Quantitative Methods with RStudio: Application for Management and Business Research

Deri Siswara Nafisa Berliana Indah Pratiwi

2025-07-05

Table of contents

Welcome

This is the code version of *Quantitative Methods with RStudio: Application for Management and Business Research*, a book released in 2024 by IPB Press. The book was written by Muhammad Firdaus, Farit M Afendi, Deri Siswara, and Nafisa Berliana Indah Pratiwi. You can order the full version here, which includes more detailed explanations.

Management quantitative analysis is widely utilized by students, lecturers, and researchers in Indonesia. This book aims to enhance the reputation of education and research in the country by presenting a variety of alternative analysis tools that are commonly used. Managers must accurately synthesize information during the decision-making process and prioritize various options precisely. Additionally, large volumes of transformed data such as customer identities and characteristics or consumer behavior survey results need to be synthesized properly.

The first chapter introduces RStudio software and the R programming language, while the second chapter focuses on nonparametric statistical analysis including correlation analysis of two nonparametric variables and causality relationships. Chapter three discusses logistic regression analysis for making practical decisions, followed by discriminant analysis in chapter four which models problems involving one dependent variable influenced by multiple independent variables.

Chapter five covers principal component analysis (PCA) and biplots to reduce a large selection of research variables into more compact dimensions. Chapter six delves into cluster analysis useful for mapping multiple entities whereas chapter seven comprehensively discusses factor analysis along with structural equation modeling (SEM), including PLS-SEM widely used for various problems involving latent variables such as prosperity, loyalty, and company performance.

The final chapter explores Analytic Hierarchy Process (AHP) aimed at determining priority choices based on hierarchical decision hierarchy using freely accessible RStudio software across all methods presented in this book. Updates will be made frequently.

This book may contain bugs/errors which readers can report at *derisiswarads@gmail.com*

Chapter 1

Basics of R

1.1 Introduction

```
A <- 2
```

```
A # Print A
```

```
[1] 2
```

```
A = 2
A
```

```
[1] 2
```

```
B <- "Halo Semua"
B
```

```
[1] "Halo Semua"
```

```
a<-10 # Space is not sensitive but lettercase is sensitive.
A
```

```
[1] 2
```

```
a
```

```
[1] 10
```

```
# Arithmetic operation
x <- 5
y <- 3
x + y
```

```
[1] 8
```

```r
x - y
```

```
[1] 2
```

```r
x * y
```

```
[1] 15
```

```r
x / y
```

```
[1] 1.666667
```

```r
# Logic operation
a <- TRUE
b <- FALSE
a & b
```

```
[1] FALSE
```

```r
a | b
```

```
[1] TRUE
```

```r
!a
```

```
[1] FALSE
```

```r
x <- 5
y <- 3
x > y
```

```
[1] TRUE
```

```r
x < y
```

```
[1] FALSE
```

```r
x == y
```

```
[1] FALSE
```

```r
x >= y
```

```
[1] TRUE
```

```r
x <= y
```

```
[1] FALSE
```

1.2 Types of Objects in R

1.2.1 Vector

```
a1 <- c(2,4,7,3) # Numeric vector
a2 <- c("one","two","three") # Character vector
a3 <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) # Logical vector
```

```
a1
```

```
[1] 2 4 7 3
```

```
a3[4]
```

```
[1] FALSE
```

```
a2[c(1,3)]
```

```
[1] "one"   "three"
```

```
a1[-1]
```

```
[1] 4 7 3
```

```
a1[2:4]
```

```
[1] 4 7 3
```

```
a <- c(1, 2, 3)
b <- c(4, 5, 6)
c <- c(a, b)
c
```

```
[1] 1 2 3 4 5 6
```

```
c[1:3]
```

```
[1] 1 2 3
```

```
d <- a + b
d
```

```
[1] 5 7 9
```

```
a4 <- 1:12
b1 <- matrix(a4,3,4)
b2 <- matrix(a4,3,4,byrow=TRUE)
b3 <- matrix(1:14,4,4)
```

```
b1
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
```

```
[3,]    3    6    9   12
```

```r
b2
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

```r
b3
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11    1
[4,]    4    8   12    2
```

```r
b2[2,3]
```

```
[1] 7
```

```r
b2[1:2,]
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
```

```r
b2[c(1,3),-2]
```

```
     [,1] [,2] [,3]
[1,]    1    3    4
[2,]    9   11   12
```

```r
dim(b2)
```

```
[1] 3 4
```

```r
m1 <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
m2 <- matrix(c(7, 8, 9, 10, 11, 12), nrow = 2, ncol = 3)
```

```r
m3 <- m1 + m2
m3
```

```
     [,1] [,2] [,3]
[1,]    8   12   16
[2,]   10   14   18
```

```r
m4 <- m1 %*% t(m2)
m4
```

```
     [,1] [,2]
[1,]   89   98
[2,]  116  128
```

1.2.2 Factor

```
a5 <- c("A","B","AB","O")
d1 <- factor(a5)
levels(d1)
```

```
[1] "A"  "AB" "B"  "O"
```

```
levels(d1) <- c("Darah A","Darah AB","Darah B","Darah O")
d1
```

```
[1] Darah A  Darah B  Darah AB Darah O
Levels: Darah A Darah AB Darah B Darah O
```

```
a6 <- c("SMA","SD","SMP","SMA","SMA","SMA","SMA","SMA","SMA","SMA","SMA","SMA","SMA")
d5 <- factor(a6, levels=c("SD","SMP","SMA")) # Skala pengukuran ordinal
levels(d5)
```

```
[1] "SD"  "SMP" "SMA"
```

```
d5
```

```
 [1] SMA SD  SMP SMA SMA SMA SMA SMA SMA SMA SMA SMA SMA
Levels: SD SMP SMA
```

1.2.3 List

```
a1; b2; d1
```

```
[1] 2 4 7 3
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

```
[1] Darah A  Darah B  Darah AB Darah O
Levels: Darah A Darah AB Darah B Darah O
```

```
e1 <- list(a1,b2,d1)
e2 <- list(vect=a1,mat=b2,fac=d1)
e1
```

```
[[1]]
[1] 2 4 7 3

[[2]]
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

```
[[3]]
[1] Darah A  Darah B  Darah AB Darah O
Levels: Darah A Darah AB Darah B Darah O
```

```r
e2
```

```
$vect
[1] 2 4 7 3

$mat
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12

$fac
[1] Darah A  Darah B  Darah AB Darah O
Levels: Darah A Darah AB Darah B Darah O
```

```r
e1[[1]][2]
```

```
[1] 4
```

```r
e2$fac
```

```
[1] Darah A  Darah B  Darah AB Darah O
Levels: Darah A Darah AB Darah B Darah O
```

```r
e2[2]
```

```
$mat
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

```r
names(e2)
```

```
[1] "vect" "mat"  "fac"
```

1.2.4 Data Frame

```r
Angka <- 11:15
Huruf <- factor(LETTERS[6:10])
f1 <- data.frame(Angka,Huruf)
f1
```

```
  Angka Huruf
1    11     F
```

```
2    12     G
3    13     H
4    14     I
5    15     J
```

```
f1[1,2]
```

```
[1] F
Levels: F G H I J
```

```
f1$Angka
```

```
[1] 11 12 13 14 15
```

```
f1[,"Huruf"]
```

```
[1] F G H I J
Levels: F G H I J
```

```
colnames(f1)
```

```
[1] "Angka" "Huruf"
```

```
str(f1)
```

```
'data.frame':   5 obs. of  2 variables:
 $ Angka: int  11 12 13 14 15
 $ Huruf: Factor w/ 5 levels "F","G","H","I",..: 1 2 3 4 5
```

1.3 Data Frame Management

```
data(iris)
```

```
head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```
tail(iris)
```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
```

```
149          6.2          3.4          5.4          2.3 virginica
150          5.9          3.0          5.1          1.8 virginica
```

```
str(iris)
```

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

1.3.1 R Package

```
# install.packages("readxl") - code to install R package
library(readxl)
```

```
#install.packages("dplyr")
library(dplyr)
```

1.3.2 Data Management With dplyr

```
head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```
irisbaru  <- mutate(iris, sepal2 = Sepal.Length + Sepal.Width)
```

```
head(irisbaru)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species sepal2
1          5.1         3.5          1.4         0.2  setosa    8.6
2          4.9         3.0          1.4         0.2  setosa    7.9
3          4.7         3.2          1.3         0.2  setosa    7.9
4          4.6         3.1          1.5         0.2  setosa    7.7
5          5.0         3.6          1.4         0.2  setosa    8.6
6          5.4         3.9          1.7         0.4  setosa    9.3
```

```
irisetosa <- filter(iris, Species=="setosa")
head(irisetosa)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
```

```
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```r
levels(iris$Species)
```

```
[1] "setosa"     "versicolor" "virginica"
```

```r
irisversicolor <- filter(iris, Species=="setosa"& Petal.Length==1.3)
head(irisversicolor)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          4.7         3.2          1.3         0.2  setosa
2          5.4         3.9          1.3         0.4  setosa
3          5.5         3.5          1.3         0.2  setosa
4          4.4         3.0          1.3         0.2  setosa
5          5.0         3.5          1.3         0.3  setosa
6          4.5         2.3          1.3         0.3  setosa
```

```r
iris3 <- select(iris, Sepal.Length, Species)
head(iris3)
```

```
  Sepal.Length Species
1          5.1  setosa
2          4.9  setosa
3          4.7  setosa
4          4.6  setosa
5          5.0  setosa
6          5.4  setosa
```

```r
iris4 <- arrange(iris, Petal.Width)
head(iris4)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          4.9         3.1          1.5         0.1  setosa
2          4.8         3.0          1.4         0.1  setosa
3          4.3         3.0          1.1         0.1  setosa
4          5.2         4.1          1.5         0.1  setosa
5          4.9         3.6          1.4         0.1  setosa
6          5.1         3.5          1.4         0.2  setosa
```

```r
iris4 <- arrange(iris, Species, desc(Petal.Width))
head(iris4)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.0         3.5          1.6         0.6  setosa
2          5.1         3.3          1.7         0.5  setosa
```

```
3           5.4         3.9          1.7         0.4  setosa
4           5.7         4.4          1.5         0.4  setosa
5           5.4         3.9          1.3         0.4  setosa
6           5.1         3.7          1.5         0.4  setosa
```

```
names(iris4)[1] <- "length"
head(iris4)
```

```
  length Sepal.Width Petal.Length Petal.Width Species
1    5.0         3.5          1.6         0.6  setosa
2    5.1         3.3          1.7         0.5  setosa
3    5.4         3.9          1.7         0.4  setosa
4    5.7         4.4          1.5         0.4  setosa
5    5.4         3.9          1.3         0.4  setosa
6    5.1         3.7          1.5         0.4  setosa
```

```
head(iris4[,c(-1,-3)])
```

```
  Sepal.Width Petal.Width Species
1         3.5         0.6  setosa
2         3.3         0.5  setosa
3         3.9         0.4  setosa
4         4.4         0.4  setosa
5         3.9         0.4  setosa
6         3.7         0.4  setosa
```

```
iris %>% group_by(Species) %>% summarise(rata2_Sepal.Width = mean(Sepal.Width))
```

```
# A tibble: 3 x 2
  Species    rata2_Sepal.Width
  <fct>                  <dbl>
1 setosa                  3.43
2 versicolor              2.77
3 virginica               2.97
```

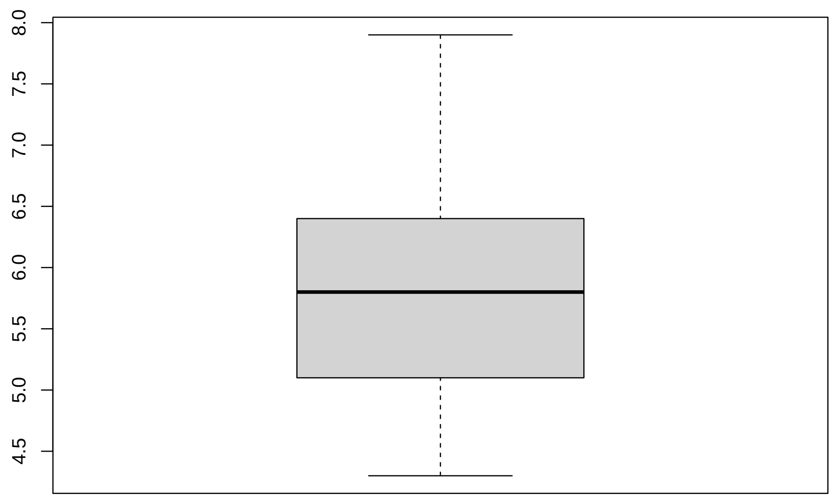
1.4 Visualization

1.4.1 Histogram

```
hist(iris$Sepal.Length)
```


```
boxplot(iris$Sepal.Length)
```

1.4.2 Box Plot

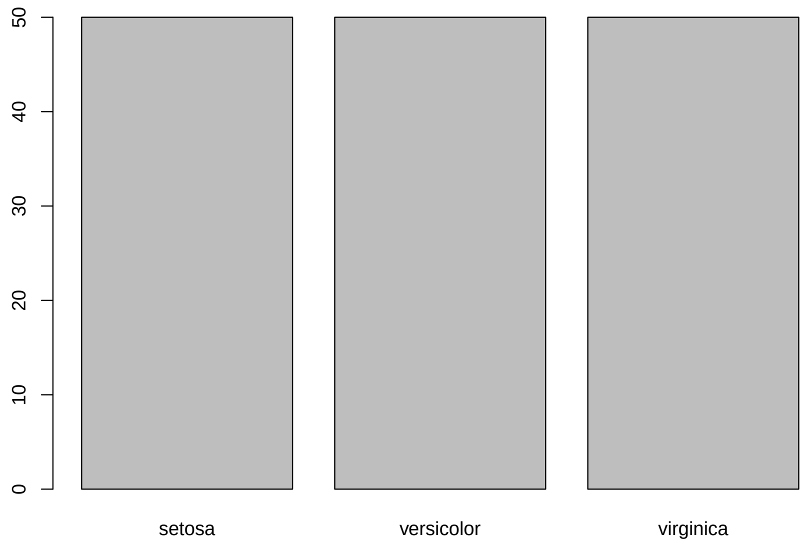



```
table(iris$Species)
```

1.4.3 Barplot

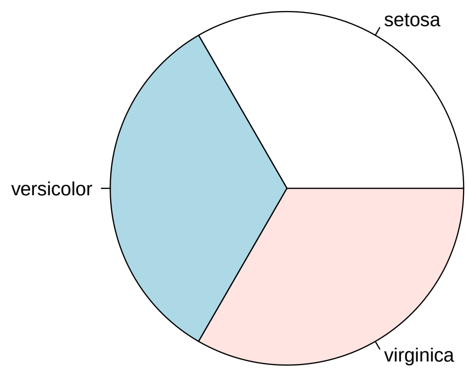
```
    setosa versicolor  virginica 
        50         50         50 
```

```
barplot(table(iris$Species))
```



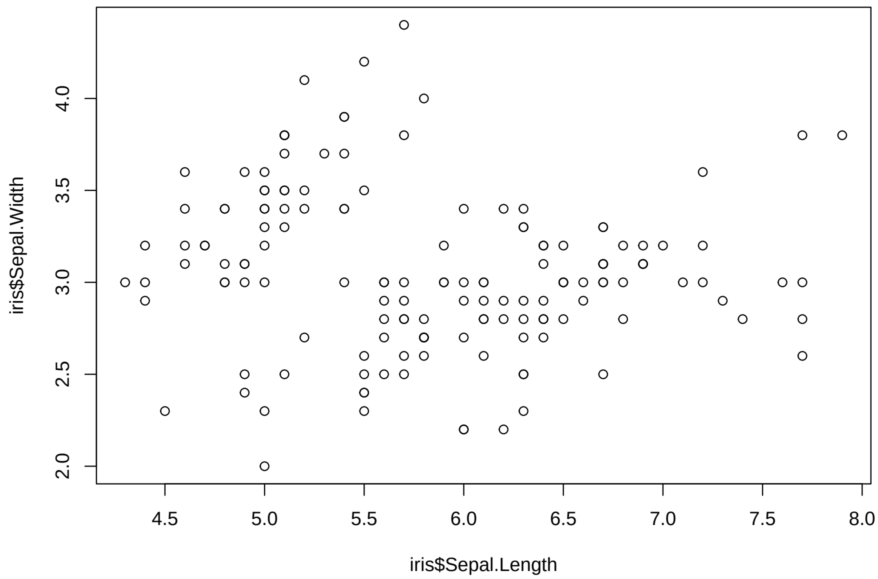

```
pie(table(iris$Species))
```

1.4.4 Pie Chart

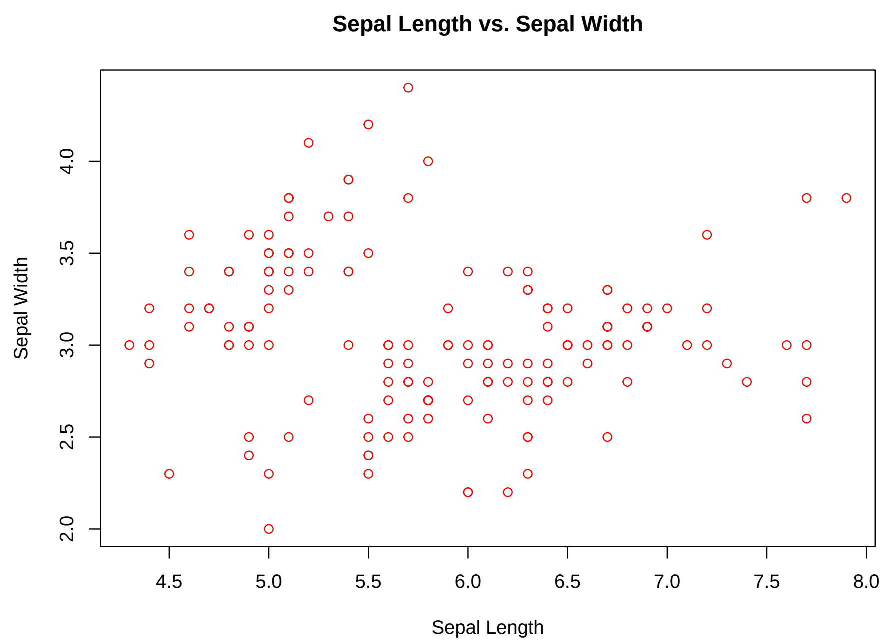



```
plot(iris$Sepal.Length,iris$Sepal.Width)
```

1.4.5 Scatter Plot




```
plot(iris$Sepal.Length, iris$Sepal.Width, main = "Sepal Length vs. Sepal Width",
     xlab = "Sepal Length", ylab = "Sepal Width", col = "red")
```


Sepal Length vs. Sepal Width
Sepal Width
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Sepal Length

Chapter 2

Nonparametric Statistics

2.1 Correlation

```r
# Membuat data contoh
# Membuat vektor untuk responden, X, dan Y
X <- c(2, 1, 6, 11, 7, 11, 1, 12, 13, 13, 11)
Y <- c(9, 8, 16, 13, 11, 12, 7, 7, 13, 17, 10)

# Membuat dataframe
dataku <- data.frame(X = X, Y = Y)

# Menampilkan data
dataku
```

```
    X  Y
1   2  9
2   1  8
3   6 16
4  11 13
5   7 11
6  11 12
7   1  7
8  12  7
9  13 13
10 13 17
11 11 10
```

```r
# Menggunakan fungsi cor.test untuk menghitung Tau-Kendall
cor.test(dataku$X, dataku$Y, method = "kendall")
```

```
    Kendall's rank correlation tau

data:  dataku$X and dataku$Y
z = 1.7529, p-value = 0.07962
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau 
0.4273658 
```

```
# Menggunakan fungsi cor.test untuk menghitung korelasi Spearman
cor.test(dataku$X, dataku$Y, method = "spearman")
```

```
    Spearman's rank correlation rho

data:  dataku$X and dataku$Y
S = 108.98, p-value = 0.1134
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho 
0.5046513 
```

2.1.1 Chi-Square Test

```
# Membuat data contoh
# Data asli dalam bentuk tabel silang
frekuensi <- matrix(c(30, 21, 30, 19, 15, 35), nrow = 2)
rownames(frekuensi) <- c("Kontrak", "Tetap")
colnames(frekuensi) <- c("Rendah", "Sedang", "Tinggi")

# Inisiasi vektor kosong untuk menyimpan data
Status_Pegawai <- c()
Tingkat_Produktivitas <- c()

# Mengulang setiap kombinasi sesuai dengan frekuensinya
for (i in 1:nrow(frekuensi)) {
  for (j in 1:ncol(frekuensi)) {
    Status_Pegawai <- c(Status_Pegawai,
                        rep(rownames(frekuensi)[i],
                            frekuensi[i, j]))

    Tingkat_Produktivitas <- c(Tingkat_Produktivitas,
                               rep(colnames(frekuensi)[j],
                                   frekuensi[i, j]))
  }
}
```

```r
# Membuat dataframe
dataku2 <- data.frame(Status_Pegawai,
                      Tingkat_Produktivitas)
# Menampilkan data
head(dataku2)
```

```
  Status_Pegawai Tingkat_Produktivitas
1        Kontrak                Rendah
2        Kontrak                Rendah
3        Kontrak                Rendah
4        Kontrak                Rendah
5        Kontrak                Rendah
6        Kontrak                Rendah
```

```r
# Transformasi menejadi factor
dataku2$Status_Pegawai <- as.factor(dataku2$Status_Pegawai)
dataku2$Tingkat_Produktivitas <- as.factor(dataku2$Tingkat_Produktivitas)

summary(dataku2)
```

```
 Status_Pegawai Tingkat_Produktivitas
 Kontrak:75     Rendah:51
 Tetap  :75     Sedang:49
                Tinggi:50
```

```r
# Melakukan tabel kontingensi
dataku2_kt <- table(dataku2$Status_Pegawai, dataku2$Tingkat_Produktivitas)
dataku2_kt
```

```
          Rendah Sedang Tinggi
  Kontrak     30     30     15
  Tetap       21     19     35
```

```r
# Melakukan uji Chi-Square
chisq.test(dataku2_kt)
```

```

    Pearson's Chi-squared test

data:  dataku2_kt
X-squared = 12.058, df = 2, p-value = 0.002408
```

2.2 Difference Test

2.2.1 Two sample test (Independent)

2.2.1.1 Mann-Whitney Test

```r
# Membuat data contoh
# Vektor data untuk efisiensi pada skala besar dan kecil
efisiensi_besar <- c(1.31, 1.25, 1.32, 1.3, 1.33, 1.31, 1.35, 1.34, 0.28, 1.34, 1.28)
efisiensi_kecil <- c(1.21, 1.28, 1.32, 1.25, 1.27, 1.31, 1.26, 1.31, 1.24, 1.22)
```

