, cv(c) = 10.7 %, Mean = 6.554415

>
```

11.5 Strip Plot Design

Strip plot design is when each of two factors require larger experimental units to be tested in the same experiment. For example, factor *A* is applied to whole plots like the usual split plot designs but factor *B* is also applied to strips which are actually a new set of whole plots orthogonal to the original plots used for factor *A*. These designs are also called Split Block Designs. Figure below is an example of strip plot design where factor A has four levels and factor B has three levels.

Factor B	Factor A			
	A4	A2	A1	A3
B2	A4B2	A2B2	A1B2	A3B2
B3	A4B3	A2B3	A1B3	A3B3
B1	A4B1	A2B1	A1B1	A3B1

Precision of the interaction effect between the two factors is higher than that of the main effect of either one of the two factors. In other words, the degrees of precision of the main effects of the two factors are sacrificed in order to improve the precision of the interaction effect. Experimental units for these design are the units for effects of factor A and B which are equal to whole plot of each factor and the experimental unit for interaction AB which is a subplot or the intersection of the two whole plots. So the model for this design is:

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \gamma_k + (\tau\gamma)_{ik} + (\beta\gamma)_{jk} + \varepsilon_{ijk} \qquad i = 1,...,a;\ j = 1,...,r;\ k = 1,...,b$$

whrere

yijk = observation of ith level of factor A, kth level of factor B and jth replication
μ = general mean
βj = jth block effect
τi = ith level of factor A effect
γk = kth level of factor B effect
(τγ)ik : interaction between ith level of factor A and the kth level of factor B
$(\tau\beta)_{ij}$, $(\tau\gamma)_{ik}$ and ε_{ijk} are the errors to be used to test Factor *A*, Factor *B* and interaction *AB*, respectively.

ANOVA table for this design will be like table below.

Source of variation	df	Sum of squares	F statistik
Replication (blocks)	r – 1	SSblock	
A	a – 1	SSA	MSA/MSwpA
Whole plot error A	(r – 1)(a – 1)	SSwpA	
B	b – 1	SSB	MSB/MSwpB
Whole plot error B	(r – 1)(b – 1)	SSwpB	
AB	(a – 1)(b – 1)	SSAB	MSAB/MSsp
Sub plot error	(r – 1)(a – 1)(b – 1)	SSsp	
Total	rab – 1	SST	

For example, we use data of grain yield of six varieties of rice, broadcast seeded and grown with three nitrogen rates in a strip plot design with three replications, as shown in table below (Gomez and Gomez, 1984).

Nitrogen rate (kg/ha)	Grain yield (kg/ha)		
	Rep.I	Rep.II	Rep.III
	IR8 (V1)		
0 (N1)	2373	3958	4384
60 (N2)	4076	6431	4889
120 (N3)	7254	6808	8582
	IR127-80 (V2)		
0 (N1)	4007	5795	5001
60 (N2)	5630	7334	7177
120 (N3)	7053	8284	6297
	IR305-4-12 (V3)		
0 (N1)	2620	4508	5621
60 (N2)	4676	6672	7019
120 (N3)	7666	7328	8611
	IR400-2-5 (V4)		
0 (N1)	2726	5630	3821

60 (N2)	4838	7007	4816
120 (N3)	6881	7735	6667
	IR665-58 (V5)		
0 (N1)	4447	3276	4582
60 (N2)	5549	5340	6011
120 (N3)	6880	5080	6076
	Peta (V6)		
0 (N1)	2572	3724	3326
60 (N2)	3896	2822	4425
120 (N3)	1556	2706	3214

In R:

```
> data=read.csv("stripplot.csv", header=T)
> head(data)
  Nitrogen Variety Replication Yield
1       N1      V1       Rep.I  2373
2       N2      V1       Rep.I  4076
3       N3      V1       Rep.I  7254
4       N1      V2       Rep.I  4007
5       N2      V2       Rep.I  5630
6       N3      V2       Rep.I  7053
> data
   Nitrogen Variety Replication Yield
1        N1      V1       Rep.I  2373
2        N2      V1       Rep.I  4076
3        N3      V1       Rep.I  7254
4        N1      V2       Rep.I  4007
5        N2      V2       Rep.I  5630
6        N3      V2       Rep.I  7053
7        N1      V3       Rep.I  2620
8        N2      V3       Rep.I  4676
9        N3      V3       Rep.I  7666
10       N1      V4       Rep.I  2726
11       N2      V4       Rep.I  4838
12       N3      V4       Rep.I  6881
13       N1      V5       Rep.I  4447
14       N2      V5       Rep.I  5549
15       N3      V5       Rep.I  6880
16       N1      V6       Rep.I  2572
17       N2      V6       Rep.I  3896
18       N3      V6       Rep.I  1556
19       N1      V1      Rep.II  3958
20       N2      V1      Rep.II  6431
21       N3      V1      Rep.II  6808
22       N1      V2      Rep.II  5795
23       N2      V2      Rep.II  7334
24       N3      V2      Rep.II  8284
25       N1      V3      Rep.II  4508
```

```
26       N2      V3      Rep.II  6672
27       N3      V3      Rep.II  7328
28       N1      V4      Rep.II  5630
29       N2      V4      Rep.II  7007
30       N3      V4      Rep.II  7735
31       N1      V5      Rep.II  3276
32       N2      V5      Rep.II  5340
33       N3      V5      Rep.II  5080
34       N1      V6      Rep.II  3724
35       N2      V6      Rep.II  2822
36       N3      V6      Rep.II  2706
37       N1      V1     Rep.III  4384
38       N2      V1     Rep.III  4889
39       N3      V1     Rep.III  8582
40       N1      V2     Rep.III  5001
41       N2      V2     Rep.III  7177
42       N3      V2     Rep.III  6297
43       N1      V3     Rep.III  5621
44       N2      V3     Rep.III  7019
45       N3      V3     Rep.III  8611
46       N1      V4     Rep.III  3821
47       N2      V4     Rep.III  4816
48       N3      V4     Rep.III  6667
49       N1      V5     Rep.III  4582
50       N2      V5     Rep.III  6011
51       N3      V5     Rep.III  6076
52       N1      V6     Rep.III  3326
53       N2      V6     Rep.III  4425
54       N3      V6     Rep.III  3214
> stripplot <- aov(Yield ~ Variety * Nitrogen +
Error(Replication      +      Replication:Variety      +
Replication:Nitrogen), data=data)
> summary(stripplot)

Error: Replication
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  2 9220962 4610481

Error: Replication:Variety
          Df   Sum Sq  Mean Sq F value  Pr(>F)
Variety    5 57100201 11420040   7.653 0.00337 **
Residuals 10 14922619  1492262
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

Error: Replication:Nitrogen
          Df   Sum Sq  Mean Sq F value  Pr(>F)
Nitrogen   2 50676061 25338031   34.07 0.00307 **
```

```
Residuals  4  2974908   743727
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

Error: Within
                 Df   Sum Sq Mean Sq F value   Pr(>F)
Variety:Nitrogen 10 23877979 2387798   5.801 0.000427 ***
Residuals        20  8232917  411646
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1
>
```

If using Agricolae package:

```
> library(agricolae)
> with(data, strip.plot(Replication, Nitrogen, Variety,
Yield))

ANALYSIS STRIP PLOT:  Yield
Class level information

Nitrogen        :  N1 N2 N3
Variety         :  V1 V2 V3 V4 V5 V6
Replication     :  Rep.I Rep.II Rep.III

Number of observations:  54

model Y: Yield ~ Replication + Nitrogen + Ea + Variety +
Eb + Variety:Nitrogen + Ec

Analysis of Variance Table

Response: Yield
                Df   Sum Sq  Mean Sq F value    Pr(>F)
Replication      2  9220962  4610481 11.2001 0.0005453 ***
Nitrogen         2 50676061 25338031 34.0690 0.0030746 **
Ea               4  2974908   743727  1.8067 0.1671590
Variety          5 57100201 11420040  7.6528 0.0033722 **
Eb              10 14922619  1492262  3.6251 0.0068604 **
Variety:Nitrogen 10 23877979  2387798  5.8006 0.0004271 ***
Ec              20  8232917   411646
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

cv(a) = 16.3 %, cv(b) = 23.1 %, cv(c) = 12.1 %, Mean = 5289.944

>
```

11.6 Strip Split Plot Design

Strip split plot design is an extension of the strip plot design where the intersection plot is divided into subplots for the third factor. For example, we use grain yields of six rice varieties which is treated under two planting methods and three nitrogen rates in a strip split plot design with three replications, as shown in table below (Gomez and Gomez, 1984).

Variety	Grain yield (kg/ha)					
	P1 (Broadcast)			P2 (Transplanted)		
	Rep.I	Rep.II	Rep.III	Rep.I	Rep.II	Rep.III
	N1 (0 kg N/ha)					
V1(IR8)	2373	3958	4384	2293	3528	2538
V2((IR127-8-1-10)	4007	5795	5001	4035	4885	4583
V3(IR305-4-12-1-3)	2620	4508	5621	4527	4866	3628
V4(IR400-2-5-3-3-2)	2726	5630	3821	5274	6200	4038
V5(IR665-58)	4447	3276	4582	4655	2796	3739
V6(Peta)	2572	3724	3326	4535	5457	3537
	N2 (60 kg N/ha)					
V1	4076	6431	4889	3085	7502	4362
V2	5630	7334	7177	3728	7424	5377
V3	4676	6672	7019	4946	7611	6142
V4	4838	7007	4816	4878	6928	4829
V5	5549	5340	6011	4646	5006	4666
V6	3896	2822	4425	4627	4461	4774
	N3 (120 kg N/ha)					
V1	7254	6808	8582	6661	6353	7759
V2	7053	8284	6297	6440	7648	5736
V3	7666	7328	8611	8632	7101	7416
V4	6881	7735	6667	6545	9838	7253
V5	6880	5080	6076	6995	4486	6564
V6	1556	2706	3214	5374	7218	6369

In R:

```
> data=read.csv("stripsplitplot.csv", header=T)
> head(data)
  Method Variety Nitrogen Replication Yield
1     MB      V1       N1       Rep.I  2373
2     MB      V2       N1       Rep.I  4007
3     MB      V3       N1       Rep.I  2620
4     MB      V4       N1       Rep.I  2726
5     MB      V5       N1       Rep.I  4447
```

```
6     MB      V6       N1       Rep.I  2572
> data
    Method Variety Nitrogen Replication Yield
1       MB      V1       N1       Rep.I  2373
2       MB      V2       N1       Rep.I  4007
3       MB      V3       N1       Rep.I  2620
4       MB      V4       N1       Rep.I  2726
5       MB      V5       N1       Rep.I  4447
6       MB      V6       N1       Rep.I  2572
7       MB      V1       N2       Rep.I  4076
8       MB      V2       N2       Rep.I  5630
9       MB      V3       N2       Rep.I  4676
10      MB      V4       N2       Rep.I  4838
11      MB      V5       N2       Rep.I  5549
12      MB      V6       N2       Rep.I  3896
13      MB      V1       N3       Rep.I  7254
14      MB      V2       N3       Rep.I  7053
15      MB      V3       N3       Rep.I  7666
16      MB      V4       N3       Rep.I  6881
17      MB      V5       N3       Rep.I  6880
18      MB      V6       N3       Rep.I  1556
19      MB      V1       N1      Rep.II  3958
20      MB      V2       N1      Rep.II  5795
21      MB      V3       N1      Rep.II  4508
22      MB      V4       N1      Rep.II  5630
23      MB      V5       N1      Rep.II  3276
24      MB      V6       N1      Rep.II  3724
25      MB      V1       N2      Rep.II  6431
26      MB      V2       N2      Rep.II  7334
27      MB      V3       N2      Rep.II  6672
28      MB      V4       N2      Rep.II  7007
29      MB      V5       N2      Rep.II  5340
30      MB      V6       N2      Rep.II  2822
31      MB      V1       N3      Rep.II  6808
32      MB      V2       N3      Rep.II  8284
33      MB      V3       N3      Rep.II  7328
34      MB      V4       N3      Rep.II  7735
35      MB      V5       N3      Rep.II  5080
36      MB      V6       N3      Rep.II  2706
37      MB      V1       N1     Rep.III  4384
38      MB      V2       N1     Rep.III  5001
39      MB      V3       N1     Rep.III  5621
40      MB      V4       N1     Rep.III  3821
41      MB      V5       N1     Rep.III  4582
42      MB      V6       N1     Rep.III  3326
43      MB      V1       N2     Rep.III  4889
44      MB      V2       N2     Rep.III  7177
45      MB      V3       N2     Rep.III  7019
46      MB      V4       N2     Rep.III  4816
```

47	MB	V5	N2	Rep.III	6011
48	MB	V6	N2	Rep.III	4425
49	MB	V1	N3	Rep.III	8582
50	MB	V2	N3	Rep.III	6297
51	MB	V3	N3	Rep.III	8611
52	MB	V4	N3	Rep.III	6667
53	MB	V5	N3	Rep.III	6076
54	MB	V6	N3	Rep.III	3214
55	MT	V1	N1	Rep.I	2293
56	MT	V2	N1	Rep.I	4035
57	MT	V3	N1	Rep.I	4527
58	MT	V4	N1	Rep.I	5274
59	MT	V5	N1	Rep.I	4655
60	MT	V6	N1	Rep.I	4535
61	MT	V1	N2	Rep.I	3085
62	MT	V2	N2	Rep.I	3728
63	MT	V3	N2	Rep.I	4946
64	MT	V4	N2	Rep.I	4878
65	MT	V5	N2	Rep.I	4646
66	MT	V6	N2	Rep.I	4627
67	MT	V1	N3	Rep.I	6661
68	MT	V2	N3	Rep.I	6440
69	MT	V3	N3	Rep.I	8632
70	MT	V4	N3	Rep.I	6545
71	MT	V5	N3	Rep.I	6995
72	MT	V6	N3	Rep.I	5374
73	MT	V1	N1	Rep.II	3528
74	MT	V2	N1	Rep.II	4885
75	MT	V3	N1	Rep.II	4866
76	MT	V4	N1	Rep.II	6200
77	MT	V5	N1	Rep.II	2796
78	MT	V6	N1	Rep.II	5457
79	MT	V1	N2	Rep.II	7502
80	MT	V2	N2	Rep.II	7424
81	MT	V3	N2	Rep.II	7611
82	MT	V4	N2	Rep.II	6928
83	MT	V5	N2	Rep.II	5006
84	MT	V6	N2	Rep.II	4461
85	MT	V1	N3	Rep.II	6353
86	MT	V2	N3	Rep.II	7648
87	MT	V3	N3	Rep.II	7101
88	MT	V4	N3	Rep.II	9838
89	MT	V5	N3	Rep.II	4486
90	MT	V6	N3	Rep.II	7218
91	MT	V1	N1	Rep.III	2538
92	MT	V2	N1	Rep.III	4583
93	MT	V3	N1	Rep.III	3628
94	MT	V4	N1	Rep.III	4038
95	MT	V5	N1	Rep.III	3739

```
96      MT      V6       N1     Rep.III  3537
97      MT      V1       N2     Rep.III  4362
98      MT      V2       N2     Rep.III  5377
99      MT      V3       N2     Rep.III  6142
100     MT      V4       N2     Rep.III  4829
101     MT      V5       N2     Rep.III  4666
102     MT      V6       N2     Rep.III  4774
103     MT      V1       N3     Rep.III  7759
104     MT      V2       N3     Rep.III  5736
105     MT      V3       N3     Rep.III  7416
106     MT      V4       N3     Rep.III  7253
107     MT      V5       N3     Rep.III  6564
108     MT      V6       N3     Rep.III  6369
> stripsplitplot <- aov(Yield ~ Method*Variety * Nitrogen
+ Error(Replication +
+         Replication:Variety  +  Replication:Nitrogen  +
Replication:Nitrogen:Variety),
+    data=data)
> summary(stripsplitplot)

Error: Replication
          Df   Sum Sq Mean Sq F value Pr(>F)
Residuals  2 15289498 7644749

Error: Replication:Variety
          Df   Sum Sq Mean Sq F value Pr(>F)
Variety    5 49119270 9823854   3.676 0.0379 *
Residuals 10 26721828 2672183
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Replication:Nitrogen
          Df    Sum Sq  Mean Sq F value  Pr(>F)
Nitrogen   2 116489166 58244583   36.62 0.00268 **
Residuals  4   6361491  1590373
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Replication:Variety:Nitrogen
                 Df   Sum Sq Mean Sq F value Pr(>F)
Variety:Nitrogen 10 24595731 2459573   2.575 0.0344 *
Residuals        20 19106733  955337
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Within
                       Df   Sum Sq Mean Sq F value   Pr(>F)
Method                  1   723079  723079   1.715   0.1986
Method:Variety          5 23761441 4752288  11.271 1.37e-06 ***
Method:Nitrogen         2  2468132 1234066   2.927   0.0664 .
Method:Variety:Nitrogen 10  7512072  751207   1.782   0.1000 .
Residuals              36 15179354  421649
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

XII. NESTED DESIGN

12.1 Introduction

Nested design which is also known as hierarchical design is used for experiments in which there is an interest in a set of treatments and the experimental units are sub-sampled. In other words, nested design is that levels of one factor is a subset of a level of another factor. For example, a researcher want to investigate the effect of different bulls and cows on birth weight of their calves. In this research, each of several bulls mated with several cows generated some calves. So, in this case several calves are nested to a cow and several cows are nested to a bull. Other example, four different seedlings have been sampled from four different flowers in three different fields A, B and C, where seedlings are nested to a flower and flowers are nested to a field.

Nested design is common in genetics, systematics, and evolutionary studies where it is important to keep track of each plant or animal obtained from specific populations, lines, or parentage. Furthermore, each parent and each offspring is given a unique identity because it is not replicated across a treatment.

12.2 Nested Design with Two Factors

For example, consider four bulls breed are levels of factor *A*, and three levels of factor *B* are different cows mated to those bulls. The cows are a random sample within the bulls. Birth weight of their offspring (three calves each) was measured where these calves represent random samples within the cows. Relationship among the cows is ignored and the cows bred by different bulls are independent, and also the offspring of different cows and bulls are independent of each other. The scheme of this nested design is follows.

Bull	A			B			C			D		
Cow	1	2	3	4	5	6	7	8	9	10	11	12
Calf	1	4	7	10	13	16	19	22	25	28	31	34
	2	5	8	11	14	17	20	23	26	29	32	35
	3	6	9	12	15	18	21	24	27	30	33	36

The model for this nested design is:

$$yijk = \mu + Ai + B(A)ij + \varepsilon ijk \quad i = 1,...,a; \quad j = 1,...,b; \quad k = 1,...,n$$

where:

$yijk$ = observation k in level i of factor A and level j of factor B

μ = overall mean

Ai = effect of level i of factor A (bull)

$B(A)ij$ = effect of level j of factor B (cow) within level i of factor A

εijk = random error

a = number of levels of A (bull)

b = the number of levels of B (cow)

n = the number of observations per level of B

Similarly to the other designs, the total sum of squares can be partitioned into the sums of squares of each source of variability. They are the sum of squares for factor A, the sum of squares for factor B within factor A, and the sum of squares within B (the residual sum of squares):

$$SST = SSA + SSB(A) + SS\ within\ B$$

The corresponding degrees of freedom are:

$$(abn\text{-}1) = (a\text{-}1) + a(b\text{-}1) + ab(n\text{-}1)$$

The sums of squares are:

where

$$\text{SST} = \sum_i \sum_j \sum_k (yijk - \bar{y}...)^2$$

$$\text{SSA} = \sum_i \sum_j \sum_k (\bar{y}i.. - \bar{y}...)^2$$

$$\text{SSB(A)} = \sum_i \sum_j \sum_k (\bar{y}ij. - \bar{y}i..)^2$$

$$\text{SS within B} = \sum_i \sum_j \sum_k (\bar{y}ijk - \bar{y}ij.)^2$$

Sums of squares above can be calculated using computation below.

$$\text{CF} = \frac{\left(\sum_i \sum_j \sum_k yijk\right)^2}{a.b.n}$$

$$\text{SST} = \sum_i \sum_j \sum_k (yijk)^2 - CF$$

$$\text{SSA} = \sum_i \frac{\left(\sum_j \sum_k yijk\right)^2}{n.b} - CF$$

$$\text{SSB(A)} = \sum_i \sum_j \frac{(\sum_k yijk)^2}{n} - SSA - CF$$

$$\text{SS } within\ B = SSE = SST - SSA - SSB(A)$$

Mean squares (*MS*) and the ANOVA table is:

Source of variation	SS	df	MS = SS/df
A	SSA	a – 1	MSA
B within A	SSB(A)	a(b – 1)	MSB(A)
Within B	SS within B	ab(n – 1)	MS within B
Total	SST	abn – 1	

The effect "*Within B"* is residual. Expectations of mean squares, *E*(*MS*), can be seen in the following table.

E(MS)	Variance component
E(MSA)	$\sigma^2 + n\sigma^2 B + nb\sigma^2 A$
E(MSB(A))	$\sigma^2 + n\sigma^2 B$
E(MS within B)	σ^2

F statistic for the effect of factor A is:

$$\text{F} = \frac{MSA}{MSB(A)}$$

F statistic for the effect of factor B is:

$$\text{F} = \frac{MSB(A)}{MS\ within\ B}$$

For example, forest geneticists want to know whether the origin of trees (different forests) affects growth (tree height). The researcher collected 5 seeds from 3 superior trees from 5 types of forest. The seeds are germinated in a greenhouse and the seedlings are measured for height growth. Data from measurements of tree height are presented in the following table.

Tree	Forest				
	A	B	C	D	E
1	15.9	18.6	12.4	19.5	16.1
	15.7	18.1	13.1	17.6	15.8
	16.1	18.5	12.8	19.2	16.2
2	14.0	18.0	14.1	18.8	15.9
	14.3	18.1	13.2	19.1	15.7
	13.6	17.5	13.6	18.9	16.4
3	14.0	17.9	13.2	18.3	15.9
	15.2	18.8	14.4	17.9	16.7
	15.8	18.3	12.9	19.4	15.7

Table above can be arranged as table below.

