Penjelmaan Data dan Pendiskretan

1. Penormalan Data

1.1 Penjelmaan skala data

1.1.1 Penormalan Min-Max:

$$V = \frac{\left[X - \min(X)\right] \times \left[baru\_\max(X) - baru\_\min(X)\right]}{\max(X) - \min(X)} + baru\_\min(X)$$

Skalakan semua p/ubah kepada unit yang sama

selang (0,1), tiada unit

```r
dataAP3 = read.csv("D:/MSc DSc/Sem 1/Data Mining/UKMShape Data-20241201/dataAP3.csv", header=T)
head(dataAP3)
```

```
##   X Month Day_of_month Day_of_week ozone_ppm pressure_height.hPA Wind_speed.mph
## 1 1     1            1           4      3.01                5480              8
## 2 2     1            2           5      3.20                5660              6
## 3 3     1            3           6      2.70                5710              4
## 4 4     1            4           7      5.18                5700              3
## 5 5     1            5           1      5.34                5760              3
## 6 6     1            6           2      5.77                5720              4
##   Temperature_Celcius Inversion_base_height.IBH Pressure_gradient.Psi.ft
## 1                  30                      5000                      -15
## 2                  38                      1601                      -14
## 3                  40                      2693                      -25
## 4                  45                       590                      -24
## 5                  54                      1450                       25
## 6                  35                      1568                       15
##   Inversion_temperature.ivC Visibility_pAerosol
## 1                     30.56                 200
## 2                     46.94                 300
## 3                     47.66                 250
## 4                     55.04                 100
## 5                     57.02                  60
## 6                     53.78                  60
```

```r
dataAP3 = dataAP3[-1]

attach(dataAP3)
names(dataAP3)
```

```
##  [1] "Month"                      "Day_of_month"
##  [3] "Day_of_week"                "ozone_ppm"
##  [5] "pressure_height.hPA"        "Wind_speed.mph"
##  [7] "Temperature_Celcius"        "Inversion_base_height.IBH"
##  [9] "Pressure_gradient.Psi.ft"  "Inversion_temperature.ivC"
## [11] "Visibility_pAerosol"
```

Ozone

```
min.03 = min(ozone_ppm)
min.03
```

```
## [1] 0.72
```

```
max.03 = max(ozone_ppm)
max.03
```

```
## [1] 37.98
```

```
new_max.03 = 1
new_min.03 = 0

new.03 = ((ozone_ppm - min.03)*(new_max.03 - new_min.03) / (max.03-min.03)) + new_min.03
head(new.03,5)
```

```
## [1] 0.06146001 0.06655931 0.05314010 0.11969941 0.12399356
```

ulang untuk semua data p/ubah

analises perlombongan data akan dijalankan terhadap data yang diskalakan. Selepas analisis perlombongan data, keputusan perlu dijelmakan semula kepada unit data asal

```
min.03 = 0
min.03
```

```
## [1] 0
```

```
max.03 = 1
max.03
```

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

```
new_max.03 = max(ozone_ppm)
new_min.03 = min(ozone_ppm)

data.asal.03 = ((new.03 - min.03)*(new_max.03 - new_min.03) / (max.03-min.03)) + new_min.03
head(data.asal.03)
```

```
## [1] 3.01 3.20 2.70 5.18 5.34 5.77
```

```r
head(ozone_ppm)
```

```
## [1] 3.01 3.20 2.70 5.18 5.34 5.77
```

1.1.2 Penormalan skor z

$$Z = \frac{X - \mu_X}{\sigma_X}$$

```r
mean.hPA = mean(pressure_height.hPA)
sd.hPA = sd(pressure_height.hPA)

z.score.hPA = (pressure_height.hPA - mean.hPA)/sd.hPA
head(z.score.hPA,10)
```

```
##  [1] -2.58185122 -0.87803114 -0.40474779 -0.49940446  0.06853557 -0.31009112
##  [7]  0.35250558  0.35250558 -0.49940446 -0.49940446
```

ulang untuk semua data p/ubah

analises perlombongan data akan dijalankan terhadap data yang diskalakan selepas analisis perlombongan data, keputusan perlu dijelmakan semula kepada unit data asal

```r
data.asal.hPA = (z.score.hPA*sd.hPA) + mean.hPA

head(data.asal.hPA)
```

```
## [1] 5480 5660 5710 5700 5760 5720
```

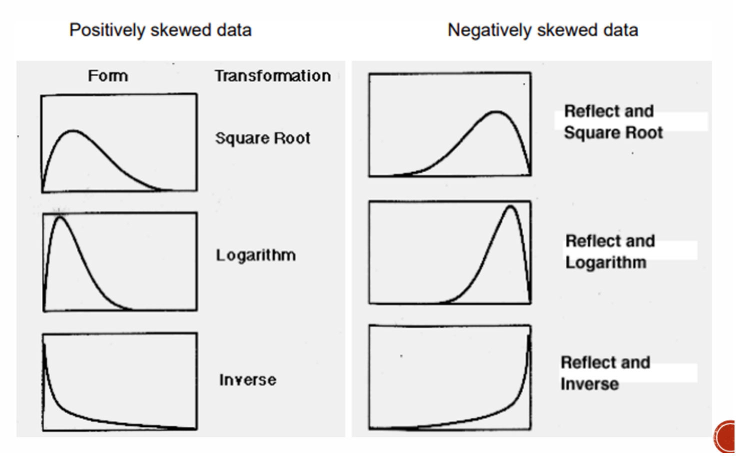
```r
head(pressure_height.hPA)
```

```
## [1] 5480 5660 5710 5700 5760 5720
```

1.1.3 Penormalan berdasarkan penskalaan perpuluhan

```r
pHnew = pressure_height.hPA/10000
vpAnew = Visibility_pAerosol/1000
```