```r
wilcox.test(efisiensi_besar, efisiensi_kecil)
```

```
	Wilcoxon rank sum test with continuity correction

data:  efisiensi_besar and efisiensi_kecil
W = 82.5, p-value = 0.05614
alternative hypothesis: true location shift is not equal to 0
```

2.2.1.2 Chi-Square Test

2.2.2 More than two sample test (Independent)

2.2.2.1 Kruskal-Wallis Test

```r
# Membuat data contoh
# Membuat vektor untuk Industri A, B, dan C
industri_A <- c(2.33, 2.79, 3.01, 2.33, 1.22, 2.79, 1.9, 1.65)
industri_B <- c(2.33, 2.33, 2.79, 3.01, 1.99, 2.45)
industri_C <- c(1.06, 1.37, 1.09, 1.65, 1.44, 1.11)

# Membuat vektor industri
industri <- c(rep("Industri A", length(industri_A)),
              rep("Industri B", length(industri_B)),
              rep("Industri C", length(industri_C)))

# Menggabungkan semua vektor value
nilai <- c(industri_A, industri_B, industri_C)

# Membuat data frame
dataku4 <- data.frame(industri, nilai)

# Menampilkan data frame
dataku4$industri <- as.factor(dataku4$industri)
dataku4
```

```
     industri nilai
```

```
1  Industri A  2.33
2  Industri A  2.79
3  Industri A  3.01
4  Industri A  2.33
5  Industri A  1.22
6  Industri A  2.79
7  Industri A  1.90
8  Industri A  1.65
9  Industri B  2.33
10 Industri B  2.33
11 Industri B  2.79
12 Industri B  3.01
13 Industri B  1.99
14 Industri B  2.45
15 Industri C  1.06
16 Industri C  1.37
17 Industri C  1.09
18 Industri C  1.65
19 Industri C  1.44
20 Industri C  1.11
```

```
# Uji kruskal wallis
kruskal.test(nilai ~ industri, data = dataku4)
```

```

	Kruskal-Wallis rank sum test

data:  nilai by industri
Kruskal-Wallis chi-squared = 10.619, df = 2, p-value = 0.004943
```

```
# Post hoc kruskal-wallis - Uji Dun
#installed.packages("FSA")
library(FSA)
dunnTest(nilai ~ industri, data = dataku4)
```

```
               Comparison          Z      P.unadj      P.adj
1 Industri A - Industri B -0.6428883 0.520296550 0.52029655
2 Industri A - Industri C  2.6109139 0.009030062 0.01806012
3 Industri B - Industri C  3.0436533 0.002337243 0.00701173
```

2.2.2.2 Chi-Square Test

2.2.3 Two sample test (Dependent)

2.2.3.1 Sign Test

```
# Membuat data contoh
# Data Skor Kepuasan
```

```
produk_lama <- c(16, 15, 18, 16, 17, 18, 20, 15, 14, 16, 19, 17)
produk_baru <- c(18, 17, 16, 19, 17, 20, 18, 16, 15, 18, 20, 18)
# Data Responden
responden <- c(1:12)
# Membuat data frame
dataku5 <- data.frame(Responden = c(rep(responden, 2)),
                      Produk = factor(c(rep("Produk Lama", length(produk_lama)),
                                        rep("Produk Baru", length(produk_baru)))),
                      Skor_Kepuasan = c(produk_lama, produk_baru))
# Menampilkan data frame
dataku5
```

```
   Responden      Produk Skor_Kepuasan
1          1 Produk Lama            16
2          2 Produk Lama            15
3          3 Produk Lama            18
4          4 Produk Lama            16
5          5 Produk Lama            17
6          6 Produk Lama            18
7          7 Produk Lama            20
8          8 Produk Lama            15
9          9 Produk Lama            14
10        10 Produk Lama            16
11        11 Produk Lama            19
12        12 Produk Lama            17
13         1 Produk Baru            18
14         2 Produk Baru            17
15         3 Produk Baru            16
16         4 Produk Baru            19
17         5 Produk Baru            17
18         6 Produk Baru            20
19         7 Produk Baru            18
20         8 Produk Baru            16
21         9 Produk Baru            15
22        10 Produk Baru            18
23        11 Produk Baru            20
24        12 Produk Baru            18
```

```
# Menghitung perbedaan
diff <- dataku5[dataku5$Produk == 'Produk Baru', ]$Skor_Kepuasan - dataku5[dataku5$Prod

# Menghitung jumlah perbedaan yang positif
jumlah_positif <- sum(diff > 0)
```

```
# Melakukan uji tanda
binom.test(jumlah_positif, length(diff),
```

```
          p = 0.5,
          alternative = "two.sided")
```

```

	Exact binomial test

data:  jumlah_positif and length(diff)
number of successes = 9, number of trials = 12, p-value = 0.146
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.4281415 0.9451394
sample estimates:
probability of success 
                  0.75 
```

Interpretation: https://www.geeksforgeeks.org/sign-test-in-r/

2.2.4 More than two sample test (Dependent)

2.2.4.1 Friedman Test

```
# Membuat data contoh
dataku6 <- matrix(c(1.24,1.50,1.62,
              1.71,1.85,2.05,
              1.37,2.12,1.68,
              2.53,1.87,2.62,
              1.23,1.34,1.51,
              1.94,2.33,2.86,
              1.72,1.43,2.86), nrow = 7, byrow = TRUE,
              dimnames = list(Person= as.character(1:7),
              Obat = c("Obat A","Obat B","Obat C")))
dataku6
```

```
      Obat
Person Obat A Obat B Obat C
     1   1.24   1.50   1.62
     2   1.71   1.85   2.05
     3   1.37   2.12   1.68
     4   2.53   1.87   2.62
     5   1.23   1.34   1.51
     6   1.94   2.33   2.86
     7   1.72   1.43   2.86
```

```
friedman.test(dataku6)
```

```

	Friedman rank sum test
```

```
data:  dataku6
Friedman chi-squared = 8.8571, df = 2, p-value = 0.01193
```

2.2.4.2 Cochran Test

```r
# Membuat data contoh
## Input data
responden <- c(1:8)
produk_A <- c("Tidak","Tidak","Ya","Ya","Ya","Tidak","Tidak","Tidak")
produk_B <- c("Tidak","Ya","Ya","Ya","Tidak","Tidak","Ya","Tidak")
produk_C <- c("Ya","Tidak","Tidak","Ya","Tidak","Ya","Ya","Tidak")
dataku7 <- data.frame(responden, produk_A, produk_B, produk_C)
dataku7$produk_A <- as.factor(dataku7$produk_A)
dataku7$produk_B <- as.factor(dataku7$produk_B)
dataku7$produk_C <- as.factor(dataku7$produk_C)
dataku7
```

```
  responden produk_A produk_B produk_C
1         1    Tidak    Tidak       Ya
2         2    Tidak       Ya    Tidak
3         3       Ya       Ya    Tidak
4         4       Ya       Ya       Ya
5         5       Ya    Tidak    Tidak
6         6    Tidak    Tidak       Ya
7         7    Tidak       Ya       Ya
8         8    Tidak    Tidak    Tidak
```

```r
dataku7 <-ifelse(dataku7=="Ya", 1,0)
```

```r
#install.packages("nonpar")
library(nonpar)
cochrans.q(as.matrix(dataku7[,-1]), alpha = 0.05)
```

```
 Cochran's Q Test

 H0: There is no difference in the effectiveness of treatments.
 HA: There is a difference in the effectiveness of treatments.

 Q = 0.333333333333333

 Degrees of Freedom = 2

 Significance Level = 0.05
 The p-value is  0.846481724890614
```

Chapter 3

Logistic Regression

3.1 Regresi Logistik Biner

3.1.1 Data

```
credit <- read.csv("Data/credit.csv")
head(credit[,1:5],10)
```

```
   creditability account.balance duration credit.amount saving.balance
1              1               1       18          1049              1
2              1               1        9          2799              1
3              1               2       12           841              2
4              1               1       12          2122              1
5              1               1       12          2171              1
6              1               1       10          2241              1
7              1               1        8          3398              1
8              1               1        6          1361              1
9              1               4       18          1098              1
10             1               2       24          3758              3
```

```
str(credit)
```

```
'data.frame':   1000 obs. of  14 variables:
 $ creditability   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ account.balance : int  1 1 2 1 1 1 1 1 4 2 ...
 $ duration        : int  18 9 12 12 12 10 8 6 18 24 ...
 $ credit.amount   : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
 $ saving.balance  : int  1 1 2 1 1 1 1 1 1 3 ...
 $ employment.year : int  2 3 4 3 3 2 4 2 1 1 ...
 $ installment.rate: int  4 2 2 3 4 1 1 2 4 1 ...
 $ marital.status  : int  2 3 2 3 3 3 3 3 2 2 ...
```

```
 $ duration.address: int  4 2 4 2 4 3 4 4 4 4 ...
 $ age             : int  21 36 23 39 38 48 39 40 65 23 ...
 $ dependents      : int  1 2 1 2 1 2 1 2 1 1 ...
 $ number.of.credit: int  1 2 1 2 2 2 2 1 2 1 ...
 $ occupation      : int  3 3 2 2 2 2 2 2 1 1 ...
 $ previous.credit : int  4 4 2 4 4 4 4 4 4 2 ...
```

```
library(dplyr)
credit <- credit %>% mutate(across(-c(duration,
                              credit.amount,
                              age),as.factor))
str(credit)
```

```
'data.frame':   1000 obs. of  14 variables:
 $ creditability   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ account.balance : Factor w/ 4 levels "1","2","3","4": 1 1 2 1 1 1 1 1 4 2 ...
 $ duration        : int  18 9 12 12 12 10 8 6 18 24 ...
 $ credit.amount   : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
 $ saving.balance  : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 1 3 ...
 $ employment.year : Factor w/ 5 levels "1","2","3","4",..: 2 3 4 3 3 2 4 2 1 1 ...
 $ installment.rate: Factor w/ 4 levels "1","2","3","4": 4 2 2 3 4 1 1 2 4 1 ...
 $ marital.status  : Factor w/ 4 levels "1","2","3","4": 2 3 2 3 3 3 3 3 2 2 ...
 $ duration.address: Factor w/ 4 levels "1","2","3","4": 4 2 4 2 4 3 4 4 4 4 ...
 $ age             : int  21 36 23 39 38 48 39 40 65 23 ...
 $ dependents      : Factor w/ 2 levels "1","2": 1 2 1 2 1 2 1 2 1 1 ...
 $ number.of.credit: Factor w/ 4 levels "1","2","3","4": 1 2 1 2 2 2 2 1 2 1 ...
 $ occupation      : Factor w/ 4 levels "1","2","3","4": 3 3 2 2 2 2 2 2 1 1 ...
 $ previous.credit : Factor w/ 5 levels "0","1","2","3",..: 5 5 3 5 5 5 5 5 5 3 ...
```

3.1.2 Pemodelan

```
logreg1 <- glm(creditability~.,data=credit,family = "binomial")
summary(logreg1)
```

```
Call:
glm(formula = creditability ~ ., family = "binomial", data = credit)

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        2.990e-01  8.942e-01   0.334 0.738097
account.balance2   4.346e-01  2.013e-01   2.159 0.030852 *
account.balance3   9.490e-01  3.602e-01   2.635 0.008421 **
account.balance4   1.804e+00  2.222e-01   8.119 4.69e-16 ***
duration          -2.705e-02  8.818e-03  -3.068 0.002156 **
credit.amount     -1.025e-04  4.161e-05  -2.465 0.013718 *
```

```
saving.balance2    1.293e-01  2.701e-01   0.479 0.632222
saving.balance3    4.144e-01  3.987e-01   1.039 0.298644
saving.balance4    1.241e+00  5.032e-01   2.467 0.013629 *
saving.balance5    8.811e-01  2.463e-01   3.577 0.000347 ***
employment.year2  -1.432e-01  4.109e-01  -0.348 0.727561
employment.year3   2.530e-01  3.957e-01   0.639 0.522582
employment.year4   7.646e-01  4.258e-01   1.796 0.072572 .
employment.year5   2.386e-01  3.962e-01   0.602 0.547012
installment.rate2 -2.841e-01  2.953e-01  -0.962 0.336089
installment.rate3 -5.122e-01  3.217e-01  -1.592 0.111374
installment.rate4 -9.279e-01  2.872e-01  -3.230 0.001236 **
marital.status2    1.744e-01  3.742e-01   0.466 0.641255
marital.status3    7.482e-01  3.670e-01   2.039 0.041468 *
marital.status4    5.577e-01  4.371e-01   1.276 0.201928
duration.address2 -7.104e-01  2.832e-01  -2.509 0.012122 *
duration.address3 -5.443e-01  3.163e-01  -1.721 0.085314 .
duration.address4 -4.386e-01  2.762e-01  -1.588 0.112244
age                1.125e-02  8.468e-03   1.329 0.183990
dependents2       -2.607e-01  2.387e-01  -1.092 0.274669
number.of.credit2 -4.177e-01  2.315e-01  -1.805 0.071133 .
number.of.credit3 -4.131e-01  5.951e-01  -0.694 0.487625
number.of.credit4 -4.589e-01  9.908e-01  -0.463 0.643240
occupation2       -8.953e-02  6.276e-01  -0.143 0.886557
occupation3       -1.487e-01  6.048e-01  -0.246 0.805804
occupation4        1.276e-02  6.087e-01   0.021 0.983277
previous.credit1  -3.136e-01  5.178e-01  -0.606 0.544686
previous.credit2   6.063e-01  4.149e-01   1.461 0.143896
previous.credit3   8.090e-01  4.531e-01   1.785 0.074205 .
previous.credit4   1.511e+00  4.169e-01   3.625 0.000288 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance:  956.0  on 965  degrees of freedom
AIC: 1026

Number of Fisher Scoring iterations: 5
```

3.1.3 Odds Ratio

```r
beta = round(coef(logreg1),2)
OR = round(exp(beta),2)
cbind(beta, OR)
```

```
                    beta   OR
(Intercept)         0.30 1.35
account.balance2    0.43 1.54
account.balance3    0.95 2.59
account.balance4    1.80 6.05
duration           -0.03 0.97
credit.amount       0.00 1.00
saving.balance2     0.13 1.14
saving.balance3     0.41 1.51
saving.balance4     1.24 3.46
saving.balance5     0.88 2.41
employment.year2   -0.14 0.87
employment.year3    0.25 1.28
employment.year4    0.76 2.14
employment.year5    0.24 1.27
installment.rate2 -0.28 0.76
installment.rate3 -0.51 0.60
installment.rate4 -0.93 0.39
marital.status2     0.17 1.19
marital.status3     0.75 2.12
marital.status4     0.56 1.75
duration.address2 -0.71 0.49
duration.address3 -0.54 0.58
duration.address4 -0.44 0.64
age                 0.01 1.01
dependents2        -0.26 0.77
number.of.credit2 -0.42 0.66
number.of.credit3 -0.41 0.66
number.of.credit4 -0.46 0.63
occupation2        -0.09 0.91
occupation3        -0.15 0.86
occupation4         0.01 1.01
previous.credit1   -0.31 0.73
previous.credit2    0.61 1.84
previous.credit3    0.81 2.25
previous.credit4    1.51 4.53
```

3.1.4 Multikolineratitas

```
library(car)
vif(logreg1)
```

```
                    GVIF Df GVIF^(1/(2*Df))
account.balance  1.283532  3        1.042480
duration         1.828834  1        1.352344
```

```
credit.amount    2.284117  1        1.511330
saving.balance   1.286469  4        1.031989
employment.year  2.406179  4        1.116005
installment.rate 1.443706  3        1.063114
marital.status   1.439516  3        1.062599
duration.address 1.502426  3        1.070201
age              1.365556  1        1.168570
dependents       1.177252  1        1.085012
number.of.credit 2.060162  3        1.128020
occupation       1.893863  3        1.112307
previous.credit  2.136438  4        1.099541
```

3.1.5 Akurasi

```r
pred_clas <- ifelse(logreg1$fitted.values > 0.5, 1, 0)
conf_matrix <- table(credit$creditability, pred_clas)
conf_matrix
```

```
   pred_clas
      0   1
  0 145 155
  1  68 632
```

```r
paste0("Akurasi Model:")
```

```
[1] "Akurasi Model:"
```

```r
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy
```

```
[1] 0.777
```

3.1.6 Kebaikan Model

```r
#install.packages("performance")
library(performance)
#Outliers
performance::check_outliers(logreg1)
```

```
OK: No outliers detected.
- Based on the following method and threshold: cook (0.8).
- For variable: (Whole model)
```

```r
#Metrik
performance(logreg1)
```

```
# Indices of model performance
```

```
AIC      |     AICc |      BIC | Tjur's R2 |  RMSE | Sigma | Log_loss
---------------------------------------------------------------------
1025.995 | 1028.609 | 1197.767 |     0.254 | 0.395 | 1.000 |    0.478

AIC      | Score_log | Score_spherical |   PCP
----------------------------------------------
1025.995 |      -Inf |           0.001 | 0.687
```

```
#Goodness Of Fit
performance_hosmer(logreg1)
```

```
# Hosmer-Lemeshow Goodness-of-Fit Test

  Chi-squared: 8.472
           df: 8
      p-value: 0.389
```

3.2 Regresi Logistik Nominal atau Multinominal

3.2.1 Data

```
library(readxl)
students <- read_excel("Data/students.xlsx")
head(students,10)
```

```
# A tibble: 10 x 6
   gender ses    prog      read write  math
   <chr>  <chr>  <chr>    <dbl> <dbl> <dbl>
 1 female low    vocation    34    35    41
 2 male   middle general     34    33    41
 3 male   high   vocation    39    39    44
 4 male   low    vocation    37    37    42
 5 male   middle vocation    39    31    40
 6 female high   general     42    36    42
 7 male   middle vocation    31    36    46
 8 male   middle vocation    50    31    40
 9 female middle vocation    39    41    33
10 male   middle vocation    34    37    46
```

```
str(students)
```

```
tibble [200 x 6] (S3: tbl_df/tbl/data.frame)
 $ gender: chr [1:200] "female" "male" "male" "male" ...
 $ ses   : chr [1:200] "low" "middle" "high" "low" ...
 $ prog  : chr [1:200] "vocation" "general" "vocation" "vocation" ...
```

```
 $ read  : num [1:200] 34 34 39 37 39 42 31 50 39 34 ...
 $ write : num [1:200] 35 33 39 37 31 36 36 31 41 37 ...
 $ math  : num [1:200] 41 41 44 42 40 42 46 40 33 46 ...
```

3.2.2 Ubah jadi faktor

```
library(dplyr)
students <- students %>% mutate(across(-c(read,write,math),as.factor))
students$prog2 <- relevel(students$prog, ref = "academic")
str(students)
```

```
tibble [200 x 7] (S3: tbl_df/tbl/data.frame)
 $ gender: Factor w/ 2 levels "female","male": 1 2 2 2 2 1 2 2 1 2 ...
 $ ses   : Factor w/ 3 levels "high","low","middle": 2 3 1 2 3 1 3 3 3 3 ...
 $ prog  : Factor w/ 3 levels "academic","general",..: 3 2 3 3 3 2 3 3 3 3 ...
 $ read  : num [1:200] 34 34 39 37 39 42 31 50 39 34 ...
 $ write : num [1:200] 35 33 39 37 31 36 36 31 41 37 ...
 $ math  : num [1:200] 41 41 44 42 40 42 46 40 33 46 ...
 $ prog2 : Factor w/ 3 levels "academic","general",..: 3 2 3 3 3 2 3 3 3 3 ...
```

```
table(students$ses, students$prog)
```

```

         academic general vocation
  high         42       9        7
  low          19      16       12
  middle       44      20       31
```

```
table(students$gender, students$prog)
```

```

         academic general vocation
  female       58      24       27
  male         47      21       23
```

3.2.3 Pemodelan

```
#install.packages("nnet")
library(nnet)
logmultinom <- multinom(prog2 ~ ses + gender + write + read, data = students)
```

```
# weights:  21 (12 variable)
initial  value 219.722458
iter  10 value 176.754587
final  value 174.725397
converged
```

```
summary(logmultinom)
```

```
Call:
multinom(formula = prog2 ~ ses + gender + write + read, data = students)

Coefficients:
         (Intercept)    seslow sesmiddle gendermale       write        read
general     2.621831 1.0038426 0.5651588  0.1273914 -0.02860308 -0.04730781
vocation    6.505182 0.6239396 1.1539447 -0.3105237 -0.08243508 -0.07108839

Std. Errors:
         (Intercept)    seslow sesmiddle gendermale      write       read
general     1.434514 0.5323398 0.4713812  0.4137756 0.02686316 0.02480868
vocation    1.524572 0.6200276 0.5231819  0.4414783 0.02793343 0.02752520

Residual Deviance: 349.4508
AIC: 373.4508
```

```
z <- summary(logmultinom)$coefficients/summary(logmultinom)$standard.errors
# 2-tailed z test
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
          (Intercept)     seslow  sesmiddle gendermale       write        read
general  6.759775e-02 0.05933302 0.23055043  0.7581770 0.286980200 0.056532815
vocation 1.982164e-05 0.31426675 0.02741006  0.4818237 0.003166173 0.009804037
```

3.2.4 Odds Ratio

```
exp(coef(logmultinom))
```

```
         (Intercept)   seslow sesmiddle gendermale     write      read
general      13.7609 2.728747  1.759727   1.135862 0.9718021 0.9537938
vocation    668.5973 1.866266  3.170676   0.733063 0.9208712 0.9313796
```

3.2.5 Multikolineratitas

```
library(car)
vif(logmultinom)
```

```
            GVIF Df GVIF^(1/(2*Df))
ses     6.640420  2        1.605273
gender  2.650955  1        1.628175
write  66.396002  1        8.148374
read   53.940932  1        7.344449
```

3.2.6 Akurasi

```
df <- students[,c("ses","gender","write","read")]
```

```
#install.packages("caret")
library(caret)
prediksi <- predict(logmultinom, df, type = "class")
confusionMatrix(as.factor(prediksi),
                students$prog2)
```

```
Confusion Matrix and Statistics

          Reference
Prediction academic general vocation
  academic       90      25       21
  general         3       7        4
  vocation       12      13       25

Overall Statistics

               Accuracy : 0.61
                 95% CI : (0.5387, 0.678)
    No Information Rate : 0.525
    P-Value [Acc > NIR] : 0.009485

                  Kappa : 0.3094

 Mcnemar's Test P-Value : 1.959e-05

Statistics by Class:

                     Class: academic Class: general Class: vocation
Sensitivity                   0.8571         0.1556          0.5000
Specificity                   0.5158         0.9548          0.8333
Pos Pred Value                0.6618         0.5000          0.5000
Neg Pred Value                0.7656         0.7957          0.8333
Prevalence                    0.5250         0.2250          0.2500
Detection Rate                0.4500         0.0350          0.1250
Detection Prevalence          0.6800         0.0700          0.2500
Balanced Accuracy             0.6865         0.5552          0.6667
```

3.2.7 Kebaikan Model

```
logmultinom0 <- multinom(prog2 ~ 1, data = students)
```

```
# weights:  6 (2 variable)
```

```
initial  value 219.722458
final  value 204.096674
converged
```

```
#install.packages("lmtest")
library(lmtest)
lrtest(logmultinom0,logmultinom)
```

```
Likelihood ratio test

Model 1: prog2 ~ 1
Model 2: prog2 ~ ses + gender + write + read
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   2 -204.10
2  12 -174.72 10 58.743  6.263e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.3 Regresi Logistik Ordinal

3.3.1 Data

```
crash <- read.csv("Data/crash.csv")
head(crash,10)
```

```
   Gender Location SeatBelt Respon
1  Female    Urban      Yes      1
2    Male    Urban      Yes      1
3    Male    Urban       No      1
4  Female    Urban       No      1
5    Male    Rural      Yes      1
6  Female    Rural      Yes      1
7    Male    Rural       No      1
8  Female    Rural       No      1
9  Female    Urban       No      3
10 Female    Rural       No      3
```

```
library(dplyr)
crash <- crash %>% mutate(across(-c(Respon),as.factor))
str(crash)
```

```
'data.frame':   80 obs. of  4 variables:
 $ Gender  : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 1 2 1 1 1 ...
 $ Location: Factor w/ 2 levels "Rural","Urban": 2 2 2 2 1 1 1 1 2 1 ...
 $ SeatBelt: Factor w/ 2 levels "No","Yes": 2 2 1 1 2 2 1 1 1 1 ...
 $ Respon  : int  1 1 1 1 1 1 1 1 3 3 ...
```

```
crash$Respon <- ordered(crash$Respon, levels=c("1","2","3","4","5"))
str(crash)
```

```
'data.frame':   80 obs. of  4 variables:
 $ Gender  : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 1 2 1 1 1 ...
 $ Location: Factor w/ 2 levels "Rural","Urban": 2 2 2 2 1 1 1 1 2 1 ...
 $ SeatBelt: Factor w/ 2 levels "No","Yes": 2 2 1 1 2 2 1 1 1 1 ...
 $ Respon  : Ord.factor w/ 5 levels "1"<"2"<"3"<"4"<..: 1 1 1 1 1 1 1 1 3 3 ...
```

3.3.2 Pemodelan

```
#install.packages("MASS")
library(MASS)
orderlog <- polr(Respon~., method='logistic',data=crash)
summary(orderlog)
```

```
Call:
polr(formula = Respon ~ ., data = crash, method = "logistic")

Coefficients:
                 Value Std. Error t value
GenderMale    -0.05369     0.3974 -0.1351
LocationUrban  0.05661     0.3958  0.1430
SeatBeltYes   -0.31102     0.3974 -0.7827

Intercepts:
    Value   Std. Error t value
1|2 -1.5425  0.4450    -3.4664
2|3 -0.5523  0.4060    -1.3603
3|4  0.2649  0.3966     0.6678
4|5  1.2472  0.4264     2.9249

Residual Deviance: 256.8444
AIC: 270.8444
```

3.3.3 Odds Ratio

```
koefisien<-coef(summary(orderlog))
exp(koefisien[,1])
```

```
   GenderMale LocationUrban   SeatBeltYes           1|2           2|3
    0.9477303     1.0582414     0.7327004     0.2138542     0.5756362
          3|4           4|5
    1.3032362     3.4805710
```

```
# menghitung pvalue
p <- pnorm(abs(koefisien[,"t value"]), lower.tail = FALSE)*2
(ctabel<-cbind(round(koefisien,2), "pvalue"=round(p,3)))
```

```
              Value Std. Error t value pvalue
GenderMale    -0.05       0.40   -0.14  0.893
LocationUrban  0.06       0.40    0.14  0.886
SeatBeltYes   -0.31       0.40   -0.78  0.434
1|2           -1.54       0.44   -3.47  0.001
2|3           -0.55       0.41   -1.36  0.174
3|4            0.26       0.40    0.67  0.504
4|5            1.25       0.43    2.92  0.003
```

3.3.4 Multikolineratitas

```
library(car)
vif(orderlog)
```

```
  Gender Location SeatBelt 
1.002035 1.001265 1.001814 
```

3.3.5 Akurasi

```
df <- crash[,1:3]
prediksi <- predict(orderlog, df, type = "class")
confusionMatrix(as.factor(prediksi),
                crash$Respon)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 10  8  7  7  8
         2  0  0  0  0  0
         3  0  0  0  0  0
         4  0  0  0  0  0
         5  6  8  9  9  8

Overall Statistics
                                          
               Accuracy : 0.225           
                 95% CI : (0.1391, 0.3321)
    No Information Rate : 0.2             
    P-Value [Acc > NIR] : 0.3292          
```

```
                  Kappa : 0.0312

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            0.6250      0.0      0.0      0.0      0.5
Specificity            0.5312      1.0      1.0      1.0      0.5
Pos Pred Value         0.2500      NaN      NaN      NaN      0.2
Neg Pred Value         0.8500      0.8      0.8      0.8      0.8
Prevalence             0.2000      0.2      0.2      0.2      0.2
Detection Rate         0.1250      0.0      0.0      0.0      0.1
Detection Prevalence   0.5000      0.0      0.0      0.0      0.5
Balanced Accuracy      0.5781      0.5      0.5      0.5      0.5
```

3.3.6 Kebaikan Model

```r
orderlog0 <-polr(Respon~1, method = "logistic", data = crash)
#install.packages("lmtest")
library(lmtest)
lrtest(orderlog0,orderlog)
```

```
Likelihood ratio test

Model 1: Respon ~ 1
Model 2: Respon ~ Gender + Location + SeatBelt
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   4 -128.75
2   7 -128.42  3 0.6657     0.8813
```

Chapter 4

Discriminant Analysis

4.1 Analisis Diskriminan Dua Grup

4.1.1 Data

```
library(readxl)
pinjaman <- read_excel("Data/pinjaman.xlsx")
head(pinjaman,10)
```

```
# A tibble: 10 x 6
      X1    X2    X3    X4    X5     Y
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1    98    35    12     4     4     1
 2    65    44     5     3     1     1
 3    22    50     0     2     7     1
 4    78    60    34     5     5     1
 5    50    31     4     2     2     1
 6    21    30     5     3     7     1
 7    42    32    21     4    11     1
 8    20    41    10     2     3     1
 9    33    25     0     3     6     1
10    57    32     8     2     5     1
```

```
str(pinjaman)
```

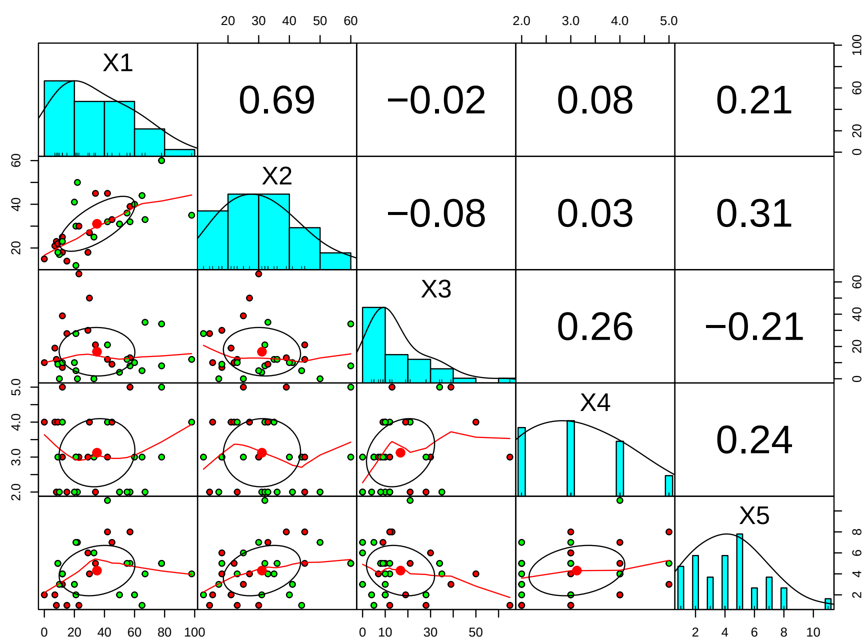
```
tibble [32 x 6] (S3: tbl_df/tbl/data.frame)
 $ X1: num [1:32] 98 65 22 78 50 21 42 20 33 57 ...
 $ X2: num [1:32] 35 44 50 60 31 30 32 41 25 32 ...
 $ X3: num [1:32] 12 5 0 34 4 5 21 10 0 8 ...
 $ X4: num [1:32] 4 3 2 5 2 3 4 2 3 2 ...
 $ X5: num [1:32] 4 1 7 5 2 7 11 3 6 5 ...
```

```
 $ Y : num [1:32] 1 1 1 1 1 1 1 1 1 1 ...
```

```
pinjaman$Y <- as.factor(pinjaman$Y)
str(pinjaman)
```

```
tibble [32 x 6] (S3: tbl_df/tbl/data.frame)
 $ X1: num [1:32] 98 65 22 78 50 21 42 20 33 57 ...
 $ X2: num [1:32] 35 44 50 60 31 30 32 41 25 32 ...
 $ X3: num [1:32] 12 5 0 34 4 5 21 10 0 8 ...
 $ X4: num [1:32] 4 3 2 5 2 3 4 2 3 2 ...
 $ X5: num [1:32] 4 1 7 5 2 7 11 3 6 5 ...
 $ Y : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

```
library(psych)
pairs.panels(pinjaman[1:5],
             gap = 0,
             bg = c("red", "green")[pinjaman$Y],
             pch = 21)
```



4.1.2 Pemodelan Linier

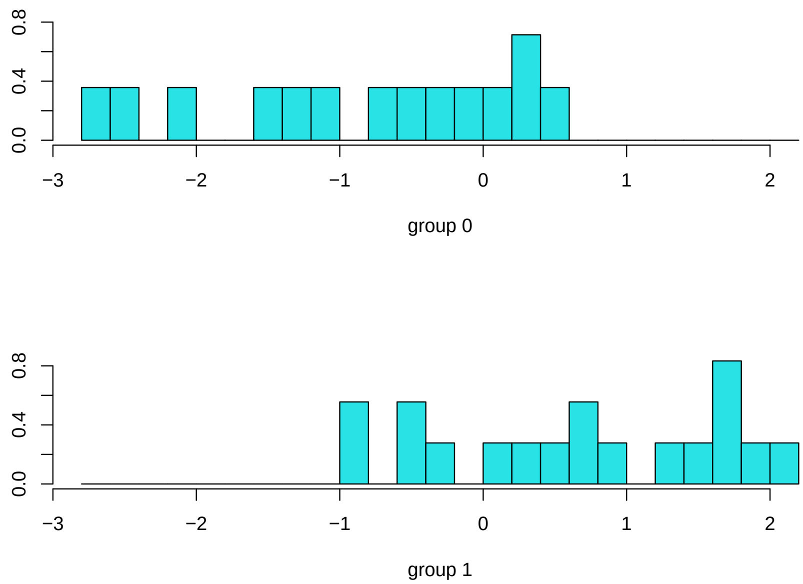
```
library(MASS)
modellda1 <- lda(Y ~ X1 + X2 + X3 + X4 + X5, data=pinjaman)
modellda1
```

```
Call:
lda(Y ~ X1 + X2 + X3 + X4 + X5, data = pinjaman)

Prior probabilities of groups:
     0      1
0.4375 0.5625

Group means:
        X1       X2       X3       X4       X5
0 23.07143 26.78571 23.21429 3.428571 4.071429
1 44.33333 34.38889 11.72222 2.888889 4.500000

Coefficients of linear discriminants:
            LD1
X1  0.037015853
X2 -0.004820049
X3 -0.043555291
X4 -0.477408359
X5 -0.008483836
```

```
m <- manova(cbind(pinjaman$X1,pinjaman$X2,pinjaman$X3,
                  pinjaman$X4,pinjaman$X5) ~ pinjaman$Y)
summary(m, test = 'Wilks')
```

```
           Df   Wilks approx F num Df den Df  Pr(>F)
pinjaman$Y  1 0.62715   3.0915      5     26 0.02544 *
Residuals  30
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
p <- predict(modellda1, pinjaman)
ldahist(data = p$x, g = pinjaman$Y)
```




```r
library(caret)
confusionMatrix(p$class,pinjaman$Y)
```

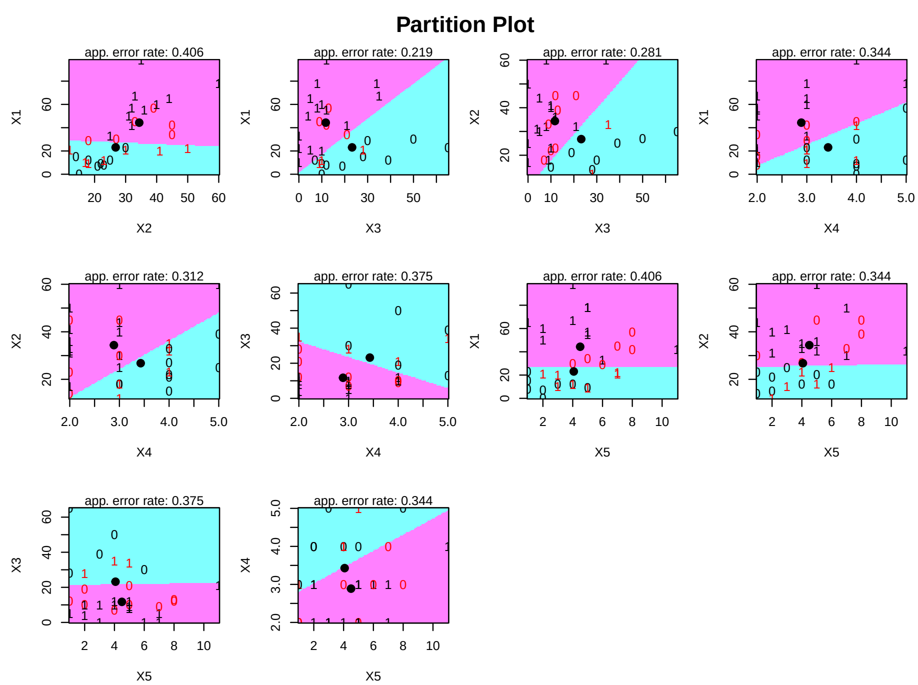
```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  9  4
         1  5 14

               Accuracy : 0.7188
                 95% CI : (0.5325, 0.8625)
    No Information Rate : 0.5625
    P-Value [Acc > NIR] : 0.0523

                  Kappa : 0.424

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.6429
            Specificity : 0.7778
         Pos Pred Value : 0.6923
         Neg Pred Value : 0.7368
             Prevalence : 0.4375
```

```
         Detection Rate : 0.2812
   Detection Prevalence : 0.4062
      Balanced Accuracy : 0.7103

       'Positive' Class : 0
```

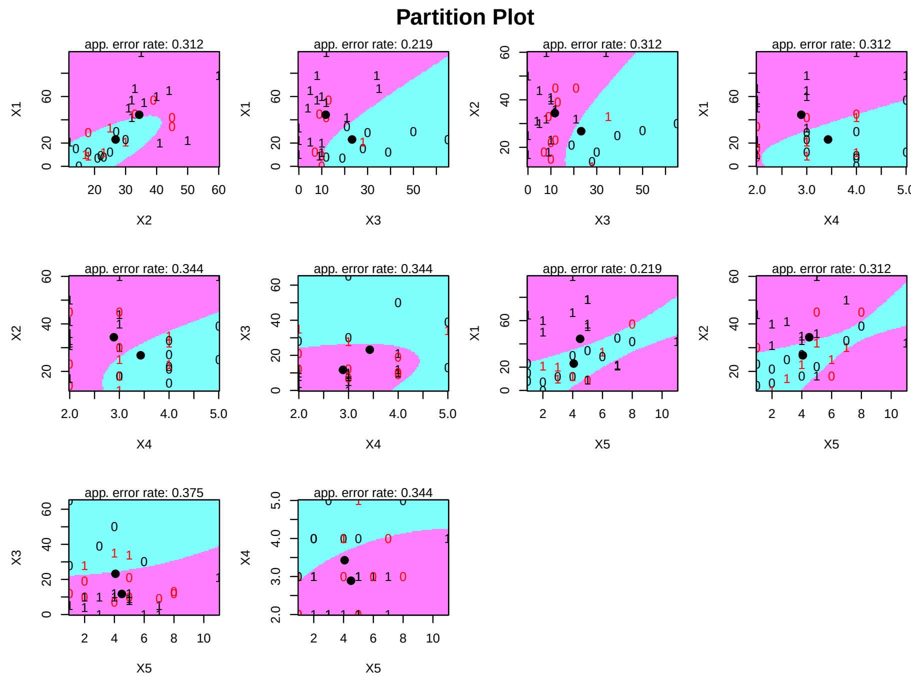
```
mean(p$class==pinjaman$Y)
```

```
[1] 0.71875
```

```
#install.packages("klaR")
library(klaR)
#Partition plot
partimat(Y~., data = pinjaman, method = "lda")
```




```
partimat(Y~., data = pinjaman, method = "qda")
```



4.1.5 Pemodelan Quadratik

```
modellda2 <- qda(Y ~ X1 + X2 + X3 + X4 + X5, data=pinjaman)
modellda2
```

```
Call:
qda(Y ~ X1 + X2 + X3 + X4 + X5, data = pinjaman)

Prior probabilities of groups:
     0      1
0.4375 0.5625

Group means:
        X1       X2       X3       X4       X5
0 23.07143 26.78571 23.21429 3.428571 4.071429
1 44.33333 34.38889 11.72222 2.888889 4.500000
```

```
p <- predict(modellda2, pinjaman)
mean(p$class==pinjaman$Y)
```

```
[1] 0.84375
```

4.1.6 Tipe Diskriminan Lainnya

```
# Mixture discriminant analysis - MDA
# install.packages("mda")
library(mda)
modellda3 <- mda(Y ~ X1 + X2 + X3 + X4 + X5, data=pinjaman)
p <- predict(modellda3, pinjaman)
mean(p==pinjaman$Y)
```

```
[1] 0.875
```

```
# Flexible discriminant analysis - FDA
modellda4 <- fda(Y ~ X1 + X2 + X3 + X4 + X5, data=pinjaman)
p <- predict(modellda4, pinjaman)
mean(p==pinjaman$Y)
```

```
[1] 0.71875
```

```
# Regularized discriminant analysis - RDA
modellda5 <- rda(Y ~ X1 + X2 + X3 + X4 + X5, data=pinjaman)
p <- predict(modellda5, pinjaman)
mean(p$class==pinjaman$Y)
```

```
[1] 0.78125
```

4.2 Analisis Diskriminan Tiga Grup

4.2.1 Data

```
data("iris")
head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```
str(iris)
```

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
library(MASS)
lda.iris <- lda(Species ~ ., iris)
lda.iris
```

```
Call:
lda(Species ~ ., data = iris)

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026

Coefficients of linear discriminants:
                    LD1         LD2
Sepal.Length  0.8293776 -0.02410215
Sepal.Width   1.5344731 -2.16452123
Petal.Length -2.2012117  0.93192121
Petal.Width  -2.8104603 -2.83918785

Proportion of trace:
   LD1    LD2
0.9912 0.0088
```

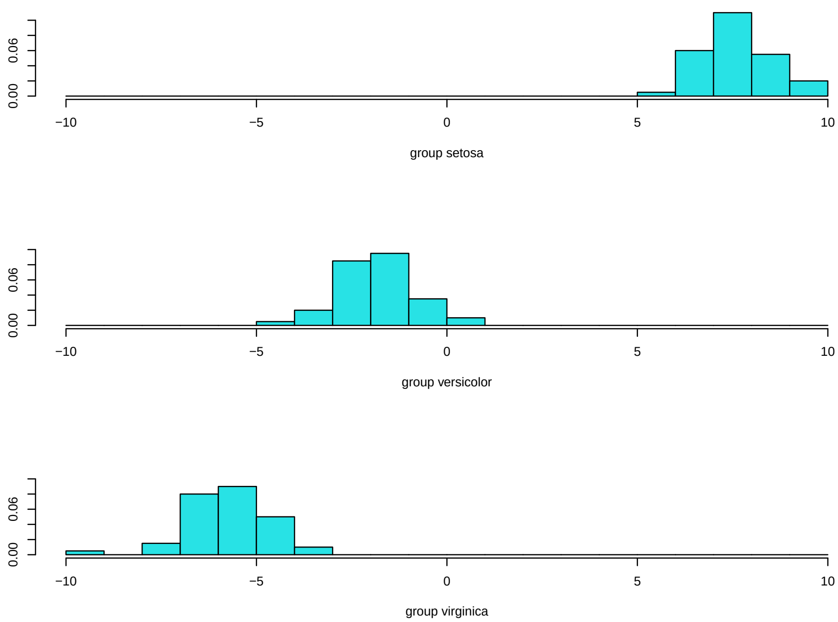
4.2.2 Uji Signifikansi Fungsi Diskriminan

```
m <- manova(cbind(iris$Sepal.Length,iris$Sepal.Width,iris$Petal.Length,
                  iris$Petal.Width) ~ iris$Species)
summary(m, test = 'Wilks')
```

```
              Df    Wilks approx F num Df den Df    Pr(>F)
iris$Species   2 0.023439   199.15      8    288 < 2.2e-16 ***
Residuals    147
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.2.3 Akurasi