Forest	Tree	Height	Sum for Forest	Sum for Tree	Total
A	T1	15.9	134.6	47.7	731.2
A	T1	15.7			
A	T1	16.1			
A	T2	14		41.9	
A	T2	14.3			
A	T2	13.6			
A	T3	14		45	
A	T3	15.2			
A	T3	15.8			
B	T1	18.6	163.8	55.2	
B	T1	18.1			
B	T1	18.5			
B	T2	18		53.6	
B	T2	18.1			
B	T2	17.5			
B	T3	17.9		55	
B	T3	18.8			
B	T3	18.3			
C	T1	12.4	119.7	38.3	
C	T1	13.1			
C	T1	12.8			
C	T2	14.1		40.9	
C	T2	13.2			
C	T2	13.6			
C	T3	13.2		40.5	
C	T3	14.4			

C	T3	12.9			
D	T1	19.5	168.7	56.3	
D	T1	17.6			
D	T1	19.2			
D	T2	18.8		56.8	
D	T2	19.1			
D	T2	18.9			
D	T3	18.3		55.6	
D	T3	17.9			
D	T3	19.4			
E	T1	16.1	144.4	48.1	
E	T1	15.8			
E	T1	16.2			
E	T2	15.9		48	
E	T2	15.7			
E	T2	16.4			
E	T3	15.9		48.3	
E	T3	16.7			
E	T3	15.7			

$$\text{CF} = \frac{\left(\sum_i \sum_j \sum_k yijk\right)^2}{a.b.n} = \frac{731.2^2}{5.3.3} = 11881.19$$

$$\text{SST} = \sum_i \sum_j \sum_k (yijk)^2 \; - CF = (15.9^2 + \ldots + 15.7^2) - CF$$

$$= 200.6124$$

$$\text{SSA} = \sum_i \frac{\left(\sum_j \sum_k yijk\right)^2}{n.b} \; - \; CF = \frac{(134.6^2 + \ldots + 144.4^2)}{3.3} - CF$$

$$= 184.0058$$

$$\text{SSB(A)} = \sum_i \sum_j \frac{(\sum_k yijk)^2}{n} \; - \; SSA - \; CF$$

$$= \frac{(47.7^2 + \ldots 48.3^2)}{3} - SSA - CF = 7.686667$$

$$\text{SS } within\ B = SSE = SST \; - \; SSA \; - \; SSB(A) = 8.92$$

In R:

```
> data=read.csv("nested2stagesForest.csv", header=TRUE)
> data
   Forest Tree Height
1       A   T1   15.9
2       A   T1   15.7
3       A   T1   16.1
4       A   T2   14.0
5       A   T2   14.3
6       A   T2   13.6
7       A   T3   14.0
8       A   T3   15.2
9       A   T3   15.8
10      B   T4   18.6
11      B   T4   18.1
12      B   T4   18.5
13      B   T5   18.0
14      B   T5   18.1
15      B   T5   17.5
16      B   T6   17.9
17      B   T6   18.8
18      B   T6   18.3
19      C   T7   12.4
20      C   T7   13.1
21      C   T7   12.8
22      C   T8   14.1
23      C   T8   13.2
24      C   T8   13.6
25      C   T9   13.2
26      C   T9   14.4
27      C   T9   12.9
28      D  T10   19.5
29      D  T10   17.6
30      D  T10   19.2
31      D  T11   18.8
32      D  T11   19.1
33      D  T11   18.9
34      D  T12   18.3
35      D  T12   17.9
36      D  T12   19.4
37      E  T13   16.1
38      E  T13   15.8
39      E  T13   16.2
40      E  T14   15.9
41      E  T14   15.7
42      E  T14   16.4
43      E  T15   15.9
44      E  T15   16.7
```

```
45       E   T15    15.7
> str(data)
'data.frame':   45 obs. of  3 variables:
 $ Forest: Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 2 ...
 $ Tree  : Factor w/ 15 levels "T1","T10","T11",..: 1 1 1 8 8 8 9 9 9 10 ...
 $ Height: num  15.9 15.7 16.1 14 14.3 13.6 14 15.2 15.8 18.6 ...

> nested=aov(Height~Forest/Tree, data=data)
> summary(nested)
             Df Sum Sq Mean Sq F value Pr(>F)
Forest        4 184.01   46.00 154.713 <2e-16 ***
Forest:Tree  10   7.69    0.77   2.585 0.0216 *
Residuals    30   8.92    0.30
---
Signif. codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘ ’ 1
```

Expectations of mean squares, *E*(*MS*), can be seen in the following table.

Source	E(MS)	Variance component
Forest	$\sigma^2 + n\sigma^2 Tree + nb\sigma^2 Forest$	$46 = 0.3 + 3(0.1567) + 9\sigma^2 Forest$ $\sigma^2 Forest = 5.025556$
Tree within Forest	$\sigma^2 + n\sigma^2 Tree$	$0.77 = 0.3 + 3\sigma^2 Tree$ $\sigma^2 Tree = 0.1567$
Seed within Tree	σ^2	$\sigma^2 seed = 0.30$

Note: Forest (a) = 5; Tree (b) = 3; Seed (n) = 3

```
> library(agricolae)
> duncan.test(model, "Forest", console=TRUE)

Study: model ~ "Forest"

Duncan's new multiple range test for Height

Mean Square Error:  0.2973333

Forest,  means

     Height       std  r  Min  Max
A  14.95556 0.9761034  9 13.6 16.1
B  18.20000 0.3968627  9 17.5 18.8
C  13.30000 0.6344289  9 12.4 14.4
D  18.74444 0.6691620  9 17.6 19.5
E  16.04444 0.3395258  9 15.7 16.7

Alpha: 0.05 ; DF Error: 30

Critical Range
        2         3         4         5
0.5249636 0.5516830 0.5690037 0.5813668
```

```
Means with the same letter are not significantly
different.

    Height   groups
D 18.74444      a
B 18.20000      a
E 16.04444      b
A 14.95556      c
C 13.30000      d
>
```

12.3 Nested Design with Three Factors

Consider an experiment was conducted to study the hardness of a metal alloy. A three-stage nested design was conducted that included two alloy chemistry compositions, three ovens for each alloy chemistry composition (6 ovens were used), four ingot molds were used to produce alloy ingots for each of the six combinations of alloy chemistry composition and oven (24 molds were used), and three ingots were produced from each of the 24 molds. Molds can only be used once. The experimental data in the table below contains alloy hardness measurements (Anonymous, 2019).

Alloy chemestry	1											
Oven	1				2				3			
Mold	1	2	3	4	1	2	3	4	1	2	3	4
Hardness	42.5	43.1	37	50.7	61.6	58.8	61.9	53.9	58	53.8	59.5	55.9
	46.5	48.1	39	53.7	59.6	62.8	60.9	59.9	59	50.8	57.5	46.9
	44.5	40.1	43	47.7	61.6	57.8	52.9	57.9	61	53.8	55.5	51.9
Alloy chemestry	2											
Oven	1				2				3			
Mold	1	2	3	4	1	2	3	4	1	2	3	4
Hardness	39.5	37.8	45	37.8	59	63.9	63.8	58	56.7	50.8	52.7	45.7
	35.5	38.8	38	38.8	59	61.9	65.8	60	50.7	50.8	56.7	47.7
	37.5	41.8	42	41.8	60	59.9	59.8	62	52.7	58.8	57.7	49.7

In R:

```
> data=read.csv("nested3stagesAlloy.csv", header=TRUE)
> data
   Alloy Oven Mold Hardness
1     A1   O1   M1     42.5
2     A1   O1   M2     43.1
```

3	A1	O1	M3	37.0
4	A1	O1	M4	50.7
5	A1	O2	M1	61.6
6	A1	O2	M2	58.8
7	A1	O2	M3	61.9
8	A1	O2	M4	53.9
9	A1	O3	M1	58.0
10	A1	O3	M2	53.8
11	A1	O3	M3	59.5
12	A1	O3	M4	55.9
13	A1	O1	M1	46.5
14	A1	O1	M2	48.1
15	A1	O1	M3	39.0
16	A1	O1	M4	53.7
17	A1	O2	M1	59.6
18	A1	O2	M2	62.8
19	A1	O2	M3	60.9
20	A1	O2	M4	59.9
21	A1	O3	M1	59.0
22	A1	O3	M2	50.8
23	A1	O3	M3	57.5
24	A1	O3	M4	46.9
25	A1	O1	M1	44.5
26	A1	O1	M2	40.1
27	A1	O1	M3	43.0
28	A1	O1	M4	47.7
29	A1	O2	M1	61.6
30	A1	O2	M2	57.8
31	A1	O2	M3	52.9
32	A1	O2	M4	57.9
33	A1	O3	M1	61.0
34	A1	O3	M2	53.8
35	A1	O3	M3	55.5
36	A1	O3	M4	51.9
37	A2	O1	M1	39.5
38	A2	O1	M2	37.8
39	A2	O1	M3	45.0
40	A2	O1	M4	37.8
41	A2	O2	M1	59.0
42	A2	O2	M2	63.9
43	A2	O2	M3	63.8
44	A2	O2	M4	58.0
45	A2	O3	M1	56.7
46	A2	O3	M2	50.8
47	A2	O3	M3	52.7
48	A2	O3	M4	45.7
49	A2	O1	M1	35.5
50	A2	O1	M2	38.8
51	A2	O1	M3	38.0

```
52     A2    O1    M4      38.8
53     A2    O2    M1      59.0
54     A2    O2    M2      61.9
55     A2    O2    M3      65.8
56     A2    O2    M4      60.0
57     A2    O3    M1      50.7
58     A2    O3    M2      50.8
59     A2    O3    M3      56.7
60     A2    O3    M4      47.7
61     A2    O1    M1      37.5
62     A2    O1    M2      41.8
63     A2    O1    M3      42.0
64     A2    O1    M4      41.8
65     A2    O2    M1      60.0
66     A2    O2    M2      59.9
67     A2    O2    M3      59.8
68     A2    O2    M4      62.0
69     A2    O3    M1      52.7
70     A2    O3    M2      58.8
71     A2    O3    M3      57.7
72     A2    O3    M4      49.7
> str(data)
'data.frame':   72 obs. of  4 variables:
 $ Alloy   : Factor w/ 2 levels "A1","A2": 1 1 1 1 1 1 1 1 1 1 ...
 $ Oven    : Factor w/ 3 levels "O1","O2","O3": 1 1 1 1 2 2 2 2 3 3 ...
 $ Mold    : Factor w/ 4 levels "M1","M2","M3",..: 1 2 3 4 1 2 3 4 1 2 ...
 $ Hardness: num  42.5 43.1 37 50.7 61.6 58.8 61.9 53.9 58 53.8 ...
> nested3=aov(Hardness~Alloy/Oven/Mold, data=data)
> summary(nested3)
                Df Sum Sq Mean Sq F value   Pr(>F)
Alloy            1     70    70.0   8.632 0.005063 **
Alloy:Oven       4   4181  1045.3 128.869  < 2e-16 ***
Alloy:Oven:Mold 18    492    27.3   3.368 0.000401 ***
Residuals       48    389     8.1
---
Signif. codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’ 0.1 ‘ ’ 1
>
```

Expectations of mean squares, *E*(*MS*), can be seen in the following table.

Source	E(MS)	Variance component
Alloy	$\sigma^2 + n\sigma^2 Mold + nc\sigma^2 Oven + nbc\sigma^2 Alloy$	$\sigma^2 Alloy =$
Oven within Alloy	$\sigma^2 + n\sigma^2 Mold + nc\sigma^2 Oven$	$\sigma^2 Oven = 84.63$
Mold within Oven	$\sigma^2 + n\sigma^2 Mold$	$\sigma^2 Mold = 6.4$
Within metal	σ^2	$\sigma^2 Metal =$ 8.1

Note: Alloy (a) = 2; Oven (b) = 3; Mold (n) = 4; Metal = 3

XIII. ANALYSIS OF COVARIANCE (ANCOVA)

13.1 Introduction

Basically analysis of covariance (ANCOVA) is a combination of regression and variance analysis. This includes measuring the other variables besides the response variable, which is to be observed from the experimental material. The other variable mentioned is covariable (accompanying variable = concomitant variable), which has a very close relationship with the response variable, and even determines it. Observation of the covariables is intended to help reduce experimental errors, through adjustments, namely by eliminating the influence of variations caused by the covariable. The results of observations of the response variable, adjusted for the results of observations of covariables (which may vary), to obtain a higher accuracy analysis results.

For example, an experiment is designed to test the effects of three diets on yearling weight of cattle. Different initial weight, different age or maybe different parity at the beginning of the experiment will affect the precision of the experiment. Thus, to increase the precision of analysis, it is important to adjust yearling weights for differences in initial weight or initial age or parity. In this case initial weight or age or parity can be defined as a covariate in the model.

13.2 Analysis of Covariance Using Completely Randomized Design

Analysis of covariance with completely randomized design is intended for correcting treatment means, controlling the experimental error and increasing precision. The ANCOVA model is:

$$yij = \beta 0 + \beta 1 xij + \tau i + \varepsilon ij \qquad i = 1,..,a; \quad j = 1,...,n$$

where:

yij = observation j in treatment i

$\beta 0$ = intercept

$\beta 1$ = coefficient of regression

xij = a continuous independent variable with mean μx (covariate)

τi = fixed effect of treatment i

εij = random error

For example, an experiment investigating the gain of bull fattened using four different diets for four months was conducted using a completely randomized design. Initial weight (kg) of the bull was recorded, but not used in the assignment of animals to the diets. Body weight gain (kg) at the end of the experiment were measured, as presented in the table below.

Diet A		Diet B		Diet C		Diet D	
Initial weight	Gain	Initial weight	Gain	Initial weight	Gain	Initial weight	Gain
330	115.2	370	117.6	380	117.6	400	118.8
380	118.8	320	112.8	300	111.6	320	112.8
340	116.4	390	116.4	310	110.4	330	111.6
330	116.4	410	117.6	370	118.8	390	120.0
320	115.2	370	116.4	400	118.8	420	120.0

In R:

```
> dat=read.csv("ancovaCRD12.csv", header=TRUE)
> dat
   Treatment InitialWeight  Gain
1       DietA            330 115.2
2       DietA            380 118.8
3       DietA            340 116.4
4       DietA            330 116.4
5       DietA            320 115.2
6       DietB            370 117.6
7       DietB            320 112.8
8       DietB            390 116.4
9       DietB            410 117.6
10      DietB            370 116.4
11      DietC            380 117.6
12      DietC            300 111.6
13      DietC            310 110.4
14      DietC            370 118.8
15      DietC            400 118.8
16      DietD            400 118.8
17      DietD            320 112.8
18      DietD            330 111.6
19      DietD            390 120.0
20      DietD            420 120.0
> str(dat)
'data.frame':   20 obs. of  3 variables:
 $ Treatment    : Factor w/ 4 levels "DietA","DietB",..: 1 1 1 1 1 2 2 2 2 2 ...
 $ InitialWeight: int  330 380 340 330 320 370 320 390 410 370 ...
 $ Gain         : num  115 119 116 116 115 ...
```

>##Note: COVARIATE (initial weight) needs to be a continuous numeric variable

```
> #without initial weight
> ancova=lm(Gain~Treatment, data=dat)
> anova(ancova)
```

```
Analysis of Variance Table

Response: Gain
          Df  Sum Sq Mean Sq F value Pr(>F)
Treatment  3   4.032   1.344  0.1353 0.9376
Residuals 16 158.976   9.936

> #initial weight included
> ancova=lm(Gain~InitialWeight+Treatment, data=dat)
> anova(ancova)
Analysis of Variance Table

Response: Gain
              Df  Sum Sq Mean Sq F value    Pr(>F)
InitialWeight  1 121.459 121.459 80.7455 2.001e-07 ***
Treatment      3  18.985   6.328  4.2071   0.02398 *
Residuals     15  22.563   1.504
---
Signif. codes: 0‘***’0.001‘**’0.01‘*’0.05 ‘.’ 0.1 ‘ ’ 1

>#Using aov is the same thing
> fit=aov(Gain~InitialWeight+Treatment, data=dat)
> summary(fit)
              Df Sum Sq Mean Sq F value Pr(>F)
InitialWeight  1 121.46  121.46  80.746  2e-07 ***
Treatment      3  18.99    6.33   4.207  0.024 *
Residuals     15  22.56    1.50
---
Signif. codes: 0‘***’0.001‘**’0.01‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Based on two analysis above it can be seen that the first model (without initial weight) was not correct (the effect of treatment is not significant). When initial weights is included in the model a significant difference between treatments was found.

13.3 Analysis of Covariance using Randomized Completely Block Design

The model for the analysis of covariance for two-way classified data with k treatments in r blocks of the randomized block design. The ANCOVA model is:

$$yij = \beta 0 + \beta 1 xijk + \tau i + bj + \varepsilon ijk \qquad i = 1,..,a; \quad j = 1,...,b; \quad kj = 1,...,n$$

where:

$yijk$ = observation k in treatment i in block j

$\beta 0$ = intercept

$\beta 1$ = coefficient of regression

$xijk$ = a continuous independent variable with mean μx (covariate)

τi = fixed effect of treatment i

bj = fixed effect of block j

εijk = random error

For example, an experiment is conducted to investigate four types of treatment in the form of milk replacer substitutes: A, B, C, and D, which are tested on the calf of FH cattle of the same age. Experiments were carried out with RCBD and each with 5 replications. The Response variable observed was body weight gain after the experiment was completed (Y). Because the initial body weight varies, an observation is also made on the initial weight of each calf (X), as a covariable. The data from the observation of the two variables are as follows.

Table. Body weight gain (Y) and initial body weight of calf treated with four different types of milk replacer

Milk Replacer	Variable	Block				
		I	II	III	IV	V
A	X	44	47	45	44	44
	Y	87	87	90	88	85
B	X	36	39	31	33	36
	Y	57	77	45	43	45
C	X	41	46	41	34	37
	Y	59	52	57	35	44
D	X	35	30	37	31	33
	Y	37	36	46	26	36

```
> data <- read.csv('anacova.csv', header=T)
> data
   MilkReplacer Block BirthWeight Gain
1             A     I           44   87
2             A    II           47   87
3             A   III           45   90
4             A    IV           44   88
5             A     V           44   85
6             B     I           36   57
7             B    II           39   77
8             B   III           31   45
9             B    IV           33   43
```

```
10             B     V          36   45
11             C     I          41   59
12             C    II          46   52
13             C   III          41   57
14             C    IV          34   35
15             C     V          37   44
16             D     I          35   37
17             D    II          30   36
18             D   III          37   46
19             D    IV          31   26
20             D     V          33   36
> str(data)
'data.frame':   20 obs. of  4 variables:
 $ MilkReplacer: Factor w/ 4 levels "A","B","C","D": 1 1 1 1 1 2 2 2 2 2 ...
 $ Block       : Factor w/ 5 levels "I","II","III",..: 1 2 3 4 5 1 2 3 4 5 ...
 $ BirthWeight : int  44 47 45 44 44 36 39 31 33 36 ...
 $ Gain        : int  87 87 90 88 85 57 77 45 43 45 ...

> ##Without covariate (Birth weight)
> fit1 <- aov(Gain ~ Block + MilkReplacer, data=data)
> summary(fit1)
             Df Sum Sq Mean Sq F value   Pr(>F)
Block         4    607   151.7   2.221    0.128
MilkReplacer  3   7134  2378.1  34.819 3.35e-06 ***
Residuals    12    820    68.3
---
Signif. codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’ 0.1 ‘ ’ 1

> ##With covariate (Birth weight)
> fit2 <- aov(Gain ~ BirthWeight + Block + MilkReplacer,
data=data)
> summary(fit2)
             Df Sum Sq Mean Sq F value   Pr(>F)
BirthWeight   1   6116    6116 119.973 2.96e-07 ***
Block         4     36       9   0.175 0.946701
MilkReplacer  3   1848     616  12.081 0.000835 ***
Residuals    11    561      51
---
Signif. codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’ 0.1 ‘ ’ 1
>
> fit3 <- aov(Gain ~ Block + BirthWeight + MilkReplacer,
data=data)
> summary(fit3)
             Df Sum Sq Mean Sq F value   Pr(>F)
Block         4    607     152   2.976 0.068350 .
BirthWeight   1   5545    5545 108.770 4.85e-07 ***
MilkReplacer  3   1848     616  12.081 0.000835 ***
Residuals    11    561      51
---
Signif. codes: 0‘***’0.001‘**’0.01‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Based on ANCOVA result it can be concluded that initial weight (birth weight) affected body weight gain and increase the precision of the analysis by reducing residual. Furhermore, different milk replacer influence body weight gain.

XIV. REPEATED MEASURES DESIGN

14.1 Introduction

Repeated measures design is usually used to compare the treatment response which is measured repeatedly on each subject, for example, milk yield measured during lactation, growth of animal or plant measured over some period or hormone concentrations in blood measured several times and soon. Experimental unit which is measured repeatedly is known subject. In crossover design, an experimental unit is assigned to different treatment, but in repeated measure design an experimental unit receives the same treatment over time.

Analysis for repeated measures experiment is similar to split plot experiments in which there are two sources of error. In repeated measures design, treatments are compared to the less precise subject to subject error whilst trends over time between treatments are compared to the more precise within subject experimental error.

14.2 Model in Repeated Measures (One Way ANOVA)

Analysis for the repeated measures design is similar to a split-plot design with whole-plots to be subjects (for example animal) and sub-plots are different observation times on each subject. For example, several dairy cows were randomized to treatment diets, so the diets are the whole-plot treatments, while weekly measurements and the interaction between diets and weekly measurements are the sub-plot treatments. Assumption for this analysis is that variance and covariance between measures is equal, independent and normally distributed. For example, suppose an experiment assign *a* treatments and *b* animals for each treatment which each animal is measured in *n* periods, the model is:

$$yijk = \mu + \tau i + \delta ij + tk + (\tau * t)ik + \varepsilon ijk \qquad i = 1,...,a; \quad j = 1,...,b; \quad k = 1,...,n$$

where:

$yijk$ = observation ijk

μ = overall mean

τi = effect of treatment i

tk = effect of period k

$(\tau * t)ik$ = the effect of interaction between treatment i and period k

δij = random error, the variance between animals (subjects) within treatment and it is equal to the covariance between repeated measurements within animals

εijk = random error, the variance between measurements within animals

a = number of treatments

b = number of subjects (animals)

n = number of periods

14.3 Simple Repeated Measures (One Within Subject Variable)

The simplest repeated measure is when measurement is within subject only. In this case a researcher just want to investigate if there is change between measurement over the period. For example, protein sample of the milk was measured weekly from ten cows.