1.2 Penormalan Bentuk Taburan Data



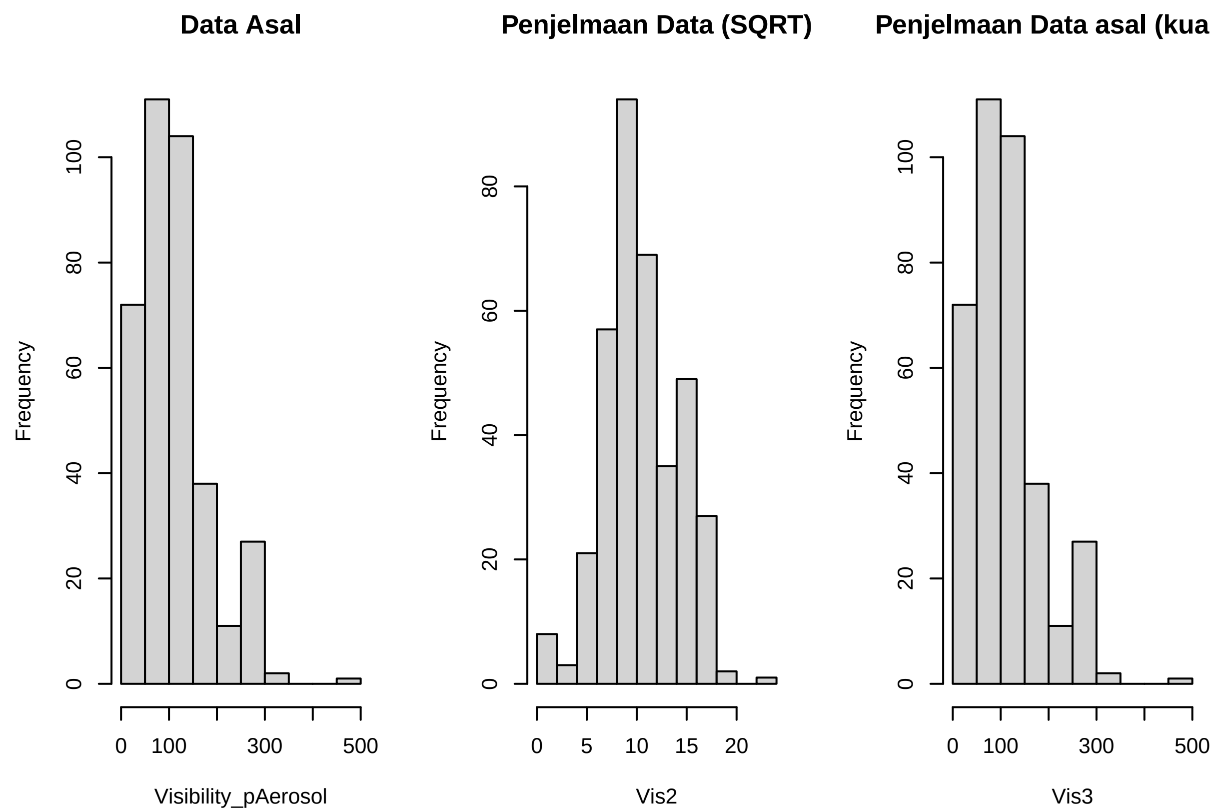
guna data dataAP3

1.2.1 Data Pencong ke kanan

```
Vis2 = sqrt(Visibility_pAerosol)

Vis3 = Vis2^2 # Tukar balik ke data asal

par(mfrow=c(1,3))
hist(Visibility_pAerosol, main="Data Asal")
hist(Vis2, main="Penjelmaan Data (SQRT)")
hist(Vis3, main="Penjelmaan Data asal (kuasa 2)")
```



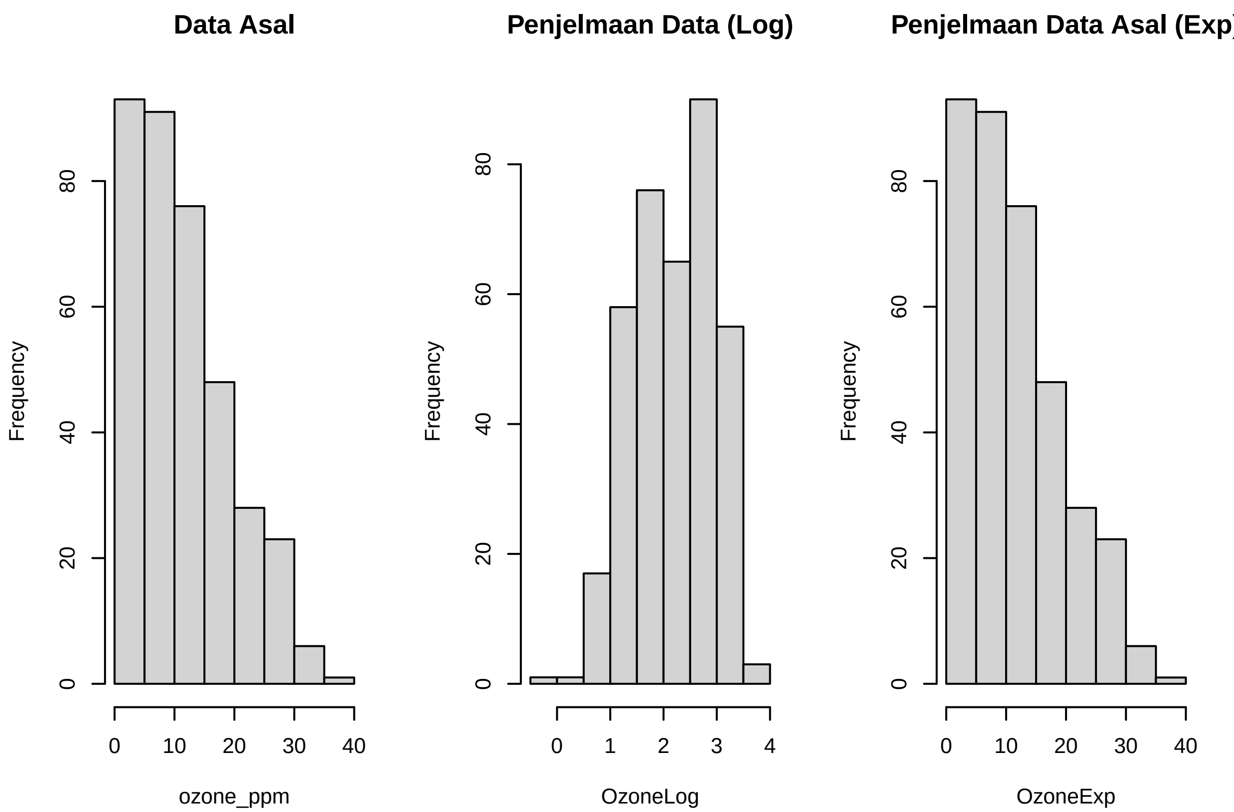
a. Fungsi SQRT

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

```
OzoneLog = log(ozone_ppm)

OzoneExp = exp(OzoneLog) # Tukar balik kepada data asal

par(mfrow=c(1,3))
hist(ozone_ppm, main="Data Asal")
hist(OzoneLog, main="Penjelmaan Data (Log)")
hist(OzoneExp, main="Penjelmaan Data Asal (Exp)")
```



b. Fungsi Log

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

Selepas analisis perlombongan data dijalankan, keputusan analisis perlu dijelmakan kepada data yang asal.

```
OzoneExp = exp(OzoneLog)
```