```
p <- predict(lda.iris, iris)
ldahist(data = p$x, g = iris$Species)
```




```
table(p$class,iris$Species)
```

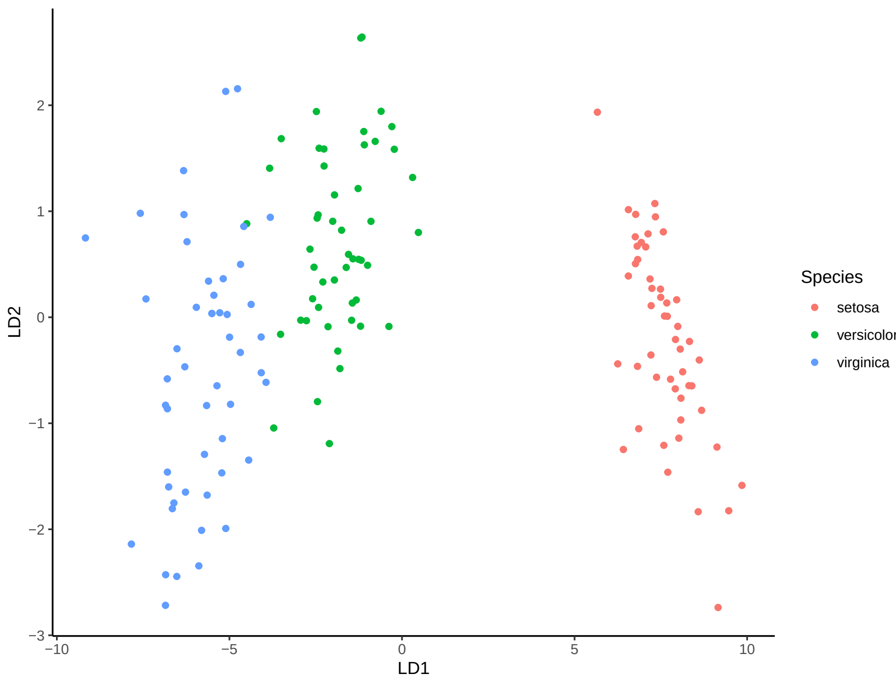
```
             setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         1
  virginica       0          2        49
```

```
mean(p$class==iris$Species)
```

```
[1] 0.98
```

4.2.4 Visualisasi

```
library(ggplot2)
lda.data <- cbind(iris,  p$x)
ggplot(lda.data, aes(LD1, LD2)) +
  geom_point(aes(color = Species)) + theme_classic()
```



4.2.5 Pemodelan Quadratik

```
qda.iris <- qda(Species ~ ., data=iris)
qda.iris
```

```
Call:
qda(Species ~ ., data = iris)

Prior probabilities of groups:
    setosa versicolor  virginica 
 0.3333333  0.3333333  0.3333333 

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026
```

```
p <- predict(qda.iris, iris)
mean(p$class==iris$Species)
```

```
[1] 0.98
```

4.2.6 Tipe Diskriminan Lainnya

```
# Mixture discriminant analysis - MDA
# install.packages("mda")
library(mda)
mda.iris <- mda(Species ~ ., data=iris)
mda.iris
```

```

Call:
mda(formula = Species ~ ., data = iris)

Dimension: 4

Percent Between-Group Variance Explained:
    v1     v2     v3     v4
 95.06  97.78  99.59 100.00

Degrees of Freedom (per dimension): 5

Training Misclassification Error: 0.01333 ( N = 150 )

Deviance: 13.302
```

```
p <- predict(mda.iris, iris)
mean(p==iris$Species)
```

```
[1] 0.9866667
```

```
# Flexible discriminant analysis - FDA
fda.iris <- fda(Species ~ ., data=iris)
fda.iris
```

```

Call:
fda(formula = Species ~ ., data = iris)

Dimension: 2

Percent Between-Group Variance Explained:
    v1     v2
 99.12 100.00

Degrees of Freedom (per dimension): 5

Training Misclassification Error: 0.02 ( N = 150 )
```

```
p <- predict(fda.iris, iris)
mean(p==iris$Species)
```

```
[1] 0.98
```

```
# Regularized discriminant analysis - RDA
rda.iris <- rda(Species ~ ., data=iris)
rda.iris
```

```
Call:
rda(formula = Species ~ ., data = iris)

Regularization parameters:
    gamma    lambda
0.1945631 0.5593066

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Misclassification rate:
       apparent: 2 %
cross-validated: 2 %
```

```
p <- predict(rda.iris, iris)
mean(p$class==iris$Species)
```

```
[1] 0.98
```

Chapter 5

Cluster Analysis

5.1 Metode berhirarki

Ref: https://rpubs.com/odenipinedo/cluster-analysis-in-R

```
library(readxl)
Provinsi <- read_excel("Data/provinsi.xlsx")
Prov.scaled = scale(Provinsi[,c(4:8)])
rownames(Prov.scaled) = Provinsi$Provinsi
head(Prov.scaled)
```

```
               IPM         UHH        RLS         PPK       Gini
Aceh    0.20822137  0.04044611  0.7444782 -0.62264434 -0.8089350
Sumut   0.20085709 -0.39282591  1.0245728 -0.11277090 -0.6516207
Sumbar  0.36532598 -0.23835502  0.4747574  0.01481560 -1.2546590
Riau    0.50033775  0.59428078  0.5162529  0.19012890 -0.9138112
Jambi   0.05848104  0.50762638 -0.1165536 -0.18648754 -0.6778397
Sumsel -0.21890679 -0.08765171 -0.2825356 -0.02582306  0.1349510
```

```
## membuat dissimilarity matrix
dprov = dist(Prov.scaled, method="euclidean")
```

```
c.comp = hclust(dprov, method = "complete")
cor(dprov , cophenetic(c.comp))
```

```
[1] 0.7853523
```

```
c.sing = hclust(dprov, method = "single")
cor(dprov , cophenetic(c.sing))
```

```
[1] 0.7905858
```

```
c.avrg = hclust(dprov, method = "average")
cor(dprov , cophenetic(c.avrg))
```

```
[1] 0.8092689
```

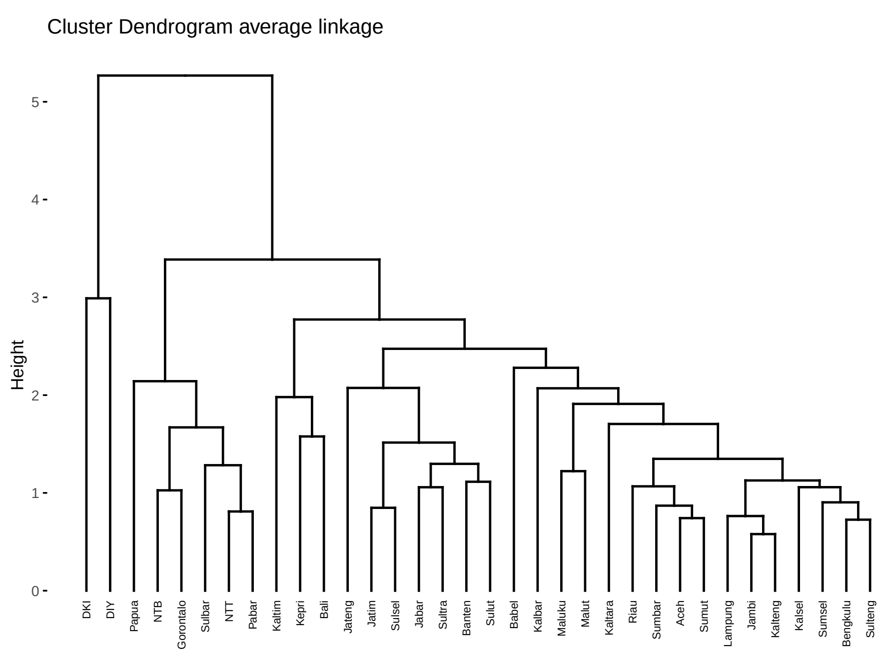
```
c.ward = hclust(dprov, method = "ward.D")
cor(dprov , cophenetic(c.ward))
```

```
[1] 0.5336018
```

```
c.ctrd = hclust(dprov, method = "centroid")
cor(dprov , cophenetic(c.ctrd))
```

```
[1] 0.7700878
```

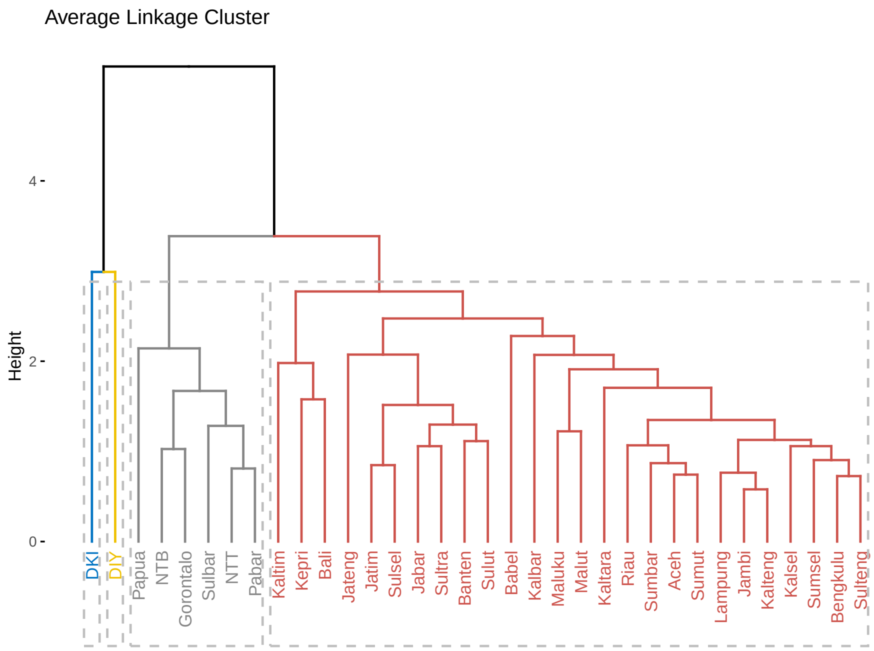
```
library(factoextra)
fviz_dend(c.avrg, cex = 0.5,
          main = "Cluster Dendrogram average linkage")
```




```
avg_coph <- cophenetic(c.avrg)
avg_clust <- cutree(c.avrg, k = 4)
table(avg_clust)
```

```
avg_clust
 1  2  3  4
26  1  1  6
```

```
fviz_dend(c.avrg, k = 4,
          k_colors = "jco",
          rect = T,
          main = "Average Linkage Cluster")
```

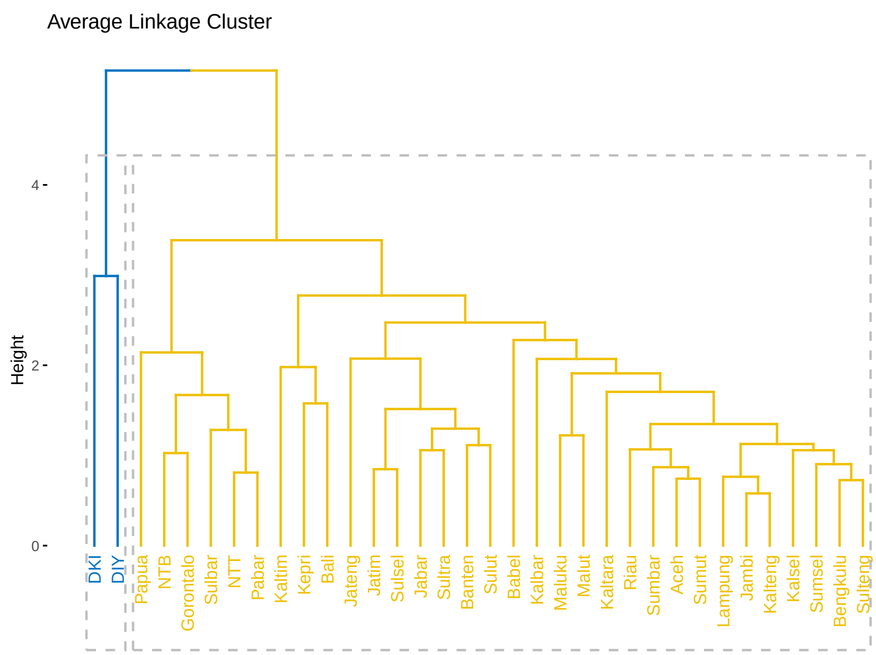



```
library(clValid)
library(cluster)
# internal measures
internal <- clValid(Prov.scaled, nClust = 2:6,
                    clMethods = "hierarchical",
                    validation = "internal",
                    metric = "euclidean",
                    method = "average")
summary(internal)
```

```
Clustering Methods:
 hierarchical

Cluster sizes:
 2 3 4 5 6

Validation Measures:
```

```
                                  2       3       4       5       6

hierarchical Connectivity   4.5246 10.3012 11.6345 18.3198 24.1508
             Dunn           0.3637  0.3703  0.3703  0.3224  0.3592
             Silhouette     0.4915  0.3484  0.3092  0.2567  0.3117

Optimal Scores:

             Score  Method       Clusters
Connectivity 4.5246 hierarchical 2
Dunn         0.3703 hierarchical 3
Silhouette   0.4915 hierarchical 2
```

```
fviz_dend(c.avrg, k = 2,
          k_colors = "jco",
          rect = T,
          main = "Average Linkage Cluster")
```

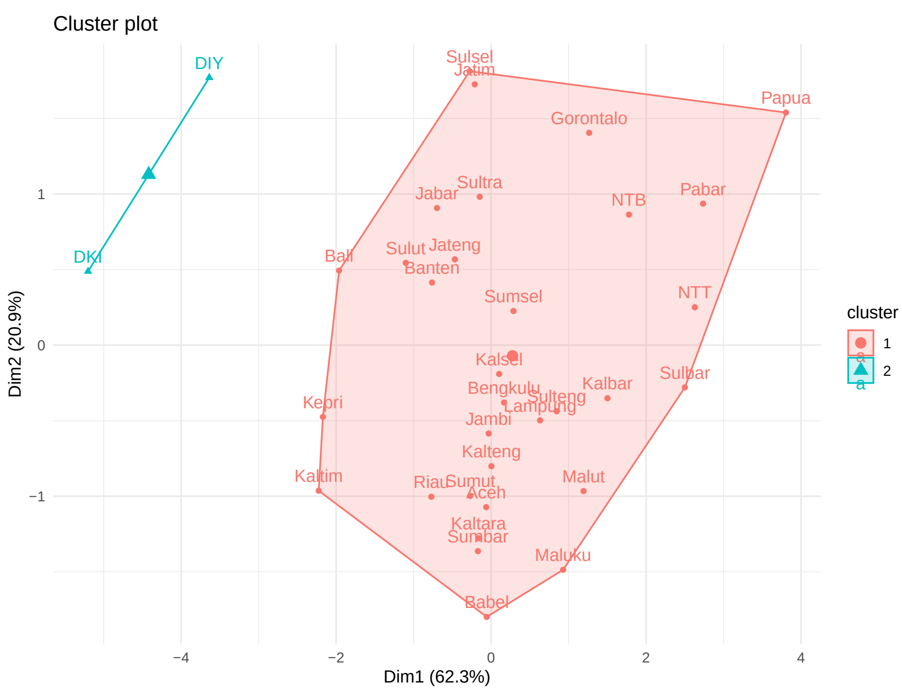



```
group = cutree(c.avrg, k = 2)
group
```

```
     Aceh     Sumut    Sumbar      Riau     Jambi    Sumsel  Bengkulu   Lampung
        1         1         1         1         1         1         1         1
    Babel     Kepri       DKI     Jabar    Jateng       DIY     Jatim    Banten
```

```
       1        1        2        1        1        2        1        1
    Bali      NTB      NTT   Kalbar  Kalteng   Kalsel   Kaltim  Kaltara
       1        1        1        1        1        1        1        1
   Sulut  Sulteng   Sulsel   Sultra Gorontalo   Sulbar   Maluku    Malut
       1        1        1        1        1        1        1        1
   Pabar    Papua
       1        1
```

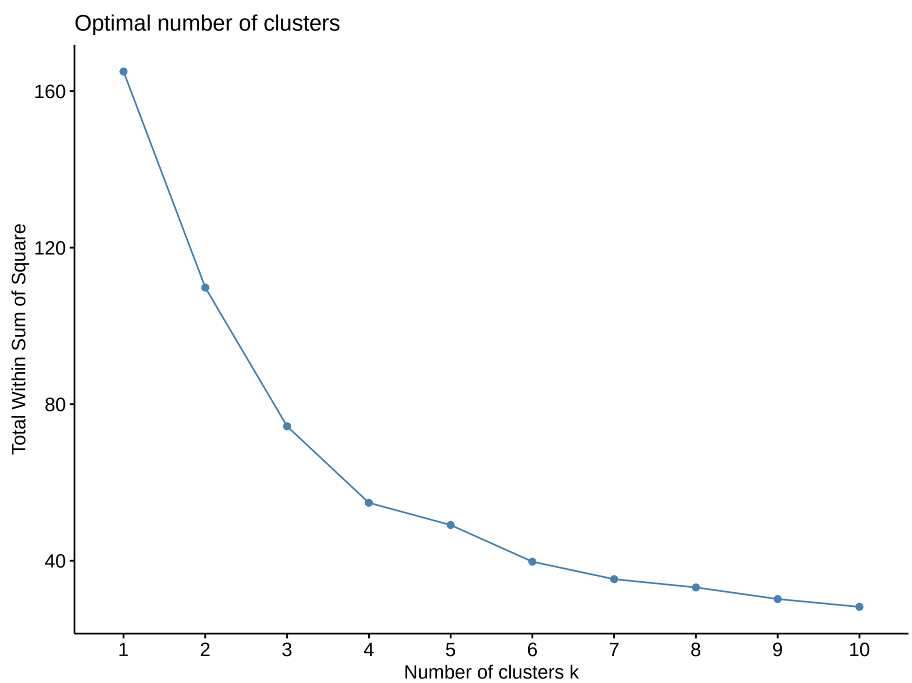
```
fviz_cluster(list(data = Prov.scaled,
                  cluster = group)) +
  theme_minimal()
```



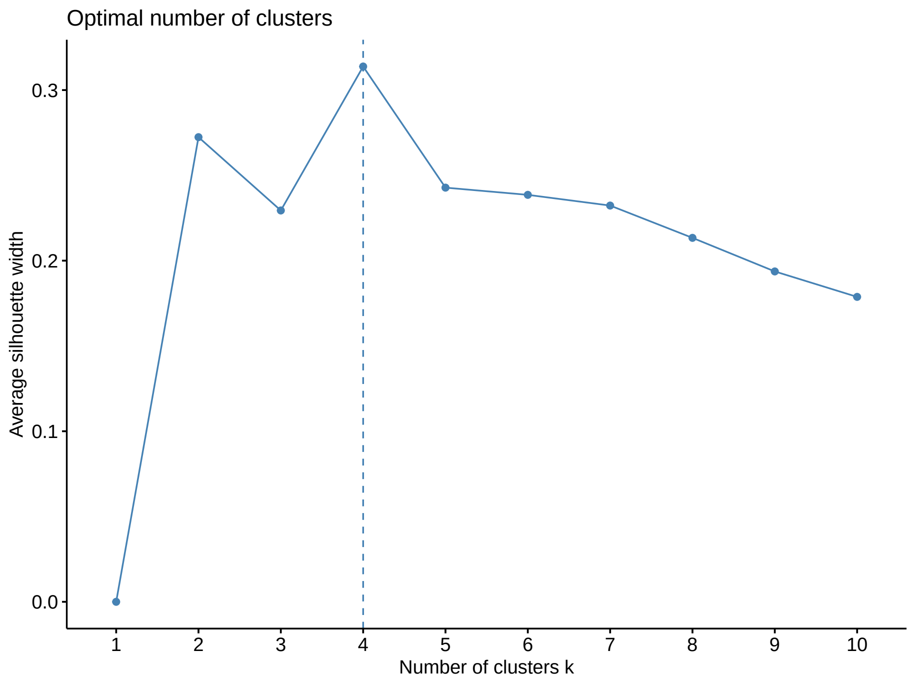

```
prcomp(Prov.scaled)
```

```
Standard deviations (1, .., p=5):
[1] 1.7653705 1.0227284 0.7270850 0.5299864 0.1671984

Rotation (n x k) = (5 x 5):
            PC1         PC2          PC3           PC4         PC5
IPM  -0.5601680 -0.05311199 -0.005227509 -0.0006949187 -0.82665781
UHH  -0.4513030  0.05646383 -0.811065327  0.2024129889  0.30714735
RLS  -0.4591728 -0.33781331  0.497619343  0.5648220282  0.32923179
PPK  -0.5069166  0.09086739  0.227624805 -0.7546468667  0.33685862
Gini -0.1213811  0.93360390  0.206658283  0.2655422416  0.02073819
```

5.2 Metode tidak berhirarki - kmeans




```
fviz_nbclust(Prov.scaled, kmeans, method = "silhouette")
```

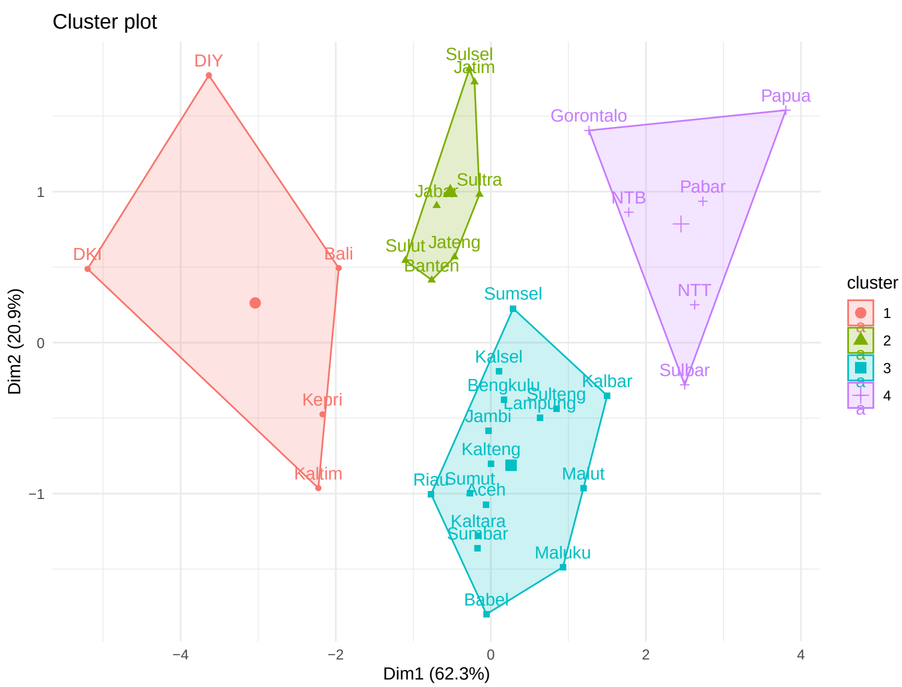



```
set.seed(1)
km = kmeans(Prov.scaled, centers=4)
km
```

```
K-means clustering with 4 clusters of sizes 5, 7, 16, 6

Cluster means:
          IPM        UHH        RLS         PPK       Gini
1  1.67223995  1.1202353  1.3689855  1.75840321  0.6331131
2  0.22785944  0.6620973 -0.1491572  0.09292014  0.9739608
3 -0.08819085 -0.1001318  0.1246391 -0.24567350 -0.7991029
4 -1.42419372 -1.4389581 -1.2991755 -0.91861351  0.4670591