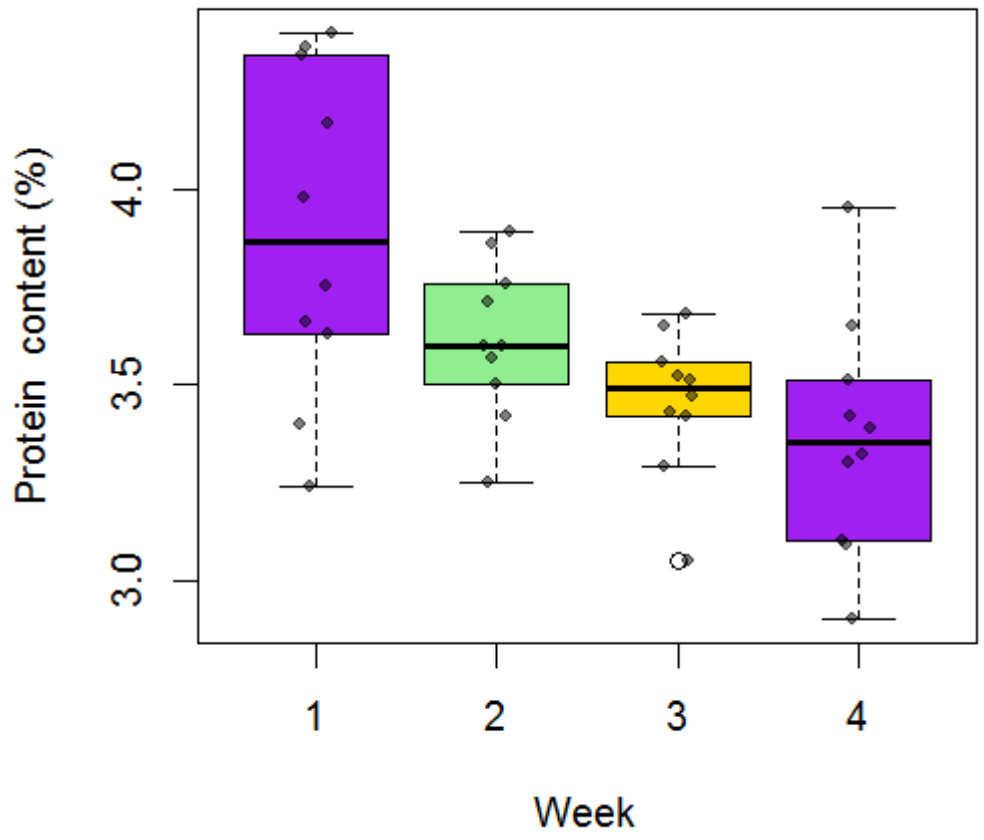
Cow	Week			
	1	2	3	4
1	3.63	3.57	3.47	3.65
2	3.24	3.25	3.29	3.09
3	3.98	3.6	3.43	3.3
4	3.66	3.5	3.05	2.9
5	4.34	3.76	3.68	3.51
6	4.36	3.71	3.42	3.95
7	4.17	3.6	3.52	3.1
8	4.4	3.86	3.56	3.32
9	3.4	3.42	3.51	3.39
10	3.75	3.89	3.65	3.42

In R:

```
> data=read.csv("repeated1.csv", header=T)
> data
   Cow Week Protein
1    1    1    3.63
2    1    2    3.57
3    1    3    3.47
4    1    4    3.65
5    2    1    3.24
6    2    2    3.25
7    2    3    3.29
8    2    4    3.09
9    3    1    3.98
10   3    2    3.60
11   3    3    3.43
12   3    4    3.30
```

```
13    4     1     3.66
14    4     2     3.50
15    4     3     3.05
16    4     4     2.90
17    5     1     4.34
18    5     2     3.76
19    5     3     3.68
20    5     4     3.51
21    6     1     4.36
22    6     2     3.71
23    6     3     3.42
24    6     4     3.95
25    7     1     4.17
26    7     2     3.60
27    7     3     3.52
28    7     4     3.10
29    8     1     4.40
30    8     2     3.86
31    8     3     3.56
32    8     4     3.32
33    9     1     3.40
34    9     2     3.42
35    9     3     3.51
36    9     4     3.39
37   10     1     3.75
38   10     2     3.89
39   10     3     3.65
40   10     4     3.42
> str(data)
'data.frame':   40 obs. of  3 variables:
 $ Cow    : int  1 1 1 1 2 2 2 2 3 3 ...
 $ Week   : int  1 2 3 4 1 2 3 4 1 2 ...
 $ Protein: num  3.63 3.57 3.47 3.65 3.24 3.25 3.29 3.09 3.98 3.6 ...
> data$Cow=as.factor(data$Cow)
> data$Week=as.factor(data$Week)
> str(data)
'data.frame':   40 obs. of  3 variables:
 $ Cow    : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
 $ Week   : Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 4 1 2 ...
 $ Protein: num  3.63 3.57 3.47 3.65 3.24 3.25 3.29 3.09 3.98 3.6 ...
> fit1=aov(Protein ~ Week + Error(Cow), data=data)
> summary(fit1)

Error: Cow
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  9  1.738  0.1931

Error: Within
          Df Sum Sq Mean Sq F value   Pr(>F)
Week       3  1.612  0.5374   11.12 6.23e-05 ***
Residuals 27  1.304  0.0483
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

```
> boxplot(Protein ~ Week,
+         data = data,
+         col = c("purple", "lightgreen", "gold"))
> stripchart(Protein ~ Week,
+            vertical = TRUE,
+            data = data,
+            method = "jitter",
+            add = TRUE,
+            pch = 20,
+            col = rgb(0,0,0,0.5))
> boxplot(Protein ~ Week,
+         data = data,
+           col = c("purple", "lightgreen", "gold"),
ylab="Protein  content (%)",
+         xlab="Week")
> stripchart(Protein ~ Week,
+            vertical = TRUE,
+            data = data,
+            method = "jitter",
+            add = TRUE,
+            pch = 20,
+            col = rgb(0,0,0,0.5))
>
```



Based on ANOVA table and boxplot above it can be concluded that protein content change significantly over the period (decreasing), or in other word, protein content is different in different week.

14.4 One Between Subject Variable, One Within Subject Variable

For example, in the following table is data of protein content in milk for the first four week samples from 10 cows in each group of diets (Lawson, 2015).

Diets	Cow	Week			
		1	2	3	4
Barley	1	3.63	3.57	3.47	3.65
Barley	2	3.24	3.25	3.29	3.09
Barley	3	3.98	3.6	3.43	3.3
Barley	4	3.66	3.5	3.05	2.9
Barley	5	4.34	3.76	3.68	3.51
Barley	6	4.36	3.71	3.42	3.95
Barley	7	4.17	3.6	3.52	3.1
Barley	8	4.4	3.86	3.56	3.32
Barley	9	3.4	3.42	3.51	3.39
Barley	10	3.75	3.89	3.65	3.42
Mixed	11	3.38	3.38	3.1	3.09
Mixed	12	3.8	3.51	3.19	3.11
Mixed	13	4.17	3.71	3.32	3.1
Mixed	14	4.59	3.86	3.62	3.6
Mixed	15	4.07	3.45	3.56	3.1
Mixed	16	4.32	3.37	3.47	3.46
Mixed	17	3.56	3.14	3.6	3.36
Mixed	18	3.67	3.33	3.2	2.72
Mixed	19	4.15	3.55	3.27	3.27
Mixed	20	3.51	3.9	2.75	3.37
Lupins	21	3.69	3.38	3	3.5
Lupins	22	4.2	3.35	3.37	3.07
Lupins	23	3.31	3.04	2.8	3.17
Lupins	24	3.13	3.34	3.34	3.25
Lupins	25	3.73	3.61	3.82	3.61
Lupins	26	4.32	3.7	3.62	3.5
Lupins	27	3.04	2.89	2.78	2.84
Lupins	28	3.84	3.51	3.39	2.88
Lupins	29	3.98	3.3	3.02	2.99
Lupins	30	4.18	4.12	3.84	3.65

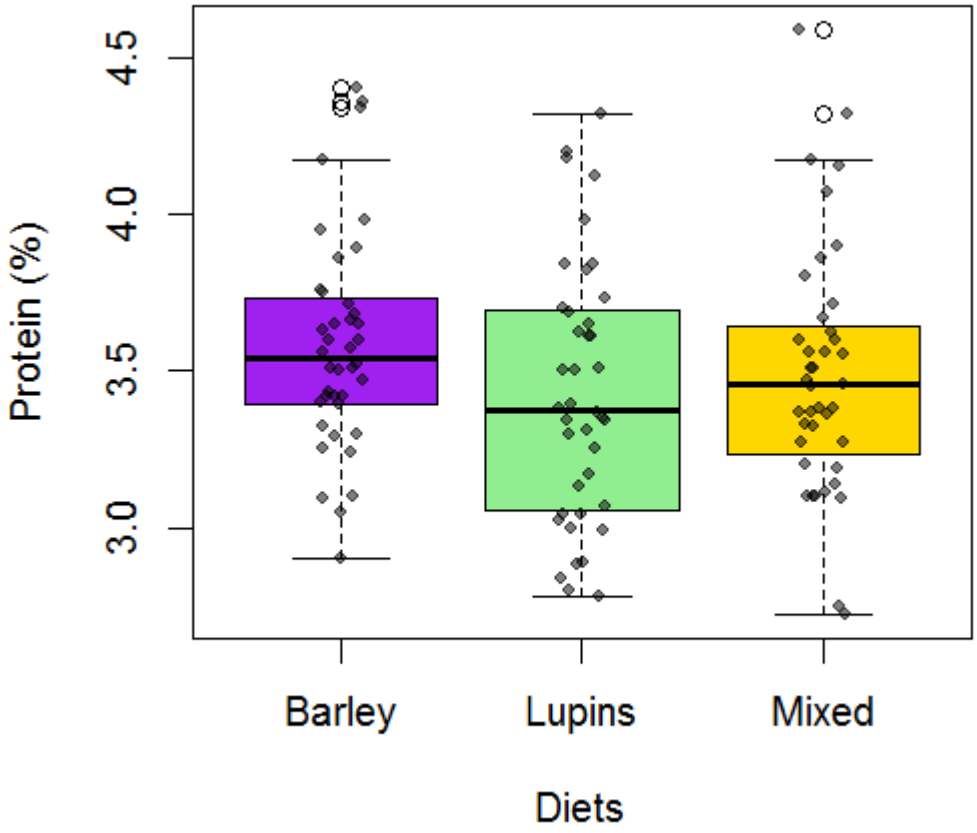
In R:

```
> dat=read.csv("repeated2.csv", header=T)
> dat
     Diets Cow Week Protein
1   Barley   1    1    3.63
2   Barley   2    1    3.24
3   Barley   3    1    3.98
4   Barley   4    1    3.66
5   Barley   5    1    4.34
6   Barley   6    1    4.36
7   Barley   7    1    4.17
8   Barley   8    1    4.40
9   Barley   9    1    3.40
10  Barley  10    1    3.75
11   Mixed  11    1    3.38
12   Mixed  12    1    3.80
13   Mixed  13    1    4.17
14   Mixed  14    1    4.59
15   Mixed  15    1    4.07
16   Mixed  16    1    4.32
17   Mixed  17    1    3.56
18   Mixed  18    1    3.67
19   Mixed  19    1    4.15
20   Mixed  20    1    3.51
21  Lupins  21    1    3.69
22  Lupins  22    1    4.20
23  Lupins  23    1    3.31
24  Lupins  24    1    3.13
25  Lupins  25    1    3.73
26  Lupins  26    1    4.32
27  Lupins  27    1    3.04
28  Lupins  28    1    3.84
29  Lupins  29    1    3.98
30  Lupins  30    1    4.18
31  Barley   1    2    3.57
32  Barley   2    2    3.25
33  Barley   3    2    3.60
34  Barley   4    2    3.50
35  Barley   5    2    3.76
36  Barley   6    2    3.71
37  Barley   7    2    3.60
38  Barley   8    2    3.86
39  Barley   9    2    3.42
40  Barley  10    2    3.89
41   Mixed  11    2    3.38
42   Mixed  12    2    3.51
43   Mixed  13    2    3.71
44   Mixed  14    2    3.86
45   Mixed  15    2    3.45
46   Mixed  16    2    3.37
```

47	Mixed	17	2	3.14
48	Mixed	18	2	3.33
49	Mixed	19	2	3.55
50	Mixed	20	2	3.90
51	Lupins	21	2	3.38
52	Lupins	22	2	3.35
53	Lupins	23	2	3.04
54	Lupins	24	2	3.34
55	Lupins	25	2	3.61
56	Lupins	26	2	3.70
57	Lupins	27	2	2.89
58	Lupins	28	2	3.51
59	Lupins	29	2	3.30
60	Lupins	30	2	4.12
61	Barley	1	3	3.47
62	Barley	2	3	3.29
63	Barley	3	3	3.43
64	Barley	4	3	3.05
65	Barley	5	3	3.68
66	Barley	6	3	3.42
67	Barley	7	3	3.52
68	Barley	8	3	3.56
69	Barley	9	3	3.51
70	Barley	10	3	3.65
71	Mixed	11	3	3.10
72	Mixed	12	3	3.19
73	Mixed	13	3	3.32
74	Mixed	14	3	3.62
75	Mixed	15	3	3.56
76	Mixed	16	3	3.47
77	Mixed	17	3	3.60
78	Mixed	18	3	3.20
79	Mixed	19	3	3.27
80	Mixed	20	3	2.75
81	Lupins	21	3	3.00
82	Lupins	22	3	3.37
83	Lupins	23	3	2.80
84	Lupins	24	3	3.34
85	Lupins	25	3	3.82
86	Lupins	26	3	3.62
87	Lupins	27	3	2.78
88	Lupins	28	3	3.39
89	Lupins	29	3	3.02
90	Lupins	30	3	3.84
91	Barley	1	4	3.65
92	Barley	2	4	3.09
93	Barley	3	4	3.30
94	Barley	4	4	2.90
95	Barley	5	4	3.51

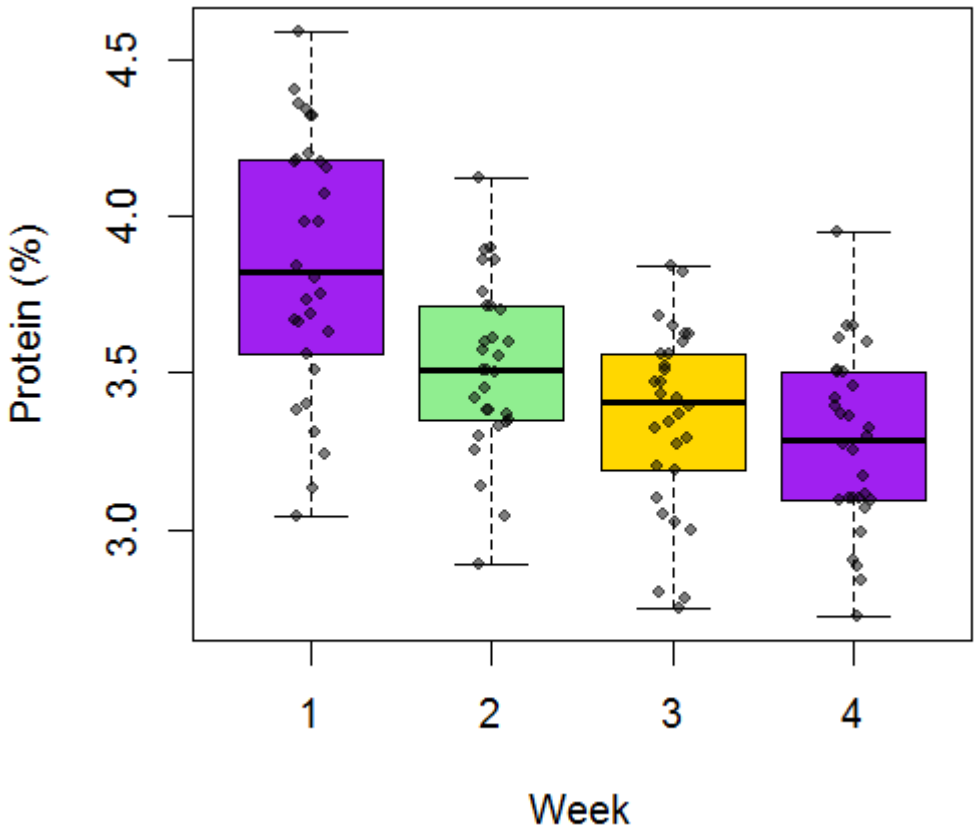
```
96  Barley   6    4    3.95
97  Barley   7    4    3.10
98  Barley   8    4    3.32
99  Barley   9    4    3.39
100 Barley  10    4    3.42
101  Mixed  11    4    3.09
102  Mixed  12    4    3.11
103  Mixed  13    4    3.10
104  Mixed  14    4    3.60
105  Mixed  15    4    3.10
106  Mixed  16    4    3.46
107  Mixed  17    4    3.36
108  Mixed  18    4    2.72
109  Mixed  19    4    3.27
110  Mixed  20    4    3.37
111 Lupins  21    4    3.50
112 Lupins  22    4    3.07
113 Lupins  23    4    3.17
114 Lupins  24    4    3.25
115 Lupins  25    4    3.61
116 Lupins  26    4    3.50
117 Lupins  27    4    2.84
118 Lupins  28    4    2.88
119 Lupins  29    4    2.99
120 Lupins  30    4    3.65
> str(dat)
'data.frame':   120 obs. of  4 variables:
 $ Diets  : Factor w/ 3 levels "Barley","Lupins",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Cow    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Week   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Protein: num  3.63 3.24 3.98 3.66 4.34 4.36 4.17 4.4 3.4 3.75 ...
> dat$Diets=as.factor(dat$Diets)
> dat$Cow=as.factor(dat$Cow)
> dat$Week=as.factor(dat$Week)
> str(data)
'data.frame':   40 obs. of  3 variables:
 $ Cow    : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
 $ Week   : Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 4 1 2 ...
 $ Protein: num  3.63 3.57 3.47 3.65 3.24 3.25 3.29 3.09 3.98 3.6 ...
>
> fit2=aov(Protein ~ Diets*Week + Error(Cow/Week), data=dat)
> summary(fit2)

Error: Cow
          Df Sum Sq Mean Sq F value Pr(>F)
Diets      2  0.485  0.2425    0.94  0.403
Residuals 27  6.962  0.2579

Error: Cow:Week
           Df Sum Sq Mean Sq F value   Pr(>F)
Week        3  5.880  1.9598  36.551 4.87e-15 ***
Diets:Week  6  0.165  0.0275   0.513    0.797
Residuals  81  4.343  0.0536
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
>
> boxplot(Protein ~ Diets,
+         data = dat,
+         col = c("purple", "lightgreen", "gold"),
+         ylab="Protein (%)", xlab="Diets")
> stripchart(Protein ~ Diets,
+            vertical = TRUE,
+            data = dat,
+            method = "jitter",
+            add = TRUE,
+            pch = 20,
+            col = rgb(0,0,0,0.5))
>
```

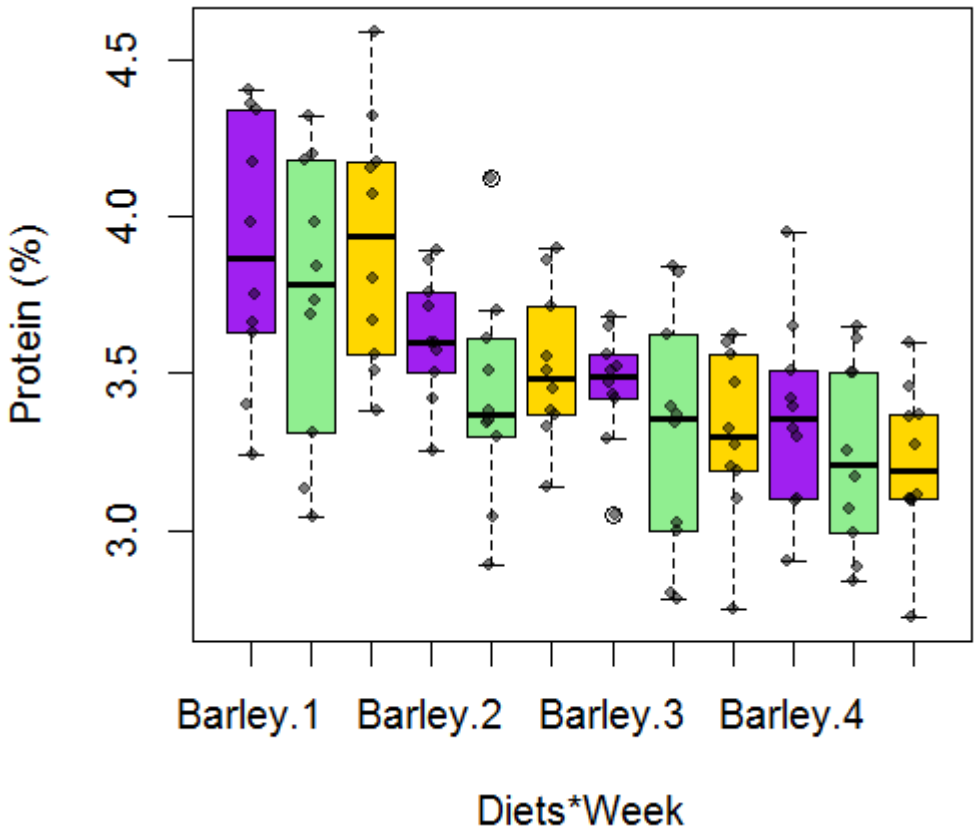



```
> boxplot(Protein ~ Week,
+         data = dat,
+         col = c("purple", "lightgreen", "gold"),
+         ylab="Protein (%)",xlab="Week")
> stripchart(Protein ~ Week,
+            vertical = TRUE,
+            data = dat,
+            method = "jitter",
+            add = TRUE,
+            pch = 20,
```

```
+             col = rgb(0,0,0,0.5))
>
```




```
> boxplot(Protein ~ Diets*Week,
+         data = dat,
+         col = c("purple", "lightgreen", "gold"),
+         ylab="Protein (%)", xlab="Diets*Week")
> stripchart(Protein ~ Diets*Week,
+            vertical = TRUE,
+            data = dat,
+            method = "jitter",
+            add = TRUE,
+            pch = 20,
+            col = rgb(0,0,0,0.5))
>
```



Based on ANOVA table and boxplot above it can be concluded that protein content did not change over the period, but protein content change significantly over the period (protein content is different in different week).

XV. ANALYSIS OF NUMERICAL TREATMENT LEVELS

15.1 Introduction

In a research, sometimes we want to find an optimum treatment level which affect a maximum response. For example, we want to evaluate the effect of different levels of mineral content in a ration on body weight gain of broiler. At the same time we want to find an optimum mineral content in ration to get maximum body weight gain. In this case the use of regression or polynomial orthogonal contrasts can be alternatives to solve this problem. The next question is that which regression model is most appropriate, either linear, quadratic or cubic. To test the appropriateness of a model can be done by lack of fit analysis. If a regression model fails to adequately describe the functional relationship between the experimental factors and the response variable means that the regression model exhibits lack of fit.

Consider simple linear regression model is:

$$yij = \beta 0 + \beta 1\ xi + \varepsilon ij$$

and $\bar{y}i$ is the mean and $\hat{y}i$ is the estimated value for level i. If the difference between $\hat{y}i$ and $\bar{y}i$ is not significant means that the model is correct.

15.2 Lack of Fit Test

The residual sum of squares in this test is divided into a pure error and a lack of fit sum of squares.

$$SSRES = SSPE + SSLOF$$

with appropriate degrees of freedom: $(n\text{-}1) = \Sigma i\ (ni\ \text{-}1) + (m\text{-}p)$ where p is the number of parameters in the model. The sums of squares are:

$$\text{SSE} = \sum_i \sum_j (yij\ -\ \hat{y}i)^2$$

$$\text{SSPE} = \sum_i \sum_j (yij\ -\ \bar{y}i)^2$$

$$\text{SSLOF} = \sum_i ni\ (\bar{y}i\ -\ \hat{y}i)^2$$

where:

$$\bar{y}\text{i} = \frac{1}{ni} \sum_j yij = mean\ for\ level\ i$$

$\hat{y}i =$ estimated value for level i

The mean square for pure error is:

$$\text{MSPE} = \frac{SSPE}{\sum_i (ni - 1)}$$

$$\text{MSLOF} = \frac{SSLOF}{m - p}$$

$$\text{F} = \frac{MSLOF}{MSPE}$$

The ANOVA table is:

Source	SS	df	MS	F statistic
Regression	SSreg	1	SSreg/1	MSreg/MSE
Error	SSE	n-2	SSE/(n-2)	
Lack of fit	SSLOF	m-2	SSLOF/(m-2)	MSLOF/SSPE
Pure error	SSPE	n-m	SSPE/(n-m)	
Total	SST			

For example, an experiment is to evaluate the effect of different levels of protein content in ration on feed conversion of broiler. Data of feed conversion for each treatment of protein level in ration at the end of the experiment is presented in the following table.

Level protein			
18%	20%	22%	24%
1.8	1.6	1.6	1.8
1.9	1.5	1.7	1.9
1.7	1.6	1.7	1.8
1.9	1.4	1.5	1.7
1.6	1.5	1.6	1.7
1.8	1.6	1.7	1.8

In R:

```
> data=read.csv("LackOfFit1.csv", header=T)
> data
   Protein FeedConversion
1       18            1.8
2       18            1.9
3       18            1.7
4       18            1.9
```

```
5       18            1.6
6       18            1.8
7       20            1.6
8       20            1.5
9       20            1.6
10      20            1.4
11      20            1.5
12      20            1.6
13      22            1.6
14      22            1.7
15      22            1.7
16      22            1.5
17      22            1.6
18      22            1.7
19      24            1.8
20      24            1.9
21      24            1.8
22      24            1.7
23      24            1.7
24      24            1.8
> data$Protein=as.factor(data$Protein)
> fit=lm(FeedConversion ~  Protein, data=data)
> anova(fit)
Analysis of Variance Table

Response: FeedConversion
          Df  Sum Sq  Mean Sq F value    Pr(>F)
Protein    3 0.27000 0.090000   11.02 0.0001736 ***
Residuals 20 0.16333 0.008167
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> #Analysis of Lack of Fit:
> Reduced <- lm(FeedConversion ~ Protein, data=data)
> data$Protein=as.numeric(data$Protein)
> Reduced <- lm(FeedConversion ~ Protein, data=data)
>  Full  <-  lm(FeedConversion  ~  0  +  as.factor(Protein),  data
=data)
> anova (Reduced, Full)
Analysis of Variance Table

Model 1: FeedConversion ~ Protein
Model 2: FeedConversion ~ 0 + as.factor(Protein)
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     22 0.43033
2     20 0.16333  2     0.267 16.347 6.204e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> #Using EnvStats package:
> library(EnvStats)
> data$Protein=as.numeric(data$Protein)
> fit=lm(FeedConversion ~ Protein, data=data)
```

```
> anovaPE(fit)
                   Df  Sum Sq  Mean Sq F value    Pr(>F)
Protein             1 0.00300 0.003000  0.3673    0.5513
Lack of Fit         2 0.26700 0.133500 16.3469 6.204e-05 ***
Pure Error         20 0.16333 0.008167
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Based on ANOVA table it can be concluded that the different of protein level affected feed conversion of broiler. However, the regression model (protein level on feed conversion response) is not linear which is shown by the significance of Lack of Fit p-value (0.000062). Which model is more appropriate can be evaluated using polynomial orthogonal contrast, either linear, quadratic, cubic or quartic (following topic).