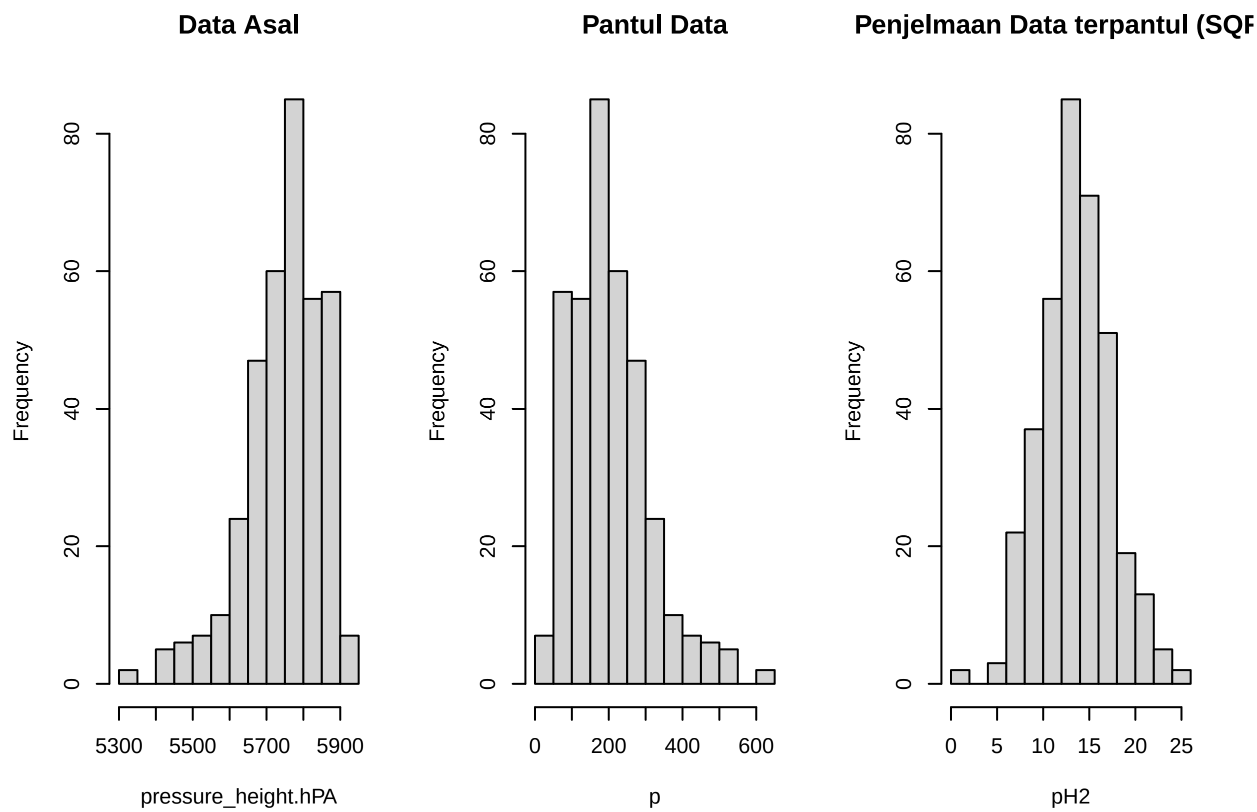
1.2.2 Data Pencong ke Kiri

$$k = (max(x) + 1)$$

```
k = max(pressure_height.hPA) +1 # Dapatkan pusingan data
p = k-pressure_height.hPA

pH2 = sqrt(p) # jelmakan data
```

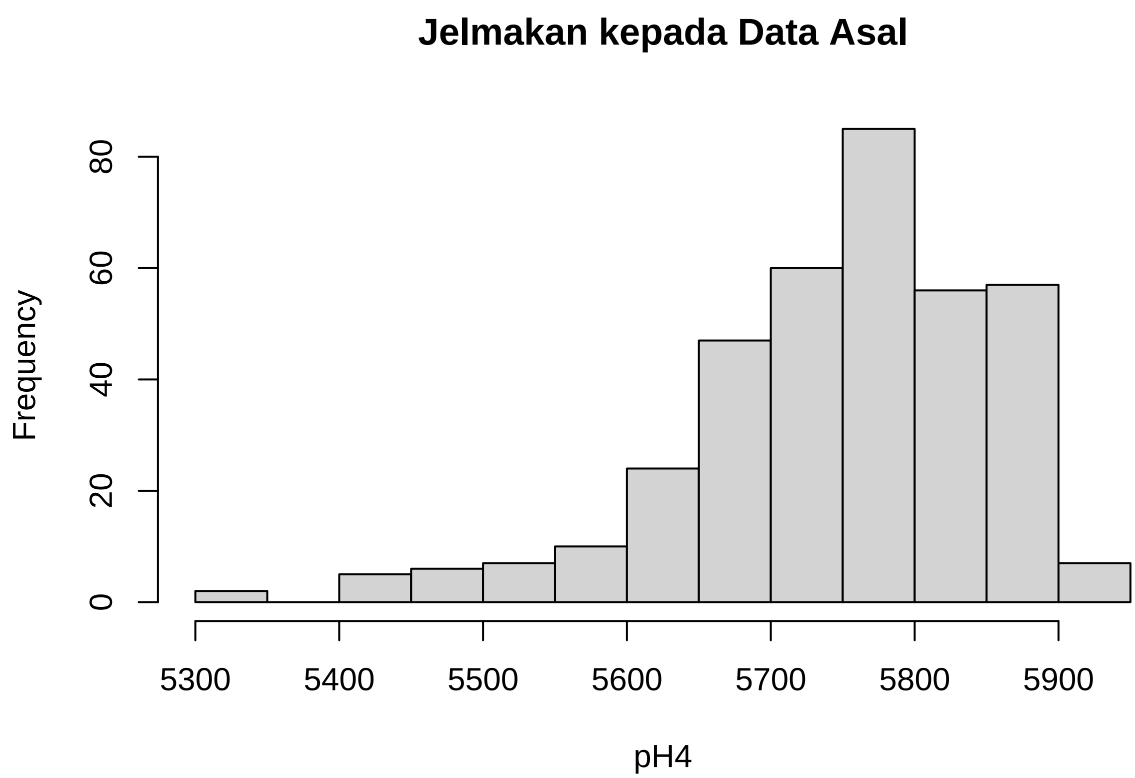
```
par(mfrow=c(1,3))
hist(pressure_height.hPA, main="Data Asal")
hist(p, main="Pantul Data")
hist(pH2, main="Penjelmaan Data terpantul (SQRT)")
```




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

Selepas analisis perlombongan dijalankan, perlu tukarkan kembali kepada data asal

```r
pH3 = pH2^2
pH4 = k-pH3
hist(pH4, main="Jelmakan kepada Data Asal")
```

