Penurunan Data

Terbahagi kepada 2:

1. Penurunan Dimensi Data
2. Penurunan Numerositi Data

1. Penurunan Dimensi Data

Data dengan dimensi besar boleh dikecilkan dimensi menerusi kaedah:

1. Mengeluarkan Atribut. (Menggunakan Domain Knowledge)
2. Analisis Komponen Utama.
3. Analisis Faktor.

1.1 Mengeluarkan Atribut

1.1.1 Atribut hampir sama sifat

1.1.2 Atribut tidak relevan

1.1.3 Atribut tidak signifikan

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.4.2
```

```
data(Hitters)
head(Hitters,5)
```

```
##                   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun
## -Andy Allanson      293   66     1   30  29    14     1    293    66      1
## -Alan Ashby         315   81     7   24  38    39    14   3449   835     69
## -Alvin Davis        479  130    18   66  72    76     3   1624   457     63
## -Andre Dawson       496  141    20   65  78    37    11   5628  1575    225
## -Andres Galarraga   321   87    10   39  42    30     2    396   101     12
##                   CRuns CRBI CWalks League Division PutOuts Assists Errors
## -Andy Allanson       30   29     14      A        E     446      33     20
## -Alan Ashby         321  414    375      N        W     632      43     10
## -Alvin Davis        224  266    263      A        W     880      82     14
```

```
## -Andre Dawson       828  838    354      N        E     200      11      3
## -Andres Galarraga    48   46     33      N        E     805      40      4
##                   Salary NewLeague
## -Andy Allanson        NA         A
## -Alan Ashby        475.0         N
## -Alvin Davis       480.0         A
## -Andre Dawson      500.0         N
## -Andres Galarraga   91.5         N
```

```
str(Hitters)
```

```
## 'data.frame':    322 obs. of  20 variables:
##  $ AtBat    : int  293 315 479 496 321 594 185 298 323 401 ...
##  $ Hits     : int  66 81 130 141 87 169 37 73 81 92 ...
##  $ HmRun    : int  1 7 18 20 10 4 1 0 6 17 ...
##  $ Runs     : int  30 24 66 65 39 74 23 24 26 49 ...
##  $ RBI      : int  29 38 72 78 42 51 8 24 32 66 ...
##  $ Walks    : int  14 39 76 37 30 35 21 7 8 65 ...
##  $ Years    : int  1 14 3 11 2 11 2 3 2 13 ...
##  $ CAtBat   : int  293 3449 1624 5628 396 4408 214 509 341 5206 ...
##  $ CHits    : int  66 835 457 1575 101 1133 42 108 86 1332 ...
##  $ CHmRun   : int  1 69 63 225 12 19 1 0 6 253 ...
##  $ CRuns    : int  30 321 224 828 48 501 30 41 32 784 ...
##  $ CRBI     : int  29 414 266 838 46 336 9 37 34 890 ...
##  $ CWalks   : int  14 375 263 354 33 194 24 12 8 866 ...
##  $ League   : Factor w/ 2 levels "A","N": 1 2 1 2 2 1 2 1 2 1 ...
##  $ Division : Factor w/ 2 levels "E","W": 1 2 2 1 1 2 1 2 2 1 ...
##  $ PutOuts  : int  446 632 880 200 805 282 76 121 143 0 ...
##  $ Assists  : int  33 43 82 11 40 421 127 283 290 0 ...
##  $ Errors   : int  20 10 14 3 4 25 7 9 19 0 ...
##  $ Salary   : num  NA 475 480 500 91.5 750 70 100 75 1100 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 1 2 1 2 2 1 1 1 2 1 ...
```

```
Hitters2 = na.omit(Hitters)
```

Suaikan model rigresi

```
model.F = lm(Salary ~., data=Hitters2)
summary(model.F)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = Hitters2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -907.62 -178.35  -31.11  139.09 1877.04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  163.10359   90.77854   1.797 0.073622 .
## AtBat         -1.97987    0.63398  -3.123 0.002008 **
```

```
## Hits           7.50077    2.37753   3.155 0.001808 **
## HmRun          4.33088    6.20145   0.698 0.485616
## Runs          -2.37621    2.98076  -0.797 0.426122
## RBI           -1.04496    2.60088  -0.402 0.688204
## Walks          6.23129    1.82850   3.408 0.000766 ***
## Years         -3.48905   12.41219  -0.281 0.778874
## CAtBat        -0.17134    0.13524  -1.267 0.206380
## CHits          0.13399    0.67455   0.199 0.842713
## CHmRun        -0.17286    1.61724  -0.107 0.914967
## CRuns          1.45430    0.75046   1.938 0.053795 .
## CRBI           0.80771    0.69262   1.166 0.244691
## CWalks        -0.81157    0.32808  -2.474 0.014057 *
## LeagueN       62.59942   79.26140   0.790 0.430424
## DivisionW   -116.84925   40.36695  -2.895 0.004141 **
## PutOuts        0.28189    0.07744   3.640 0.000333 ***
## Assists        0.37107    0.22120   1.678 0.094723 .
## Errors        -3.36076    4.39163  -0.765 0.444857
## NewLeagueN   -24.76233   79.00263  -0.313 0.754218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 315.6 on 243 degrees of freedom
## Multiple R-squared:  0.5461, Adjusted R-squared:  0.5106
## F-statistic: 15.39 on 19 and 243 DF,  p-value: < 2.2e-16
```

P/ubah yang diperlukan untuk analisis Salary

```
attach(Hitters2)
Hitters3 = cbind(Salary, AtBat, Hits, Walks, CWalks, Division, PutOuts)
head(Hitters3, 10)
```

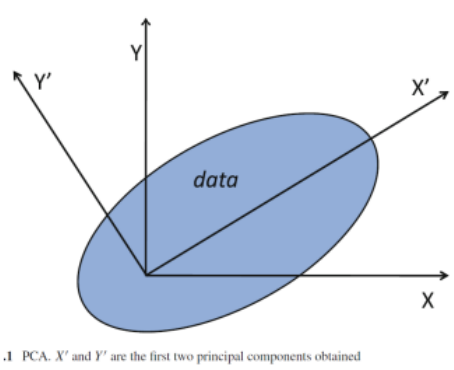
```
##         Salary AtBat Hits Walks CWalks Division PutOuts
##  [1,]  475.000   315   81    39    375        2     632
##  [2,]  480.000   479  130    76    263        2     880
##  [3,]  500.000   496  141    37    354        1     200
##  [4,]   91.500   321   87    30     33        1     805
##  [5,]  750.000   594  169    35    194        2     282
##  [6,]   70.000   185   37    21     24        1      76
##  [7,]  100.000   298   73     7     12        2     121
##  [8,]   75.000   323   81     8      8        2     143
##  [9,] 1100.000   401   92    65    866        1       0
## [10,]  517.143   574  159    59    488        1     238
```