15.3 Polynomial Orthogonal Contrast

Treatment levels analysis and evaluating linear, quadratic, and higher order effects can be tested by polynomial orthogonal contrasts. Degree of polynomial contrast and its coefficient of treatment levels are shown in the following table.

Number of treatment levels	Degree of polynomial	Coefficients (c)	Total ci^2
2	linear	+1 -1	2
3	linear	+1 0 -1	2
	quadratic	+1 -2 +1	6
4	linear	+3 +1 -1 -3	20
	quadratic	+1 -1 -1 +1	4
	cubic	+1 +3 -3 -1	20

Using the same example as previous topic (Lack of Fit Test), polynomial orthogonal contrast can be done like below.

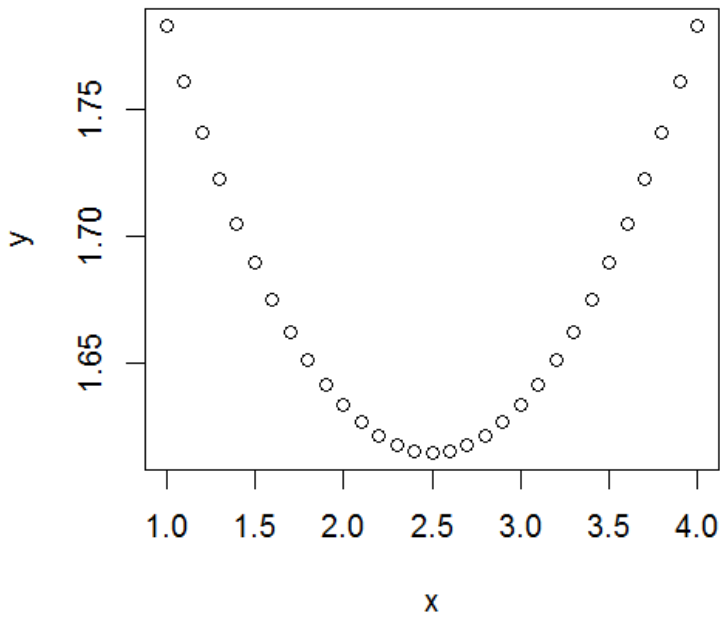
In R:

```
> data=read.csv("LackOfFit1.csv", header=T)
> head(data)
  Protein FeedConversion
1      18            1.8
2      18            1.9
3      18            1.7
4      18            1.9
5      18            1.6
6      18            1.8
> data$Protein=as.factor(data$Protein)
```

```
> fit=lm(FeedConversion ~  Protein, data=data)
> anova(fit)
Analysis of Variance Table

Response: FeedConversion
          Df  Sum Sq  Mean Sq F value    Pr(>F)
Protein    3 0.27000 0.090000   11.02 0.0001736 ***
Residuals 20 0.16333 0.008167
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> #Analysis of Lack of Fit:
> data$Protein=as.numeric(data$Protein)
> library(EnvStats)

Attaching package: ‘EnvStats’

The following objects are masked from ‘package:stats’:

    predict, predict.lm

The following object is masked from ‘package:base’:

    print.default

Warning message:
package ‘EnvStats’ was built under R version 3.5.2

> data$Protein=as.numeric(data$Protein)
> fit=lm(FeedConversion ~ Protein, data=data)
> anovaPE(fit)
                   Df  Sum Sq  Mean Sq F value    Pr(>F)
Protein             1 0.00300 0.003000  0.3673    0.5513
Lack of Fit         2 0.26700 0.133500 16.3469 6.204e-05 ***
Pure Error         20 0.16333 0.008167
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
fit2=lm(FeedConversion~Protein+I(Protein^2)+I(Protein^3)+I(Pro
tein^4)+
+       as.factor(Protein),data=data)
> anova(fit2)
Analysis of Variance Table

Response: FeedConversion
             Df  Sum Sq  Mean Sq F value    Pr(>F)
Protein       1 0.00300 0.003000  0.3673   0.55127
I(Protein^2)  1 0.24000 0.240000 29.3878 2.632e-05 ***
I(Protein^3)  1 0.02700 0.027000  3.3061   0.08403 .
Residuals    20 0.16333 0.008167
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Based on ANOVA table it can be concluded that the most appropriate model for the treatment levels is quadratic. The optimum level of protein is 21% (see in figure below denoted by 2.5 of scale 1-4, where 1 = 18%; 2 = 20%; 3 =22%; and 4 = 24%) with feed conversion of 1.60.

```
> fit3=lm(FeedConversion~Protein+I(Protein^2)+
+       as.factor(Protein),data=data)
> anova(fit3)
Analysis of Variance Table

Response: FeedConversion
                   Df  Sum Sq  Mean Sq F value    Pr(>F)
Protein             1 0.00300 0.003000  0.3673   0.55127
I(Protein^2)        1 0.24000 0.240000 29.3878 2.632e-05 ***
as.factor(Protein)  1 0.02700 0.027000  3.3061   0.08403 .
Residuals          20 0.16333 0.008167
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> b=coef(fit3)
> b
 (Intercept)    Protein  I(Protein^2)  as.factor(Protein)2
   2.083333   -0.375000   0.075000       -0.100000

as.factor(Protein)3 as.factor(Protein)4
                 NA                  NA
>
> x=seq(from=1, to=4, by=0.1)
> y=2.08333-0.375*x+0.075*x^2
> plot(x,y)
```



Or we can calculate optimum point by doing first derivative of the quadratic equation, as below.

$Y = 2.08333-0.375*x+0.075*x^2$

$0 = -0.375 + 2*0.075x$

$x = 0.375/0.15$

$x = 2.5$

XVI. LINEAR REGRESSION

16.1 Introduction

A linear relationship between independent variable(s) (x) or predictor variable(s) and dependent variable (y) or response variable can be formulated using mathematical model which is known as a linear regression model. The goal of linear regression model is to predict the response y, when the predictors values (x) are known. Mathematical equation of the linear regression can be generalized as follows:

$$y = a + bx + e \quad \text{or} \quad y = \beta_0 + \beta_1 x + \epsilon$$

where a or β_0 is the intercept and b or β_1 is the slope or coefficient of regression, and e or ϵ is the error term or residual error.

16.2 Simple Linear Regression

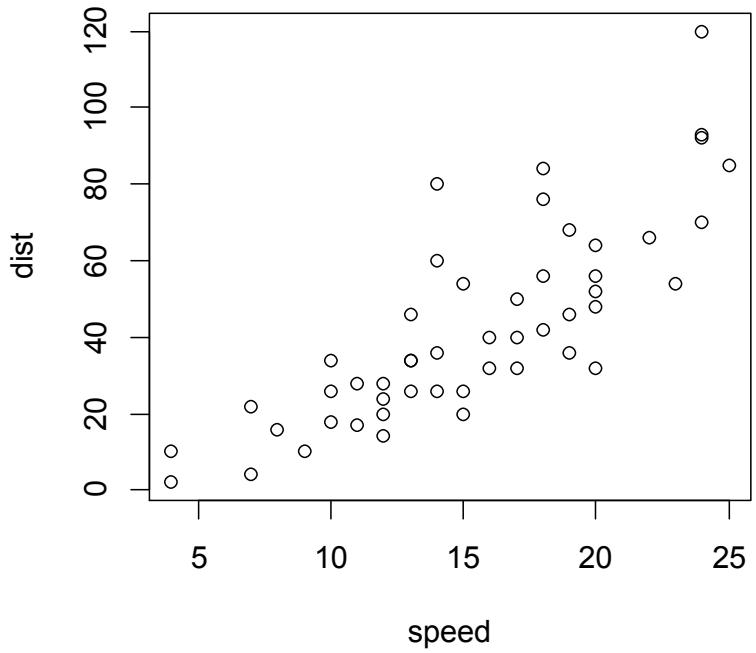
For example, we will use data from cars dataset from the package cars. So there are two variables, namely speed and distance. Speed shows how fast the car goes (x) in miles per hour and the distance (y) measures how far the car goes from start to stop, in feet. We can make scatter plot for the data with the command plot(dist ~ speed, data = data). But previously we see a glimpse of the data.

```
> data=cars
head(data)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
```

Or the overall data is as follows.

```
> data
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
```

```
  7      10   18
  8      10   26
  9      10   34
  10     11   17
  11     11   28
  12     12   14
  13     12   20
  14     12   24
  15     12   28
  16     13   26
  17     13   34
  18     13   34
  19     13   46
  20     14   26
  21     14   36
  22     14   60
  23     14   80
  24     15   20
  25     15   26
  26     15   54
  27     16   32
  28     16   40
  29     17   32
  30     17   40
  31     17   50
  32     18   42
  33     18   56
  34     18   76
  35     18   84
  36     19   36
  37     19   46
  38     19   68
  39     20   32
  40     20   48
  41     20   52
  42     20   56
  43     20   64
  44     22   66
  45     23   54
  46     24   70
  47     24   92
  48     24   93
  49     24  120
  50     25   85

>
> plot(dist ~ speed, data = data)
```

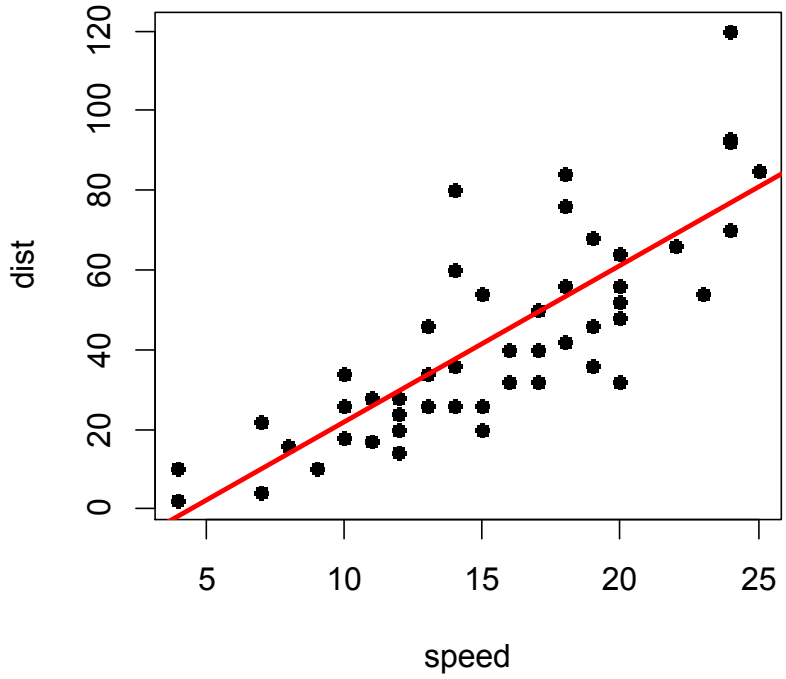


Next, how to make a line in the graph above is to use the lm command. We first save the results in the first example in an object, for example car like the following.

```
> car <- lm(dist ~ speed, data = data)
> coef(car)
       (Intercept)             speed
         -17.579095      3.932409
```

So the intercept of the regression line above is -17.58 with a regression coefficient of 3.93. Thus the regression line equation above is Distance = -17.58 + 3.93 speed, or y = -17.58 + 3.93x.

```
> plot(dist ~ speed, data = data, pch = 16)
> abline(coef(car), col="red", lwd=3)
```



Actually intercept (a) and regression coefficient (b) can be calculated manually as follows.

$$\mathrm{b} = \frac{N\sum XY - (\sum X)(\sum Y)}{N\sum X^2 - (\sum X)^2}$$

$$\mathrm{a} = \bar{Y} - b * \bar{X}$$

In R manually:

```
> b <- ((length(data$dist)*sum(data$speed*
+       data$dist))-(sum(data$speed)*sum(data$dist)))/
+       ((length(data$dist)*sum(data$speed^2))-
+       (sum(data$speed))^2) ## the same as formula
> b
[1] 3.932409
> ## or we can use script below, b=cov(x,y)/var(x)
> b <- cov(data$speed,data$dist)/var(data$speed)
> b
[1] 3.932409
> a <- mean(data$dist) - b*mean(data$speed)
> a
[1] -17.57909
>
```

How to predict the value of y if the value of x is known, for example in the above example we got the equation of the regression line y = -17.58 + 3.93x. What is the value of y if x = 15 and x = 7?, then we can calculate by hand or use a calculator as follows:

```
> x=15
> y = -17.58 + 3.93*x
> y
[1] 41.37
>
> x=7
> y = -17.58 + 3.93*x
> y
[1] 9.93
>
```

Or by using the predict () function, as follows.

```
> predict(car, newdata = data.frame(speed = c(15, 7)))
        1         2
41.407036  9.947766
> #different result is because of rounding in the hand
calculation
```

Or if we want to find out the value of y in the first five x values (4, 4, 7, 7, 8) as follows.

```
> fitted(car)[1:5]
        1         2         3         4         5
-1.849460 -1.849460  9.947766  9.947766 13.880175
>
```

The third and the fourth y value, x = 7 is 9.947766, as in the previous calculation. If we want to know the deviation between the actual observation value and the predicted value, it can be done with the following command.

```
> residuals(car)[1:5]
        1         2         3         4         5
 3.849460 11.849460 -5.947766 12.052234  2.119825
>
```

So the deviation, for example, for the third and the fourth observation where the value of both x = 7 with observations (y) = 4 and 22 and the predicted value = 9.947766, then the deviation or bias are -5.947766 and 12.052234, respectively.

We can also see the results of the overall regression analysis as follows.

```
> carsumry <- summary(car)
> carsumry

Call:
lm(formula = dist ~ speed, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif.codes: 0‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘ ’1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,    Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12

>
```

Based on summary it can be concluded that regression coefficient for speed is significant with p-value 1.49e-12 or less than 0.05 meaning that variation of distance can be explained by speed significantly. Coefficient of determination or R-squared in this regression equation is 0.65 meaning that variation of distance can be explained for about 65% while the rest (35%) by other factors. Overall the regression equation can be used to predict distance if the speed is known with F-statistic 89.57 and p-value 1.49e-12. If we want to know the standard error value specifically, even though it actually appears in the summary, it is as follows:

```
> carsumry$sigma
[1] 15.37959
>
```

and the confidence interval of the regression equation above is as follows.

```
> confint(car)
                 2.5 %    97.5 %
(Intercept) -31.167850 -3.990340
```

```
speed          3.096964   4.767853
>
```

The coefficient of determination (R^2) and the value of the correlation coefficient (r) are as follows, which are the same as those in the summary.

```
> carsumry$r.squared ##Coefficient of determination (R^2)
[1] 0.6510794
> cor(cars$dist,cars$speed)^2 ##or this script for R^2
[1] 0.6510794
> sqrt(carsumry$r.squared) ##Coefficient of correlation (r)
[1] 0.8068949
> cor(cars$dist,cars$speed) ##or this script for r
[1] 0.8068949
>
```

Actually the formula of the correlation coefficient (r) is as follows.

$$r = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{[N\sum X^2 - (\sum X)^2][N\sum Y^2 - (\sum Y)^2]}}$$

```
> r=(cov(data$speed,data$dist))/sqrt(var(data$speed)*var(data$dist))
> r
[1] 0.8068949
> R2=r^2
> R2
[1] 0.6510794
>
```

So it can be seen that the correlation coefficient (r) between distance (dist) and speed (speed) is 0.81 which means that the relationship is quite tight. The determination coefficient (R^2) of the relationship is 0.65, which means that 65% of the distance variation can be explained by speed, that is, distance is affected by a speed of 65%, while the rest (35%) is influenced by other factors.

Another example, a research investigating the relationship between live weight (kg) before slaughtered and carcass weight (kg) (Pandiangan, 2016). The research question is that how to model the relationship between the two variables.

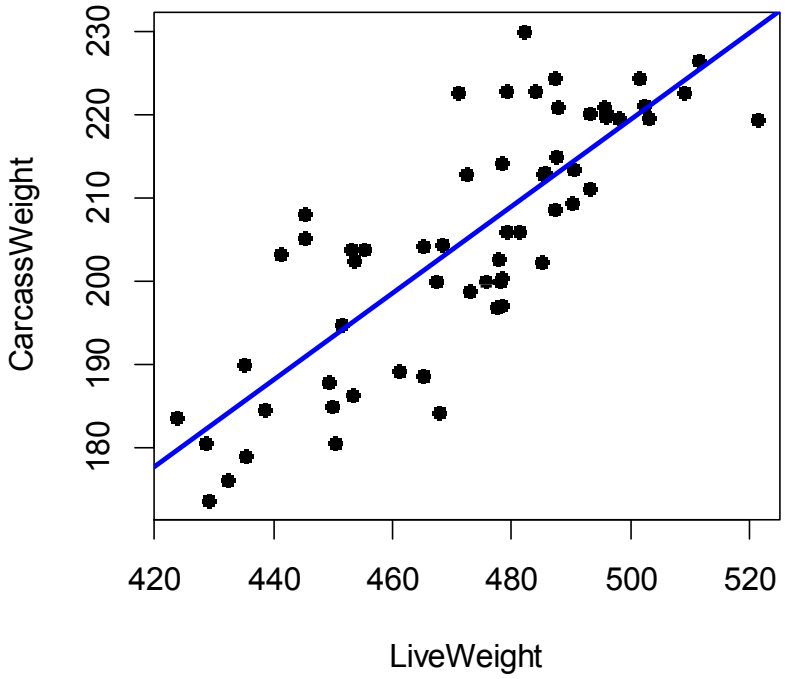
```
> data=read.csv("LinearRegression.csv", header=T)
> dim(data)
[1] 60  3
> head(data)
  LiveWeight CarcassWeight CarcassPercentage
1      423.8         183.7             43.35
```

```
2      428.5          180.7              42.17
3      429.0          173.7              40.49
4      435.3          179.2              41.17
5      435.1          190.1              43.69
6      432.4          176.3              40.77
> data
   LiveWeight CarcassWeight CarcassPercentage
1      423.80         183.70              43.35
2      428.50         180.70              42.17
3      429.00         173.70              40.49
4      435.30         179.20              41.17
5      435.10         190.10              43.69
6      432.40         176.30              40.77
7      438.50         184.60              42.10
8      441.30         203.20              46.05
9      445.30         208.00              46.71
10     449.70         185.10              41.16
11     449.20         188.00              41.85
12     445.20         205.20              46.09
13     450.40         180.70              40.12
14     453.30         186.40              41.12
15     453.60         202.50              44.64
16     453.10         203.90              45.00
17     451.30         194.90              43.19
18     455.20         203.80              44.77
19     465.20         204.20              43.90
20     467.30         200.00              42.80
21     465.10         188.70              40.57
22     461.10         189.20              41.03
23     468.30         204.50              43.67
24     467.70         184.36              39.42
25     475.60         200.00              42.05
26     478.30         197.20              41.23
27     477.90         200.00              41.85
28     477.40         196.90              41.24
29     478.30         214.20              44.78
30     478.40         200.40              41.89
31     479.15         206.00              42.99
32     477.80         202.80              42.44
33     472.81         198.80              42.05
34     479.21         222.80              46.49
35     471.10         222.60              47.25
36     472.50         212.80              45.04
37     487.80         220.90              45.28
38     487.30         214.90              44.10
39     487.20         224.40              46.06
40     483.80         222.90              46.07
41     485.50         213.10              43.89
42     481.30         205.90              42.78
43     487.10         208.70              42.85
44     482.10         230.00              47.71
45     485.10         202.30              41.70
46     485.30         212.80              43.85
```

```
47     490.20        209.40             42.72
48     495.50        221.00             44.60
49     490.40        213.40             43.52
50     495.70        220.30             44.44
51     493.20        211.20             42.82
52     501.50        224.30             44.73
53     503.10        219.50             43.63
54     511.40        226.50             24.74
55     495.70        219.80             44.34
56     493.20        220.10             44.63
57     502.20        221.20             44.05
58     521.20        219.40             42.10
59     508.90        222.70             43.76
60     497.80        219.50             44.09

> model <- lm(CarcassWeight ~ LiveWeight, data = data)
> summary(model)

Call:
lm(formula = CarcassWeight ~ LiveWeight, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-18.2578  -6.5941  -0.2086   4.2680  19.8847

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -40.89424   22.87043  -1.788    0.079 .
LiveWeight    0.52066    0.04837  10.764 1.89e-15 ***
---
Signif. codes: 0 ‘***’0.001‘**’0.01‘*’0.05‘.’0.1‘ ’1

Residual standard error: 8.581 on 58 degrees of freedom
Multiple R-squared:  0.6664,    Adjusted R-squared:  0.6607
F-statistic: 115.9 on 1 and 58 DF,  p-value: 1.889e-15
```

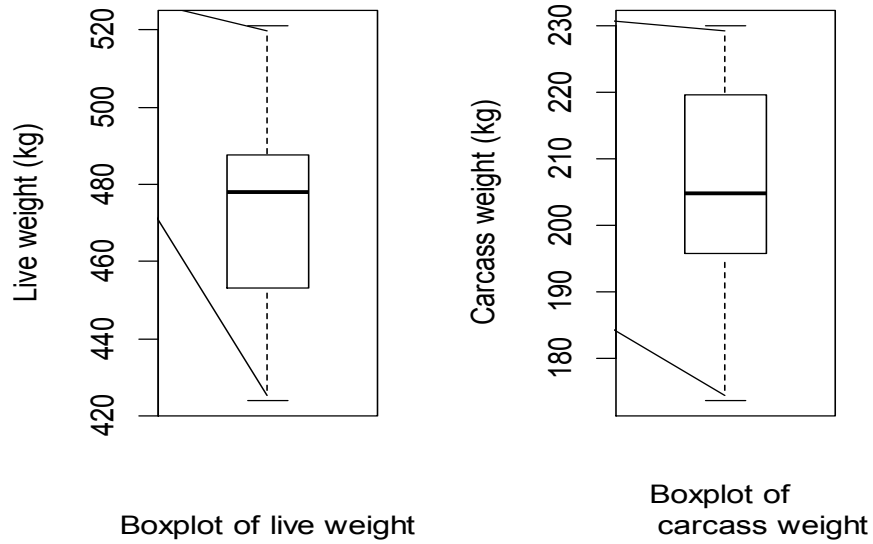
Based on summary we can report that the relationship between live weight and carcass weight can be formulated as carcass weight = -40.89 + 0.52 * live weight with R^2 0.67 and p-value less than 0.05. This result tells us that the model can be used to predict carcass weight if live weight is known.