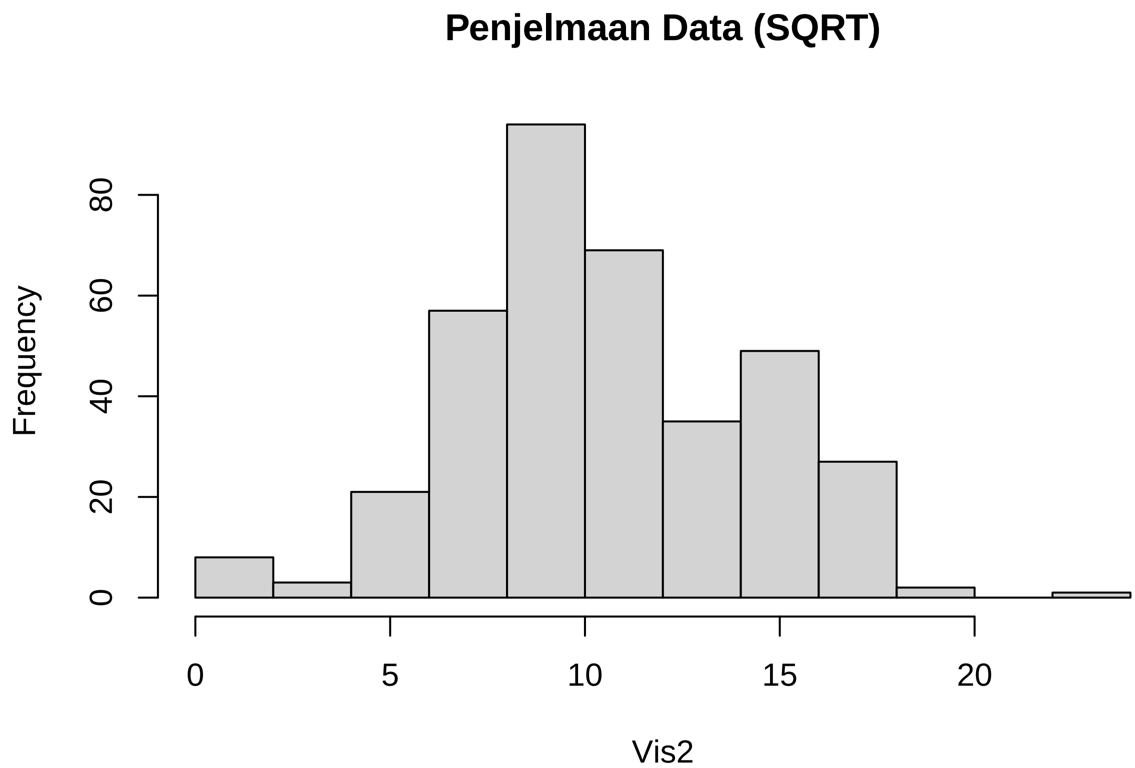


1.3 Kaedah menilai kenormalan taburan data

1.3.1 Pendekatan naif

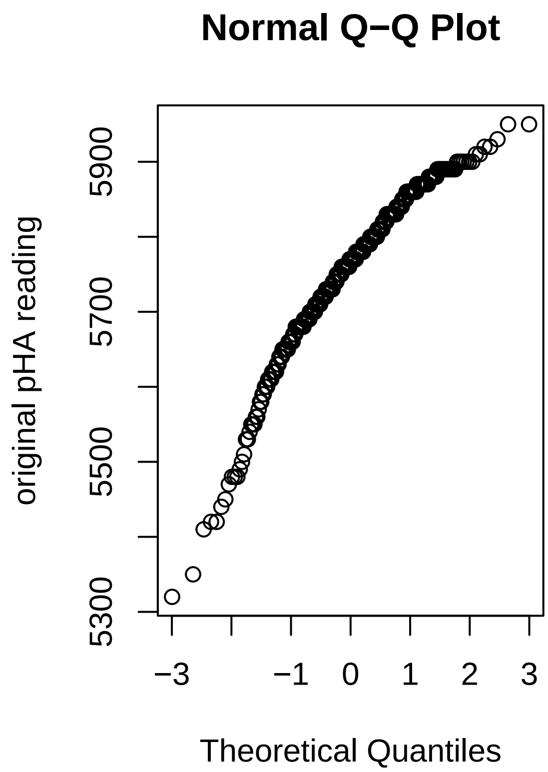
Plot histogram atau plot kotak

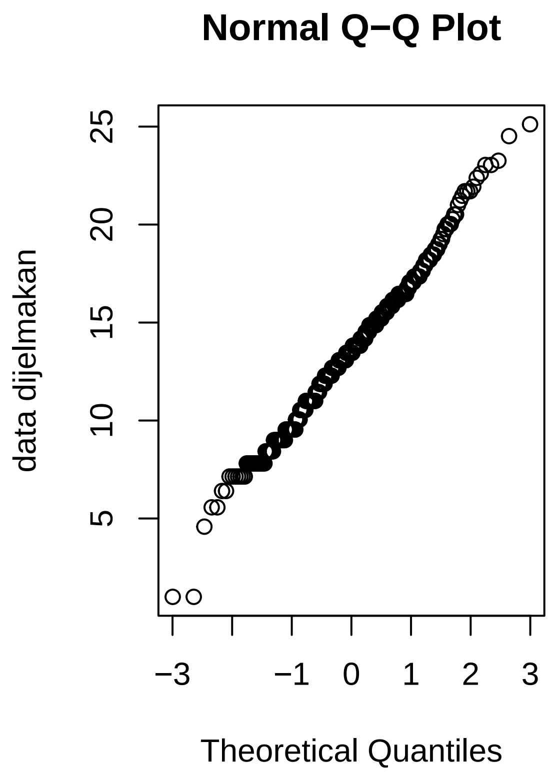
```
hist(Vis2, main="Penjelmaan Data (SQRT)")
```



1.3.2 Plot normal kuantil

```
par(mfrow=c(1,2))
qqnorm(pressure_height.hPA, ylab='original pHA reading')
qqnorm(pH2, ylab='data dijelmakan')
```






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

1.3.3 Ujian Statistik

Ujian Kolmogorov-Smirnov

$$H_0 : Data\ cerapan = taburan\ normal H_1 : Data\ cerapan \neq taburan\ normal$$

```r
n = length(pH2)
x = rnorm(n)
ks.test(pH2, x) # bandingkan data dijelmakan dengan data taburan
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  pH2 and x
## D = 0.99454, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

2. Pendiskretan

Membahagikan data atribut kepada beberapa selang (terselia vs tidak terselia)

- Contoh : pokok-keputusan (*Decision Tree*)
- Selanjar:Jumlah pendapatan,1000< X< 10000.
- Selang:1000-2000,2000-3000,>3000.
- Diskrit/berkategori: 1=pendapatan rendah, 2=pendapatan, sederhana,3=pendapatan tinggi

2.1 Pendiskretan tidak terselia

2.1.1 Menerusi pengetahuan domain - dibuat secara manual

```r
library(infotheo)
data("USArrests")

attach(USArrests)
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

```r
cutoff = 10 # perlu hujah dalam domain knowledge
Status.M = ifelse(Murder<=10, "Low Risk", "High Risk")
```

a. Dua Kategori

b. Lebih dari 2 kategori

1. Boleh guna pernyataan nested ifelse
2. Cara kodkan data

```r
library(car)
```

```
## Loading required package: carData
```

```r
Status.Den = Recode(UrbanPop, "0:50 = 'Low Density';
                              51:70 = 'Moderate Density';
                              else = 'High Density'")
head(Status.Den,5)
```

```
## [1] "Moderate Density" "Low Density"      "High Density"     "Low Density"
## [5] "High Density"
```

2.1.2 Pendiskretan sama lebar (*Equal-width*) - $W = (B - A)/N$.

maklumat minimum (A) dan maksiumum (B) data (X)

```
Assault.Status = discretize(Assault, "equalwidth",4)
unique(Assault.Status)
```

```
##   X
## 1 3
## 3 4
## 4 2
## 7 1
```

2.1.3 Pendiskretan sama kekrapan (*Equal-Frequency*)

```
Rape.Stat = discretize(Rape,'equalfreq', 4)
unique(Rape.Stat)
```

```
##   X
## 1 3
## 2 4
## 4 2
## 7 1
```

Gabungkan Data

```
Discr_data = data.frame(Status.M, Status.Den, Assault.Status,Rape.Stat)
head(Discr_data,10)
```

```
##     Status.M       Status.Den X X.1
## 1  High Risk Moderate Density 3   3
## 2   Low Risk      Low Density 3   4
## 3   Low Risk     High Density 4   4
## 4   Low Risk      Low Density 2   2
## 5   Low Risk     High Density 4   4
## 6   Low Risk     High Density 3   4
## 7   Low Risk     High Density 1   1
## 8   Low Risk     High Density 3   2
## 9  High Risk     High Density 4   4
## 10 High Risk Moderate Density 3   3
```

2.2 Pendiskretan terselia

```
library(discretization)
data(iris)
iris2 = chi2(iris,alp = 0.05,del = 0.05)$Disc.data
head(iris2,5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1            1           3            1           1  setosa
## 2            1           2            1           1  setosa
## 3            1           2            1           1  setosa
## 4            1           2            1           1  setosa
## 5            1           3            1           1  setosa
```