```
model.G = lm(Salary~ AtBat+Hits+Walks+CWalks+Division+PutOuts)
summary(model.G)
```

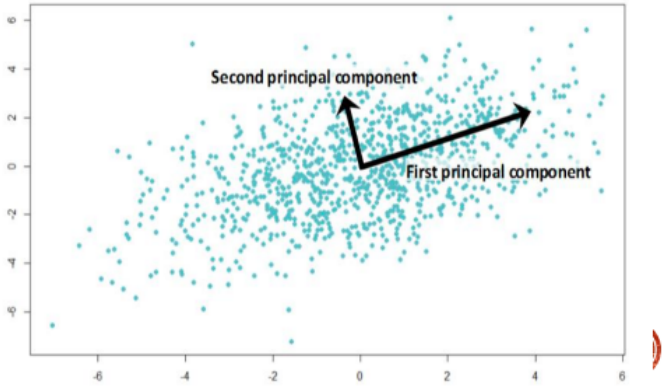
```
##
## Call:
## lm(formula = Salary ~ AtBat + Hits + Walks + CWalks + Division +
##     PutOuts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -1018.8  -180.8   -45.2   139.0  2059.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   68.60280   69.33235   0.989 0.323364
## AtBat         -1.69314    0.55802  -3.034 0.002660 **
## Hits           8.12481    1.75374   4.633 5.75e-06 ***
## Walks          1.54020    1.38994   1.108 0.268856
## CWalks         0.70860    0.08922   7.942 6.29e-14 ***
## DivisionW   -112.66871   42.04740  -2.680 0.007850 **
## PutOuts        0.29003    0.07899   3.671 0.000294 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 337.7 on 256 degrees of freedom
## Multiple R-squared:  0.4525, Adjusted R-squared:  0.4397
## F-statistic: 35.26 on 6 and 256 DF,  p-value: < 2.2e-16
```

1.2 Analisis Komponen Utama



.1 PCA. X' and Y' are the first two principal components obtained



Prosedur PCA

1. Skalakan data input dengan mempiawaikan julat bagi setiap atribut (skor-z).
2. Dapatkan k-set vektor ortogonal berdasarkan data yang telah di piawaikan.
3. Komponen utama disusun secara sumbangan menurun berdasarkan maklumat nilai eigen. Komponen utama berfungsi sebagai set paksi-paksi baru untuk data yang diselaraskan mengikut varians data asal.
4. Pengurangan dimensi data dibuat dengan membuang komponen yang memberikan sumbangan varians yang rendah:
    - Hanya komponen utama yang menerangkan sumbangan varians yang tinggi dikekalkan sebagai set p/ubah baharu.

```
data = read.csv("D:/MSc DSc/Sem 1/Data Mining/Data/READING120n.csv", header=T)
head(data,10)
```

```
##    GEN rhyme Begsnd ABC LS Spelling COW
## 1    M    10     10   6  7        4   7
## 2    F    10     10  22 19        9  15
## 3    M     9     10  23 15        5   6
## 4    F     5     10  10  3        2   3
## 5    F     2     10   4  0        0   2
## 6    M     5      6  22  8       17   6
## 7    M     8      5  25 20       12   4
## 8    M     4      3  26 16        3   0
## 9    F     3      7  18  8        3   0
## 10   F     9     10  26 17       15  15
```

Keluarkan atribut bukan nombor

```
dat = data[,-1]
```

Huraikan data

```
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.4.2
```

```
describe(dat)
```

```
##          vars   n  mean   sd median trimmed  mad min max range  skew kurtosis
## rhyme       1 120  7.29 2.99      9    7.65 1.48   0  10    10 -0.65    -1.02
## Begsnd      2 120  7.94 2.74     10    8.41 0.00   0  10    10 -1.03    -0.23
## ABC         3 120 20.92 6.89     24   22.36 2.97   1  26    25 -1.54     1.19
## LS          4 120 14.46 7.45     16   14.92 7.41   0  26    26 -0.53    -0.77
## Spelling    5 120  7.55 5.96      6    7.18 7.41   0  20    20  0.39    -1.03
## COW         6 120 10.15 7.21     10    9.96 9.64   0  22    22  0.13    -1.33
##            se
## rhyme    0.27
## Begsnd   0.25
## ABC      0.63
## LS       0.68
## Spelling 0.54
## COW      0.66
```

Skalakan data

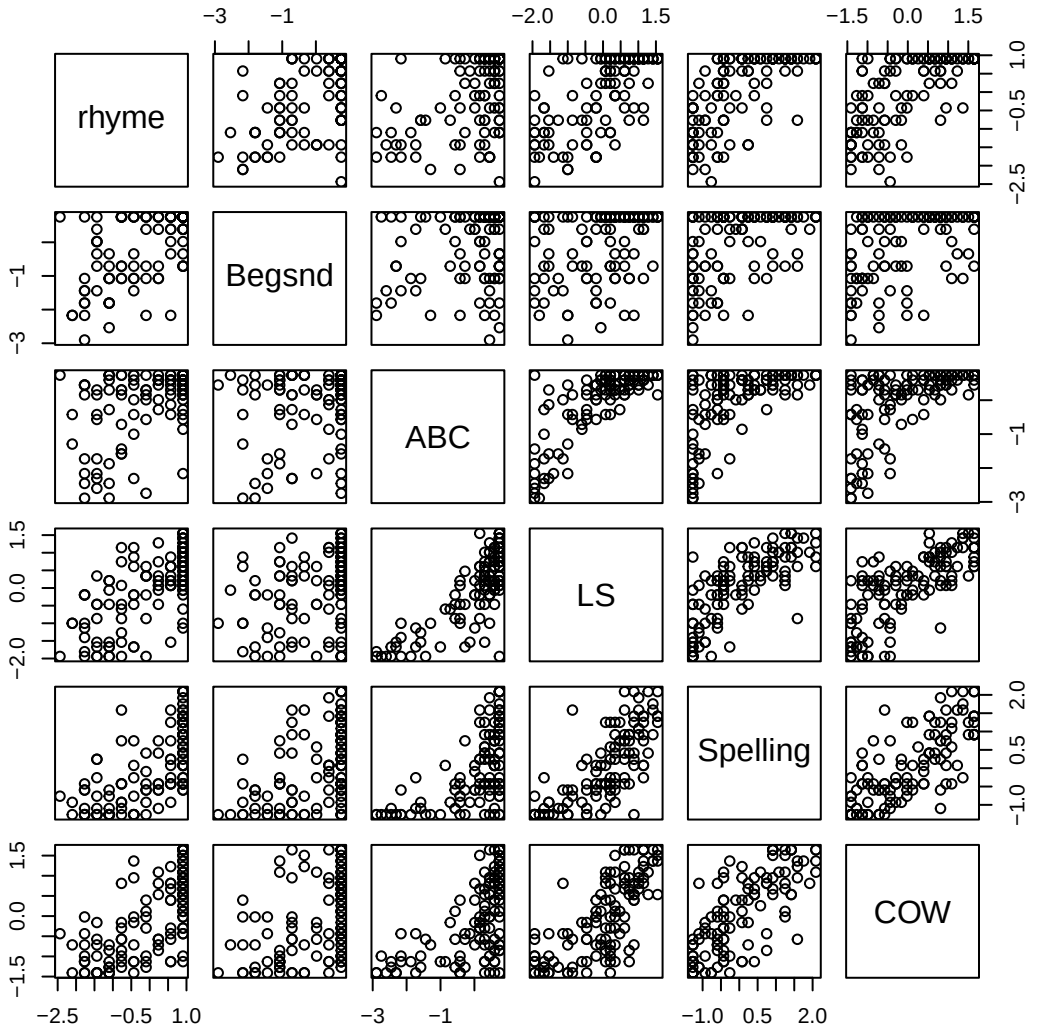
```
zdata = scale(dat)
head(zdata,10)
```