```
> plot(CarcassWeight ~ LiveWeight, data = data, pch=16)
> abline(coef(model), col="blue", lwd=3)
>
```




```
> par(mfrow=(1:2))
> boxplot(data$LiveWeight,xlab="Boxplot of live weight",
+          ylab="Live weight (kg)")
> boxplot(data$CarcassWeight,xlab="Boxplot of
+          carcass weight",ylab="Carcass weight (kg)")
>
```

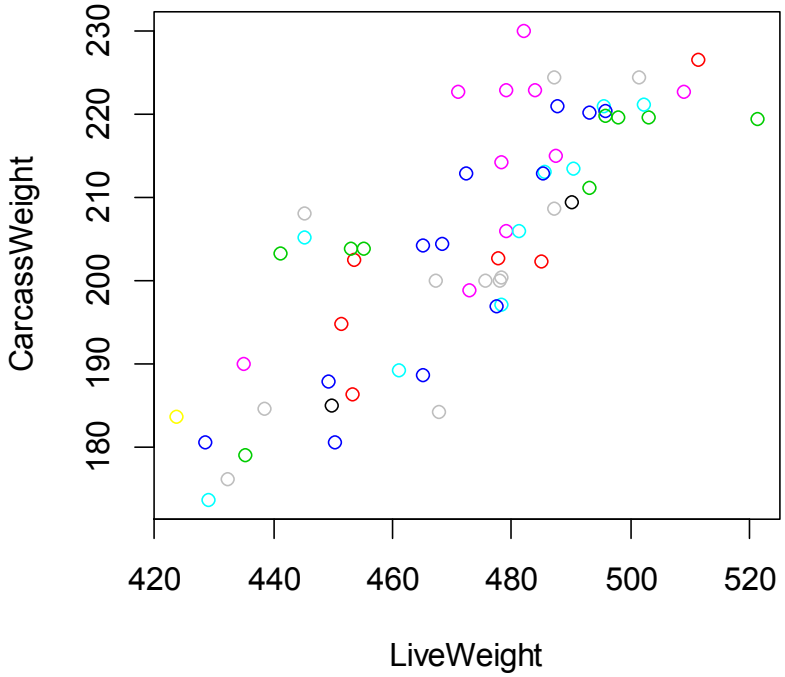
The following graphics are just to explore the data before analysing and making decision to use the equation resulted in. Boxplot is to see the spread of data of the two variables. Plotting graphic is similar with boxplot to see the tendency of the two variables. Whilst the quantile-quantile plot (Q-Q plot) is a scatterplot of two sets of quantiles against one another. If the points forming a line that's roughly straight means that both sets of quantiles came from the same distribution.



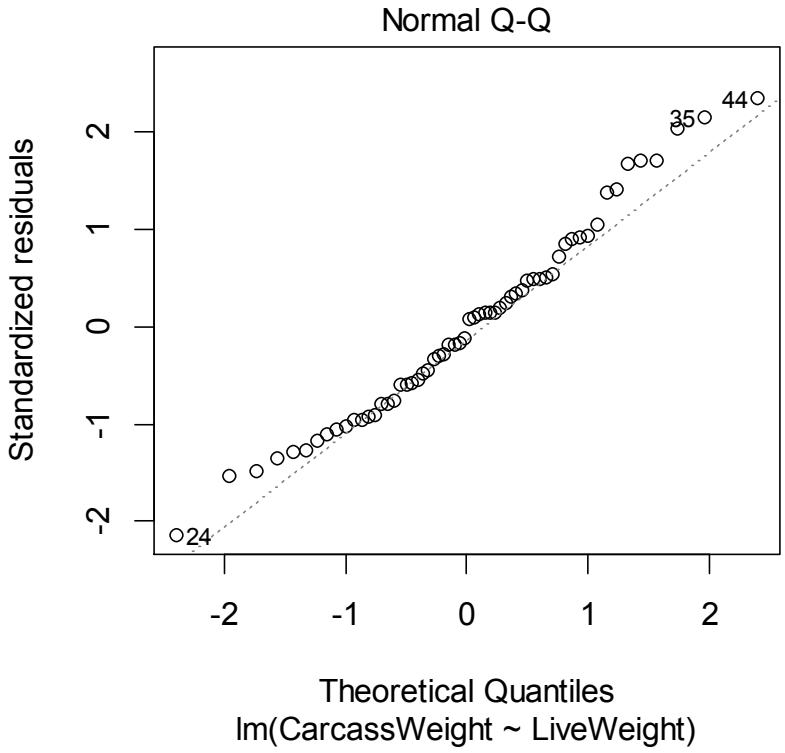
Boxplot of live weight

Boxplot of carcass weight

```
> plot(data, col=data$CarcassWeight)
```




```
> plot(model, 2)
```

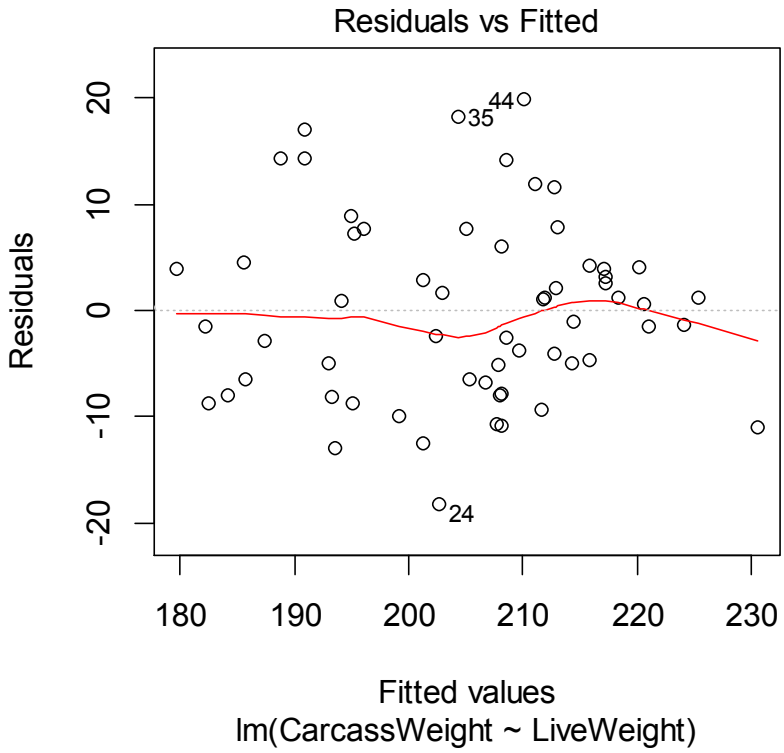


16.3 Assumption in Simple Linear Regression

Assumption for the linear regression model is linearity meaning that the relationship between the predictors (xs) and the outcome variable is linear because sometimes the relationship could be polynomial or logarithmic. The second assumption is normality meaning that residual errors should be normally distributed. The third assumption is homogeneity of residual variance meaning that residuals error are constant (homoscedasticity). The fourth assumption is independency of residual error meaning that data are independent from each other between variables. Therefore, we should check whether the regression model that we built has potential problems or not and whether the linear regression model met the assumption or not. Generally, examining the distribution of residuals can tell us more about our data.

The linearity can be diagnosed by evaluating the plot of residuals and fitted, like below (using previous data):

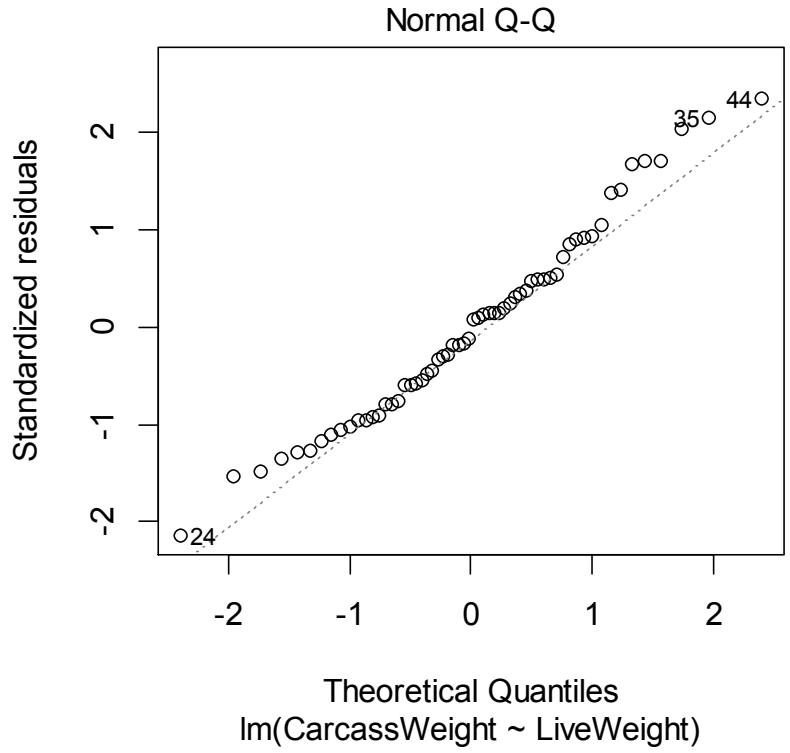
```
> plot(model, 1)
```



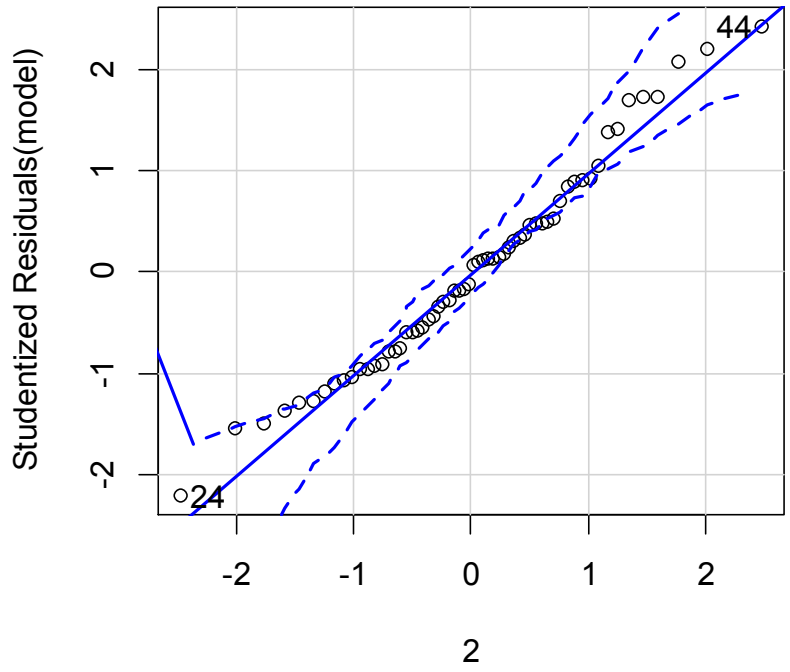
Based on the plot above the red line approximately close to horizontal at zero. This suggests that relationship between the predictors and the response variables is linear. Ideally, there is no fitted pattern for the residual plot, the presence of a pattern may indicate a problem with some aspect of the linear model. If the residual plot indicates a non-linear relationship, the predictor variables should be non-linear transformed (for instance, log(x), sqrt(x) and x^2).

Normality can be checked visually using Q-Q plot where residuals should approximately follow a straight line. Q-Q plot can use default in r or use package car.

```
> plot(model, 2)
```



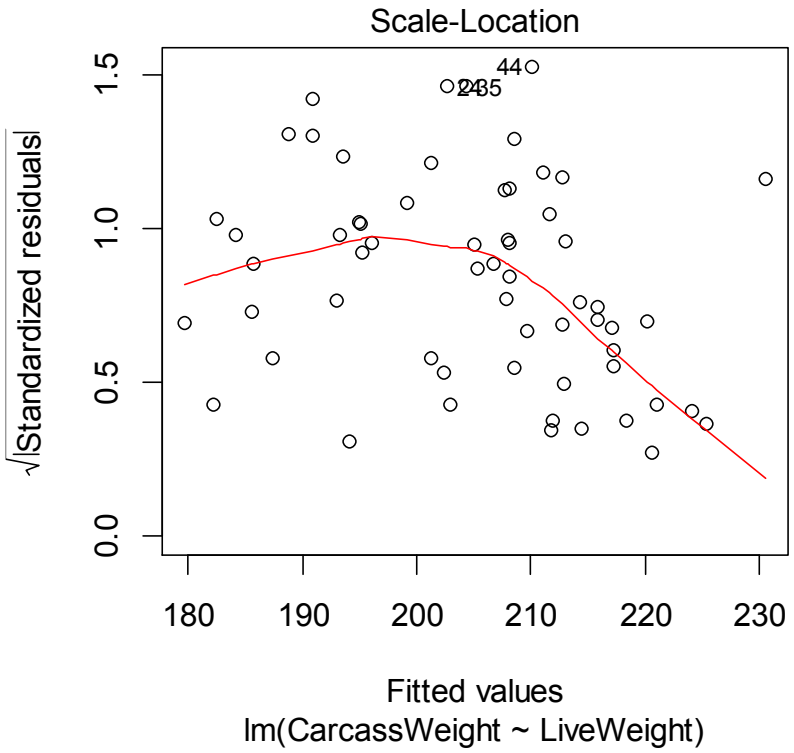

```
> library("car")
Loading required package: carData
> qqPlot(model,2)
[1] 24 44
```



Based on QQ plot above we can see that all the points fall approximately along the line, so we can assume that the data came from sample of population which is normally distributed.

Homogeneity variance can be diagnosed by inspecting the scale-location plot, or the spread-location plot.

```
> plot(model, 3)
```



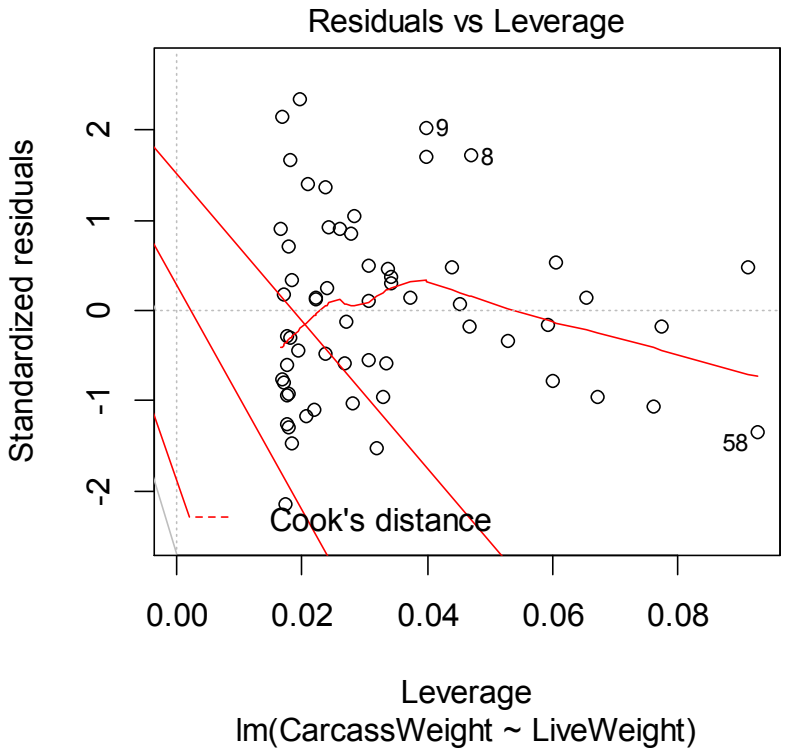
This plot shows if residuals are spread equally along the ranges of predictors. It's good if the line is horizontal with equally spread points.

It can be seen that the variability (variances) of the residual points increases a little bit and decrease with the value of the fitted response variable, suggesting non-constant variances in the residuals errors. A solution to reduce this heteroscedasticity problem is to use a log or square root transformation of the response variable (y).

Residuals versus leverage is used to identify outlier that is extreme values that might influence the regression model reliability. Outliers can be identified by inspecting the standardized residual, which is the residual divided by its estimated standard error. Standardized residuals can be interpreted as the number of standard errors away from the regression line. Observations whose standardized residuals are

greater than 3 in absolute value are possible outliers (James et al. 2014). While if data has extreme predictor x values meaning that the data has high leverage. Outliers and high leverage points can be identified by inspecting the residuals versus leverage plot:

```
> plot(model, 5)
```

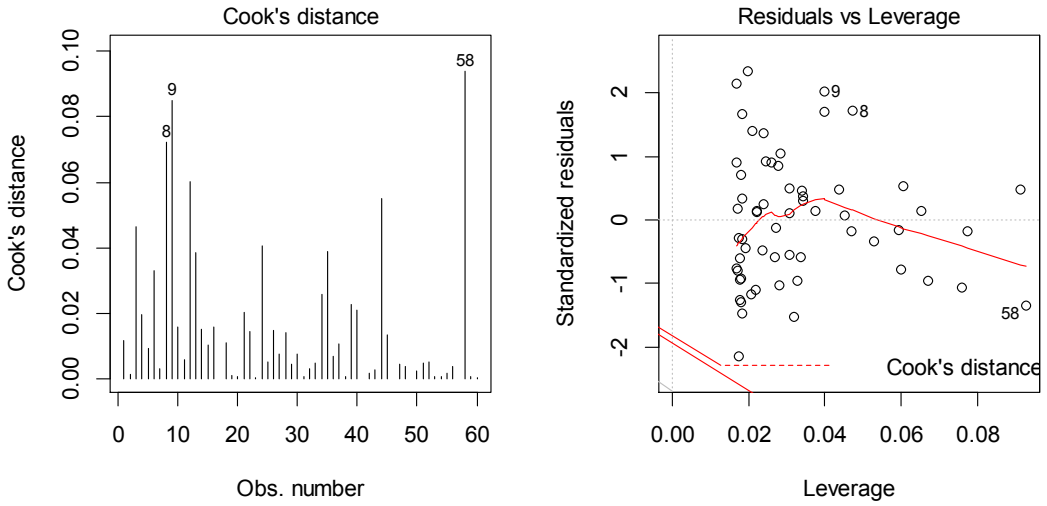


The plot above highlights the top 3 most extreme points (data number #8, #9 and #58), with a standardized residuals below 2 or -1. However, there is no outliers that exceed 3 or below -3 standard deviations, which is good. In addition, there is no high leverage point in the data, where all data points have a leverage statistic below 2(p + 1)/n = 4/60 = 0.067.

A value that is associated with a large residual is known as an influential value, because inclusion or exclusion the value can alter the results of the regression analysis. However, not all outliers are influential value in linear regression analysis. Statisticians have developed a metric called Cook's distance to determine the influence of a value. This metric defines influence as a combination of leverage and residual size. An observation has high influence if Cook's distance exceeds 4/(n - p - 1) (Bruce and Bruce 2017), where n is the number of observations and `p` the number of predictor variables.

Residuals versus leverage plot can help us to find if there is any influential observations. Outlying values are generally located at the upper right corner or at the lower right corner where data points can be influential against a regression line. The following plots illustrate the Cook's distance and the leverage of model discussed before:

```
> par(mfrow=c(1,2))
> plot(model, 4)
> plot(model, 5)
```



Based on the plot above the data don't present any influential points. Cook's distance lines (a red dashed line) are not shown on the Residuals vs Leverage plot because all points are well inside of the Cook's distance lines.

16.4 Multiple Linear Regression

Multiple linear regression is an extension of simple linear regression used to predict the outcome variable (y) based on different predictor variables (x). In other words, multiple Linear Regression explains how a single response variable y depends linearly on a number of predictor variables (x). In simple linear regression we have one predictor (x) and one response variable (y), but in multiple linear regression we have more than one predictor variable (x1, x2, ..., xn) and one response variable (y).

For example, we have three predictor variables (x), the predictive value of y can be expressed by the following equation:

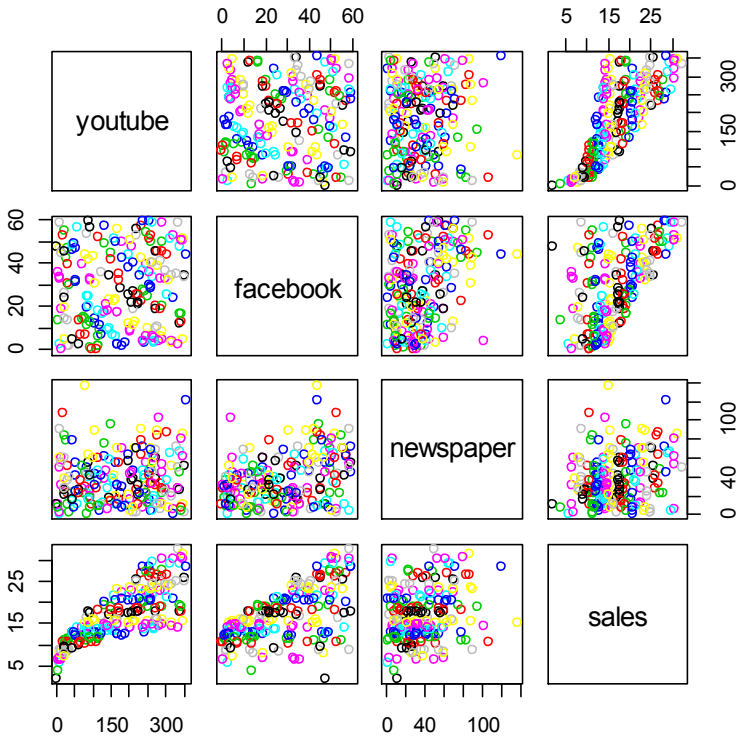
$y = b0 + b1*x1 + b2*x2 + ... + bn*xn + e$ or $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$

The value "b" is called regression weight (or beta coefficient), "bi" can be interpreted as the effect of xi in average on y from the increase of one unit "xi", if all other predictors are considered constant. To understand multiple linear regression we use sales data using the marketing / advertising method of Youtube, Facebook, and Newspaper (Shekar, 2018). Previously we made a model to estimate sales based on the advertising budget invested in Youtube, Facebook and newspapers, as follows:

sales = b0 + b1*youtube + b2*facebook + b3*newspaper

16.4.1 Exploring and Understanding Data

```
> data=read.csv("sales.csv", header=T)
> dim(data)
[1] 200   4
> head(data)
  youtube facebook newspaper sales
1  276.12    45.36     83.04 26.52
2   53.40    47.16     54.12 12.48
3   20.64    55.08     83.16 11.16
4  181.80    49.56     70.20 22.20
5  216.96    12.96     70.08 15.48
6   10.44    58.68     90.00  8.64
> plot(data, col=data$sales)
```




```
> plot(data[c(1,2,3)]) ## pairwise plot data inter predictor
    variables
```

Based on the plot above, it does not appear that any of predictor variables are highly correlated, or have a strong linear relationship with one another. Additional assumption for multiple linear regression is that between two predictor or independent variables does not highly correlate each other. This assumption is called multicollinearity. Furthermore, between predictor and response variable we can see that there is relationship between the two variables, between youtube and sales and between facebook and sales appear to have relationship. While between newspaper and sales does not seem to have relationship.

To make sure how big the correlation between two predictor variables we use cor(), as follows.