Clustering vector:
     Aceh     Sumut    Sumbar      Riau     Jambi    Sumsel  Bengkulu   Lampung
        3         3         3         3         3         3         3         3
    Babel     Kepri       DKI     Jabar    Jateng       DIY     Jatim    Banten
        3         1         1         2         2         1         2         2
     Bali       NTB       NTT    Kalbar   Kalteng    Kalsel    Kaltim   Kaltara
        1         4         4         3         3         3         1         3
    Sulut   Sulteng    Sulsel    Sultra Gorontalo    Sulbar    Maluku     Malut
        2         3         2         2         4         4         3         3
    Pabar     Papua
        4         4
```

```
Within cluster sum of squares by cluster:
[1] 17.054859  7.933134 22.111511  7.711994
 (between_SS / total_SS =  66.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

```
fviz_cluster(list(data = Prov.scaled, cluster = km$cluster)) + theme_minimal()
```



Chapter 6

PCA Analysis and Biplot

6.1 PCA

```
# impor data dari excel, beri nama: Provinsi
library(readxl)
Provinsi = read_excel("Data/provinsi.xlsx")
Prov.scaled = scale(Provinsi[,c(4:8)])
round(cor(Prov.scaled),3)
```

```
       IPM   UHH    RLS   PPK   Gini
IPM  1.000 0.780  0.811 0.872  0.159
UHH  0.780 1.000  0.447 0.581  0.153
RLS  0.811 0.447  1.000 0.637 -0.059
PPK  0.872 0.581  0.637 1.000  0.249
Gini 0.159 0.153 -0.059 0.249  1.000
```

```
# PCA langkah manual
Prov.eigen = eigen(cov(Prov.scaled))
Prov.eigen
```

```
eigen() decomposition
$values
[1] 3.11653307 1.04597347 0.52865259 0.28088555 0.02795532

$vectors
           [,1]        [,2]         [,3]          [,4]        [,5]
[1,] -0.5601680 -0.05311199  0.005227509 -0.0006949187  0.82665781
[2,] -0.4513030  0.05646383  0.811065327  0.2024129889 -0.30714735
[3,] -0.4591728 -0.33781331 -0.497619343  0.5648220282 -0.32923179
[4,] -0.5069166  0.09086739 -0.227624805 -0.7546468667 -0.33685862
[5,] -0.1213811  0.93360390 -0.206658283  0.2655422416 -0.02073819
```

```r
Prov.eigen$values
```

```
[1] 3.11653307 1.04597347 0.52865259 0.28088555 0.02795532
```

```r
Prov.eigen$values/5
```

```
[1] 0.623306615 0.209194694 0.105730518 0.056177109 0.005591064
```

```r
cumsum(Prov.eigen$values/5)
```

```
[1] 0.6233066 0.8325013 0.9382318 0.9944089 1.0000000
```

```r
Prov.pc = as.matrix(Prov.scaled) %*% Prov.eigen$vectors
round(Prov.pc,3)
```

```
        [,1]   [,2]   [,3]   [,4]   [,5]
 [1,] -0.063 -1.072 -0.028  0.684  0.141
 [2,] -0.269 -0.998 -0.667  0.411  0.001
 [3,] -0.170 -1.363 -0.172 -0.125  0.240
 [4,] -0.771 -1.003  0.373  0.025  0.016
 [5,] -0.032 -0.585  0.653 -0.002  0.008
 [6,]  0.289  0.226  0.046 -0.122 -0.055
 [7,]  0.167 -0.380 -0.246  0.160  0.149
 [8,]  0.632 -0.498  0.644 -0.115 -0.054
 [9,] -0.057 -1.798  0.675 -1.464 -0.089
[10,] -2.171 -0.475 -1.111 -0.280 -0.100
[11,] -5.201  0.488 -1.517 -0.455 -0.423
[12,] -0.699  0.908  0.818  0.388 -0.141
[13,] -0.467  0.567  1.901 -0.226 -0.064
[14,] -3.637  1.770  0.377  0.349  0.361
[15,] -0.211  1.726  0.527 -0.300  0.118
[16,] -0.763  0.414 -0.365 -0.198  0.007
[17,] -1.962  0.494  0.024 -0.719  0.052
[18,]  1.779  0.864 -0.537 -0.824  0.322
[19,]  2.630  0.251 -0.136  0.131  0.011
[20,]  1.501 -0.351  1.137 -0.243 -0.049
[21,]  0.004 -0.801  0.195 -0.277 -0.039
[22,]  0.104 -0.191 -0.358 -0.828  0.030
[23,] -2.225 -0.963  0.752  0.305  0.020
[24,] -0.162 -1.278  1.179  0.698 -0.173
[25,] -1.101  0.544 -0.154  0.823 -0.143
[26,]  0.847 -0.438 -0.472  0.097  0.061
[27,] -0.276  1.813 -0.105  0.254  0.105
[28,] -0.147  0.982  0.109  0.925 -0.004
[29,]  1.266  1.405 -0.356 -0.169  0.136
[30,]  2.501 -0.280 -0.787 -0.540  0.062
[31,]  0.929 -1.487 -1.397  0.735  0.080
```

```
[32,]  1.193 -0.966 -0.326  0.739 -0.008
[33,]  2.735  0.936 -0.533  0.218 -0.091
[34,]  3.806  1.539 -0.145 -0.057 -0.487
```

```
# dengan fungsi prcomp
pc = prcomp(x = Prov.scaled, center=TRUE, scale=TRUE)
summary(pc)
```

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5
Standard deviation     1.7654 1.0227 0.7271 0.52999 0.16720
Proportion of Variance 0.6233 0.2092 0.1057 0.05618 0.00559
Cumulative Proportion  0.6233 0.8325 0.9382 0.99441 1.00000
```

```
round(pc$x,3)#scores
```

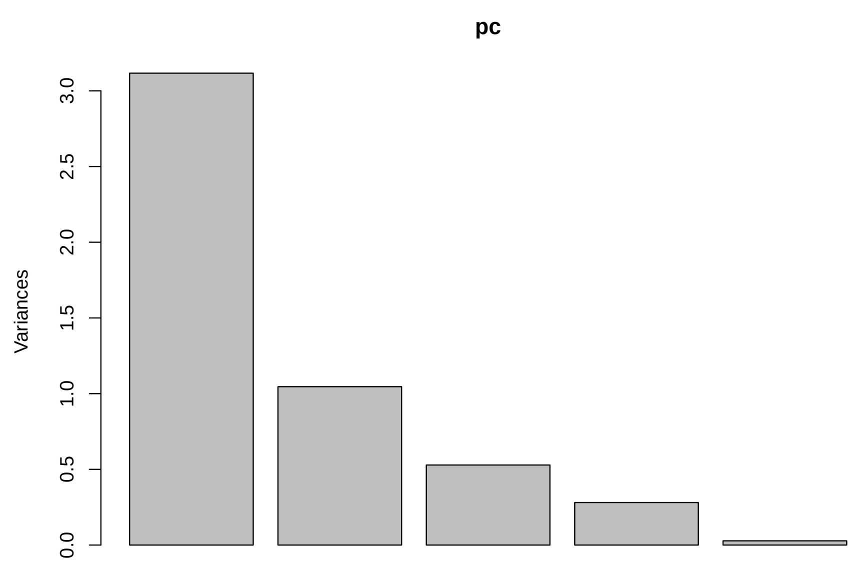
```
         PC1    PC2    PC3    PC4    PC5
 [1,] -0.063 -1.072  0.028  0.684 -0.141
 [2,] -0.269 -0.998  0.667  0.411 -0.001
 [3,] -0.170 -1.363  0.172 -0.125 -0.240
 [4,] -0.771 -1.003 -0.373  0.025 -0.016
 [5,] -0.032 -0.585 -0.653 -0.002 -0.008
 [6,]  0.289  0.226 -0.046 -0.122  0.055
 [7,]  0.167 -0.380  0.246  0.160 -0.149
 [8,]  0.632 -0.498 -0.644 -0.115  0.054
 [9,] -0.057 -1.798 -0.675 -1.464  0.089
[10,] -2.171 -0.475  1.111 -0.280  0.100
[11,] -5.201  0.488  1.517 -0.455  0.423
[12,] -0.699  0.908 -0.818  0.388  0.141
[13,] -0.467  0.567 -1.901 -0.226  0.064
[14,] -3.637  1.770 -0.377  0.349 -0.361
[15,] -0.211  1.726 -0.527 -0.300 -0.118
[16,] -0.763  0.414  0.365 -0.198 -0.007
[17,] -1.962  0.494 -0.024 -0.719 -0.052
[18,]  1.779  0.864  0.537 -0.824 -0.322
[19,]  2.630  0.251  0.136  0.131 -0.011
[20,]  1.501 -0.351 -1.137 -0.243  0.049
[21,]  0.004 -0.801 -0.195 -0.277  0.039
[22,]  0.104 -0.191  0.358 -0.828 -0.030
[23,] -2.225 -0.963 -0.752  0.305 -0.020
[24,] -0.162 -1.278 -1.179  0.698  0.173
[25,] -1.101  0.544  0.154  0.823  0.143
[26,]  0.847 -0.438  0.472  0.097 -0.061
[27,] -0.276  1.813  0.105  0.254 -0.105
[28,] -0.147  0.982 -0.109  0.925  0.004
[29,]  1.266  1.405  0.356 -0.169 -0.136
[30,]  2.501 -0.280  0.787 -0.540 -0.062
```

```
[31,]  0.929 -1.487  1.397  0.735 -0.080
[32,]  1.193 -0.966  0.326  0.739  0.008
[33,]  2.735  0.936  0.533  0.218  0.091
[34,]  3.806  1.539  0.145 -0.057  0.487
```

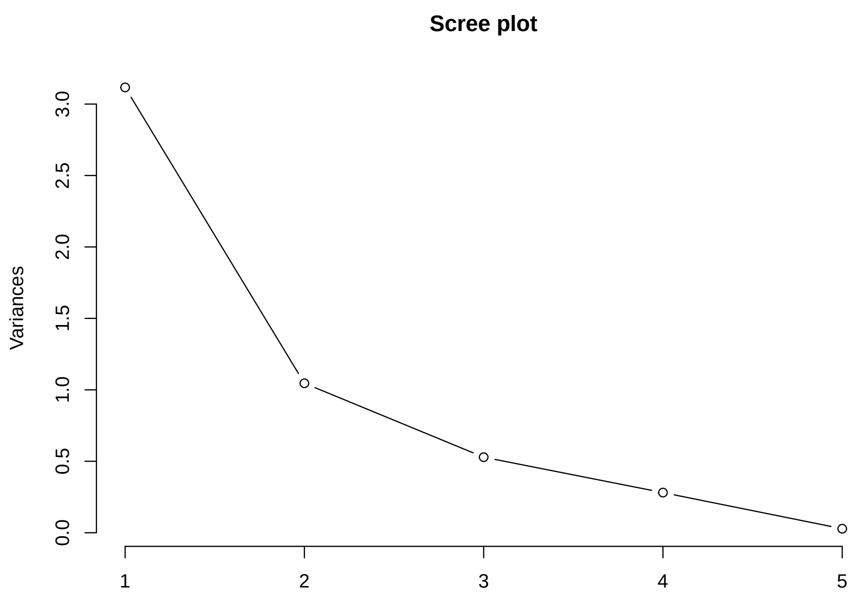
```
round(pc$rotation,3)  #loadings
```

```
        PC1    PC2    PC3    PC4    PC5
IPM  -0.560 -0.053 -0.005 -0.001 -0.827
UHH  -0.451  0.056 -0.811  0.202  0.307
RLS  -0.459 -0.338  0.498  0.565  0.329
PPK  -0.507  0.091  0.228 -0.755  0.337
Gini -0.121  0.934  0.207  0.266  0.021
```

```
plot(pc)
```




```
screeplot(x = pc, type="line", main="Scree plot")
```

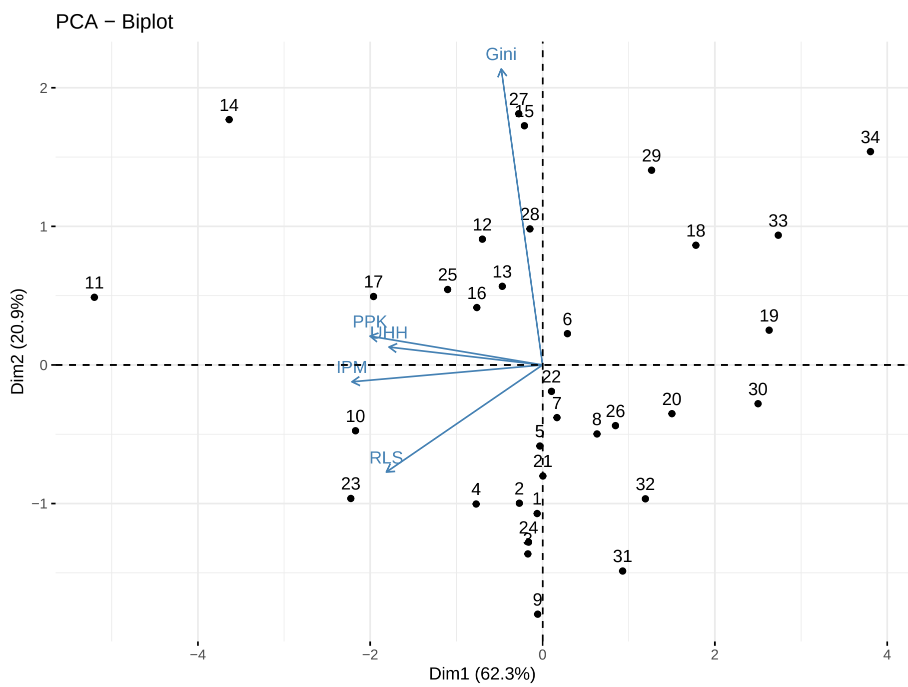



```
# korelasi variabel asli dengan PC
data = cbind(Prov.pc, Prov.scaled)
korelasi = cor(data)
korelasi[6:10,1:2]
```

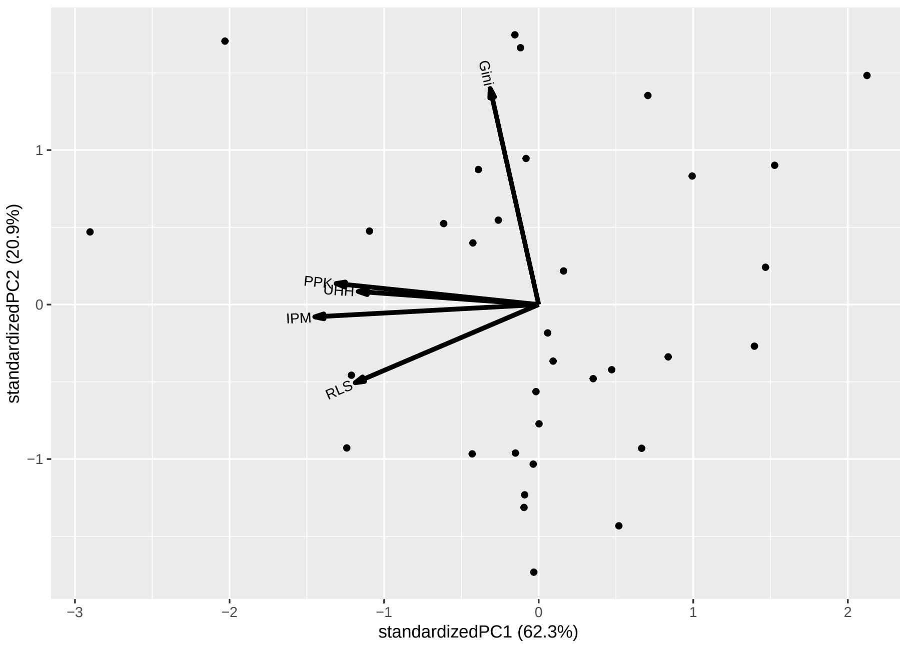
```
IPM  -0.9889040 -0.05431915
UHH  -0.7967169  0.05774717
RLS  -0.8106101 -0.34549128
PPK  -0.8948956  0.09293267
Gini -0.2142826  0.95482326
```

6.2 Biplot

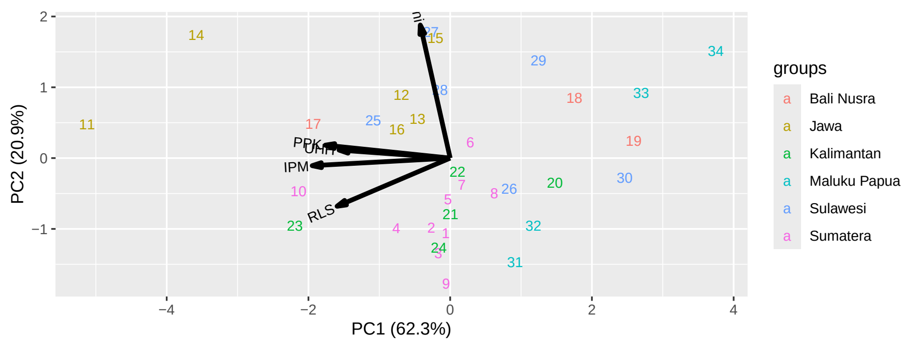
```
# biplot
library(factoextra)
fviz_pca(pc)
```



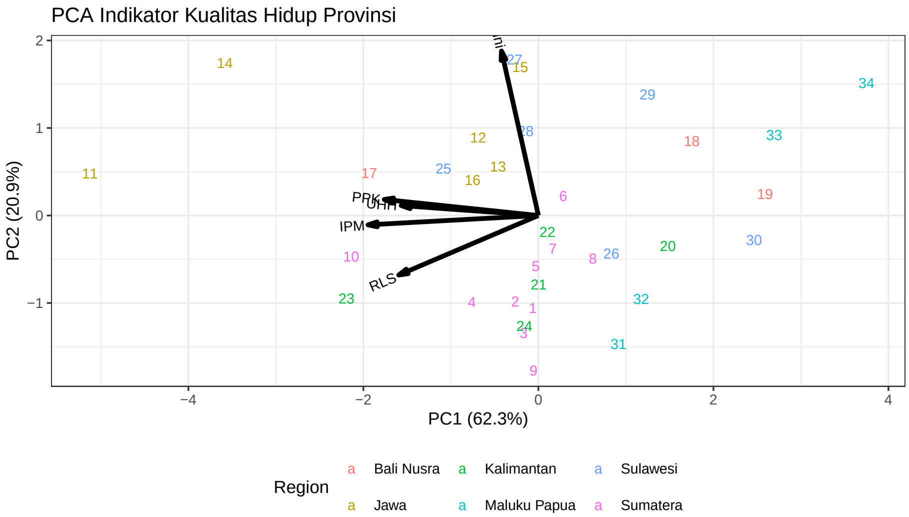

```
# alternatif bentuk biplot
# install.packages("remotes")
# remotes::install_github("vqv/ggbiplot")
library(ggbiplot)
ggbiplot(pc)
```




```r
biplot = ggbiplot(pcobj = pc,
                  choices = c(1,2),
                  obs.scale = 1, var.scale = 1,
                  labels = row.names(Provinsi),
                  varname.size = 3,
                  varname.abbrev = FALSE,
                  var.axes = TRUE,
                  group = Provinsi$Region)
biplot
```




```
biplot2 = biplot + theme_bw() +
  theme(legend.position="bottom") +
  labs(
  title = "PCA Indikator Kualitas Hidup Provinsi",
  color = "Region")
biplot2
```


PCA Indikator Kualitas Hidup Provinsi
PC1 (62.3%)
PC2 (20.9%)
PRK
UHH
IPM
RLS
Region
Bali Nusra
Kalimantan
Sulawesi
Jawa
Maluku Papua
Sumatera

Chapter 7

Factor Analysis and Structural Equation Modeling (SEM)

7.1 Analisis Faktor

```
harga <- read.csv("Data/harga.csv")
head(harga)
```

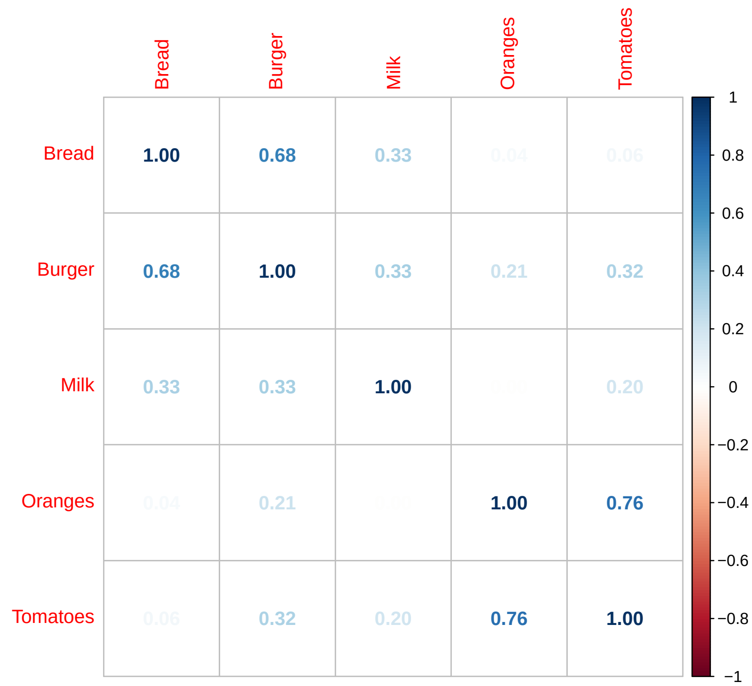
```
         City Bread Burger Milk Oranges Tomatoes
1    Atlanta   24.5   94.5 73.9    80.1     41.6
2  Baltimore   26.5   91.0 67.5    74.6     33.3
3     Boston   29.7  100.8 61.4   104.0     59.6
4    Buffalo   22.8   86.6 65.3   118.4     61.2
5    Chicago   26.7   86.7 62.7   105.9     60.2
6 Cincinnati   25.3  102.5 63.3    99.3     45.6
```

```
str(harga)
```

```
'data.frame':	23 obs. of  6 variables:
 $ City    : chr  "Atlanta " "Baltimore " "Boston " "Buffalo " ...
 $ Bread   : num  24.5 26.5 29.7 22.8 26.7 25.3 22.8 23.3 24.1 29.3 ...
 $ Burger  : num  94.5 91 100.8 86.6 86.7 ...
 $ Milk    : num  73.9 67.5 61.4 65.3 62.7 63.3 52.4 62.5 51.5 80.2 ...
 $ Oranges : num  80.1 74.6 104 118.4 105.9 ...
 $ Tomatoes: num  41.6 33.3 59.6 61.2 60.2 45.6 60.1 60.8 60.5 71.7 ...
```

7.1.1 EFA