```
##            rhyme     Begsnd        ABC          LS   Spelling        COW
##  [1,]  0.9054058  0.7524019 -2.1649759 -1.00074029 -0.5955932 -0.4368511
##  [2,]  0.9054058  0.7524019  0.1572329  0.60938934  0.2432705  0.6726120
##  [3,]  0.5711021  0.7524019  0.3023709  0.07267946 -0.4278205 -0.5755340
##  [4,] -0.7661126  0.7524019 -1.5844237 -1.53745016 -0.9311387 -0.9915827
##  [5,] -1.7690236  0.7524019 -2.4552520 -1.93998257 -1.2666842 -1.1302656
##  [6,] -0.7661126 -0.7097556  0.1572329 -0.86656282  1.5854524 -0.5755340
```

```
##  [7,]  0.2367984 -1.0752950  0.5926470  0.74356681  0.7465887 -0.8528998
##  [8,] -1.1004163 -1.8063738  0.7377851  0.20685693 -0.7633660 -1.4076314
##  [9,] -1.4347200 -0.3442162 -0.4233193 -0.86656282 -0.7633660 -1.4076314
## [10,]  0.5711021  0.7524019  0.7377851  0.34103440  1.2499069  0.6726120
```

Plot korelasi & taburan data

```
pairs(zdata)
```



Lihat korelasi data

```
R = cor(zdata)
R
```

```
##               rhyme     Begsnd        ABC         LS  Spelling       COW
## rhyme     1.0000000 0.6161831 0.4994385 0.6769710 0.6682135 0.6929980
```

```
## Begsnd   0.6161831 1.0000000 0.2850706 0.3467132 0.4688980 0.4694738
## ABC      0.4994385 0.2850706 1.0000000 0.7955943 0.5888044 0.5981786
## LS       0.6769710 0.3467132 0.7955943 1.0000000 0.7579600 0.7492896
## Spelling 0.6682135 0.4688980 0.5888044 0.7579600 1.0000000 0.7668598
## COW      0.6929980 0.4694738 0.5981786 0.7492896 0.7668598 1.0000000
```

Bil atribut

```
p = ncol(zdata)
p
```

```
## [1] 6
```

Nilai & Vektor Eigen

```
e = eigen(R)
ev = e$values
evr = e$vectors
e
```

```
## eigen() decomposition
## $values
## [1] 4.0417265 0.8725973 0.4200022 0.2990629 0.2322152 0.1343960
##
## $vectors
##            [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
## [1,] -0.4202540  0.29934149 -0.09269853  0.80020266 -0.12282334  0.26415587
## [2,] -0.3068973  0.75974276  0.43140561 -0.33291224  0.02990871 -0.17541132
## [3,] -0.3849778 -0.46782622  0.65714698 -0.08464742  0.06601473  0.43539111
## [4,] -0.4458305 -0.33461651  0.06528679  0.14407269 -0.13081189 -0.80444760
## [5,] -0.4358068 -0.03894126 -0.43902727 -0.43785381 -0.60459527  0.24199105
## [6,] -0.4385206 -0.02897612 -0.42005561 -0.17090073  0.77266730  0.06474189
```

peratusan varians bagi setiap p/ubah PCA

```
Prop.var = ev/length(ev)
cumsum(Prop.var)
```

```
## [1] 0.6736211 0.8190540 0.8890543 0.9388981 0.9776007 1.0000000
```

Kita akan kekalkan 2 p/ubah PCA yang dapat menerangkan lebih 80% variasi data asal

```
y = zdata%*%evr
head(y,5)
```

```
##            [,1]      [,2]         [,3]       [,4]        [,5]        [,6]
## [1,]  1.1193495 2.2262055 -0.802398485  0.8485456 -0.07816090 -0.20279050
## [2,] -1.3445992 0.5362251 -0.005566031  0.3270445  0.21458787 -0.21216009
## [3,] -0.1808963 0.6101463  0.904678375  0.4770719 -0.22322576 -0.04872692
## [4,]  2.2270880 1.6629857  0.079348607 -0.3737515  0.00991986 -0.07692304
## [5,]  3.3703245 1.9219481 -0.220657112 -0.9899433  0.22398274 -0.48736015
```

```r
colnames(y) = c('PCA1','PCA2','PCA3','PCA4','PCA5','PCA6')
head(y,5)
```

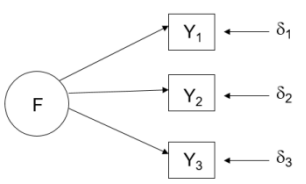
```
##            PCA1      PCA2         PCA3       PCA4        PCA5        PCA6
## [1,]  1.1193495 2.2262055 -0.802398485  0.8485456 -0.07816090 -0.20279050
## [2,] -1.3445992 0.5362251 -0.005566031  0.3270445  0.21458787 -0.21216009
## [3,] -0.1808963 0.6101463  0.904678375  0.4770719 -0.22322576 -0.04872692
## [4,]  2.2270880 1.6629857  0.079348607 -0.3737515  0.00991986 -0.07692304
## [5,]  3.3703245 1.9219481 -0.220657112 -0.9899433  0.22398274 -0.48736015
```

Data yang dikekalkan unutk analisis perlombongan data

```r
data2 =y[,c(1,2)]
head(data2,5)
```

```
##            PCA1      PCA2
## [1,]  1.1193495 2.2262055
## [2,] -1.3445992 0.5362251
## [3,] -0.1808963 0.6101463
## [4,]  2.2270880 1.6629857
## [5,]  3.3703245 1.9219481
```

1.3 Analisis Faktor



$$Y_1 = \lambda_1 F + \delta_1$$
$$Y_2 = \lambda_2 F + \delta_2$$
$$Y_3 = \lambda_3 F + \delta_3$$

1.3.1 Model satu-faktor

1.3.2 Model dua-faktor

1.3.4 Putaran faktor

```r
d = read.csv("D:/MSc DSc/Sem 1/Data Mining/Data/food-texture.csv", header=T, row.names ="X")
head(d)
```

```
##       Oil Density Crispy Fracture Hardness
## B110 16.5    2955     10       23       97
## B136 17.7    2660     14        9      139
## B171 16.2    2870     12       17      143
## B192 16.7    2920     10       31       95
## B225 16.3    2975     11       26      143
## B237 19.1    2790     13       16      189
```