```
> cor(data[c(1,2, 3)]) ## to make sure the correlation
                youtube   facebook  newspaper
youtube    1.00000000 0.05480866 0.05664787
```

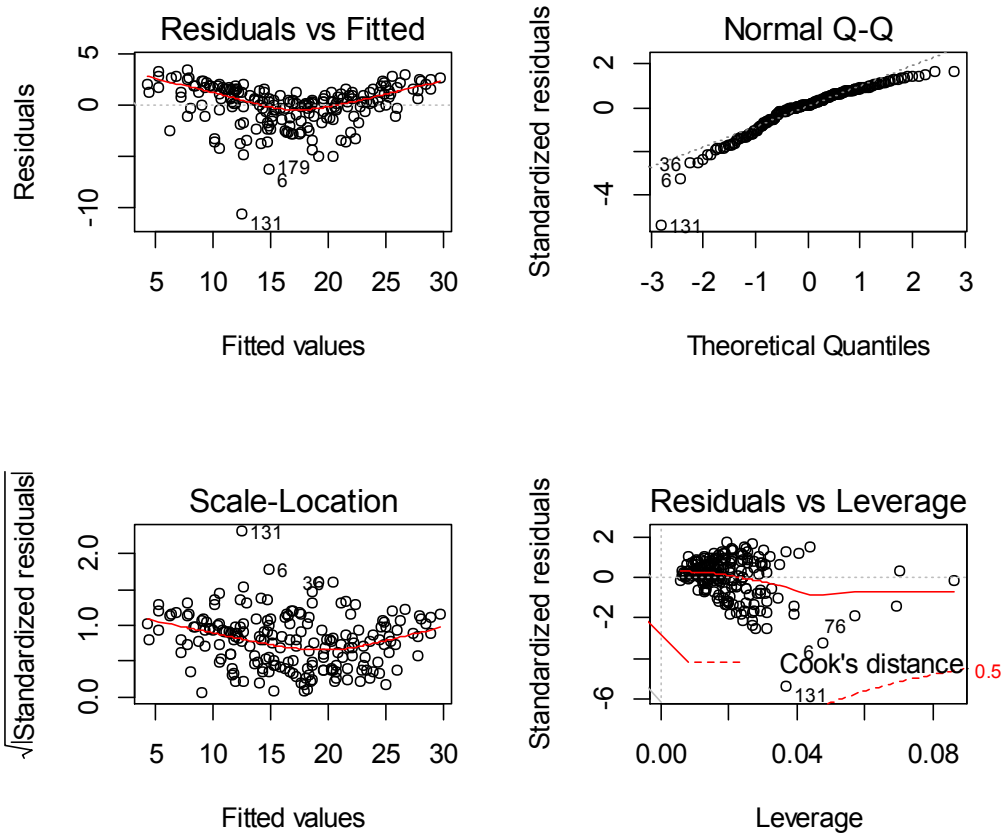
```
facebook  0.05480866 1.00000000 0.35410375
newspaper 0.05664787 0.35410375 1.00000000
```

We can see that the pairwise correlations between predictor variables are very low. It will be a problem if the correlation is greater than 0.9 meaning that there is multicollinearity issue in the model. Other method to evaluate multicollinearity is by calculating VIF (variance Inflation Factor). If the VIF > 10 means that there is multicollinearity of the data set. The VIF scores should be close to 1 but under 5 is fine and 10+ indicates that the variable is not needed and can be removed from the model. Based on VIF analysis it can be reported that all the VIF values in this analysis have scores close to 1, so that we can continue to the next steps.

```
> library(car)
Loading required package: carData
> vif(model) ##other way to check multicollinearity
  youtube  facebook newspaper
 1.004611  1.144952  1.145187
```

Other assumption for multiple linear regression is the same as assumption for simple linear regression, including that dataset plausibly came from similar normal distribution. If the distributions are similar, the points in the Q-Q plot will approximately lie on a line of y = x. The following plots are plots to check assumption for multiple linear regression discussed in assumption in simple linear regression, including normality.

```
> par(mfrow=c(2,2))
> plot(model, 1)
> plot(model, 2)
> plot(model, 3)
> plot(model, 5)
```



Based on plots above there is outlier for data number #6, #76 and #131 and high leverage. We should check further in the analysis.

16.4.2 Building Regression Model

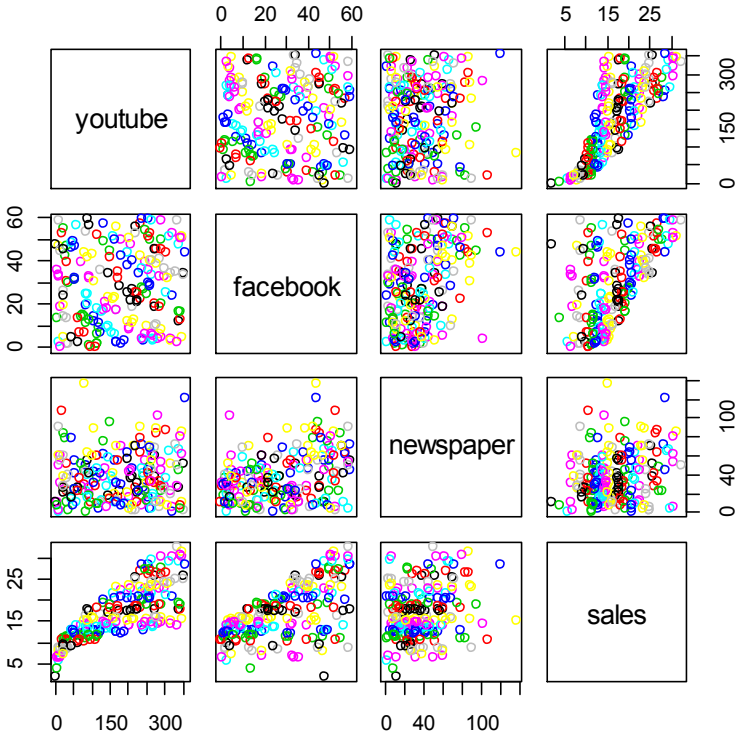
```
> model <- lm(sales ~ youtube + facebook + newspaper, data =
      data)
> summary(model)

Call:
lm(formula = sales ~ youtube + facebook + newspaper, data =
      data)

Residuals:
     Min       1Q   Median       3Q      Max
-10.5932  -1.0690   0.2902   1.4272   3.3951

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.526667   0.374290   9.422   <2e-16 ***
youtube      0.045765   0.001395  32.809   <2e-16 ***
```

```
facebook     0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’
      1

Residual standard error: 2.023 on 196 degrees of freedom
Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8956
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16

>
> anova(model)
Analysis of Variance Table

Response: sales
           Df Sum Sq Mean Sq   F value Pr(>F)
youtube     1 4773.1  4773.1 1166.7308 <2e-16 ***
facebook    1 2225.7  2225.7  544.0501 <2e-16 ***
newspaper   1    0.1     0.1    0.0312 0.8599
Residuals 196  801.8     4.1
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

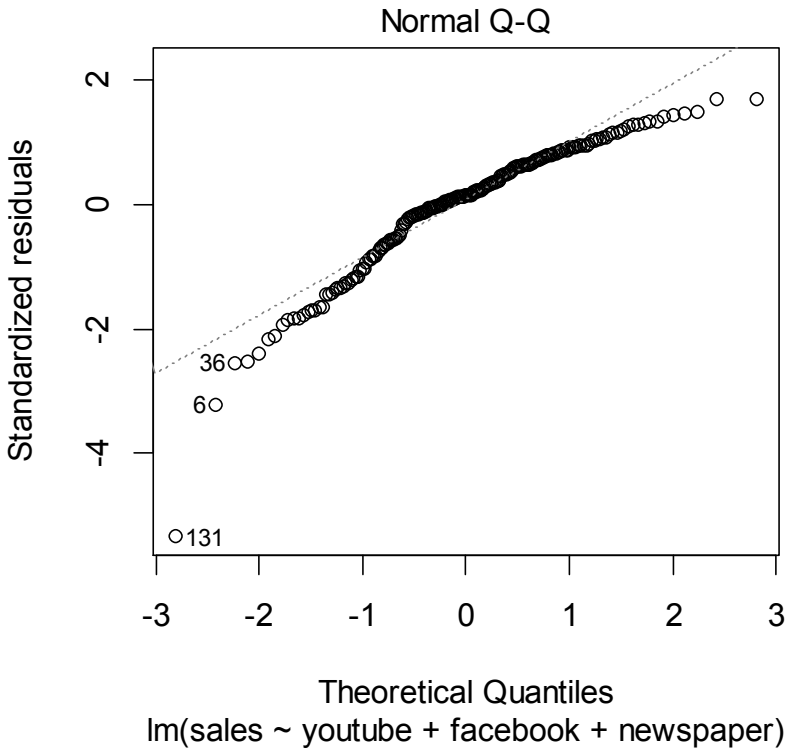
Based on the summary and ANOVA above, the p-value of the F-statistic is <2.2e-16, which is very significant. This means that, at least, one of the predictor variables is significantly related to the outcome variable (sales).

It can be seen that, changes in the youtube and facebook advertising budgets are significantly related to changes in sales, while changes in newspaper budgets are not significantly related to sales. Based on the three predictor variables, the coefficient (b) can be interpreted as the mean effect on y due to the increase in one unit in the predictor, assuming other predictors are considered constant. For example, if the youtube and newspaper advertising budgets are considered constant, every addition of 1,000 dollars to Facebook ads will increase sales by an average of about 0.1885 * 1000 = 189 sales units. Likewise, with every increase of 1000 dollars in advertising on YouTube, where other ads are constant, we can expect an average increase of 0.045 * 1000 = 45 sales units.

```
> plot(data, col=data$sales)
```


youtube
facebook
newspaper
sales
0 20 40 60
5 15 25
0 150 300
0 20 40 60
0 40 100
5 15 25
0 150 300
0 40 100

```
> plot(model, 2)
```


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(sales ~ youtube + facebook + newspaper)
36
6
131
-3 -2 -1 0 1 2 3
-4 -2 0 2

16.4.3 Finding The Best Model in Multiple Linear Regression

Finding predictors that influence or support response variable for the best model in multiple linear regression is by comparing the two models using the adjusted R^2, using Akaike's Information Criterion (AIC) value, using anova command or by doing stepwise regression.

Adjusted R^2 which is computed from the ANOVA table or computed as follows can be used to compare regression models with:

$$R^2adj = 1 - \left[\frac{(n\ -\ 1)}{(n\ -\ p)}.(1\ -\ R^2)\right]$$

```
> data=read.csv("sales.csv", header=T)
> model1 <- lm(sales ~ youtube + facebook + newspaper, data = data)
> model2 <- lm(sales ~ youtube + facebook, data = data)
> model3 <- lm(sales ~ youtube + newspaper, data = data)
> model4 <- lm(sales ~ facebook + newspaper, data = data)
> model5 <- lm(sales ~ youtube, data = data)
> model6 <- lm(sales ~ facebook, data = data)
> model7 <- lm(sales ~ newspaper, data = data)
> summary(model1)$adj.r.squared
[1] 0.8956373
> summary(model2)$adj.r.squared
[1] 0.8961505
> summary(model3)$adj.r.squared
[1] 0.6422399
> summary(model4)$adj.r.squared
[1] 0.3259306
> summary(model5)$adj.r.squared
[1] 0.6099148
> summary(model6)$adj.r.squared
[1] 0.3286589
> summary(model7)$adj.r.squared
[1] 0.04733317
>
```

Based on the value of adjusted R2, model 2 (sales = youtube + facebook) is the best among the seven model.

Akaike's Information Criterion (AIC) is more general measure to apply to to compare model, which is the lower value is the better.

```
    > AIC(model1); AIC(model2); AIC(model3); AIC(model4);
AIC(model5);AIC(model6); AIC(model7)
    [1] 855.2909
```

```
[1] 853.3227
[1] 1100.707
[1] 1227.401
[1] 1117.02
[1] 1225.602
[1] 1295.6
>
```

So model 2 is the best because its AIC is the smallest.

Command anova can be used to find the best model by comparing the two models. For example, we use previous example:

```
> data=read.csv("sales.csv", header=T)
> dim(data)
[1] 200   4
> head(data)
  youtube facebook newspaper sales
1  276.12    45.36     83.04 26.52
2   53.40    47.16     54.12 12.48
3   20.64    55.08     83.16 11.16
4  181.80    49.56     70.20 22.20
5  216.96    12.96     70.08 15.48
6   10.44    58.68     90.00  8.64
> model1 <- lm(sales~youtube+facebook+newspaper,data=data)
> model2 <- lm(sales ~ youtube + facebook, data = data)
> model3 <- lm(sales ~ youtube + newspaper, data = data)
> model4 <- lm(sales ~ facebook + newspaper, data = data)
> model5 <- lm(sales ~ youtube, data = data)
> model6 <- lm(sales ~ facebook, data = data)
> model7 <- lm(sales ~ newspaper, data = data)
> anova(model1,model2,model3,model4,model5,model6,model7)
Analysis of Variance Table

Model 1: sales ~ youtube + facebook + newspaper
Model 2: sales ~ youtube + facebook
Model 3: sales ~ youtube + newspaper
Model 4: sales ~ facebook + newspaper
Model 5: sales ~ youtube
Model 6: sales ~ facebook
Model 7: sales ~ newspaper
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    196  801.8
2    197  802.0 -1     -0.13 0.0312 0.8599
3    197 2762.7  0  -1960.77
4    197 5205.4  0  -2442.63
5    198 3027.6 -1   2177.72
6    198 5210.6  0  -2182.97
7    198 7394.1  0  -2183.51
>
```

Based on ANOVA result it can be concluded that model 1 and model 2 is not different significantly with RSS 801 and 802, respectively. Whilst the rest of the model (model 3-7) is not better than model 1 and model 2 with big different of RSS. For parsimonious model, model 2 is the best because model 1 and model 1 is not different significantly. In addition, coefficient of predictor for newspaper is not significant (see previous discussion).

Stepwise method can be used for finding the best model, as follow.

```
> fit <- step(lm(sales ~ youtube+facebook+newspaper, data=data))
Start:  AIC=285.72
sales ~ youtube + facebook + newspaper

             Df Sum of Sq    RSS    AIC
- newspaper  1       0.1  802.0 283.75
<none>                    801.8 285.72
- facebook   1    1960.9 2762.7 531.13
- youtube    1    4403.5 5205.4 657.83

Step:  AIC=283.75
sales ~ youtube + facebook

           Df Sum of Sq    RSS    AIC
<none>                   802.0 283.75
- facebook  1    2225.7 3027.6 547.44
- youtube   1    4408.7 5210.6 656.03
>
```

Based on stepwise analysis the best model is sales ~ youtube + facebook with lower AIC (Akaike Information Criterion), the lower AIC the best the model. The result is the same as anova command.

16.4.4 Comparing Two Slopes in Multiple Linear Regression

To compare two slopes of linear regression model can use analysis of covariance (ANCOVA) method. By testing the effect of a categorical factor on a response variable (y) and controlling for the effect of a continuous covariable (x) we can compare the two lines or slopes. If there is interaction between the categorical variable (i.e. treatment effect) and the continuous independent variable (x) means that the regression lines have different slopes. If the slopes are not different or parallel but with significant effect of treatment means that the two regression model have different intercept. Furthermore, if the treatment effect is not different significantly and there is no

interaction between categorical and continuous variable means that there is only a single regression line.

The following example is to investigate whether the regression of carcass weight (pounds) on back fat thickness (mm) in pig fed with different ration (Ration A and Ration B) have the same slopes (Steel and Torrie, 1989). Data of carcass weight and back fat thickness is presented in the following table.

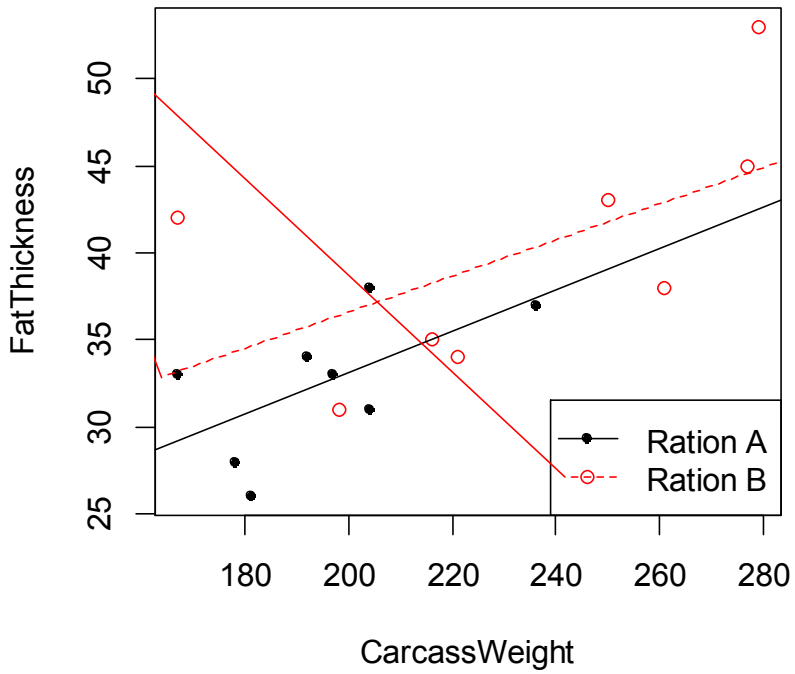
Ration A		Ration B	
Carcass weight	Back fat thickness	Carcass weight	Back fat thickness
167	33	167	42
192	34	261	38
204	38	279	53
197	33	221	34
181	26	216	35
178	28	198	31
236	37	277	45
204	31	250	43

```
> data=read.csv("TwoSlopes.csv", header=T)
> data
   Ration CarcassWeight FatThickness
1
2       A           192           34
3       A           204           38
4       A           197           33
5       A           181           26
6       A           178           28
7       A           236           37
8       A           204           31
9       B           167           42
10      B           261           38
11      B           279           53
12      B           221           34
13      B           216           35
14      B           198           31
15      B           277           45
16      B           250           43
> mod1 <- aov(FatThickness~CarcassWeight*Ration, data=data)
> mod2 <- aov(FatThickness~CarcassWeight+Ration, data=data)
> anova(mod1,mod2)
Analysis of Variance Table

Model 1: FatThickness ~ CarcassWeight * Ration
Model 2: FatThickness ~ CarcassWeight + Ration
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     12 307.69
2     13 308.25 -1  -0.55502 0.0216 0.8855
```

Based on comparison between model 1 (interaction) and model 2 (without interaction) it can be concluded that the two models are not different significantly with Fstatisic `0.0216` and p-value `0.8855` meaning that the slopes of the two regression model are the same. We can check further for the intercept visually or by doing ANOVA for investigating treatment effect (ration).

```
>
> RationA <- subset(data, Ration=="A")
> RationB <- data[data$Ration=='B',]
> RationA
  Ration CarcassWeight FatThickness
1      A           167           33
2      A           192           34
3      A           204           38
4      A           197           33
5      A           181           26
6      A           178           28
7      A           236           37
8      A           204           31
> RationB
   Ration CarcassWeight FatThickness
9       B           167           42
10      B           261           38
11      B           279           53
12      B           221           34
13      B           216           35
14      B           198           31
15      B           277           45
16      B           250           43
> reg1 <- lm(FatThickness~CarcassWeight, data=RationA); summary(reg1)

Call:
lm(formula = FatThickness ~ CarcassWeight, data = RationA)

Residuals:
   Min     1Q Median     3Q    Max
-4.855 -2.520 -0.064  2.332  4.418

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.39470   12.31155   0.763    0.474
CarcassWeight  0.11856    0.06285   1.886    0.108

Residual standard error: 3.514 on 6 degrees of freedom
Multiple R-squared:  0.3723,    Adjusted R-squared:  0.2677
F-statistic: 3.558 on 1 and 6 DF,  p-value: 0.1082

> reg2 <- lm(FatThickness~CarcassWeight, data=RationB); summary(reg2)

Call:
lm(formula = FatThickness ~ CarcassWeight, data = RationB)

Residuals:
   Min     1Q Median     3Q    Max
-5.438 -4.853 -1.457  2.930  8.770
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   15.94876   13.98970    1.14    0.298
CarcassWeight  0.10348    0.05913    1.75    0.131

Residual standard error: 6.24 on 6 degrees of freedom
Multiple R-squared:  0.3379,    Adjusted R-squared:  0.2276
F-statistic: 3.063 on 1 and 6 DF,  p-value: 0.1307

>
> plot(FatThickness~CarcassWeight, data=data, type='n')
> points(RationA$CarcassWeight,RationA$FatThickness, pch=20)
> points(RationB$CarcassWeight,RationB$FatThickness, col="red",pch=1)
> abline(reg1, lty=1)
> abline(reg2, lty=2, col="red")
> legend("bottomright", c("Ration A","Ration B"),
+  lty=c(1,2),col=c("black","red"), pch=c(20,1) )
>
```




```
> summary(mod1)
                    Df Sum Sq Mean Sq F value  Pr(>F)
CarcassWeight        1  361.0   361.0  14.077 0.00276 **
Ration               1   34.2    34.2   1.335 0.27046
CarcassWeight:Ration 1    0.6     0.6   0.022 0.88548
Residuals           12  307.7    25.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> summary(mod2)
             Df Sum Sq Mean Sq F value  Pr(>F)
CarcassWeight  1  361.0   361.0  15.223 0.00182 **
Ration         1   34.2    34.2   1.443 0.25104
Residuals     13  308.3    23.7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Based on ANOVA it can be seen that ration effect is not different significantly, meaning that the intercepts of the two regression model is not different statistically although visually it is a bit different.

16.5 Nonlinear Regression

Nonlinear regression is a form of regression analysis where data of observation are modelled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables. Statistical model of the nonlinear model can be formulated as follows:

$$y \sim f(x,\beta)$$

Where y is dependent variable, x is independent variable, f is nonlinear function with parameter β. For example, the Michaelis-Menten model for enzyme kinetics with two parameters (β_1 and β_2) and one independent variable, as follows :

$$f(x,\beta) = y = \frac{\beta 1 x}{\beta 2 + x}$$

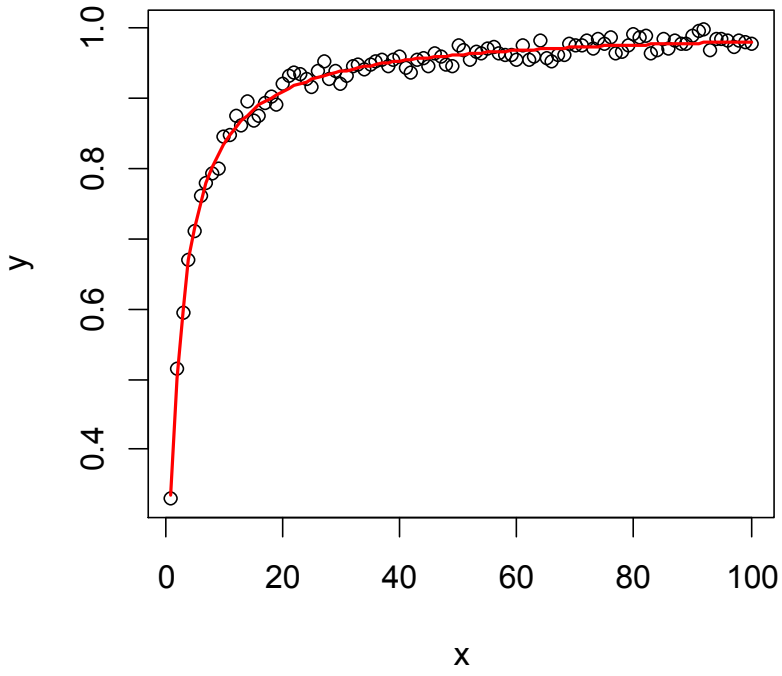


Figure above is scripted as follows using default R with function nls.