```
library(corrplot)
corrplot(cor(harga[,2:6]), method="number")
```




```
library(psych)
KMO(harga[,2:6])
```

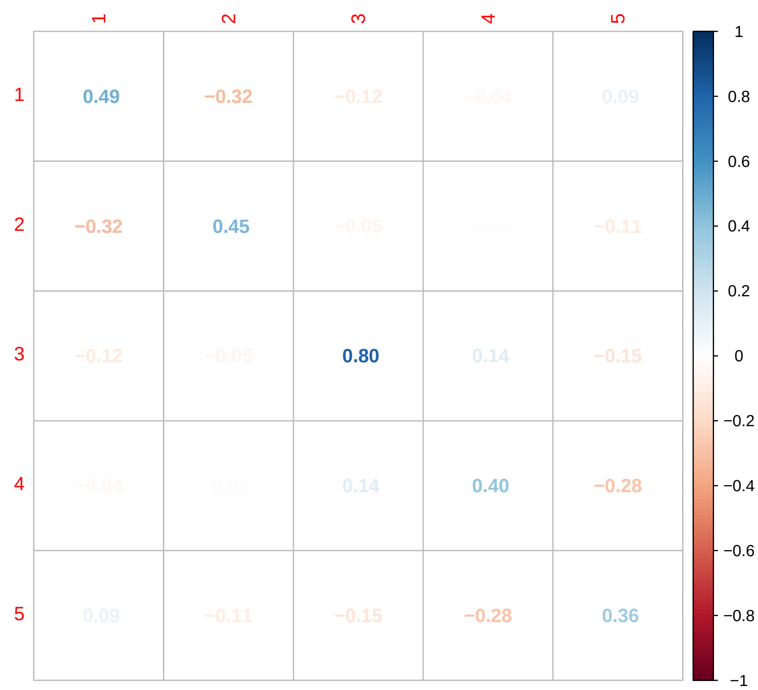
```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = harga[, 2:6])
Overall MSA =  0.52
MSA for each item =
   Bread   Burger     Milk  Oranges Tomatoes
    0.52     0.58     0.59     0.49     0.48
```

```
# Bartlett's Test of Sphericity
cortest.bartlett(harga[,2:6])
```

```
$chisq
[1] 36.46285

$p.value
[1] 7.006877e-05

$df
```

```
[1] 10
```

```
# Anti image correlation (AIC)
corrplot(KMO(harga[,2:6])$ImCo, method="number")
```



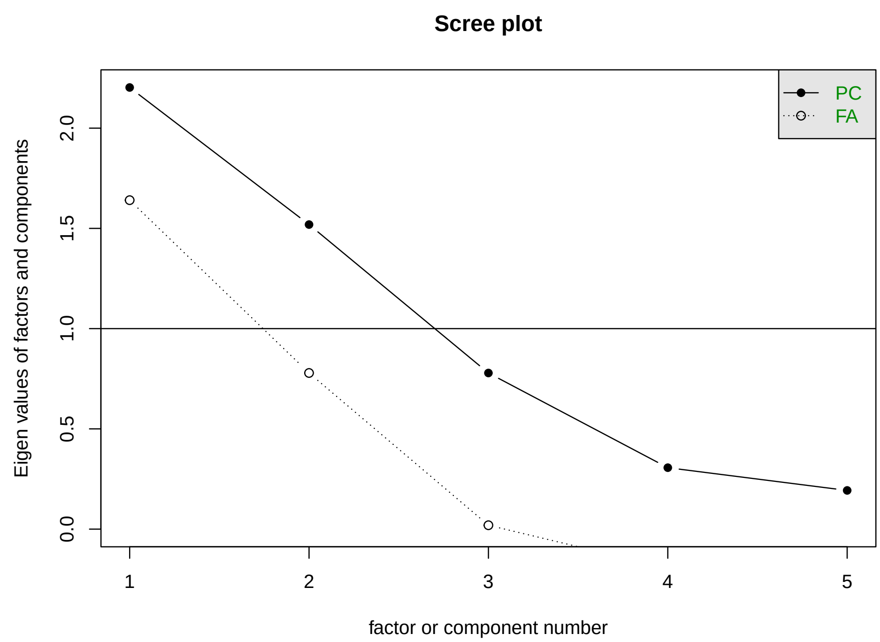

```
# Determinan positif
det(cor(harga[,2:6]))
```

```
[1] 0.1541406
```

```
# Principal component analysis (PCA)
pca1 = princomp(harga[,2:6], scores=TRUE, cor=TRUE)
summary(pca1)
```

```
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
Standard deviation     1.4841538 1.2325047 0.8824610 0.55357732 0.43935672
Proportion of Variance 0.4405425 0.3038136 0.1557475 0.06128957 0.03860687
Cumulative Proportion  0.4405425 0.7443561 0.9001036 0.96139313 1.00000000
```

```
scree(harga[,2:6])
```




```
# Menentukan faktor loading Analisis faktor loading
loadings(pca1)
```

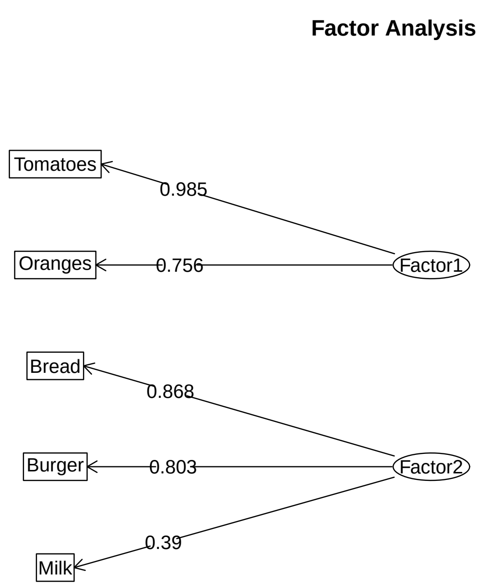
```
Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
Bread     0.436  0.484  0.354  0.597  0.306
Burger    0.542  0.292  0.307 -0.657 -0.309
Milk      0.346  0.308 -0.866        -0.163
Oranges   0.410 -0.579  0.108  0.399 -0.571
Tomatoes  0.478 -0.500 -0.137 -0.211  0.677

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
SS loadings       1.0    1.0    1.0    1.0    1.0
Proportion Var    0.2    0.2    0.2    0.2    0.2
Cumulative Var    0.2    0.4    0.6    0.8    1.0
```

```
# Rotasi untuk mengkonfirmasi hasil analisis loading
fa1 = factanal(harga[,2:6], factor=2, rotation="varimax")
fa1
```

```
Call:
factanal(x = harga[, 2:6], factors = 2, rotation = "varimax")
```

```
Uniquenesses:
   Bread   Burger     Milk  Oranges Tomatoes
   0.239    0.318    0.830    0.420    0.005

Loadings:
         Factor1 Factor2
Bread             0.868
Burger    0.195   0.803
Milk      0.135   0.390
Oranges   0.756
Tomatoes  0.985   0.157

               Factor1 Factor2
SS loadings      1.605   1.583
Proportion Var   0.321   0.317
Cumulative Var   0.321   0.638

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 1.16 on 1 degree of freedom.
The p-value is 0.282
```

```
# Diagram jalur hasil analisis EFA dan menampilkan faktor loading-nya
fa.diagram(fa1$loadings, digits = 3)
```



7.1.2 CFA

```
# Spesifikasi model
attach(harga)
model1 <- "
F1 =~ Tomatoes + Oranges
F2 =~ Bread + Burger + Milk
F1 ~~ F2 "
```

```
library(lavaan)
fitmod = cfa(model1, data = harga)
summary(fitmod, fit.measures = TRUE, standardized = TRUE)
```

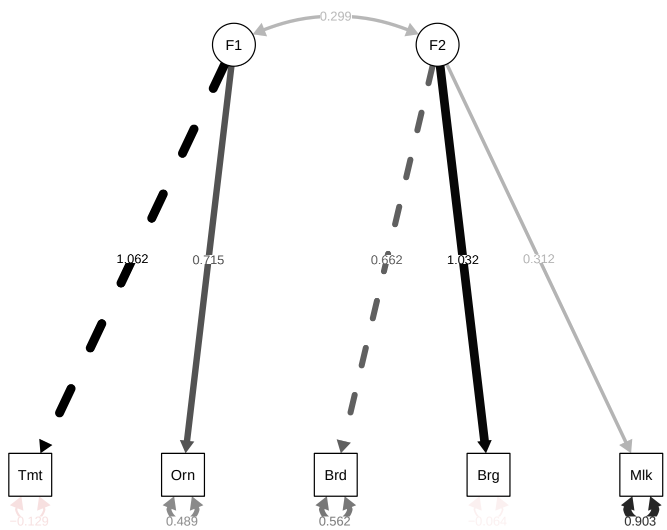
```
lavaan 0.6-19 ended normally after 85 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        11

  Number of observations                            23

Model Test User Model:

  Test statistic                                 3.642
  Degrees of freedom                                 4
  P-value (Chi-square)                           0.457

Model Test Baseline Model:

  Test statistic                                43.007
  Degrees of freedom                                10
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    1.000
  Tucker-Lewis Index (TLI)                       1.027

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)               -367.812
  Loglikelihood unrestricted model (H1)       -365.991

  Akaike (AIC)                                 757.623
  Bayesian (BIC)                               770.114
  Sample-size adjusted Bayesian (SABIC)        736.072
```

```
Root Mean Square Error of Approximation:

  RMSEA                                          0.000
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.302
  P-value H_0: RMSEA <= 0.050                    0.487
  P-value H_0: RMSEA >= 0.080                    0.469

Standardized Root Mean Square Residual:

  SRMR                                           0.065

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  F1 =~
    Tomatoes          1.000                               10.659    1.062
    Oranges           0.934    0.580    1.611    0.107    9.952    0.715
  F2 =~
    Bread             1.000                                1.622    0.662
    Burger            4.700    2.464    1.907    0.056    7.623    1.032
    Milk              1.307    0.858    1.523    0.128    2.119    0.312

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  F1 ~~
    F2                5.161    4.482    1.151    0.250    0.299    0.299

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Tomatoes        -12.966   66.742   -0.194    0.846  -12.966   -0.129
   .Oranges          94.906   64.476    1.472    0.141   94.906    0.489
   .Bread             3.381    1.581    2.138    0.033    3.381    0.562
   .Burger           -3.518   27.199   -0.129    0.897   -3.518   -0.064
   .Milk             41.714   12.439    3.354    0.001   41.714    0.903
    F1              113.605   72.842    1.560    0.119    1.000    1.000
    F2                2.631    1.912    1.376    0.169    1.000    1.000
```

```
fitmeasures(fitmod)
```

```
              npar                 fmin                chisq 
            11.000                0.079                3.642 
                df               pvalue       baseline.chisq 
             4.000                0.457               43.007 
       baseline.df      baseline.pvalue                  cfi 
            10.000                0.000                1.000 
               tli                 nnfi                  rfi 
             1.027                1.027                0.788 
               nfi                 pnfi                  ifi 
             0.915                0.366                1.009 
               rni                 logl    unrestricted.logl 
             1.011             -367.812             -365.991 
               aic                  bic               ntotal 
           757.623              770.114               23.000 
              bic2                rmsea       rmsea.ci.lower 
           736.072                0.000                0.000 
    rmsea.ci.upper       rmsea.ci.level         rmsea.pvalue 
             0.302                0.900                0.487 
    rmsea.close.h0 rmsea.notclose.pvalue    rmsea.notclose.h0 
             0.050                0.469                0.080 
               rmr           rmr_nomean                 srmr 
             2.823                2.823                0.065 
      srmr_bentler  srmr_bentler_nomean                 crmr 
             0.065                0.065                0.080 
       crmr_nomean           srmr_mplus    srmr_mplus_nomean 
             0.080                0.065                0.065 
             cn_05                cn_01                  gfi 
            60.915               84.843                0.947 
              agfi                 pgfi                  mfi 
             0.803                0.253                1.008 
              ecvi 
             1.115 
```

```r
library(semPlot)
semPaths(fitmod, what='std', layout='tree', title = TRUE,
         posCol = 1, nDigits = 3,
         edge.label.cex=0.7,
         exoVar = FALSE,
         sizeMan = 5,
         sizeLat = 5)
```




```r
# Estimasi Reliabilitas alpha cronbach
psych::alpha(harga[,2:6])
```

```
Reliability analysis
Call: psych::alpha(x = harga[, 2:6])

  raw_alpha std.alpha G6(smc) average_r S/N ase mean  sd median_r
      0.63      0.67    0.77      0.29 2.1 0.1   67 5.8     0.26

    95% confidence boundaries
         lower alpha upper
Feldt     0.32  0.63  0.82
Duhachek  0.42  0.63  0.83

 Reliability if an item is dropped:
         raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
Bread         0.64      0.63    0.68      0.30 1.7    0.110 0.065  0.26
Burger        0.56      0.54    0.63      0.23 1.2    0.107 0.083  0.13
Milk          0.64      0.68    0.78      0.34 2.1    0.091 0.096  0.26
Oranges       0.55      0.65    0.66      0.32 1.9    0.140 0.043  0.32
Tomatoes      0.37      0.59    0.61      0.26 1.4    0.197 0.062  0.27

 Item statistics
```

```
          n raw.r std.r r.cor r.drop mean   sd
Bread    23  0.38  0.64  0.56   0.30   25  2.5
Burger   23  0.62  0.77  0.72   0.41   92  7.6
Milk     23  0.42  0.56  0.36   0.20   62  7.0
Oranges  23  0.82  0.61  0.56   0.49  103 14.2
Tomatoes 23  0.86  0.71  0.68   0.71   52 10.3
```

7.2 Model Persamaan Struktural (SEM)

```
library(lavaan)
library(semPlot)
```

```
library(readxl)
datasem <- read_excel("Data/Datalikert.xlsx")
head(datasem[,1:5])
```

```
# A tibble: 6 x 5
  Perusahaan Provinsi   Pulau    A1    A2
       <dbl> <chr>      <chr> <dbl> <dbl>
1          1 Jawa Barat Jawa      4     5
2          2 Jawa Timur Jawa      5     5
3          3 Jawa Timur Jawa      4     4
4          4 Jawa Barat Jawa      4     4
5          5 Jawa Timur Jawa      4     4
6          6 Jawa Timur Jawa      4     4
```

```
str(datasem)
```

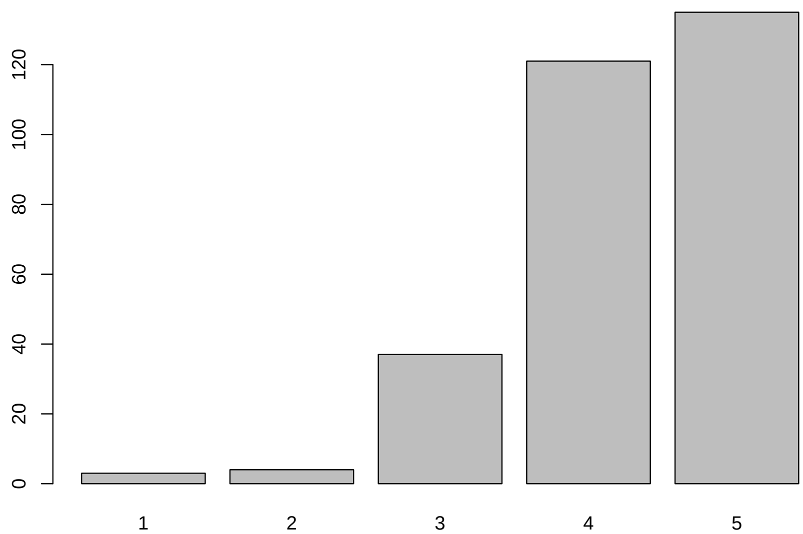
```
tibble [300 x 45] (S3: tbl_df/tbl/data.frame)
 $ Perusahaan: num [1:300] 1 2 3 4 5 6 7 8 9 10 ...
 $ Provinsi  : chr [1:300] "Jawa Barat" "Jawa Timur" "Jawa Timur" "Jawa Barat" ...
 $ Pulau     : chr [1:300] "Jawa" "Jawa" "Jawa" "Jawa" ...
 $ A1        : num [1:300] 4 5 4 4 4 4 4 5 4 5 ...
 $ A2        : num [1:300] 5 5 4 4 4 4 4 5 4 5 ...
 $ A3        : num [1:300] 5 5 4 3 4 5 4 5 3 5 ...
 $ A4        : num [1:300] 4 5 4 4 3 4 4 5 3 5 ...
 $ A5        : num [1:300] 4 4 4 4 4 4 4 5 3 5 ...
 $ A6        : num [1:300] 4 5 4 4 4 4 4 5 3 4 ...
 $ A7        : num [1:300] 5 5 5 4 4 4 4 5 3 5 ...
 $ A8        : num [1:300] 5 5 5 4 4 4 4 5 3 4 ...
 $ Atotal    : num [1:300] 36 39 34 31 31 33 32 40 26 38 ...
 $ B1        : num [1:300] 4 4 4 4 3 5 3 3 3 4 ...
 $ B2        : num [1:300] 4 4 4 3 4 4 3 3 2 4 ...
 $ Btotal    : num [1:300] 8 8 8 7 7 9 6 6 5 8 ...
 $ C1        : num [1:300] 4 4 4 4 4 4 4 5 3 4 ...
 $ C2        : num [1:300] 4 4 4 4 4 4 4 4 3 4 ...
```

```
 $ Ctotal    : num [1:300] 8 8 8 8 8 8 8 9 6 8 ...
 $ D1        : num [1:300] 4 5 4 4 4 4 4 4 3 4 ...
 $ D2        : num [1:300] 4 5 4 3 4 5 4 4 2 4 ...
 $ D3        : num [1:300] 4 5 4 4 4 4 4 4 3 4 ...
 $ D4        : num [1:300] 4 5 4 5 4 4 4 4 3 4 ...
 $ Dtotal    : num [1:300] 16 20 16 16 16 17 16 16 11 16 ...
 $ E1        : num [1:300] 5 5 4 4 4 4 4 4 3 5 ...
 $ E2        : num [1:300] 5 5 4 4 4 5 4 4 3 5 ...
 $ E3        : num [1:300] 5 5 4 4 4 5 4 5 4 5 ...
 $ E4        : num [1:300] 4 5 4 3 4 5 4 4 3 4 ...
 $ E5        : num [1:300] 4 5 4 4 3 5 4 4 3 4 ...
 $ E6        : num [1:300] 4 5 4 4 4 4 4 4 3 4 ...
 $ E7        : num [1:300] 4 5 4 4 4 5 4 4 3 4 ...
 $ E8        : num [1:300] 4 5 4 4 3 5 4 4 3 4 ...
 $ E9        : num [1:300] 4 5 4 4 4 4 4 4 3 4 ...
 $ E10       : num [1:300] 4 5 4 4 4 5 4 5 3 4 ...
 $ E11       : num [1:300] 4 5 4 3 3 5 4 5 3 4 ...
 $ E12       : num [1:300] 5 5 4 4 4 5 4 5 3 5 ...
 $ Etotal    : num [1:300] 52 60 48 46 45 57 48 52 37 52 ...
 $ F1        : num [1:300] 5 5 4 4 4 5 4 4 2 4 ...
 $ F2        : num [1:300] 4 5 4 4 4 5 4 4 3 3 ...
 $ F3        : num [1:300] 4 5 4 4 4 4 4 4 2 3 ...
 $ F4        : num [1:300] 4 5 4 4 4 5 4 5 3 4 ...
 $ F5        : num [1:300] 4 5 4 4 3 5 4 4 3 3 ...
 $ F6        : num [1:300] 4 5 4 4 3 4 4 5 3 4 ...
 $ F7        : num [1:300] 4 5 4 4 3 4 4 4 3 4 ...
 $ F8        : num [1:300] 4 5 4 4 4 5 4 4 3 4 ...
 $ Ftotal    : num [1:300] 33 40 32 32 29 37 32 34 22 29 ...
```

```r
attach(datasem)
table(A1)
```

```
A1
  1   2   3   4   5 
  3   4  37 121 135 
```

```r
barplot(table(A1))
```




```
# Spesifikasi Model
sem.model = "
faktor =~ A1 + A2 + A3 + A4
permintaan =~ B1 + B2
industri =~ C1 + C2
strategi =~ D1 + D2 + D3 + D4
regulasi =~ E1 + E2 + E3 + E4 + E5 + E6
kesempatan =~ F1 + F2 + F3 + F4
kesempatan ~ faktor + permintaan + industri + strategi + regulasi"
```

```
sem.fit = sem(sem.model, data = datasem)
summary(sem.fit, fit.measures=TRUE)
```

```
lavaan 0.6-19 ended normally after 90 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        59

  Number of observations                           300

Model Test User Model:

  Test statistic                               555.757
```

```
  Degrees of freedom                               194
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                              7355.210
  Degrees of freedom                               231
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.949
  Tucker-Lewis Index (TLI)                       0.940

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -4608.159
  Loglikelihood unrestricted model (H1)      -4330.280

  Akaike (AIC)                                9334.318
  Bayesian (BIC)                              9552.841
  Sample-size adjusted Bayesian (SABIC)       9365.728

Root Mean Square Error of Approximation:

  RMSEA                                          0.079
  90 Percent confidence interval - lower         0.071
  90 Percent confidence interval - upper         0.087
  P-value H_0: RMSEA <= 0.050                    0.000
  P-value H_0: RMSEA >= 0.080                    0.410

Standardized Root Mean Square Residual:

  SRMR                                           0.035

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  faktor =~
    A1                1.000
    A2                1.266    0.089   14.271    0.000
```

```
    A3                1.312    0.094   13.991    0.000
    A4                1.261    0.091   13.913    0.000
  permintaan =~
    B1                1.000
    B2                1.020    0.063   16.072    0.000
  industri =~
    C1                1.000
    C2                1.035    0.044   23.446    0.000
  strategi =~
    D1                1.000
    D2                0.973    0.033   29.472    0.000
    D3                0.972    0.043   22.590    0.000
    D4                0.817    0.042   19.325    0.000
  regulasi =~
    E1                1.000
    E2                0.929    0.039   23.666    0.000
    E3                0.950    0.043   22.088    0.000
    E4                1.015    0.039   25.697    0.000
    E5                0.985    0.042   23.464    0.000
    E6                0.913    0.045   20.186    0.000
  kesempatan =~
    F1                1.000
    F2                1.006    0.038   26.712    0.000
    F3                1.033    0.042   24.672    0.000
    F4                0.943    0.046   20.414    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  kesempatan ~
    faktor            0.016    0.111    0.146    0.884
    permintaan        0.042    0.059    0.705    0.481
    industri          0.129    0.133    0.976    0.329
    strategi          0.131    0.091    1.449    0.147
    regulasi          0.685    0.077    8.860    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  faktor ~~
    permintaan        0.233    0.034    6.785    0.000
    industri          0.327    0.037    8.729    0.000
    strategi          0.292    0.035    8.242    0.000
    regulasi          0.343    0.039    8.730    0.000
  permintaan ~~
    industri          0.366    0.043    8.447    0.000
    strategi          0.391    0.045    8.713    0.000
    regulasi          0.332    0.043    7.797    0.000
```