```
describe(d)
```

```
##           vars  n    mean     sd median trimmed    mad    min    max range  skew
## Oil          1 50   17.20   1.59   16.9   17.14   1.19   13.7   21.2   7.5  0.41
## Density      2 50 2857.60 124.50 2867.5 2860.50 129.73 2570.0 3125.0 555.0 -0.18
## Crispy       3 50   11.52   1.78   12.0   11.57   1.48    7.0   15.0   8.0 -0.28
## Fracture     4 50   20.86   5.47   21.0   20.98   5.93    9.0   33.0  24.0 -0.11
## Hardness     5 50  128.18  31.13  126.0  128.32  27.43   63.0  192.0 129.0  0.01
##           kurtosis    se
## Oil          0.30  0.23
## Density     -0.46 17.61
## Crispy      -0.48  0.25
## Fracture    -0.61  0.77
## Hardness    -0.42  4.40
```

Piawaikan Data

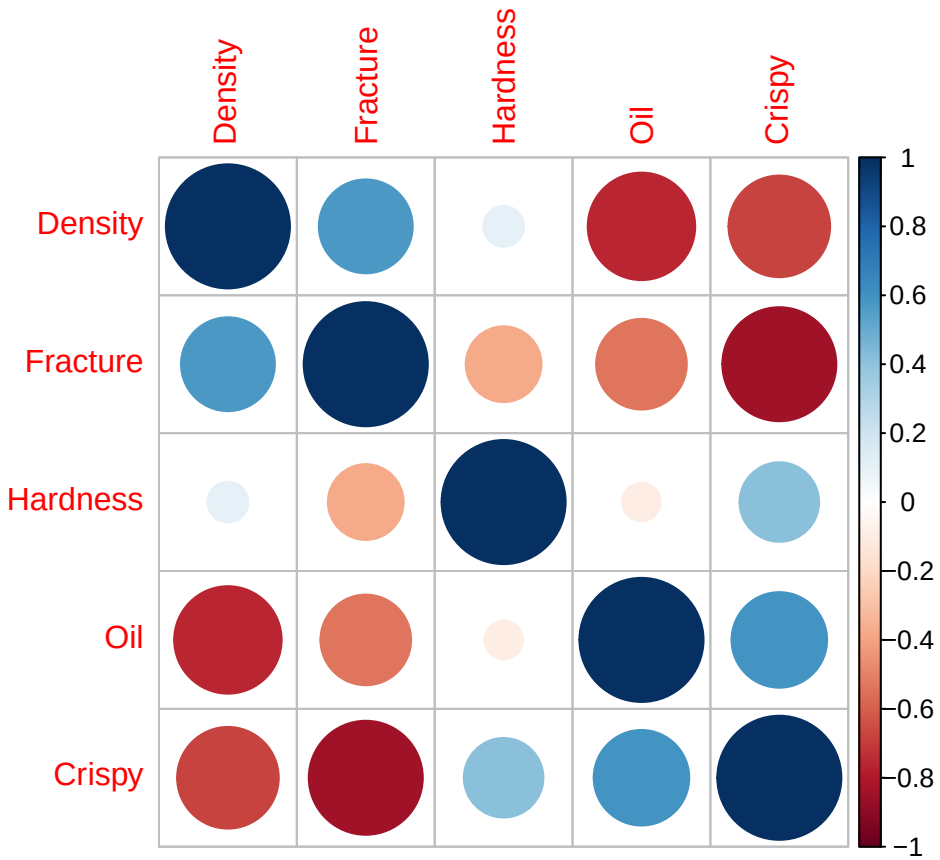
```
z_skor = scale(d)
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.2
```

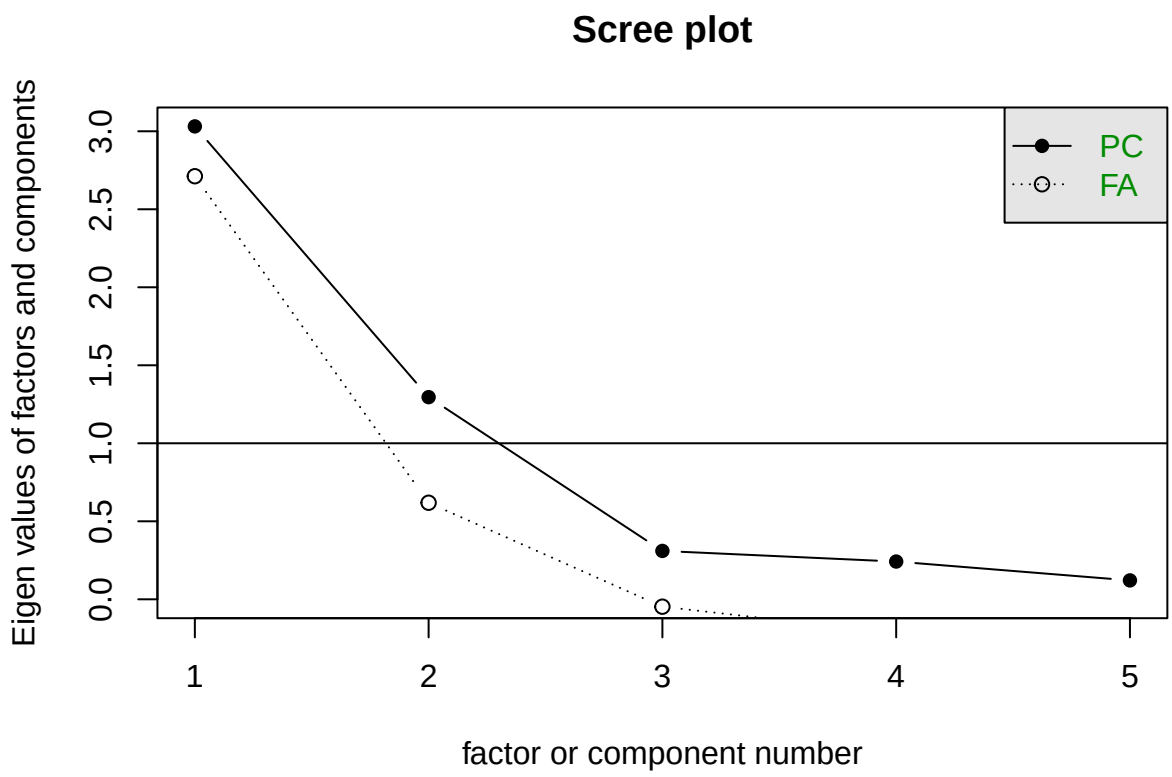
```
## corrplot 0.95 loaded
```

```
corrplot(cor(z_skor), order='hclust')
```



Scree Plot

```
scree(z_skor)
```



Jalankan analisis faktor

```
F.A = factanal(z_skor, factors=2, rotation='varimax')
F.A
```

```
##
## Call:
## factanal(x = z_skor, factors = 2, rotation = "varimax")
##
## Uniquenesses:
##      Oil  Density   Crispy Fracture Hardness
##    0.334    0.156    0.042    0.256    0.407
##
## Loadings:
##          Factor1 Factor2
## Oil      -0.816
## Density   0.919
## Crispy   -0.745   0.635
## Fracture  0.645  -0.573
## Hardness          0.764
##
##                Factor1 Factor2
```

```
## SS loadings      2.490   1.316
## Proportion Var   0.498   0.263
## Cumulative Var   0.498   0.761
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 0.27 on 1 degree of freedom.
## The p-value is 0.603
```

```
F.A2 = factanal(z_skor, factors=2, rotation='varimax')
```

Data skor analisis faktor

```
FA.skor = factanal(z_skor, factors=2, scores='regression', rotation='varimax')
head(FA.skor$scores, 10)
```