```
> x <- 1:100
> y <- 1*x/(2+x) + rnorm(100, 0, 0.01)
```

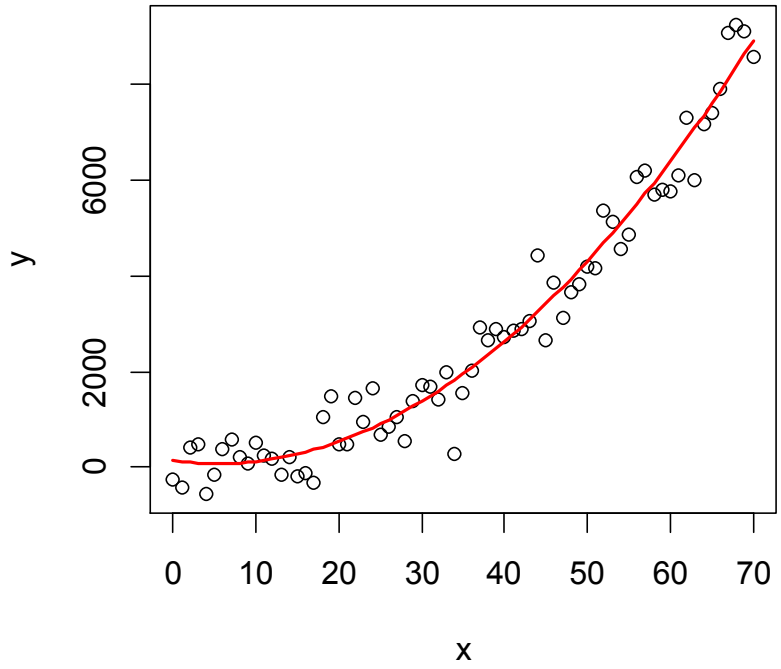
```
> nm <- nls(y ~ a*x/(b+x), start = list(a = 1, b = 1))
> nm
Nonlinear regression model
  model: y ~ a * x/(b + x)
   data: parent.frame()
      a      b
0.9972 1.9830
 residual sum-of-squares: 0.01008

Number of iterations to convergence: 4
Achieved convergence tolerance: 1.181e-06
> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```

The following nonlinear equation is only example using data generated by random function.

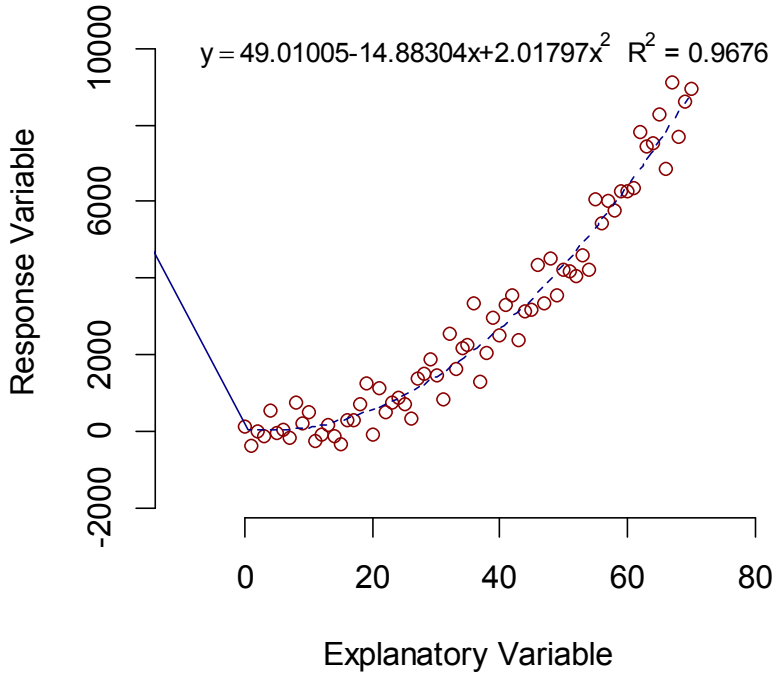
16.5.1 Quadratic

Quadratic equation is y~a+b*x+c*x^2.

```
> x <- 0:70
> y <- 3 - 14*x + 2*x^2 + rnorm(70, 20, 500)

Warning message:
In 3 - 14 * x + 2 * x^2 + rnorm(70, 20, 500) :
  longer object length is not a multiple of shorter
object length

> nm <- nls(y ~ a + b*x + c*x^2, start=list(a=4,b=5,c=5))
> nm

Nonlinear regression model
  model: y ~ a + b * x + c * x^2
   data: parent.frame()
      a       b       c
130.096 -21.316   2.096
 residual sum-of-squares: 18210790

Number of iterations to convergence: 1
Achieved convergence tolerance: 3.801e-08

> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```



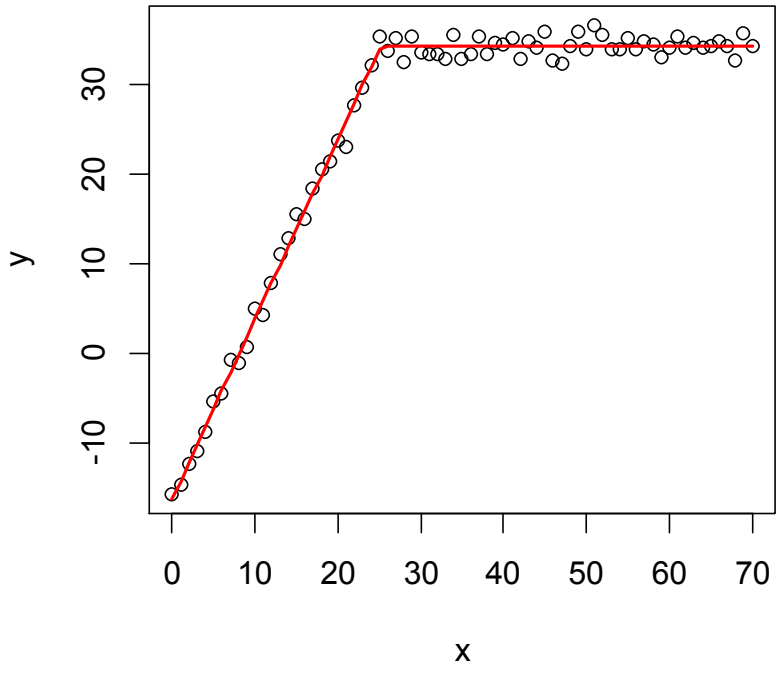

```
> library(easynls)
> data1=data.frame(x,y)
> nlsplot(data1, model=2)
```



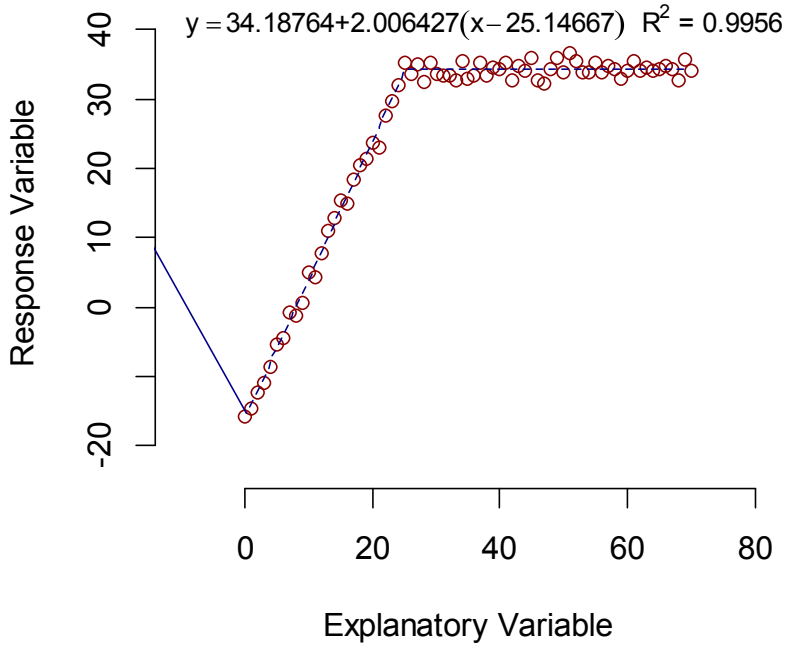
16.5.2 Linear Plateu

In agricultural research, especially in soil fertility and soil chemistry, the response function usually exhibits a plateau effect. In such situations, it is often appropriate to approximate the underlying function with two intersecting linear lines. Linear plateu equation is y ~ a + b * (x - c) * (x <= c).

```
> x <- 0:70
> y <- 14 + 2 * (x - 25) * (x <= 25) + rnorm(70, 20, 1)
Warning message:
In 14 + 2 * (x - 25) * (x <= 25) + rnorm(70, 20, 1) :
  longer object length is not a multiple of shorter object length
> nm <- nls(y ~ a + b * (x - c) * (x <= c), start=list(a=1,b=1,c=1))
> nm
Nonlinear regression model
  model: y ~ a + b * (x - c) * (x <= c)
   data: parent.frame()
     a      b      c
34.188  2.006 25.147
 residual sum-of-squares: 73.69

Number of iterations to convergence: 6
Achieved convergence tolerance: 4.525e-10
> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```



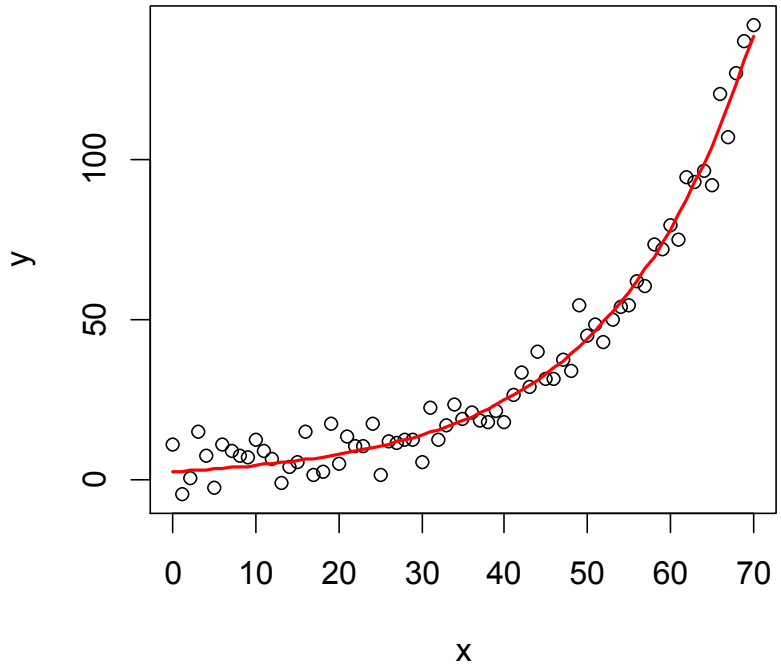

```
> data1 <- data.frame(x,y)
> nlsplot(data1, model=3)
>
```



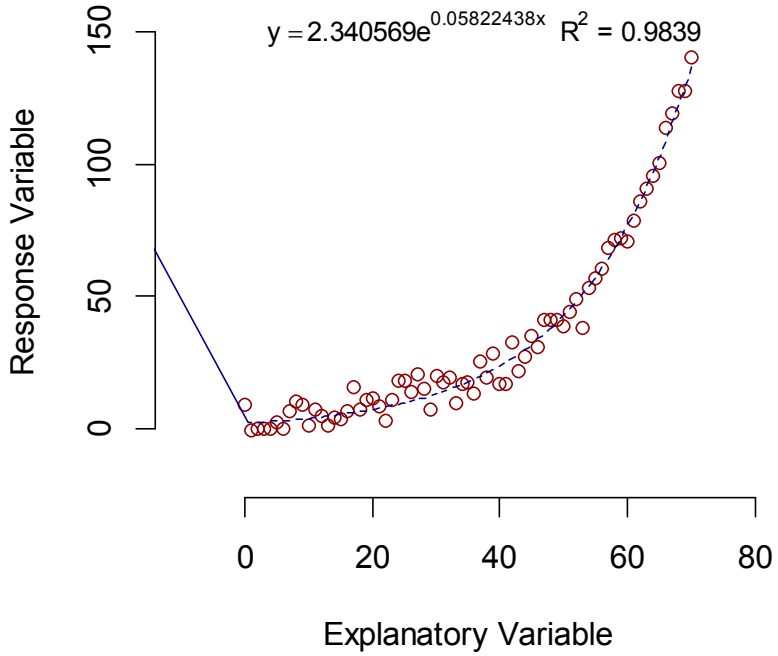
16.5.3 Exponential

The use of exponential regression is to model data or situations that start to growth slowly and then increases rapidly without bound, or begins rapidly and then slows down to get closer and closer to zero. Exponential equation is y~a*exp(b*x).

```
> x <- 0:70
> y <- 2*exp(0.06*x) + rnorm(70, 3, 5)
Warning message:
In 2 * exp(0.06 * x) + rnorm(70, 3, 5) :
  longer object length is not a multiple of shorter
object length
> nm <- nls(y ~ a*exp(b*x), start=list(a=1,b=0.01))
> nm
Nonlinear regression model
  model: y ~ a * exp(b * x)
   data: parent.frame()
      a       b
2.49092 0.05744
 residual sum-of-squares: 2153

Number of iterations to convergence: 7
Achieved convergence tolerance: 3.495e-06
> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```



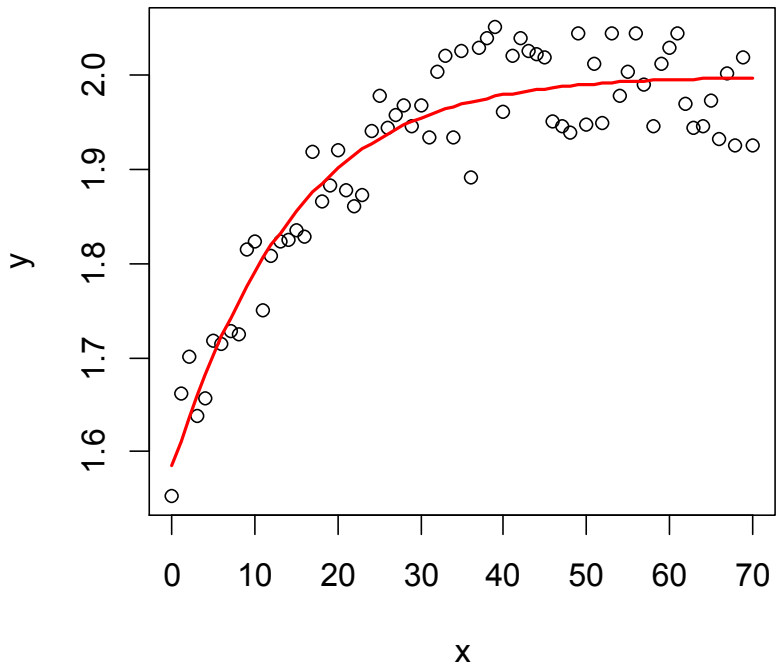

```
> data1=data.frame(x,y)
> nlsplot(data1, model=6, start=c(250,0.05))
>
```



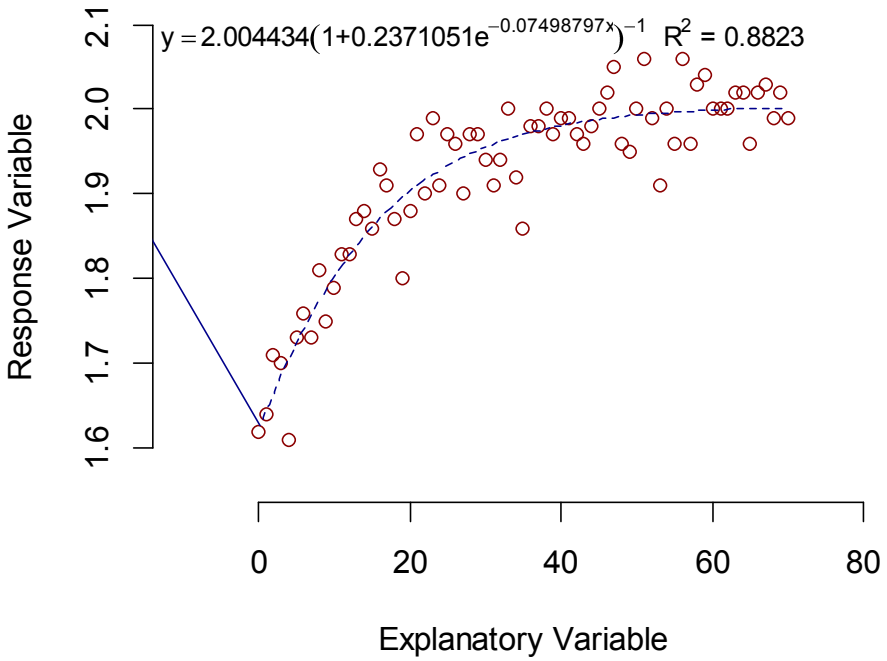
16.5.4 Logistic

Logistic equation is y~a*(1+b*(exp(-c*x)))^-1

```
> x <- 0:70
> y <- 1*(1+ 0.6*(exp(-0.08*x)))^-1 + rnorm(70, 1, 0.04)
Warning message:
In 1 * (1 + 0.6 * (exp(-0.08 * x)))^-1 + rnorm(70, 1, 0.04) :
  longer object length is not a multiple of shorter object length
> nm <- nls(y ~ a*(1+b*(exp(-c*x)))^-1, start=list(a=10,b=0.1,c=0.1))
> nm
Nonlinear regression model
  model: y ~ a * (1 + b * (exp(-c * x)))^-1
   data: parent.frame()
      a       b       c
1.99937 0.26140 0.08126
 residual sum-of-squares: 0.1153

Number of iterations to convergence: 9
Achieved convergence tolerance: 3.655e-06
> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```



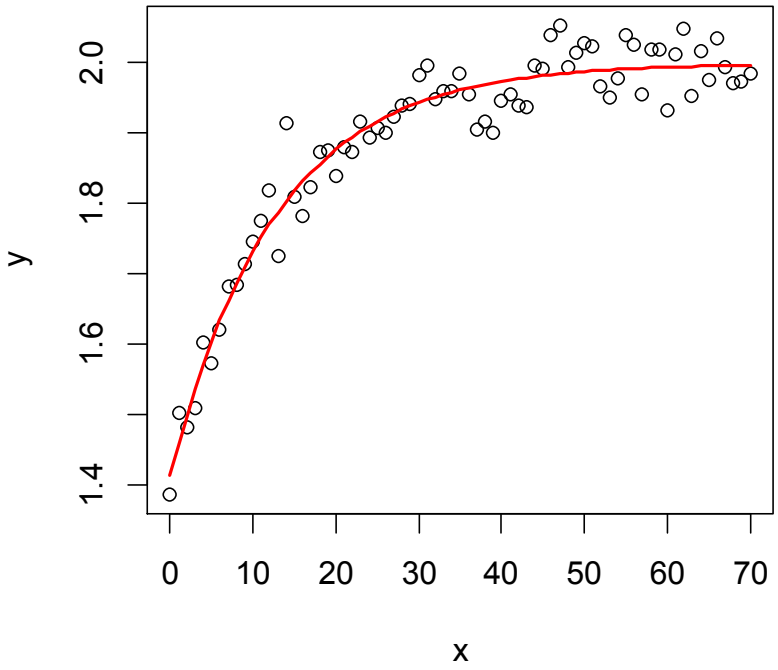

```
> data1=data.frame(x,y)
> nlsplot(data1, model=7, start=c(600,4,0.05))
>
```