```
  industri ~~
    strategi          0.437    0.043   10.274    0.000
    regulasi          0.416    0.043    9.764    0.000
  strategi ~~
    regulasi          0.405    0.042    9.580    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .A1                0.323    0.029   11.229    0.000
   .A2                0.161    0.018    8.902    0.000
   .A3                0.205    0.022    9.430    0.000
   .A4                0.198    0.021    9.552    0.000
   .B1                0.269    0.032    8.457    0.000
   .B2                0.078    0.025    3.161    0.002
   .C1                0.122    0.014    8.515    0.000
   .C2                0.106    0.014    7.549    0.000
   .D1                0.093    0.011    8.749    0.000
   .D2                0.063    0.008    7.476    0.000
   .D3                0.182    0.017   10.625    0.000
   .D4                0.200    0.018   11.219    0.000
   .E1                0.145    0.014   10.563    0.000
   .E2                0.114    0.011   10.395    0.000
   .E3                0.156    0.014   10.845    0.000
   .E4                0.091    0.010    9.488    0.000
   .E5                0.133    0.013   10.462    0.000
   .E6                0.198    0.018   11.224    0.000
   .F1                0.139    0.014    9.697    0.000
   .F2                0.090    0.011    8.221    0.000
   .F3                0.140    0.015    9.540    0.000
   .F4                0.233    0.021   10.912    0.000
    faktor            0.321    0.047    6.841    0.000
    permintaan        0.525    0.065    8.048    0.000
    industri          0.480    0.049    9.751    0.000
    strategi          0.522    0.050   10.406    0.000
    regulasi          0.542    0.055    9.811    0.000
   .kesempatan        0.122    0.015    8.068    0.000
```

```r
sem.fit = sem(sem.model, data = datasem, std.lv=TRUE)
summary(sem.fit, fit.measures=TRUE, standardized=TRUE)
```

```
lavaan 0.6-19 ended normally after 90 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        59
```

```
  Number of observations                           300

Model Test User Model:

  Test statistic                               555.757
  Degrees of freedom                               194
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                              7355.210
  Degrees of freedom                               231
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.949
  Tucker-Lewis Index (TLI)                       0.940

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -4608.159
  Loglikelihood unrestricted model (H1)      -4330.280

  Akaike (AIC)                                9334.318
  Bayesian (BIC)                              9552.841
  Sample-size adjusted Bayesian (SABIC)       9365.728

Root Mean Square Error of Approximation:

  RMSEA                                          0.079
  90 Percent confidence interval - lower         0.071
  90 Percent confidence interval - upper         0.087
  P-value H_0: RMSEA <= 0.050                    0.000
  P-value H_0: RMSEA >= 0.080                    0.410

Standardized Root Mean Square Residual:

  SRMR                                           0.035

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  faktor =~
    A1                0.566    0.041   13.681    0.000    0.566    0.706
    A2                0.717    0.038   18.699    0.000    0.717    0.872
    A3                0.743    0.041   18.064    0.000    0.743    0.854
    A4                0.714    0.040   17.894    0.000    0.714    0.849
  permintaan =~
    B1                0.725    0.045   16.097    0.000    0.725    0.813
    B2                0.739    0.038   19.509    0.000    0.739    0.935
  industri =~
    C1                0.692    0.036   19.503    0.000    0.692    0.893
    C2                0.717    0.036   20.132    0.000    0.717    0.911
  strategi =~
    D1                0.723    0.035   20.812    0.000    0.723    0.922
    D2                0.703    0.033   21.615    0.000    0.703    0.941
    D3                0.702    0.038   18.344    0.000    0.702    0.855
    D4                0.590    0.036   16.459    0.000    0.590    0.797
  regulasi =~
    E1                0.736    0.038   19.623    0.000    0.736    0.888
    E2                0.684    0.034   19.941    0.000    0.684    0.897
    E3                0.699    0.037   18.967    0.000    0.699    0.870
    E4                0.747    0.035   21.120    0.000    0.747    0.927
    E5                0.725    0.037   19.819    0.000    0.725    0.894
    E6                0.673    0.038   17.720    0.000    0.673    0.834
  kesempatan =~
    F1                0.350    0.022   16.135    0.000    0.771    0.900
    F2                0.352    0.021   16.722    0.000    0.776    0.933
    F3                0.361    0.022   16.227    0.000    0.796    0.905
    F4                0.330    0.022   14.833    0.000    0.727    0.833

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  kesempatan ~
    faktor            0.026    0.180    0.146    0.884    0.012    0.012
    permintaan        0.086    0.123    0.705    0.481    0.039    0.039
    industri          0.256    0.263    0.973    0.331    0.116    0.116
    strategi          0.272    0.188    1.447    0.148    0.123    0.123
    regulasi          1.443    0.190    7.608    0.000    0.654    0.654

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  faktor ~~
    permintaan        0.568    0.046   12.297    0.000    0.568    0.568
    industri          0.833    0.025   33.258    0.000    0.833    0.833
    strategi          0.715    0.033   21.548    0.000    0.715    0.715
```

```
    regulasi          0.822    0.023   35.175    0.000    0.822    0.822
  permintaan ~~
    industri          0.729    0.035   20.610    0.000    0.729    0.729
    strategi          0.746    0.032   23.194    0.000    0.746    0.746
    regulasi          0.623    0.041   15.291    0.000    0.623    0.623
  industri ~~
    strategi          0.874    0.020   44.744    0.000    0.874    0.874
    regulasi          0.816    0.024   33.446    0.000    0.816    0.816
  strategi ~~
    regulasi          0.762    0.027   27.976    0.000    0.762    0.762

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .A1                0.323    0.029   11.229    0.000    0.323    0.502
   .A2                0.161    0.018    8.902    0.000    0.161    0.239
   .A3                0.205    0.022    9.430    0.000    0.205    0.271
   .A4                0.198    0.021    9.552    0.000    0.198    0.280
   .B1                0.269    0.032    8.457    0.000    0.269    0.339
   .B2                0.078    0.025    3.161    0.002    0.078    0.126
   .C1                0.122    0.014    8.515    0.000    0.122    0.203
   .C2                0.106    0.014    7.549    0.000    0.106    0.171
   .D1                0.093    0.011    8.749    0.000    0.093    0.151
   .D2                0.063    0.008    7.476    0.000    0.063    0.114
   .D3                0.182    0.017   10.625    0.000    0.182    0.270
   .D4                0.200    0.018   11.219    0.000    0.200    0.365
   .E1                0.145    0.014   10.563    0.000    0.145    0.211
   .E2                0.114    0.011   10.395    0.000    0.114    0.195
   .E3                0.156    0.014   10.845    0.000    0.156    0.242
   .E4                0.091    0.010    9.488    0.000    0.091    0.141
   .E5                0.133    0.013   10.462    0.000    0.133    0.201
   .E6                0.198    0.018   11.224    0.000    0.198    0.304
   .F1                0.139    0.014    9.697    0.000    0.139    0.190
   .F2                0.090    0.011    8.221    0.000    0.090    0.130
   .F3                0.140    0.015    9.540    0.000    0.140    0.181
   .F4                0.233    0.021   10.912    0.000    0.233    0.306
    faktor            1.000                               1.000    1.000
    permintaan        1.000                               1.000    1.000
    industri          1.000                               1.000    1.000
    strategi          1.000                               1.000    1.000
    regulasi          1.000                               1.000    1.000
   .kesempatan        1.000                               0.206    0.206
```

```
#sem.fit = sem(sem.model, data = datasem, std.lv=TRUE, orthogonal=TRUE)
#summary(sem.fit, fit.measures=TRUE, standardized=TRUE)
```

```r
# Modification Indices
modificationIndices(sem.fit, minimum.value = 10)
```

```
           lhs op rhs     mi    epc sepc.lv sepc.all sepc.nox
72      faktor =~  D3 10.792  0.143   0.143    0.174    0.174
82      faktor =~  F3 14.022 -0.170  -0.170   -0.193   -0.193
99  permintaan =~  E6 13.919  0.142   0.142    0.176    0.176
112   industri =~  D3 19.393  0.315   0.315    0.383    0.383
134   strategi =~  E3 11.975 -0.144  -0.144   -0.179   -0.179
152   regulasi =~  D3 18.808  0.197   0.197    0.240    0.240
157   regulasi =~  F4 13.142  0.272   0.272    0.312    0.312
168 kesempatan =~  D3 22.896  0.100   0.220    0.268    0.268
175 kesempatan =~  E6 25.214  0.153   0.337    0.418    0.418
176         A1 ~~  A2 15.863  0.068   0.068    0.298    0.298
270         B1 ~~  F4 14.265  0.063   0.063    0.253    0.253
317         D1 ~~  D3 10.752 -0.035  -0.035   -0.272   -0.272
331         D2 ~~  E1 11.098  0.025   0.025    0.257    0.257
347         D3 ~~  E6 12.029  0.042   0.042    0.223    0.223
351         D3 ~~  F4 10.217 -0.043  -0.043   -0.208   -0.208
352         D4 ~~  E1 11.953 -0.038  -0.038   -0.223   -0.223
362         E1 ~~  E2 17.329  0.038   0.038    0.294    0.294
363         E1 ~~  E3 10.360  0.033   0.033    0.220    0.220
364         E1 ~~  E4 12.186 -0.031  -0.031   -0.266   -0.266
371         E2 ~~  E3 11.663  0.032   0.032    0.236    0.236
373         E2 ~~  E5 10.449 -0.028  -0.028   -0.231   -0.231
381         E3 ~~  E6 11.439 -0.039  -0.039   -0.221   -0.221
386         E4 ~~  E5 25.380  0.043   0.043    0.388    0.388
398         E6 ~~  F2 14.478 -0.037  -0.037   -0.275   -0.275
399         E6 ~~  F3 20.998  0.052   0.052    0.310    0.310
405         F2 ~~  F4 24.019 -0.058  -0.058   -0.404   -0.404
406         F3 ~~  F4 14.294  0.050   0.050    0.279    0.279
```

```r
sem.model2 = "
faktor =~ A1 + A2 + A3 + A4
permintaan =~ B1 + B2
industri =~ C1 + C2
strategi =~ D1 + D2 + D3 + D4
regulasi =~ E1 + E2 + E3 + E4 + E5 + E6
kesempatan =~ F1 + F2 + F3 + F4
kesempatan ~ faktor + permintaan + industri + strategi + regulasi
A1  ~~  A2
"
```

```r
sem.fit = sem(sem.model2, data = datasem, std.lv=TRUE)
summary(sem.fit, fit.measures=TRUE, standardized=TRUE)
```

```
lavaan 0.6-19 ended normally after 94 iterations
```

```
  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        60

  Number of observations                           300

Model Test User Model:

  Test statistic                               540.535
  Degrees of freedom                               193
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                              7355.210
  Degrees of freedom                               231
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.951
  Tucker-Lewis Index (TLI)                       0.942

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -4600.548
  Loglikelihood unrestricted model (H1)      -4330.280

  Akaike (AIC)                                9321.095
  Bayesian (BIC)                              9543.322
  Sample-size adjusted Bayesian (SABIC)       9353.038

Root Mean Square Error of Approximation:

  RMSEA                                          0.077
  90 Percent confidence interval - lower         0.070
  90 Percent confidence interval - upper         0.085
  P-value H_0: RMSEA <= 0.050                    0.000
  P-value H_0: RMSEA >= 0.080                    0.303

Standardized Root Mean Square Residual:

  SRMR                                           0.035

Parameter Estimates:
```

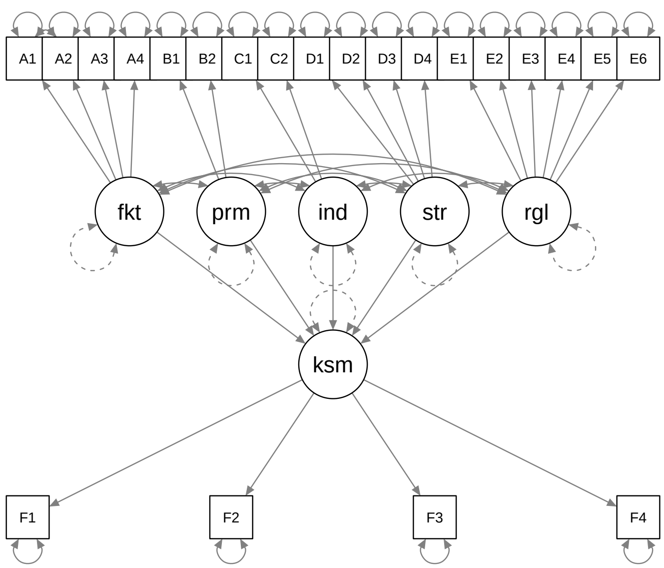
```
  Standard errors                               Standard
  Information                                   Expected
  Information saturated (h1) model            Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  faktor =~
    A1                0.539    0.043   12.660    0.000    0.539    0.672
    A2                0.702    0.039   18.009    0.000    0.702    0.854
    A3                0.752    0.041   18.363    0.000    0.752    0.864
    A4                0.720    0.040   18.060    0.000    0.720    0.855
  permintaan =~
    B1                0.724    0.045   16.093    0.000    0.724    0.813
    B2                0.739    0.038   19.507    0.000    0.739    0.935
  industri =~
    C1                0.692    0.036   19.469    0.000    0.692    0.892
    C2                0.717    0.036   20.171    0.000    0.717    0.912
  strategi =~
    D1                0.723    0.035   20.813    0.000    0.723    0.922
    D2                0.703    0.033   21.613    0.000    0.703    0.941
    D3                0.702    0.038   18.345    0.000    0.702    0.855
    D4                0.590    0.036   16.460    0.000    0.590    0.797
  regulasi =~
    E1                0.736    0.038   19.615    0.000    0.736    0.888
    E2                0.684    0.034   19.943    0.000    0.684    0.897
    E3                0.699    0.037   18.964    0.000    0.699    0.870
    E4                0.747    0.035   21.115    0.000    0.747    0.927
    E5                0.726    0.037   19.826    0.000    0.726    0.894
    E6                0.673    0.038   17.728    0.000    0.673    0.834
  kesempatan =~
    F1                0.350    0.022   16.137    0.000    0.771    0.900
    F2                0.352    0.021   16.726    0.000    0.776    0.933
    F3                0.361    0.022   16.232    0.000    0.796    0.905
    F4                0.330    0.022   14.836    0.000    0.727    0.833

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  kesempatan ~
    faktor            0.031    0.186    0.167    0.867    0.014    0.014
    permintaan        0.087    0.122    0.709    0.478    0.039    0.039
    industri          0.253    0.267    0.947    0.344    0.115    0.115
    strategi          0.272    0.189    1.442    0.149    0.123    0.123
    regulasi          1.441    0.190    7.578    0.000    0.654    0.654

Covariances:
```

```
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
 .A1 ~~
   .A2                0.068    0.019    3.588    0.000    0.068    0.269
  faktor ~~
    permintaan        0.573    0.046   12.417    0.000    0.573    0.573
    industri          0.837    0.025   33.458    0.000    0.837    0.837
    strategi          0.716    0.033   21.421    0.000    0.716    0.716
    regulasi          0.824    0.024   34.919    0.000    0.824    0.824
  permintaan ~~
    industri          0.729    0.035   20.581    0.000    0.729    0.729
    strategi          0.746    0.032   23.189    0.000    0.746    0.746
    regulasi          0.623    0.041   15.292    0.000    0.623    0.623
  industri ~~
    strategi          0.874    0.020   44.757    0.000    0.874    0.874
    regulasi          0.816    0.024   33.429    0.000    0.816    0.816
  strategi ~~
    regulasi          0.762    0.027   27.982    0.000    0.762    0.762

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .A1                0.353    0.032   11.133    0.000    0.353    0.549
   .A2                0.182    0.020    9.132    0.000    0.182    0.270
   .A3                0.192    0.022    8.905    0.000    0.192    0.253
   .A4                0.190    0.021    9.171    0.000    0.190    0.268
   .B1                0.270    0.032    8.454    0.000    0.270    0.339
   .B2                0.078    0.025    3.155    0.002    0.078    0.125
   .C1                0.123    0.014    8.573    0.000    0.123    0.205
   .C2                0.104    0.014    7.494    0.000    0.104    0.169
   .D1                0.093    0.011    8.748    0.000    0.093    0.151
   .D2                0.063    0.008    7.481    0.000    0.063    0.114
   .D3                0.182    0.017   10.624    0.000    0.182    0.270
   .D4                0.200    0.018   11.218    0.000    0.200    0.365
   .E1                0.145    0.014   10.565    0.000    0.145    0.211
   .E2                0.114    0.011   10.392    0.000    0.114    0.195
   .E3                0.157    0.014   10.844    0.000    0.157    0.242
   .E4                0.092    0.010    9.490    0.000    0.092    0.141
   .E5                0.132    0.013   10.456    0.000    0.132    0.201
   .E6                0.197    0.018   11.222    0.000    0.197    0.304
   .F1                0.140    0.014    9.700    0.000    0.140    0.190
   .F2                0.090    0.011    8.219    0.000    0.090    0.130
   .F3                0.140    0.015    9.538    0.000    0.140    0.181
   .F4                0.233    0.021   10.912    0.000    0.233    0.306
    faktor            1.000                               1.000    1.000
    permintaan        1.000                               1.000    1.000
    industri          1.000                               1.000    1.000
    strategi          1.000                               1.000    1.000
```

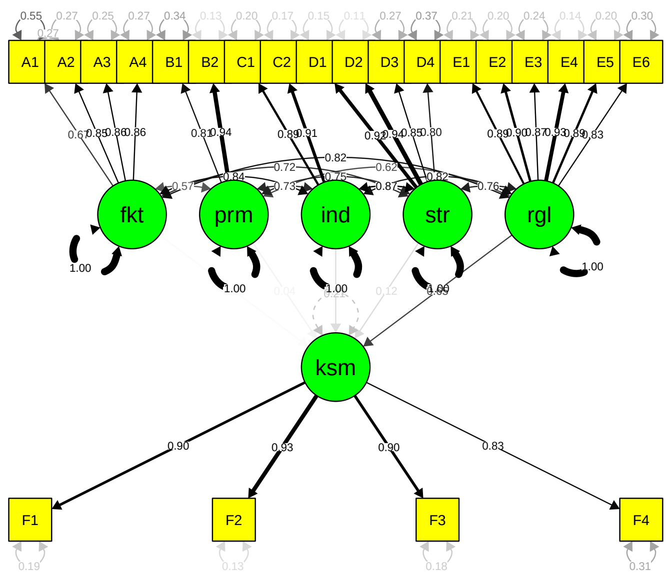
```
   regulasi          1.000                              1.000    1.000
  .kesempatan        1.000                              0.206    0.206
```

```r
semPaths(sem.fit)
```

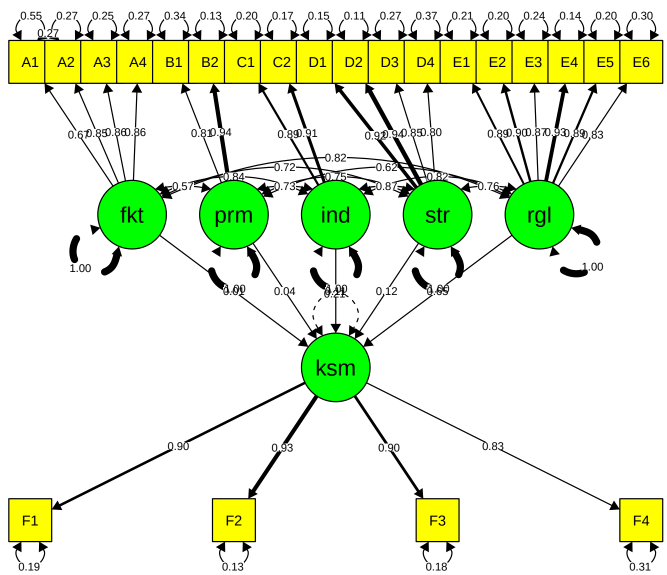
7.2.1 Visualisasi SEM



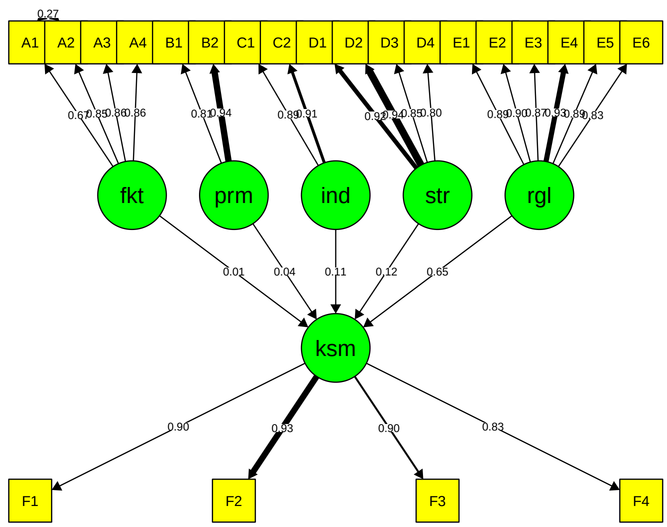

```r
semPaths(sem.fit, "std",
         color = list(lat = "green", man = "yellow"),
         edge.color="black")
```




```r
semPaths(sem.fit, "std",
         color = list(lat = "green", man = "yellow"),
         edge.color="black", fade=FALSE)
```




```r
semPaths(sem.fit, "std",
         color = list(lat = "green", man = "yellow"),
         edge.color="black",
         fade=FALSE, residuals=FALSE, exoCov=FALSE)
```



7.3 PLS SEM

```
# source:https://rpubs.com/ifn1411/PLS
# install plspm
#install.packages("plspm")
# load plspm
library(plspm)
```

```
# load data spainmodel
data(spainfoot)
# first 5 row of spainmodel data
head(spainfoot)
```

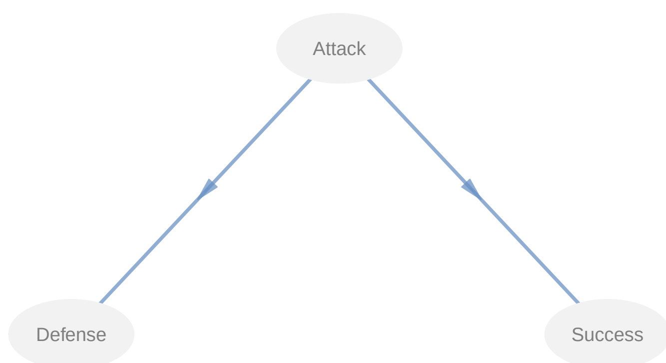
```
           GSH GSA  SSH  SSA GCH GCA  CSH  CSA WMH WMA LWR LRWL  YC RC
Barcelona   61  44 0.95 0.95  14  21 0.47 0.32  14  13  10   22  76  6
RealMadrid  49  34 1.00 0.84  29  23 0.37 0.37  14  11  10   18 115  9
Sevilla     28  26 0.74 0.74  20  19 0.42 0.53  11  10   4    7 100  8
AtleMadrid  47  33 0.95 0.84  23  34 0.37 0.16  13   7   6    9 116  5
Villarreal  33  28 0.84 0.68  25  29 0.26 0.16  12   6   5   11 102  5
Valencia    47  21 1.00 0.68  26  28 0.26 0.26  12   6   5    8 120  6
```