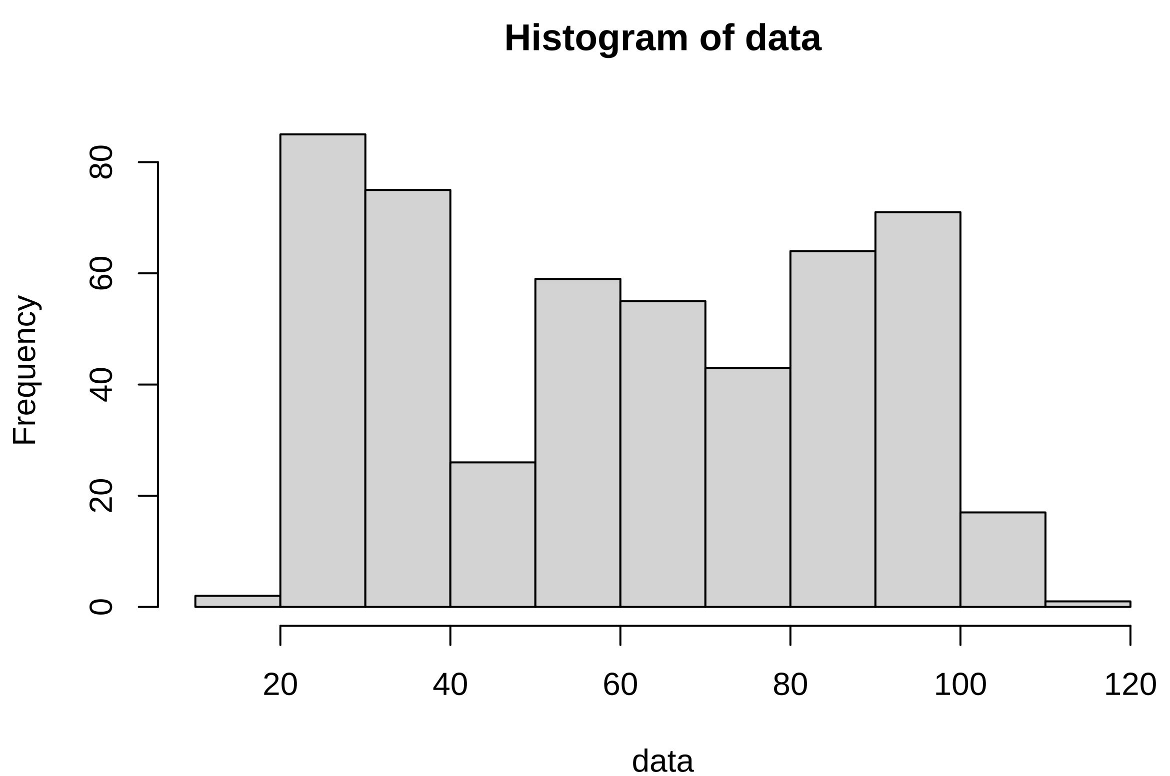
2.3 Pendiskretan Setempat

apabila kita berurusan dengan **data** yang **bersifat heterogen**, dengan segmen data yang berbeza memerlukan strategi pendiskretan yang berbeza

Contoh: Dalam set data dengan **ketumpatan taburan data yang berbeza-beza**, pendiskretan setempat perlu untuk membina;

1. bin-bin yang lebih kecil di kawasan yang mempunyai ketumpatan data yang tinggi
2. bin-bin yang lebih besar di kawasan ketumpatan data yang rendah.

```
datalocal = read.csv("D:/MSc DSc/Sem 1/Data Mining/UKMShape Data-20241201/dataLocal.csv", header=T, sep
data = datalocal[,1]
hist(data)
```



Andaikan ada 3 kategori dalam data

Boleh kenal pasti melalui kaedah pengkelompokan

2.3.1 Pendiskretan data berdasarkan kategori k-means

```r
k = 3
kmeans_C = kmeans(data, centers = k)
kmeans_C
```

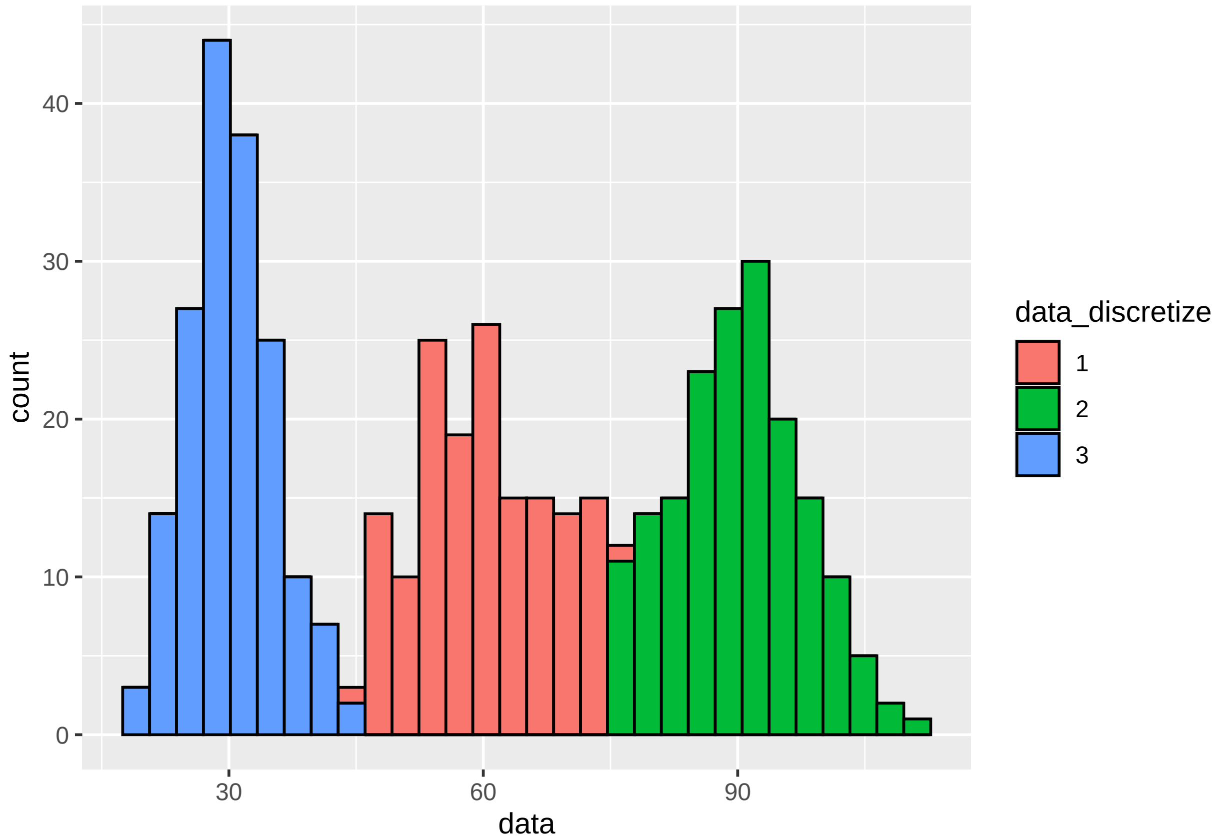
```
## K-means clustering with 3 clusters of sizes 155, 173, 170
##
## Cluster means:
##   [,1]
## 1 60.1
## 2 90.1
## 3 30.3
##
## Clustering vector:
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [112] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [149] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1
## [223] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1
## [260] 3 1 1 1 1 2 2 1 3 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1
## [297] 2 1 1 1 1 1 1 1 1 1 3 1 1 2 1 1 3 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2
## [334] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
## [371] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [408] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [445] 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [482] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1]  9286 10194  4359
##  (between_SS / total_SS =  92.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
data_discretize = as.factor(kmeans_C$cluster)
head(data_discretize,10)
```

```
##  [1] 3 3 3 3 3 3 3 3 3 3
## Levels: 1 2 3
```

```r
data2 = cbind(data, data_discretize)

library(ggplot2)
ggplot(data.frame(data,data_discretize), aes(x=data, fill=data_discretize))+
  geom_histogram(bins=30, color='black')
```