```
##         Factor1     Factor2
## B110  0.6081789 -0.60485652
## B136 -1.3383534  0.74121570
## B171  0.1514441  0.63655719
## B192  0.4830427 -0.97998211
## B225  0.8077371  0.33673197
## B237 -0.4969723  0.90014287
## B261 -0.9642382  0.09473458
## B264 -0.4441417 -1.84403591
## B353  0.6769554  0.21726360
## B360  0.5985900  0.15208902
```

2. Penurunam Numerositi Data

2.1 Model Berparameter

2.1.1 Model Regresi

```
data = read.csv("D:/MSc DSc/Sem 1/Data Mining/Data/data.csv", header=T, sep=';')
head(data,10)
```

```
##      income education_level work_experience  expenditure
## 1  45435.43               3      13.5681996 2.743065e+10
## 2  36910.20               1       6.4077324 4.532608e+08
## 3  16836.11               1       7.9438134 2.658155e+04
## 4  47458.35               5      20.4785260 8.593200e+11
## 5  17016.09               2      15.4508807 4.224400e+04
## 6  46910.73               1       4.2731317 2.991605e+10
## 7  38406.65               2      -0.2740395 6.653858e+08
## 8  57641.14               3      16.2903239 5.306520e+13
## 9  46750.24               1       6.7623052 3.175008e+10
## 10 39883.36               5       7.5471444 2.684209e+09
```

Kita berminat terhadap hubungan, $y =$ *expenditure*, terhadap fitur yang lain

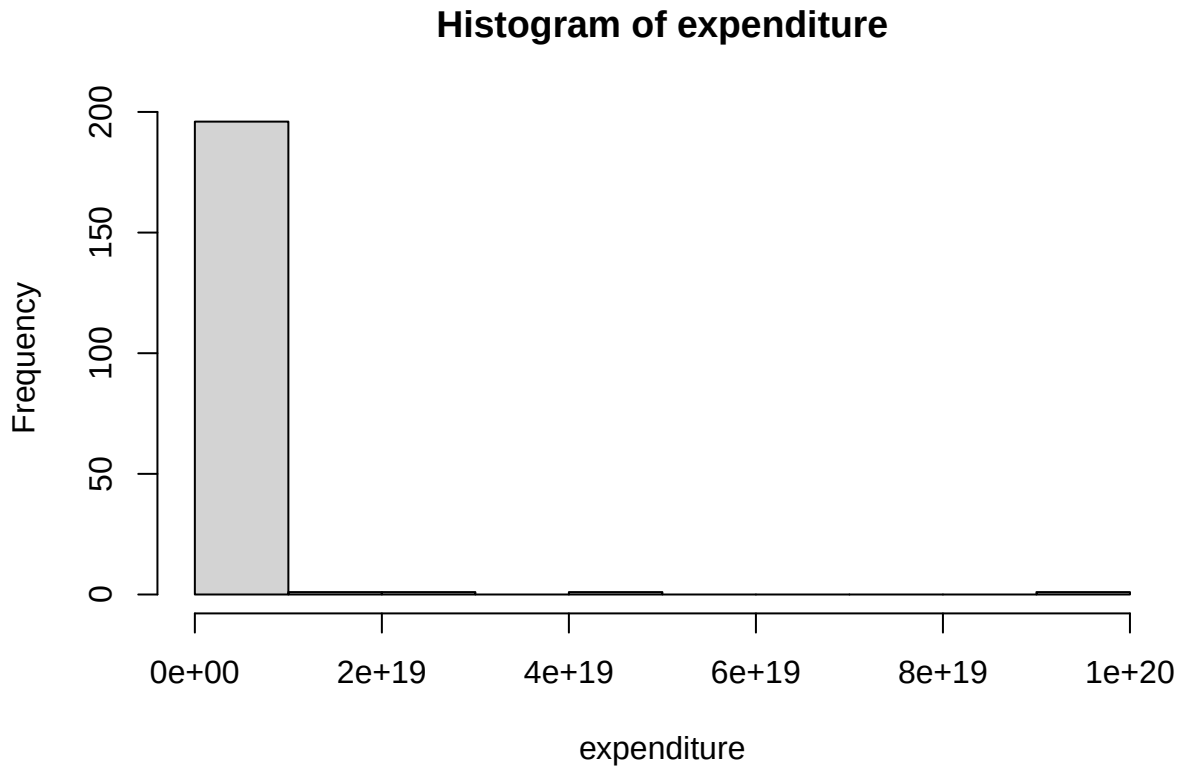
x1 = income

x2 = education_level

x3 = work_experience

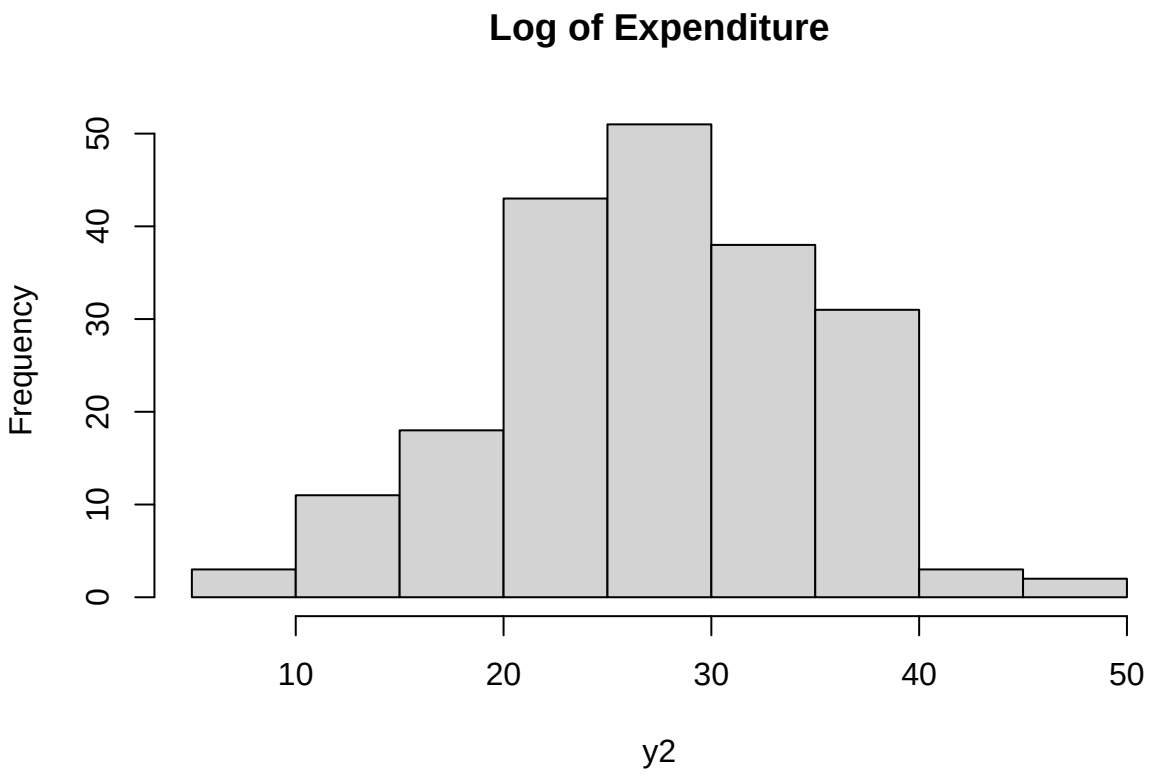
Andaian model regresi = Y menghampiri normal