```
Attack <-  c(0, 0, 0)
Defense <- c(1, 0, 0)
```

```
Success <- c(1, 0, 0)

model_path <- rbind(Attack, Defense, Success)
colnames(model_path) <- rownames(model_path)

model_path
```

```
        Attack Defense Success
Attack       0       0       0
Defense      1       0       0
Success      1       0       0
```

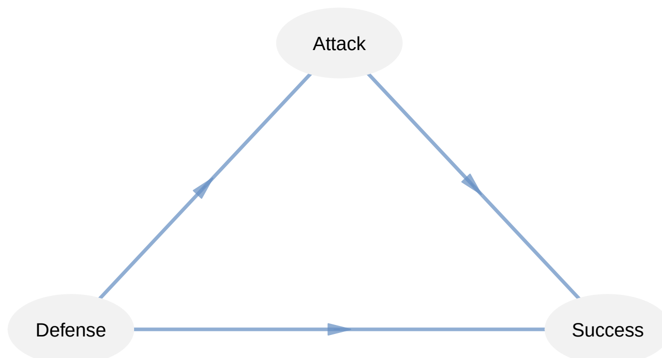
```
# graph structural model
innerplot(model_path)
```




```
Attack <-  c(0, 1, 0)
Defense <- c(0, 0, 0)
Success <- c(1, 1, 0)

model_path2 <- rbind(Attack, Defense, Success)
colnames(model_path2) <- rownames(model_path2)

model_path2
```

```
        Attack Defense Success
Attack       0       1       0
Defense      0       0       0
Success      1       1       0
```

```
# graph structural model
innerplot(model_path2, txt.col = "black")
```




```
# define latent variable associated with
model_blocks <- list(1:4, 5:8, 9:12)

# vector of modes (reflective)
model_modes <- c("A", "A", "A")

# run plspm analysis
model_pls <- plspm(Data = spainfoot, path_matrix = model_path, blocks = model_blocks, m

model_pls
```

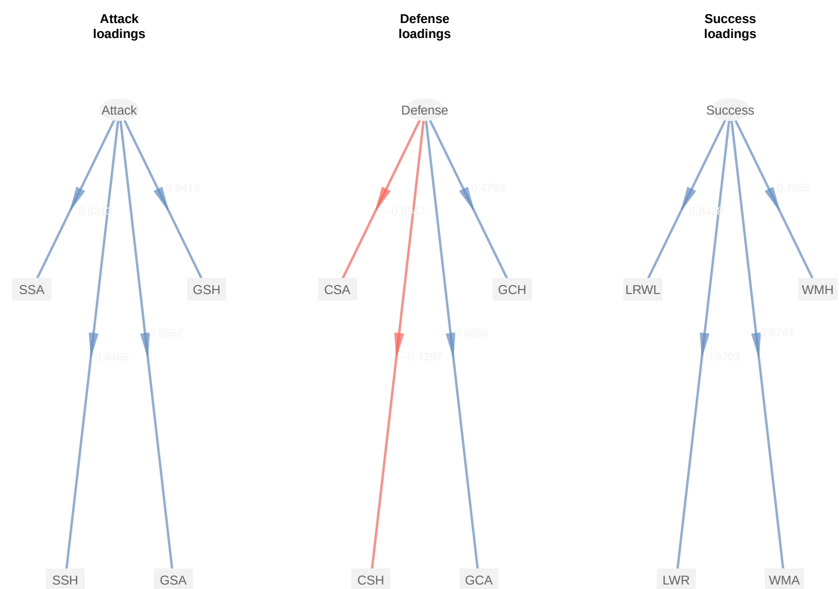
```
Partial Least Squares Path Modeling (PLS-PM)
---------------------------------------------
   NAME             DESCRIPTION
1  $outer_model     outer model
2  $inner_model     inner model
```

```
3  $path_coefs      path coefficients matrix
4  $scores          latent variable scores
5  $crossloadings   cross-loadings
6  $inner_summary   summary inner model
7  $effects         total effects
8  $unidim          unidimensionality
9  $gof             goodness-of-fit
10 $boot            bootstrap results
11 $data            data matrix
---------------------------------------------
You can also use the function 'summary'
```

```r
# Unidimensionality
model_pls$unidim
```

```
        Mode MVs   C.alpha     DG.rho  eig.1st   eig.2nd
Attack     A   4 0.8905919 0.92456079 3.017160 0.7923055
Defense    A   4 0.0000000 0.02601677 2.393442 1.1752781
Success    A   4 0.9165491 0.94232868 3.217294 0.5370492
```

```r
plot(model_pls, what = "loadings")
```




```r
# Loadings and Communilaties
model_pls$outer_model
```

```
   name   block     weight    loading communality redundancy
1   GSH  Attack  0.3474771  0.9412506   0.8859527 0.00000000
2   GSA  Attack  0.2671782  0.8562398   0.7331465 0.00000000
3   SSH  Attack  0.2922077  0.8466039   0.7167381 0.00000000
4   SSA  Attack  0.2396012  0.8212987   0.6745316 0.00000000
5   GCH Defense -0.1198790  0.4762965   0.2268583 0.05071506
6   GCA Defense -0.4264164  0.8885714   0.7895590 0.17650898
7   CSH Defense  0.2949470 -0.7297095   0.5324759 0.11903706
8   CSA Defense  0.3898039 -0.8947452   0.8005689 0.17897028
9   WMH Success  0.2484276  0.7884562   0.6216632 0.49452090
10  WMA Success  0.2691511  0.8747163   0.7651285 0.60864477
11  LWR Success  0.2947322  0.9703409   0.9415614 0.74899365
12 LRWL Success  0.2998524  0.9428112   0.8888929 0.70709694
```

```r
# Crossloadings
model_pls$crossloadings
```

```
   name   block     Attack    Defense    Success
1   GSH  Attack  0.9412506 -0.5139001  0.9019257
2   GSA  Attack  0.8562398 -0.3403294  0.7483558
3   SSH  Attack  0.8466039 -0.4124617  0.7781795
4   SSA  Attack  0.8212987 -0.3455460  0.6308989
5   GCH Defense -0.1302683  0.4762965 -0.1620567
6   GCA Defense -0.4633220  0.8885714 -0.5640722
7   CSH Defense  0.3204993 -0.7297095  0.4850456
8   CSA Defense  0.4235465 -0.8947452  0.5811253
9   WMH Success  0.7126127 -0.4120502  0.7884562
10  WMA Success  0.7720228 -0.7147787  0.8747163
11  LWR Success  0.8454164 -0.5345709  0.9703409
12 LRWL Success  0.8600973 -0.5910943  0.9428112
```

```r
# Coefficient of Determination
model_pls$inner_model
```

```
$Defense
               Estimate Std. Error       t value   Pr(>|t|)
Intercept  5.504973e-17  0.2076918  2.650549e-16 1.00000000
Attack    -4.728148e-01  0.2076918 -2.276521e+00 0.03526176

$Success
              Estimate Std. Error      t value     Pr(>|t|)
Intercept 7.783183e-17  0.1065936 7.301735e-16 1.000000e+00
Attack    8.918971e-01  0.1065936 8.367266e+00 1.285711e-07
```

```r
# Redundancy
model_pls$inner_summary
```

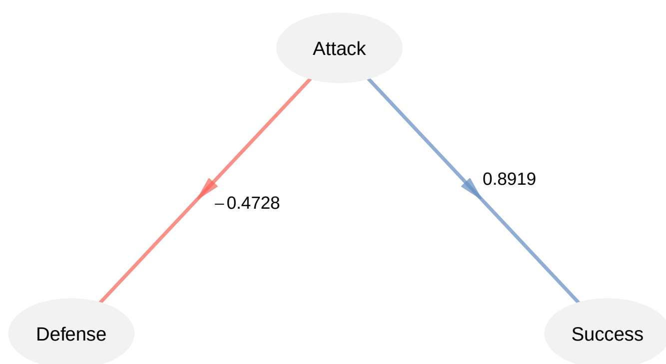
```
              Type        R2 Block_Communality Mean_Redundancy       AVE
```

```
Attack   Exogenous 0.0000000         0.7525922       0.0000000 0.7525922
Defense Endogenous 0.2235539         0.5873656       0.1313078 0.5873656
Success Endogenous 0.7954804         0.8043115       0.6398141 0.8043115
```

```
# Goodness-of-fit
model_pls$gof
```

```
[1] 0.6034738
```

```
plot(model_pls, what = "inner", colpos = "#6890c4BB", colneg = "#f9675dBB", txt.col = "black", ar
```



Chapter 8

Analytic Hierarchy Process (AHP)

8.1 Prosedur Pengolahan AHP

8.1.1 Data

```
ahpdata <- read.csv("Data/ahp.csv")
ahpdata
```

```
  Responden SAL_QL SAL_IW SAL_LC QL_IW QL_LC IW_LC J1_J2 J1_j3 J2_J3 J1_J2.1
1         1     -5     -2     -4     2     2    -2    -2    -4    -2       2
2         2     -7     -3     -3     3     3    -4    -3    -7    -1       3
  J1_j3.1 J2_J3.1 J1_J2.2 J1_j3.2 J2_J3.2 J1_J2.3 J1_j3.3 J2_J3.3
1       3       3       7       3      -3       4       7      -2
2       3       3       4      -1      -3       5       2      -4
```

8.1.2 Analisis

8.1.2.1 Faktor

```
# Mendefinisikan faktor
faktor <- c("SAL", "QL", "IW", "LC")
# Menampilkan data frame
faktor_data <- ahpdata[, 2:7]
faktor_data
```

```
  SAL_QL SAL_IW SAL_LC QL_IW QL_LC IW_LC
1     -5     -2     -4     2     2    -2
2     -7     -3     -3     3     3    -4
```

```
# install.packages("ahpsurvey")
library(ahpsurvey)
faktor_data_mat <- ahp.mat(df = faktor_data, faktor,
                           negconvert = TRUE)
faktor_data_mat
```

```
[[1]]
     SAL QL  IW  LC
SAL 1.00  5 2.0 4.0
QL  0.20  1 0.5 0.5
IW  0.50  2 1.0 2.0
LC  0.25  2 0.5 1.0

[[2]]
          SAL QL        IW        LC
SAL 1.0000000  7 3.0000000 3.0000000
QL  0.1428571  1 0.3333333 0.3333333
IW  0.3333333  3 1.0000000 4.0000000
LC  0.3333333  3 0.2500000 1.0000000
```

```
# Consistency
ri <- ahp.ri(nsims = 10000, dim = 4, seed = 42)
ahp.cr(faktor_data_mat, faktor, ri)
```

```
[1] 0.01780548 0.09677931
```

```
#Treatement Consistency (Jika Tidak Konsisten)
#faktor_data_mat <- ahp.harker(faktor_data_mat, faktor, iterations = 10, stopcr = 0.1)
#ahp.cr(faktor_data_mat, faktor)
```

The ahp.cr function calculates the consistency ratio of each decision-maker, defined by the following equation:

CR = (− n)/((n − 1)(RI))

Where is the maximum eigenvalue of the pairwise comparison matrix, n is the number of attributes, and RI is the random index. Following Saaty and Tran (2007), the RI is a function of n and is the consistency ratio of randomly generated pairwise comparison matrices.

Saaty showed that when the CR is higher than 0.1, the choice is deemed to be inconsistent

8.1.2.2 Individual Rangking Faktor

```
library(tidyverse)
library(tibble)
faktor_ind <- ahp.indpref(faktor_data_mat,
```

```
                          faktor,
                          method = "arithmetic")
round(faktor_ind, 3) %>% rownames_to_column('ID')
```

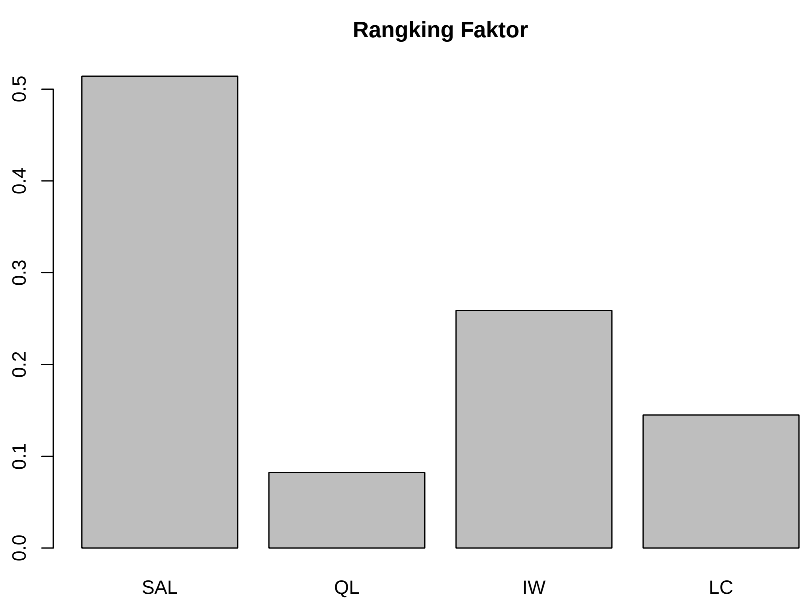
```
  ID   SAL    QL    IW    LC
1  1 0.512 0.099 0.243 0.147
2  2 0.517 0.066 0.274 0.143
```

8.1.2.3 Aggregate Rangking Faktor

```
faktor_agg <- ahp.aggpref(faktor_data_mat,
                          faktor,
                          method = "arithmetic",
                          aggmethod = "arithmetic")
round(faktor_agg, 3) %>% t()
```

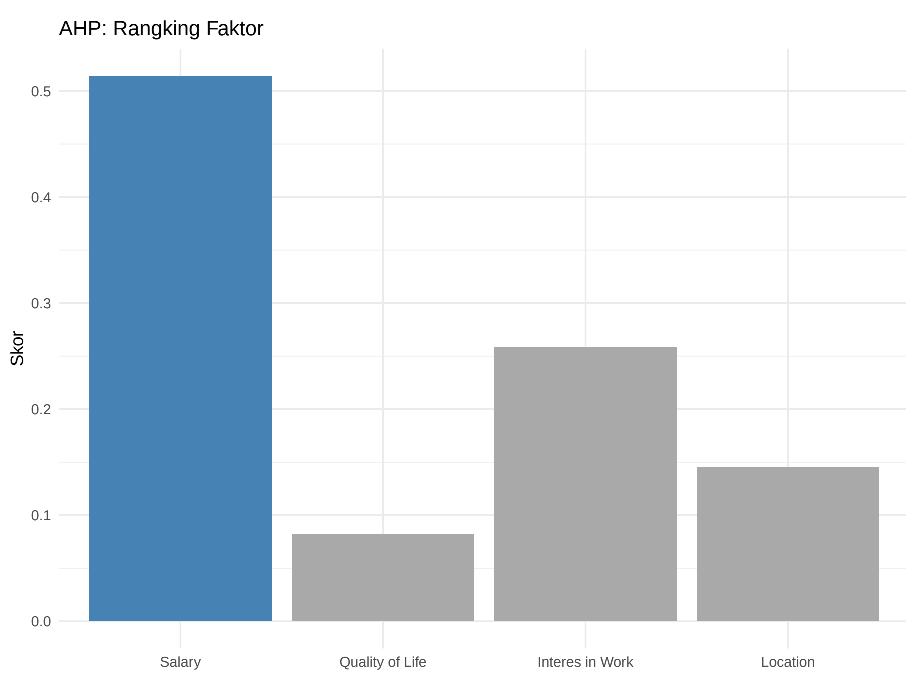
```
       SAL    QL    IW    LC
[1,] 0.514 0.082 0.259 0.145
```

```
barplot(faktor_agg,main="Rangking Faktor")
```




```
library(ggplot2)
# Mengubah Cat menjadi factor dengan label yang diinginkan
data = data.frame("Cat"=row.names(data.frame(faktor_agg)),
```

```
                  data.frame(faktor_agg))
data$Cat <- factor(data$Cat,
                   levels = c("SAL", "QL", "IW", "LC"),
                   labels = c("Salary", "Quality of Life",
                              "Interes in Work", "Location"))
# Mengurutkan
data$warna <- ifelse(data$faktor_agg ==
                       max(data$faktor_agg),
                     "terbesar", "lainnya")
# Buat grafik batang
ggplot(data, aes(x = Cat,
                 y = faktor_agg,
                 fill = warna)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("terbesar" = "#4682B4",
                               "lainnya" = "#A9A9A9")) +
  theme_minimal() +
  theme(legend.position = "none") +  # Sembunyikan legenda
  labs(
    title = "AHP: Rangking Faktor",
    y = "Skor",
    x = "")
```



8.1.2.4 Alternatif

8.1.2.5 Alternatif untuk Faktor Salary

```
library(dplyr)
alternatif <- c("J1", "J2", "J3")

# Menampilkan data frame
alternatif_data1 <- ahpdata[,8:10]
alternatif1 <- ahp.mat(df = alternatif_data1,
                       atts = alternatif,
                       negconvert = TRUE)
alternatif1_agg <- ahp.aggpref(alternatif1,
                               alternatif,
                               method = "arithmetic",
                               aggmethod = "arithmetic")
round(alternatif1_agg, 3) %>% t()
```

```
        J1    J2    J3
[1,] 0.628 0.232 0.139
```

```
#Consistency
ri <- ahp.ri(nsims = 10000, dim = 3, seed = 42)
ahp.cr(alternatif1, alternatif, ri)
```

```
[1] 0.00000000 0.07669698
```

8.1.2.6 Alternatif untuk Faktor Quality of Life

```
alternatif_data2 <- ahpdata[,11:13]
alternatif2 <- ahp.mat(df = alternatif_data2,
                       atts = alternatif,
                       negconvert = TRUE)
alternatif2_agg <- ahp.aggpref(alternatif2,
                               alternatif,
                               method = "arithmetic",
                               aggmethod = "arithmetic")
round(alternatif2_agg, 3) %>% t()
```

```
       J1    J2    J3
[1,] 0.15 0.269 0.581
```

```
#Consistency
ahp.cr(alternatif2, alternatif, ri)
```

```
[1] 0.05121571 0.12952632
```

8.1.2.7 Alternatif untuk Faktor Interest in Work

```
alternatif_data3 <- ahpdata[,14:16]
alternatif3 <- ahp.mat(df = alternatif_data3,
                       atts = alternatif,
                       negconvert = TRUE)
alternatif3_agg <- ahp.aggpref(alternatif3,
                               alternatif,
                               method = "arithmetic",
                               aggmethod = "arithmetic")
round(alternatif3_agg, 3) %>% t()
```

```
        J1    J2    J3
[1,] 0.132 0.651 0.218
```

```
#Consistency
ahp.cr(alternatif3, alternatif, ri)
```

```
[1] 0.006706716 0.008789809
```

8.1.2.8 Alternatif untuk Faktor Location

```
alternatif_data4 <- ahpdata[,17:19]
alternatif4 <- ahp.mat(df = alternatif_data4,
                       atts = alternatif,
                       negconvert = TRUE)
alternatif4_agg <- ahp.aggpref(alternatif4,
                               alternatif,
                               method = "arithmetic",
                               aggmethod = "arithmetic")
round(alternatif4_agg, 3) %>% t()
```

```
        J1    J2    J3
[1,] 0.104 0.597 0.299
```

```
#Consistency
ahp.cr(alternatif4, alternatif, ri)
```

```
[1] 0.16898990 0.02349155
```

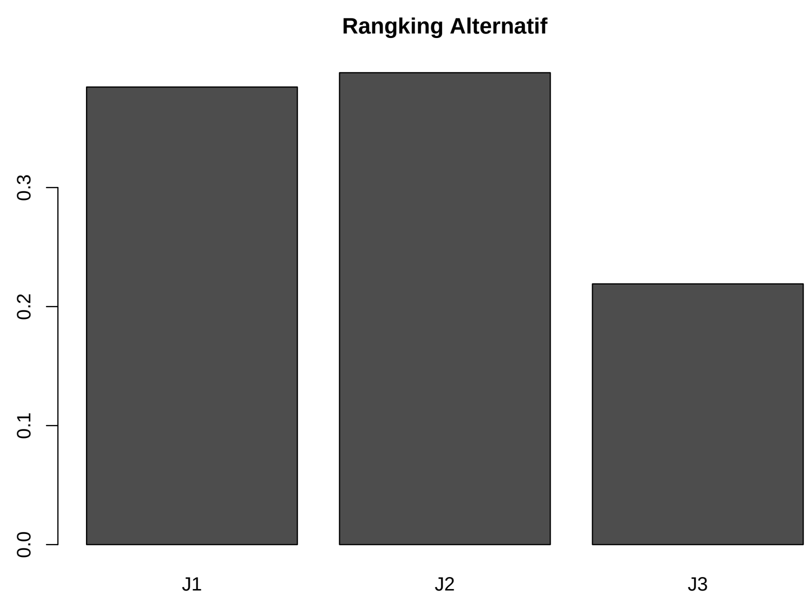
8.1.2.9 Gabungan Alternatif

```
alternatif_agg <- cbind(alternatif1_agg,alternatif2_agg,
                        alternatif3_agg,alternatif4_agg) %*% faktor_agg
alternatif_agg
```

```
        [,1]
```

```
J1 0.3844544
J2 0.3964920
J3 0.2190537
```

```
barplot(t(alternatif_agg) ,main="Rangking Alternatif")
```




```
data = data.frame("Cat"=row.names(data.frame(alternatif_agg)),
                  data.frame(alternatif_agg))
data$Cat <- factor(data$Cat,
                   levels = c( "J1" , "J2" ,"J3"),
                   labels = c("Job1", "Job2","Job3"))
# Buat grafik batang
data$warna <- ifelse(data$alternatif_agg == max(data$alternatif_agg),
                     "terbesar", "lainnya")
# Buat grafik batang
ggplot(data, aes(x = Cat, y = alternatif_agg, fill = warna)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("terbesar" = "#4682B4",
                               "lainnya" = "#A9A9A9")) +
  theme_minimal() +
  theme(legend.position = "none") +
  labs(
    title = "AHP: Rangking Alternatif",
    y = "Skor",
```

```
    x = "")
```