—Latihan—

Jalankan pendiskretan data terhadap data - datalocal p/ubah dataL2

sepatutnya ada 3 class

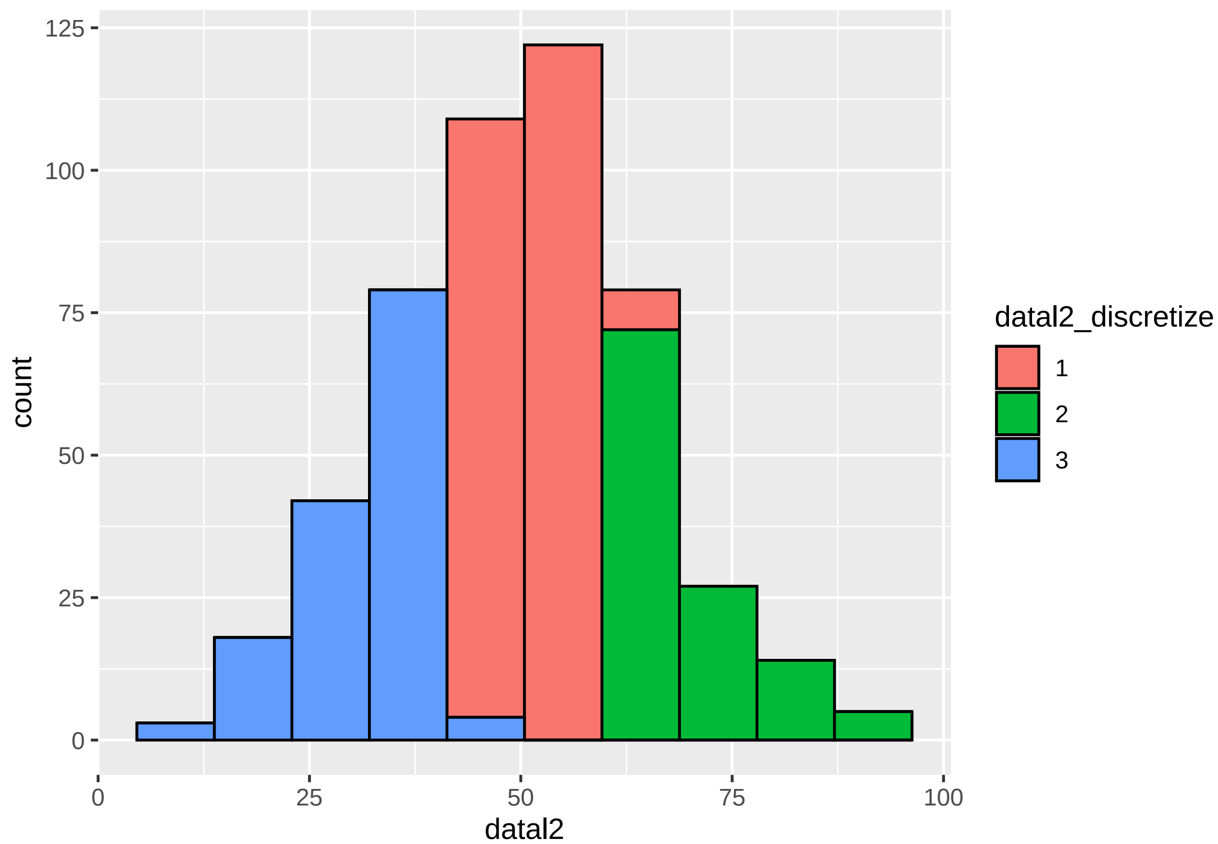
Kaedah K-Means

```
datal2 = datalocal[,2]
#hist(datal2)
k=3

kmeans_dl2 = kmeans(datal2, centers = k)
datal2_discretize = as.factor(kmeans_dl2$cluster)

data3 = cbind(datal2, datal2_discretize)

ggplot(data.frame(datal2,datal2_discretize), aes(x=datal2, fill=datal2_discretize))+
  geom_histogram(bins=10, color='black')
```