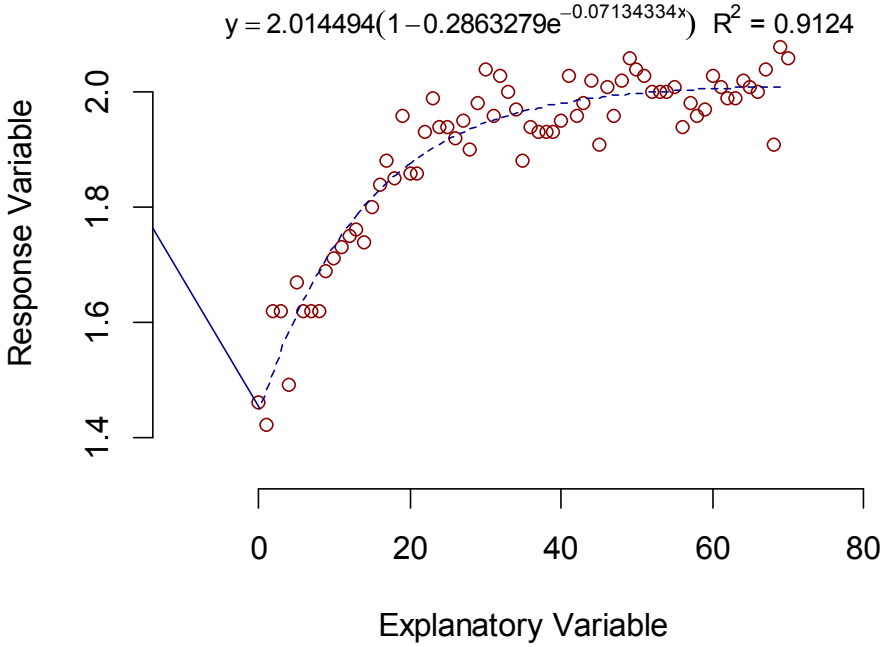
16.5.5 Brody

Brody equation is y~a*(1-b*(exp(-c*x)))

```
> x <- 0:70
> y <- 1*(1- 0.6*(exp(-0.08*x))) + rnorm(70, 1, 0.04)
Warning message:
In 1 * (1 - 0.6 * (exp(-0.08 * x))) + rnorm(70, 1, 0.04) :
  longer object length is not a multiple of shorter object length
> nm <- nls(y ~ a*(1-b*(exp(-c*x))), start=list(a=10,b=0.1,c=0.1))
> nm
Nonlinear regression model
  model: y ~ a * (1 - b * (exp(-c * x)))
   data: parent.frame()
      a       b       c
1.99787 0.29252 0.07829
 residual sum-of-squares: 0.0852

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.475e-06
> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```



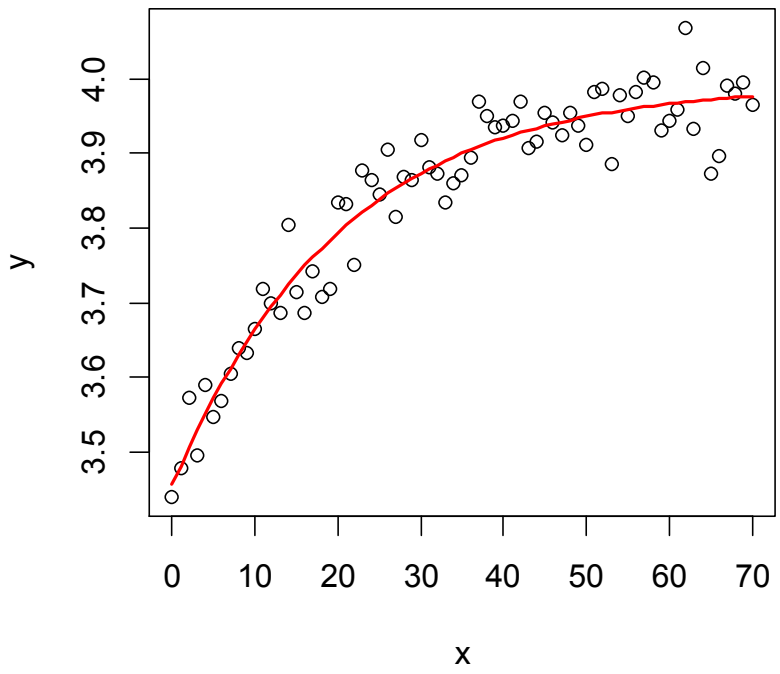

```
> data1=data.frame(x,y)
> nlsplot(data1, model=9, start=c(600,4,0.05))
>
```



16.5.6 Gompertz

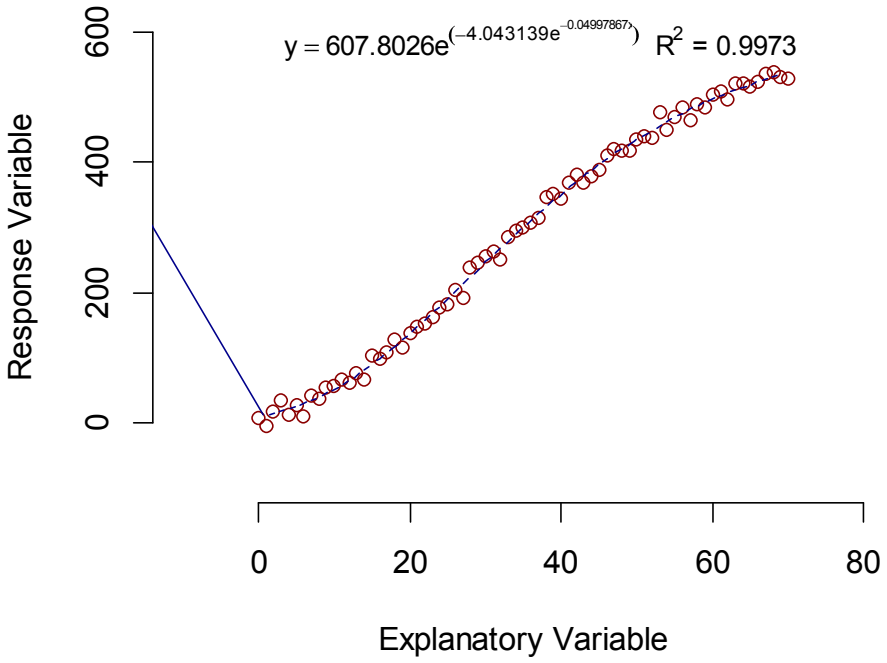
Gompertz equation is y~a*exp(-b*exp(-c*x)

```
> x <- 0:70
> y <- 3*exp(-0.2*exp(-0.05*x)) + rnorm(70, 1, 0.04)

Warning message:
In 3 * exp(-0.2 * exp(-0.05 * x)) + rnorm(70, 1, 0.04) :
  longer object length is not a multiple of shorter object length

> nm <- nls(y ~ a*exp(-b*exp(-c*x)), start=list(a=10,b=0.1,c=0.1))
> nm

Nonlinear regression model
  model: y ~ a * exp(-b * exp(-c * x))
   data: parent.frame()
      a       b       c
3.99145 0.14405 0.05223
 residual sum-of-squares: 0.107

Number of iterations to convergence: 5
Achieved convergence tolerance: 9.837e-06

> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```




```
> data1=data.frame(x,y)
> nlsplot(data1, model=10, start=c(600,4,0.05))
```

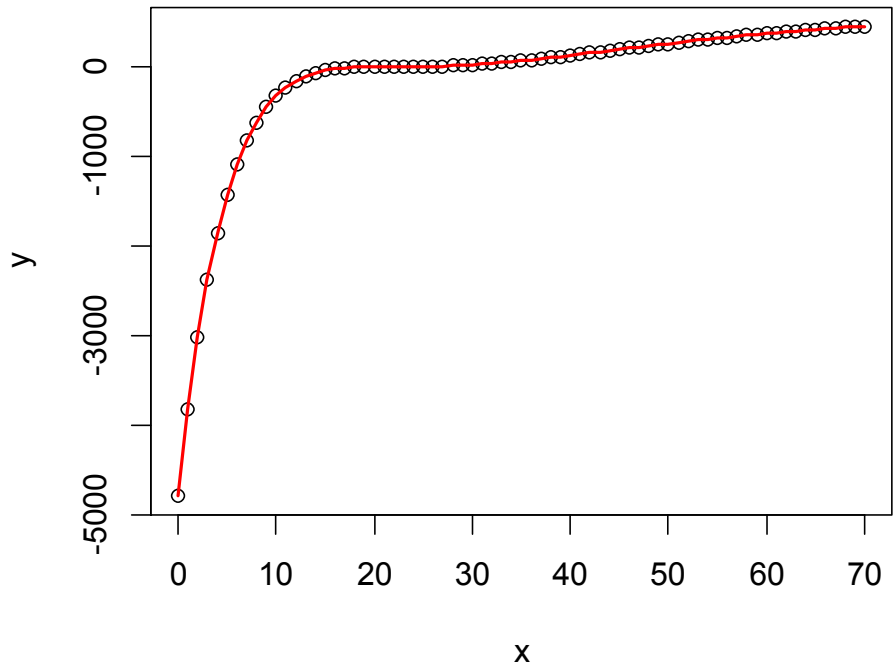
```
>
```



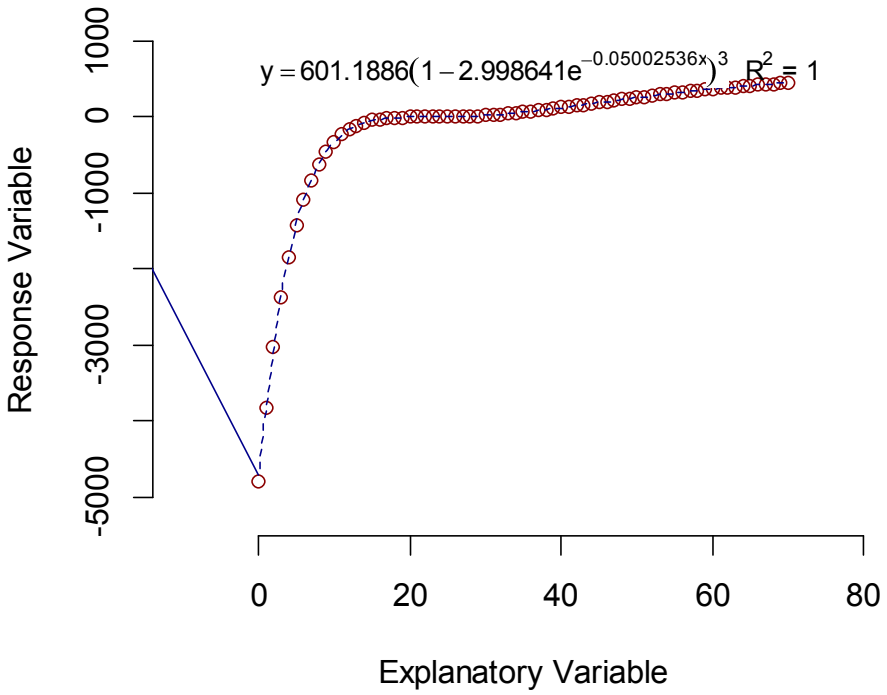
16.5.7 Van Bertalanffy

Van Bertalanffy equation is y~a*(1-b*(exp(-c*x)))^3

```
> x <- 0:70
> y <- 600*(1-3*(exp(-0.05*x)))^3 + rnorm(70, 1, 0.04)

Warning message:
In 600 * (1 - 3 * (exp(-0.05 * x)))^3 + rnorm(70, 1, 0.04) :
  longer object length is not a multiple of shorter object length

> nm <- nls(y ~ a*(1-b*(exp(-c*x)))^3, start=list(a=600,b=2,c=0.05))
> nm

Nonlinear regression model
  model: y ~ a * (1 - b * (exp(-c * x)))^3
   data: parent.frame()
        a         b         c
601.18865   2.99864   0.05003
 residual sum-of-squares: 20.3

Number of iterations to convergence: 5
Achieved convergence tolerance: 2.682e-07

> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```



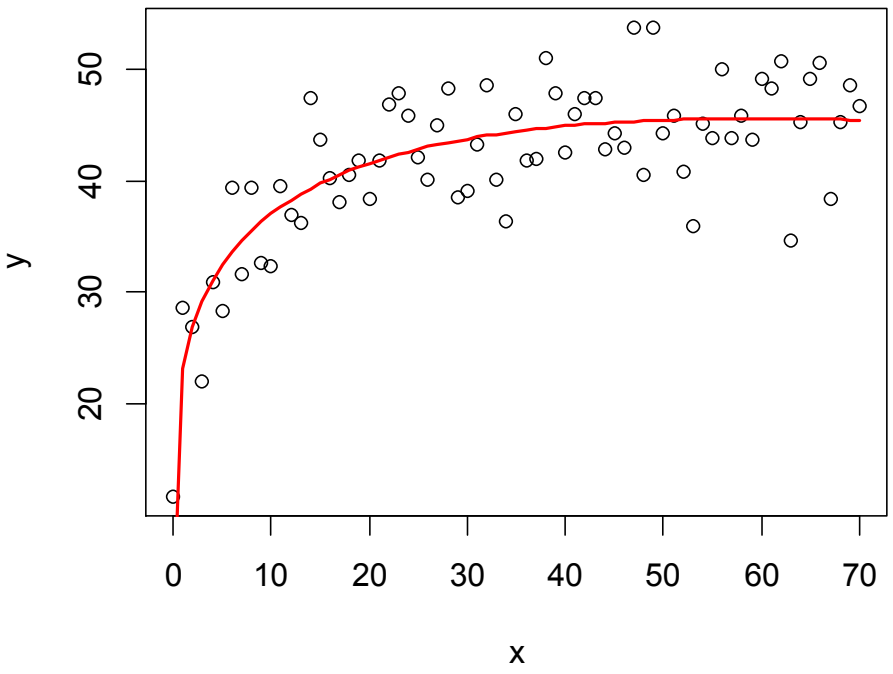

```
> data1=data.frame(x,y)
> nlsplot(data1, model=8, start=c(600,2,0.05))
>
```



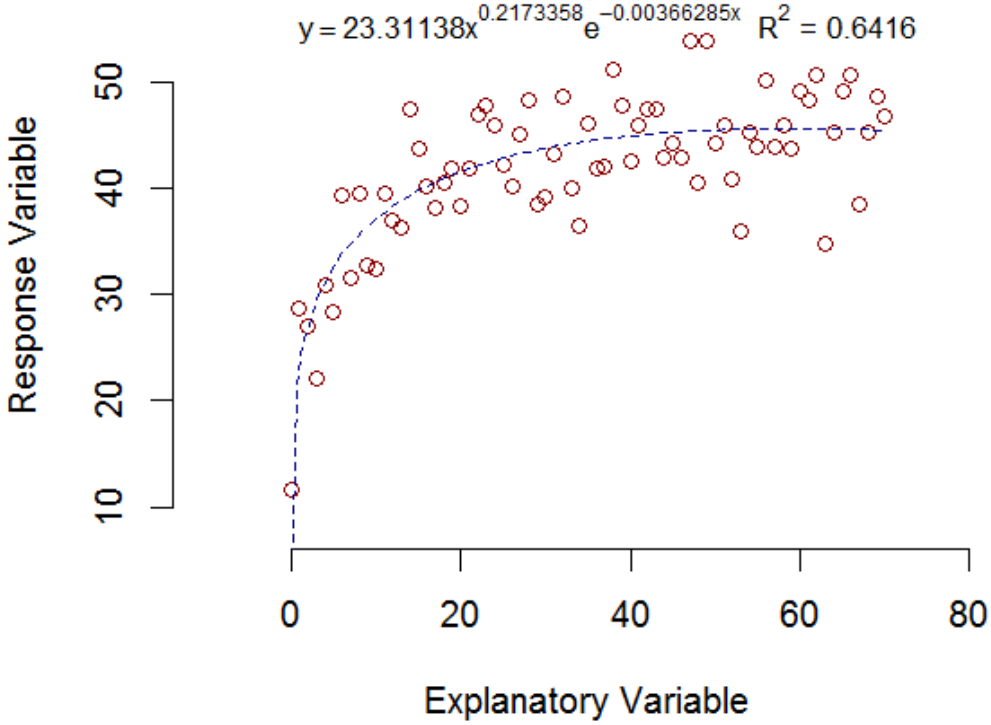
16.5.8 Lactation Curve

Lactation curve equation is y~(a*x^b)*exp(-c*x)

```
> x <- 0:70
> y <- ((16*x^0.25)*exp(-0.004*x)) + rnorm(70, 10, 4)

Warning message:
In ((16 * x^0.25) * exp(-0.004 * x)) + rnorm(70, 10, 4) :
  longer object length is not a multiple of shorter object length

> nm <- nls(y ~ ((a*x^b)*exp(-c*x)), start=list(a=16,b=0.25,c=0.004))
> nm

Nonlinear regression model
  model: y ~ ((a * x^b) * exp(-c * x))
   data: parent.frame()
        a         b         c
23.311383  0.217336  0.003663
 residual sum-of-squares: 1350

Number of iterations to convergence: 3
Achieved convergence tolerance: 7.274e-06

> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```



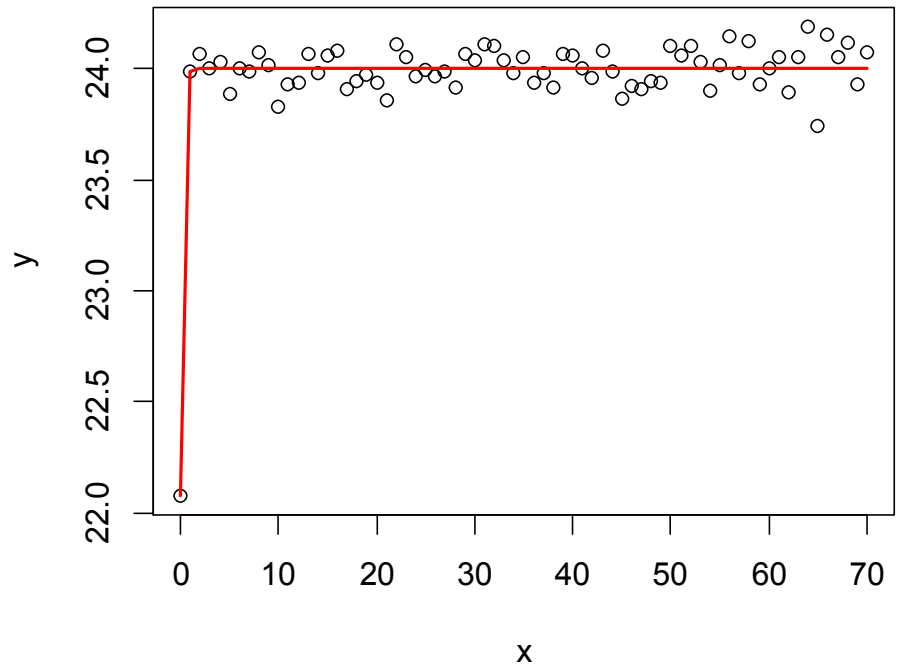

```
> data1=data.frame(x,y)
> nlsplot(data1, model=11, start=c(16,0.25,0.004))
>
```



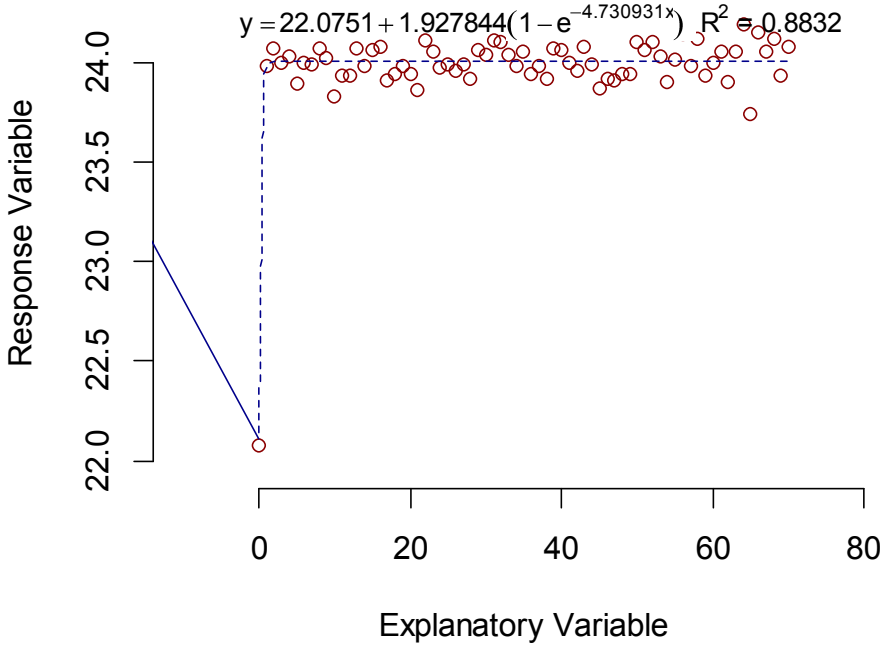
16.5.9 Ruminal Degradation Curve

Ruminal degradation curve equation is y ~ a + b * (1 - exp(-c * x))

```
> x <- 0:70
> y <- 20+2*(1-exp(-4.4*x) + rnorm(70, 1, 0.04))

Warning message:
In 1 - exp(-4.4 * x) + rnorm(70, 1, 0.04) :
  longer object length is not a multiple of shorter object length

> nm <- nls(y ~ a+b*(1-exp(-c*x)), start=list(a=14,b=3,c=2.4))
> nm

Nonlinear regression model
  model: y ~ a + b * (1 - exp(-c * x))
   data: parent.frame()
     a      b      c
22.075  1.928  4.731
 residual sum-of-squares: 0.4846

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.128e-06
> plot(y ~ x)
> lines(x, fitted(nm), lty = 1, col = "red", lwd = 2)
>
```




```
> data1=data.frame(x,y)
> nlsplot(data1, model=12)
>
```



REFERENCES

Anonymous, 2019. Chapter 8. Split-Plot Designs. https://stat.ethz.ch/~meierluk/teaching/anova/split-plot-designs.html (Access 24 January 2019)

Anonymous, 2019. Three-Stage Nested Design Example. http://www.math.montana.edu/jobo/st541/sec5b.pdf (Access 26 January 2019)

Anonymous. 2019. Simple Linear Regression in R. http://www.sthda.com/english/articles/40-regression-analysis/167-simple-linear-regression-in-r/ (Accessed 31 January 2019)

Anonymous. 2019. Linear Regression. http://r-statistics.co/Linear-Regression.html (Access 31 January 2019)

Anonymous. 2019. Performing Nonlinear Least Square and Nonlinear Regressions in R. https://dataconomy.com/2017/08/nonlinear-least-square-nonlinear-regression-r/ (Accessed 18 February 2019)

Anonymous. 2019. Regression Analysis (Nonlinear Regression Essentials in R: Polynomial and Spline Regression Models). http://www.sthda.com/english/articles/40-regression-analysis/162-nonlinear-regression-essentials-in-r-polynomial-and-spline-regression-models/ (Accessed 18 February 2019)

Gomez, K.A. and A.A. Gomez. 1984. Statistical Procedures for Agricultural Research. Second edition. An International Rice Research Institute Book, A Wiley-Interscience Publication, John Wiley & Sons. New York, Chichester, Brisbane, Toronto, Singapore.

Kaps, M. and Lamberson, W.R. 2014. Biostatistics for Animal Science. CABI Publishing, CAB International, Wallingford, Oxfordshire OX10 8DE, UK

Ken Li. 2017. Crossover Design. https://www.youtube.com/watch?v=pGZ1O8fdHAE

Lawson, J. 2015. Design and Analysis of Experiments with R. CRC Press, Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

Montgomery, D.C. 2001. Design and Analysis of Experiment. Fifth Edition. John Wiley & Sons. New York, Chichester, Weinheim, Brisbane, Toronto, Singapore.

Pandiangan, J. 2016. Penetapan Rumus Regresi terhaap Penentuan Bobot Karkas Berdasarkan Bobot Hidup Sapi Persilangan Simental di Rumah Pemotongan Hewan (RPH) Lubuk Buayako Padang. Skripsi. Fakultas Peternakan Universitas Andalas, Padang. http://scholar.unand.ac.id/19167/5/full%20skripsi.pdf (Accessed 1 February 2019).

Sehgal, D.K. 2019. Split Plot and Strip Plot Design. Indian Agricultural Statistics Research Institute Library Avenue, New Delhi-110 012. http://www.iasri.res.in/ebook/eb_smar/e-book_pdf%20files/Manual%20III/9-Split%20and%20Strip%20Plot%20Designs.pdf (Access 24 January 2019)

Shekar. 2018. Model Building on Marketing. https://rpubs.com/shekar07/399194

Tabachnick, B. G. and L.S. Fidell. 2007. Using multivariate statistics (5th ed.). Boston: Allyn and Bacon.

Vadali, S.G. 2017. Day 8: Data transformation - Skewness, normalization and much more in "Getting started with data science in 30 days using R programming": https://medium.com/@TheDataGyan/day-8-data-transformation-skewness-normalization-and-much-more-4c144d370e55 (accessed 11 November 2018).