```
attach(data)
hist(expenditure)
```



Perlu jelmakan kepada normal

```
y2 = log(expenditure)
hist(y2, main = "Log of Expenditure")
```




```
data$education_level = as.factor(education_level)

model_reg = lm(log(expenditure) ~ income + data$education_level + work_experience)
summary(model_reg)
```

```
##
## Call:
## lm(formula = log(expenditure) ~ income + data$education_level +
##     work_experience)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05108 -0.35025 -0.00016  0.31069  1.21984
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             9.621e-01  1.544e-01   6.233 2.82e-09 ***
## income                  4.987e-04  2.316e-06 215.315  < 2e-16 ***
## data$education_level2 1.873e-01  1.080e-01   1.734 0.084517 .
## data$education_level3 3.292e-01  1.055e-01   3.120 0.002087 **
## data$education_level4 4.297e-01  1.186e-01   3.623 0.000373 ***
## data$education_level5 8.723e-01  1.111e-01   7.852 2.76e-13 ***
## work_experience         8.311e-02  7.239e-03  11.479  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.5001 on 193 degrees of freedom
## Multiple R-squared:  0.9959, Adjusted R-squared:  0.9958
## F-statistic:  7782 on 6 and 193 DF,  p-value: < 2.2e-16
```

```
coef(model_reg)
```

```
##           (Intercept)                income data$education_level2
##          0.9621264920          0.0004986787          0.1872654922
## data$education_level3 data$education_level4 data$education_level5
##          0.3292185117          0.4296501743          0.8722609773
##       work_experience
##          0.0831053838
```

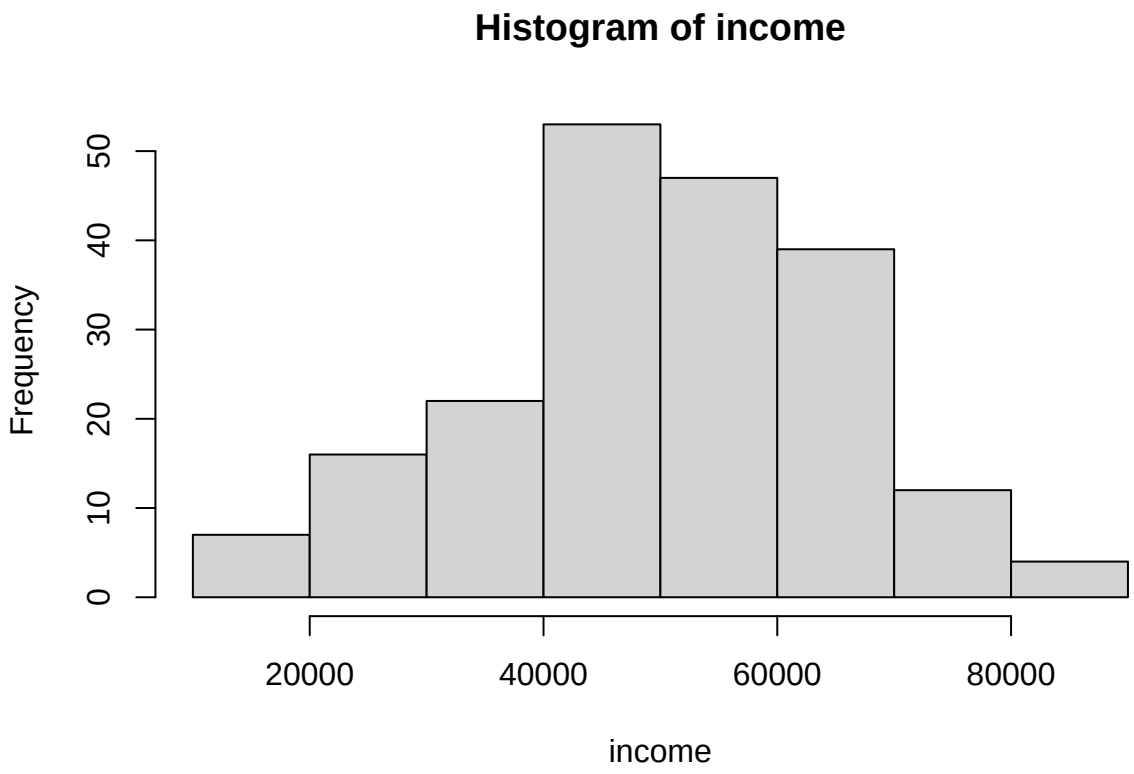
R^2 >0.99, menunjukkan model ini seesuai untuk mewakili data asal. Simpan maklumat berkaitan model;

1. Parameter model - coefficient

$$log(expenditure) = 0.9621 + 0.0005(income) + 0.1872(education\_level2) + 0.3292(education\_level3) + 0.4297(educat$$

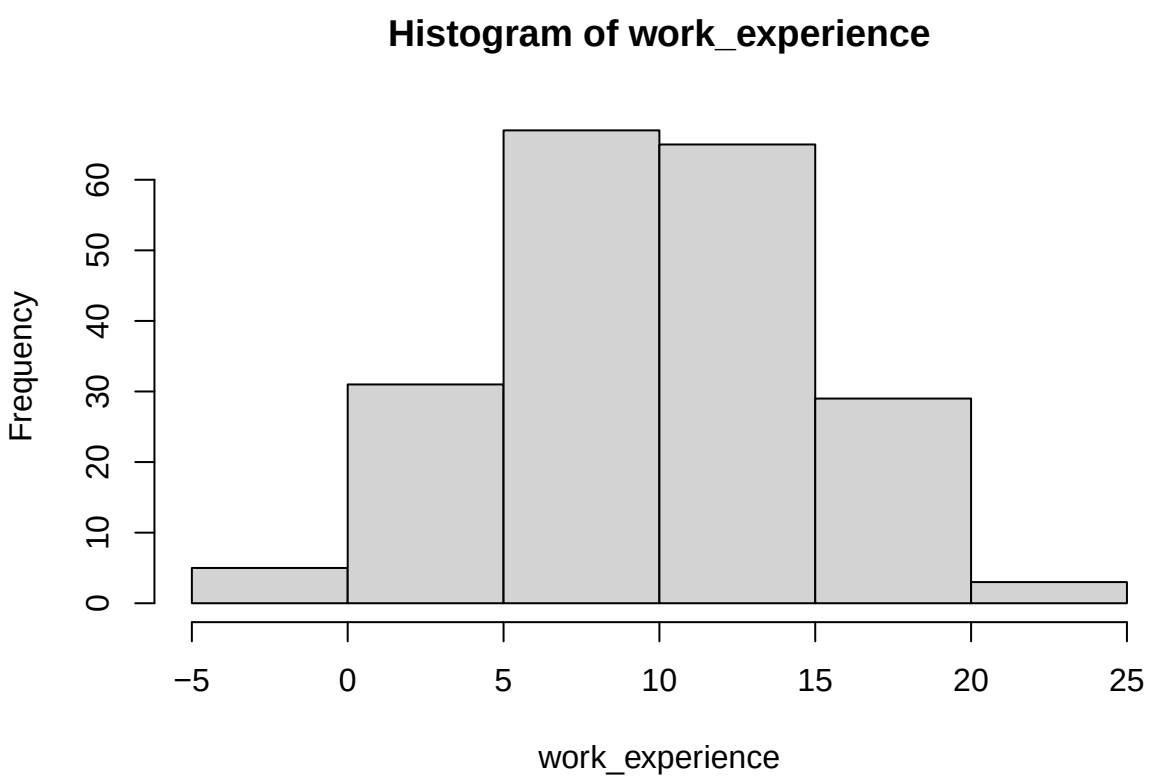
2. Maklumat fitur Xi