Kaedah *Equal-frequency*

```
datal2
```

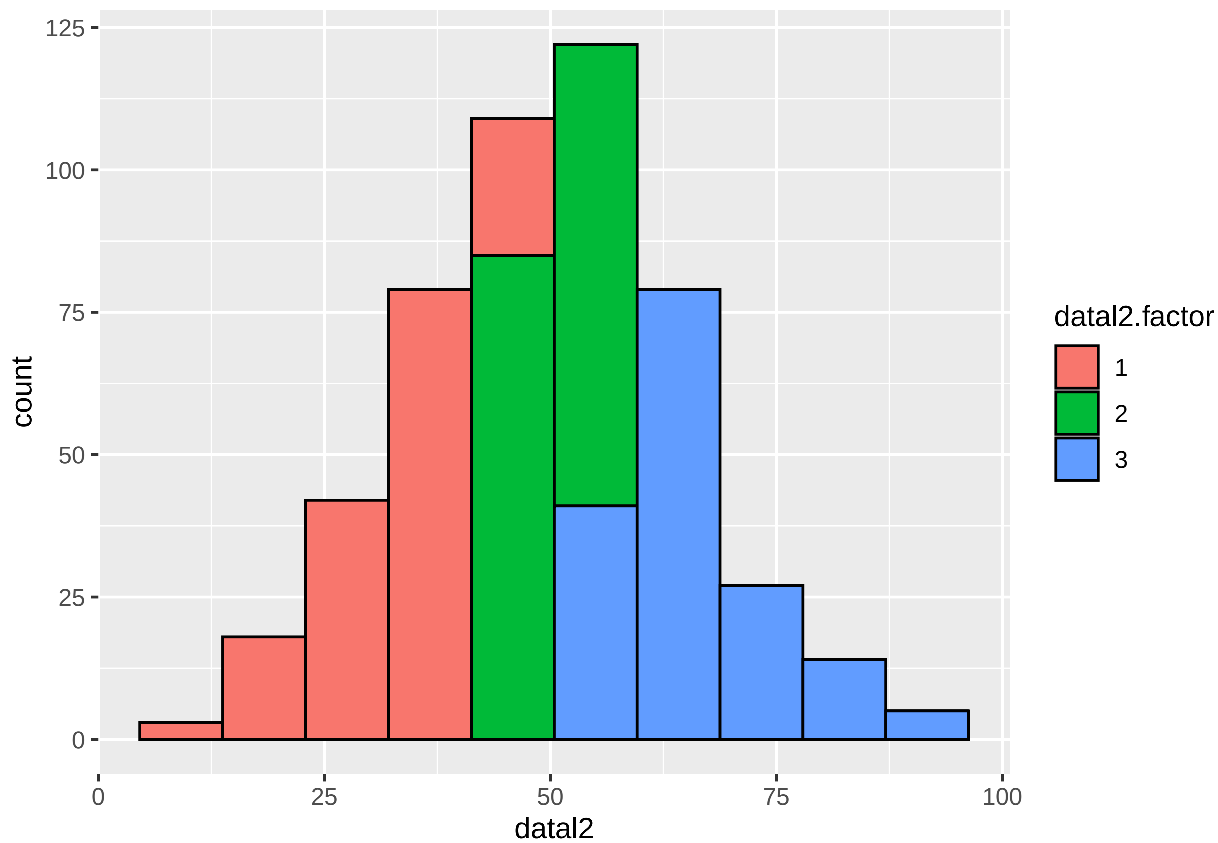
```
##   [1] 55.46 58.28 40.97 35.09 65.40 61.27 27.36 48.57 36.56 18.94 52.25 48.81
##  [13] 48.54 53.24 63.24 53.08 40.75 38.98 48.02 54.65 34.40 47.24 64.51 48.38
##  [25] 39.52 45.86 66.72 58.25 68.55 52.09 56.15 41.62 59.08 42.40 28.69 51.92
##  [37] 79.19 62.01 67.48 55.38 40.87 46.97 45.90 42.97 60.56 32.04 63.00 62.96
##  [49] 32.02 59.59 86.45 41.64 62.67 38.27 66.66 53.75 74.78 28.12 49.23 42.10
##  [61] 47.04 40.56 37.49 58.68 33.69 72.26 32.21 51.52 57.99 58.80 45.47 51.19
##  [73] 64.42 28.15 38.27 54.81 43.33 70.55 60.10 51.08 27.38 50.39 45.25 48.46
##  [85] 32.28 57.48 34.42 46.61 55.72 38.25 58.74 30.25  7.85 56.97 62.61 45.71
##  [97] 57.56 32.66 48.09 20.88 67.72 77.90 66.11 49.59 49.50 27.26 61.86 46.84
## [109] 40.15 28.82 45.50 37.26 44.04 31.74 75.31 49.76 66.12 10.97 43.20 39.87
## [121] 31.66 73.20 28.77 54.78 62.70 52.67 36.87 64.12 52.56 34.05 29.18 81.30
## [133] 39.82 22.17 58.00 54.65 29.69 20.86 48.26 67.09 59.54 42.61 37.49 54.07
## [145] 52.36 59.45 44.06 63.49 37.54 45.04 61.11 64.85 20.92 51.61 59.13 28.24
## [157] 57.21 37.58 65.30 58.08 61.54 51.81 62.95 70.71 79.49 49.57 16.26 50.47
## [169] 53.08 47.67 58.52 65.16 42.23 45.59 55.97 41.75 51.37 20.57 33.20 30.08
## [181] 37.20 39.60 55.73 64.73 39.09 35.05 34.37 43.78 46.41 57.25 45.18 18.82
## [193] 48.63 67.81 67.87 38.17 26.78 86.87 47.56 48.54 56.31 25.79 39.08 26.89
## [205] 39.60 51.78 29.53 58.85 54.34 36.44 53.39 61.22 65.92 46.81 48.60 48.70
## [217] 71.62 66.88 62.52 45.69 55.60 56.05 34.37 24.08 59.63 27.06 50.03 53.75
## [229] 58.46 52.84 39.01 64.80 76.08 63.22 20.85 70.99 49.16 57.87 59.33 48.55
```

```
## [241] 48.87 65.29 60.67 64.85 85.74 59.97 53.11 16.84 90.38 42.76 85.62 55.62
## [253] 73.08 48.35 57.67 53.21 47.21 48.19 65.19 46.98 19.43 47.06 58.10 59.25
## [265] 59.25 24.62 55.53 64.52 69.15 46.63 45.17 72.32 24.98 43.45 56.86 25.73
## [277] 54.19 78.17 49.94 45.82 57.12 45.81 62.20 63.57 50.04 32.35 30.23 41.11
## [289] 61.96 20.63 21.71 40.19 55.92 36.30 63.30 55.00 47.44 62.28 55.83 43.31
## [301] 53.47 59.71 55.34 40.13 62.83 67.29 54.14 52.16 48.87 82.42 54.14 47.63
## [313] 12.38 26.52 48.83 53.09 54.15 62.32 47.09 68.22 36.18 31.87 31.57 61.13
## [325] 48.76 61.85 45.98 41.12 44.47 22.21 32.46 28.37 65.81 41.04 61.84 72.75
## [337] 47.12 54.26 23.73 37.72 50.84 54.49 38.61 90.27 43.12 50.96 59.75 49.61
## [349] 40.35 65.68 74.23 49.55 58.43 48.54 65.25 32.66 84.81 40.95 28.12 44.74
## [361] 52.20 74.35 63.67 52.14 29.16 37.01 47.55 88.30 22.10 66.97 42.09 74.99
## [373] 32.91 52.15 33.51 63.55 72.26 79.26 61.96 77.65 68.70 48.02 57.16 35.42
## [385] 47.22 68.31 58.12 56.86 34.43 40.93 38.53 55.93 35.14 58.43 33.25 77.43
## [397] 56.91 39.48 53.62 44.71 55.57 53.65 34.79 38.13 54.49 74.59 66.27 40.63
## [409] 62.39 49.27 54.52 53.91 88.63 32.22 51.51 23.30 58.85 66.45 71.68 21.12
## [421] 56.19 73.90 43.79 46.82 49.45 55.48 59.98 69.77 48.57 52.94 87.32 56.47
## [433] 52.83 29.87 50.04 46.68 49.83 41.37 39.70 39.19 46.78 70.52 65.74 44.60
## [445] 24.71 37.33 43.13 51.55 40.06 80.10 45.92 31.79 47.88 34.92 52.34 53.50
## [457] 55.33 25.67 53.31 54.66 28.68 64.33 61.76 84.49 52.35 50.70 51.45 51.05
## [469] 22.27 24.93 48.84 41.28 50.82 18.33 27.52 33.48 64.79 33.52 38.01 51.20
## [481] 45.16 52.20 84.58 33.13 45.42 42.25 72.69 38.46 48.77 61.81 34.12 74.83
## [493] 60.14 33.89 56.82 46.80 54.70 48.65
```

```r
datal2.Stat = discretize(datal2,'equalfreq', 3)
datal2.factor = as.factor(datal2.Stat[,1])

data4 = cbind(datal2, datal2.factor)

ggplot(data.frame(datal2,datal2.factor), aes(x=datal2, fill=datal2.factor))+
  geom_histogram(bins=10, color='black')
```



3. Penjelmaan data membentuk atribut baru

3.1 Penjelmaan linear

Teknik ini melibatkan penjelmaan algebra mudah seperti hasil tambah, purata, putaran, dll

Misalkan $A = A_1, A_2, ..., A_n$ ialah set atribut, dan misalkan $B = B_1, B_2, ...B_m$ ialah subset bagi set atribut lengkap A

Atribut baru Z boleh dibentuk menerusi penjelmaan linear berikut:

$$Z = r_i B_i + r_2 B_2 + ... + r_M B_M$$

3.2 Penjelmaan data menerusi pengekodan

3.2.1 Pengekodan satu-hot

```
df = read.csv("D:/MSc DSc/Sem 1/Data Mining/UKMShape Data-20241201/df.csv", header=T, sep=';')
head(df,10)
```

```
##   team points
## 1    A     25
```

```
## 2    A      12
## 3    B      15
## 4    B      14
## 5    B      19
## 6    B      23
## 7    C      25
## 8    C      29
```