```
# X1 = income
muIn = mean(income)
sdIn = sd(income)
IncomeR = range(income)
hist(income)
```




```r
# X2 = education_level
education_range = 1:5
```

```r
# X3 = work_experience
muWE = mean(work_experience)
sdWE = sd(work_experience)
workExpR = range(work_experience)
hist(work_experience)
```

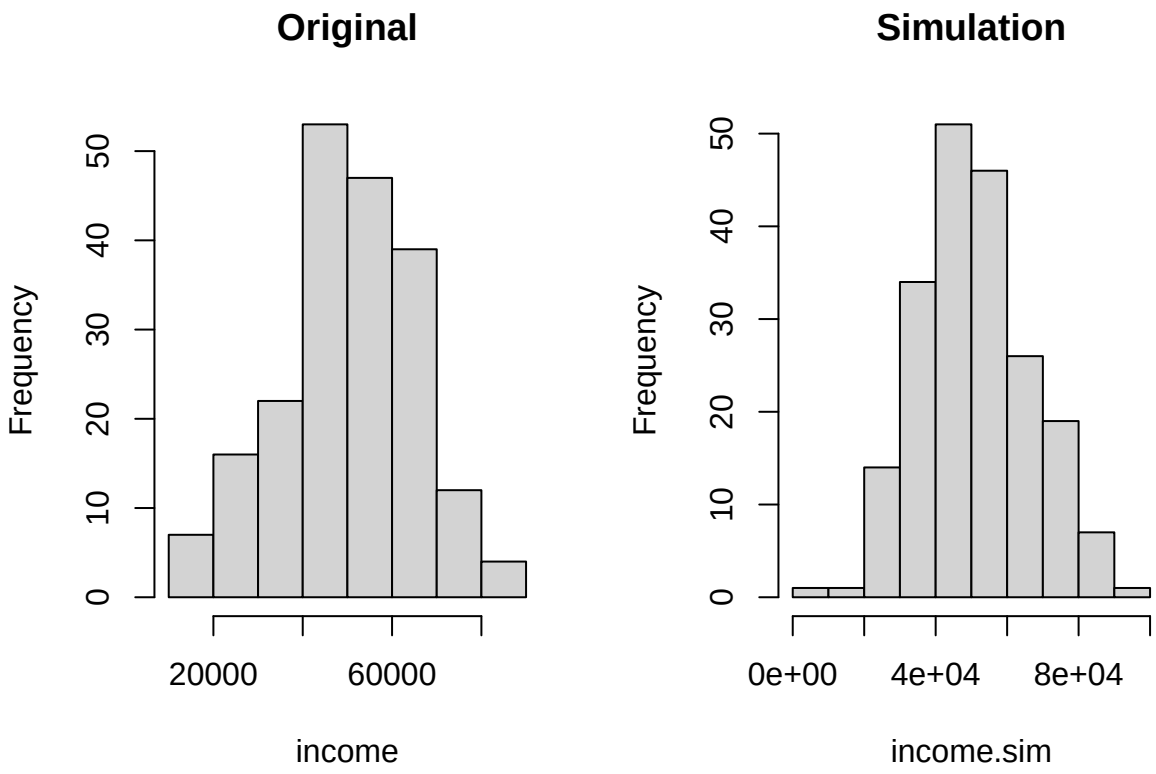


Jika mahu jalankan analisis terhadap data, boleh janakan data simulasi menggunakan model & maklumat fitur

Simulasi fitur Income

```
n = 200
income.sim = rnorm(n, mean=muIn, sd=sdIn)

par(mfrow=c(1,2))
hist(income , main='Original')
hist(income.sim , main='Simulation')
```




```
par(mfrow=c(1,1))
```

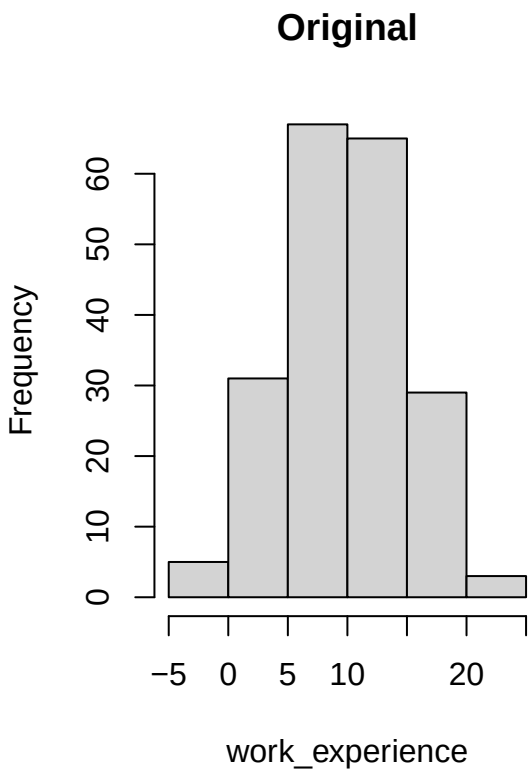
Education Level

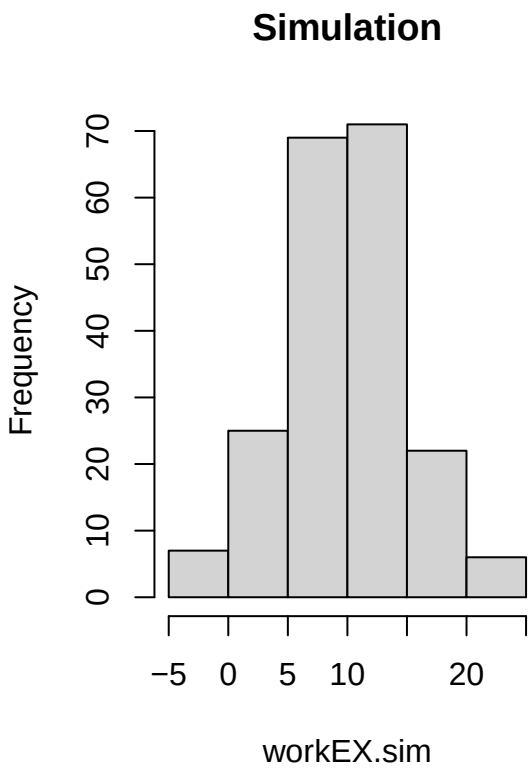
```
education.sim = sample(1:5, n, replace=T)
education.sim = as.factor(education.sim)
```

Work Experience

```
n = 200
workEX.sim = rnorm(n, mean=muWE, sd=sdWE)

par(mfrow=c(1,2))
hist(work_experience , main='Original')
hist(workEX.sim , main='Simulation')
```






```
par(mfrow=c(1,1))
```

Gabungan fitur simulation ke data frame

```
n = 200
fitur.sim = data.frame(workEX.sim, education.sim, income.sim)
```

Data simulasi y

```
sim.expend = predict(model_reg,fitur.sim)
head(sim.expend, 10)
```

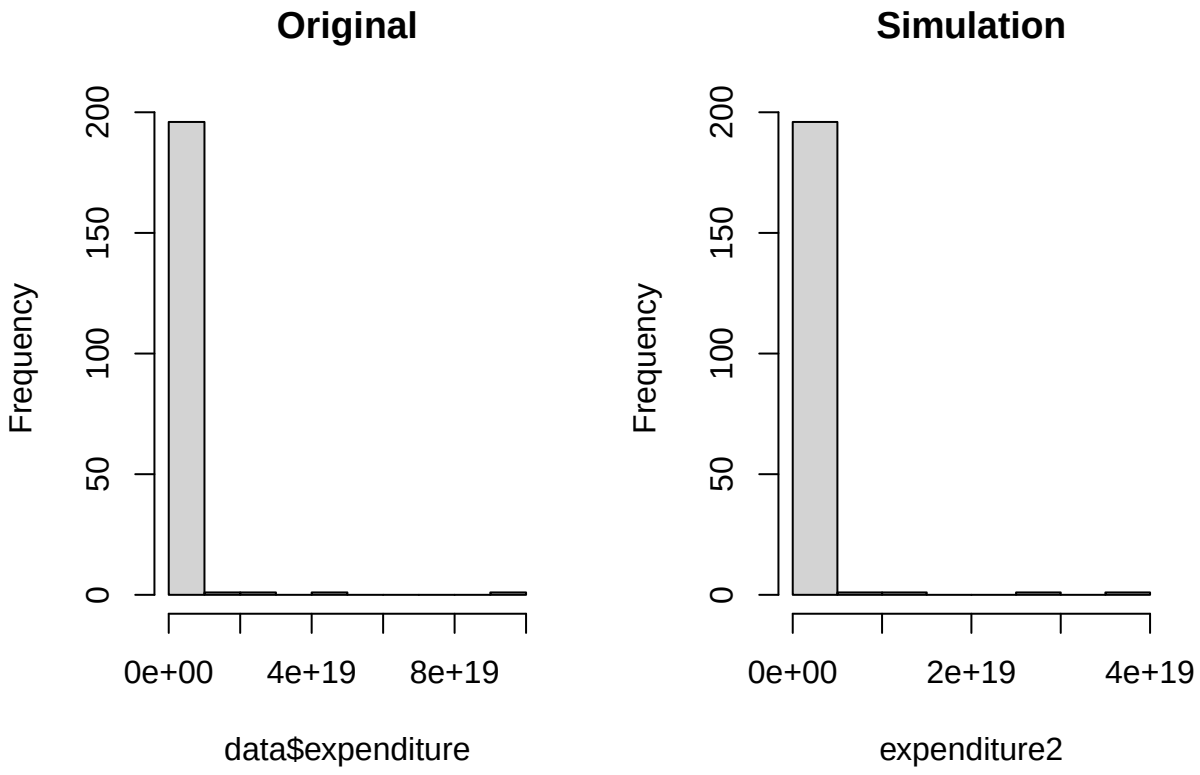
```
##        1        2        3        4        5        6        7        8
## 25.07662 19.90097 10.01811 27.20273 10.91901 24.71062 20.27920 31.38956
##        9       10
## 24.83746 22.35058
```

Data simulasi lengkap

```
expenditure2 = exp(sim.expend) # jelmakan balik kepada data asal
new_df = cbind(fitur.sim, expenditure2)
head(new_df,10)
```

```
##    workEX.sim education.sim income.sim expenditure2
## 1    8.567895             4   52207.84 7.773845e+10
## 2    7.605199             1   55184.50 4.394231e+08
## 3    3.914492             3   59555.86 2.242895e+04
## 4   12.719179             3   38163.28 6.516228e+11
## 5    9.965531             3   62023.66 5.521583e+04
## 6   14.360587             2   69348.32 5.391230e+10
## 7   10.489904             3   21886.78 6.414202e+08
## 8    7.771730             1   37401.51 4.288590e+13
## 9   -2.237988             4   35683.12 6.120267e+10
## 10   9.127209             5   43088.50 5.090167e+09
```

```
par(mfrow=c(1,2))
hist(data$expenditure ,main='Original')
hist(expenditure2 ,main='Simulation')
```




```
par(mfrow=c(1,1))
```

2.2 Model Tak Berparameter

1. Histogram/pendisketan
2. Pengkelompokan
3. Pensampelan semula (butstrap)

2.2.1 Teknik Pensampelan semula

```
dataKe = read.table("D:/MSc DSc/Sem 1/Data Mining/Data/Kewangan.D.txt", header=T)
head(dataKe, 10)
```

```
##    ID Bangsa      Hutang Pendapatan.Tahunan
## 1   1   Cina  -255.41849           3919.225
## 2   2  India  -550.95988           2023.781
## 3   3   Cina   -74.77182           2480.774
## 4   4 Melayu -3144.75019           2907.829
## 5   5   Cina -1423.13386           2481.821
## 6   6  India -1092.24217           3750.682
## 7   7 Melayu  -662.81763           3312.495
## 8   8 Melayu -2875.17346           2465.722
## 9   9 Melayu -1325.57963           4609.340
## 10 10 Melayu  -770.33155           2596.760
```

Strata Bangsa

```
table(dataKe$Bangsa)/length(dataKe$Bangsa) # data tak bagus, tak represent population
```

```
##
##   Cina  India Melayu
## 0.3301 0.3382 0.3317
```

60% Malay, 30% Cina, 10% India

```
sM = 3000*0.6
sC = 3000*0.3
sI = 3000*0.1
```

Subset berdasarkan strata

```
d1 = subset(dataKe, Bangsa == "Melayu")
d2 = subset(dataKe, Bangsa == "Cina")
d3 = subset(dataKe, Bangsa == "India")
```

Pensampelan semula

```
N1 = sample(nrow(d1), size=sM, replace = FALSE)
SN1 = d1[N1,]

N2 = sample(nrow(d2), size=sC, replace = FALSE)
SN2 = d2[N2,]

N3 = sample(nrow(d3), size=sI, replace = FALSE)
SN3 = d3[N3,]

samp = rbind(SN1,SN2,SN3)

str(samp)
```

```
## 'data.frame':    3000 obs. of  4 variables:
##  $ ID                : int  9798 48092 15 49364 42328 19051 30781 30574 8488 22415 ...
##  $ Bangsa            : chr  "Melayu" "Melayu" "Melayu" "Melayu" ...
##  $ Hutang            : num  -1356 -2424 -1552 -487 -429 ...
##  $ Pendapatan.Tahunan: num  3625 2420 3736 3825 2863 ...
```