Takrifkan fungsi pengekodan satu-hot

```r
library(caret)
```

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

```
## Loading required package: lattice
```

```r
dummy = dummyVars("~.", data=df)
```

Jalankan pengekodan satu-hot

```r
df2 = data.frame(predict(dummy, newdata = df))
head(df2,10)
```

```
##   teamA teamB teamC points
## 1     1     0     0     25
## 2     1     0     0     12
## 3     0     1     0     15
## 4     0     1     0     14
## 5     0     1     0     19
## 6     0     1     0     23
## 7     0     0     1     25
## 8     0     0     1     29
```

3.2.2 Pengekodan Ordinal

```r
demographic = read.csv("D:/MSc DSc/Sem 1/Data Mining/UKMShape Data-20241201/demographic.csv", header=T,
head(demographic,10)
```

```
##   ID   Education_Level Income_Group
## 1  1       High School          Low
## 2  2 Bachelor's Degree       Medium
## 3  3   Master's Degree         High
## 4  4         Doctorate       Medium
## 5  5   Master's Degree         High
## 6  6 Bachelor's Degree          Low
```

kodkan data dalam kategori mengikut hierarki,

tukarkan struktur data kepada yang betul chr –> ord.factor

```r
attach(demographic)
str(demographic) # education_level in chr --> tukar kpd kategori
```

```
## 'data.frame':    6 obs. of  3 variables:
##  $ ID             : int  1 2 3 4 5 6
##  $ Education_Level: chr  "High School" "Bachelor's Degree" "Master's Degree" "Doctorate" ...
##  $ Income_Group   : chr  "Low" "Medium" "High" "Medium" ...
```

```r
Edu_Level = c("High School", "Bachelor's Degree", "Master's Degree", "Doctorate")

demographic$Education_Level = factor(demographic$Education_Level, levels = Edu_Level, ordered = T )
str(demographic)
```

```
## 'data.frame':    6 obs. of  3 variables:
##  $ ID             : int  1 2 3 4 5 6
##  $ Education_Level: Ord.factor w/ 4 levels "High School"<..: 1 2 3 4 3 2
##  $ Income_Group   : chr  "Low" "Medium" "High" "Medium" ...
```

```r
Inc_group = c('Low','Medium','High')

demographic$Income_Group = factor(demographic$Income_Group, levels = Inc_group, ordered = T)
str(demographic)
```

```
## 'data.frame':    6 obs. of  3 variables:
##  $ ID             : int  1 2 3 4 5 6
##  $ Education_Level: Ord.factor w/ 4 levels "High School"<..: 1 2 3 4 3 2
##  $ Income_Group   : Ord.factor w/ 3 levels "Low"<"Medium"<..: 1 2 3 2 3 1
```

Jelmakan nilai faktor kepada berangka

```r
demographic$Education_Level = as.numeric(demographic$Education_Level)
demographic$Income_Group = as.numeric(demographic$Income_Group)
# str(demographic)
head(demographic,10)
```

```
##   ID Education_Level Income_Group
## 1  1               1            1
## 2  2               2            2
## 3  3               3            3
## 4  4               4            2
## 5  5               3            3
## 6  6               2            1
```

3.3 Penjelmaan Pangkat

$$y_i = \Phi^{-1}\left(\frac{r_i - \frac{3}{8}}{m + \frac{1}{4}}\right)$$

```r
car_prices = read.table("D:/MSc DSc/Sem 1/Data Mining/UKMShape Data-20241201/car_prices.txt", header=T)
head(car_prices)
```

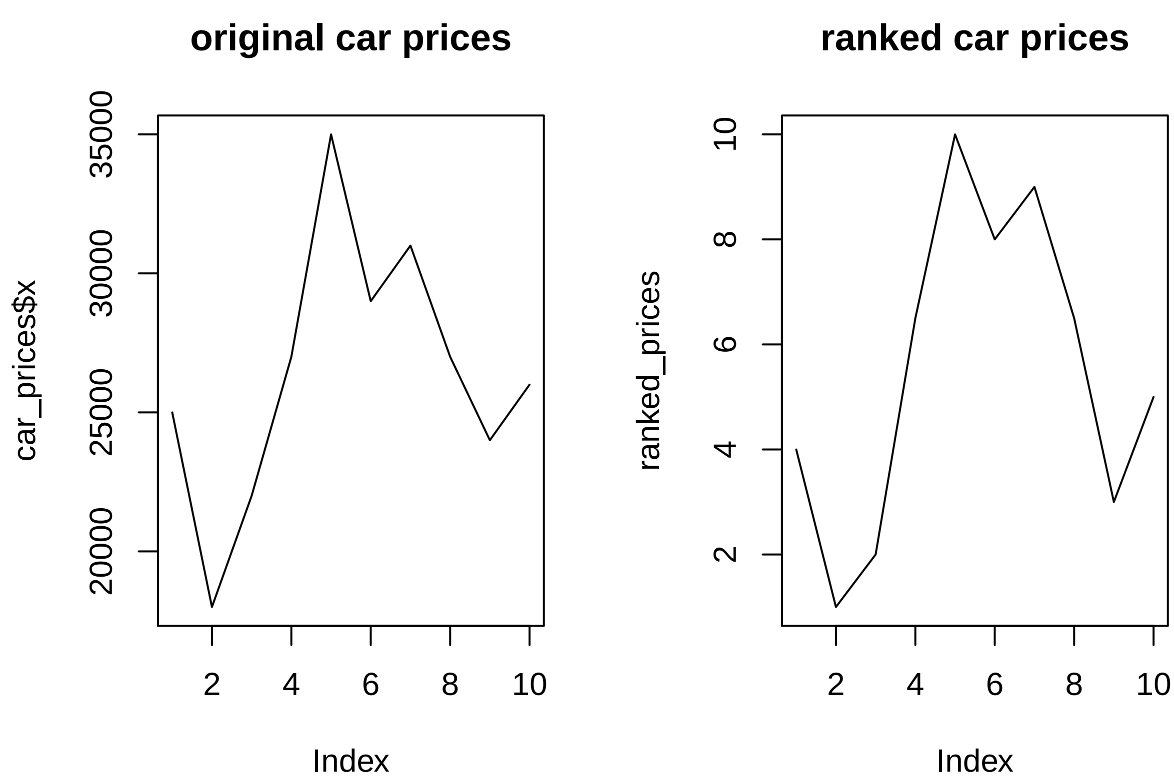
```
##       x
## 1 25000
## 2 18000
## 3 22000
## 4 27000
## 5 35000
## 6 29000
```

Pangkat yang lebih tinggi merujuk kepada harda kereta yang lebih mahal

```r
ranked_prices = rank(car_prices$x)
ranked_prices
```

```
##  [1]  4.0  1.0  2.0  6.5 10.0  8.0  9.0  6.5  3.0  5.0
```

```r
par(mfrow=c(1,2))
plot(car_prices$x, type='l', pch=16,
     main = 'original car prices')
plot(ranked_prices, type='l', pch=16,
     main = 'ranked car prices')
```



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

3.3 Penjelmaan Box-Cox

$$y = \begin{cases} (x^{\lambda} - 1)/\lambda \ , & \lambda = 0 \\ \log(x), & \lambda \neq 0 \end{cases}$$

Terhad kepada data bukan negatif

Nilai λ terbaik dipilih jika didapati taburan menghampiri normal.

```r
library(AID)
```

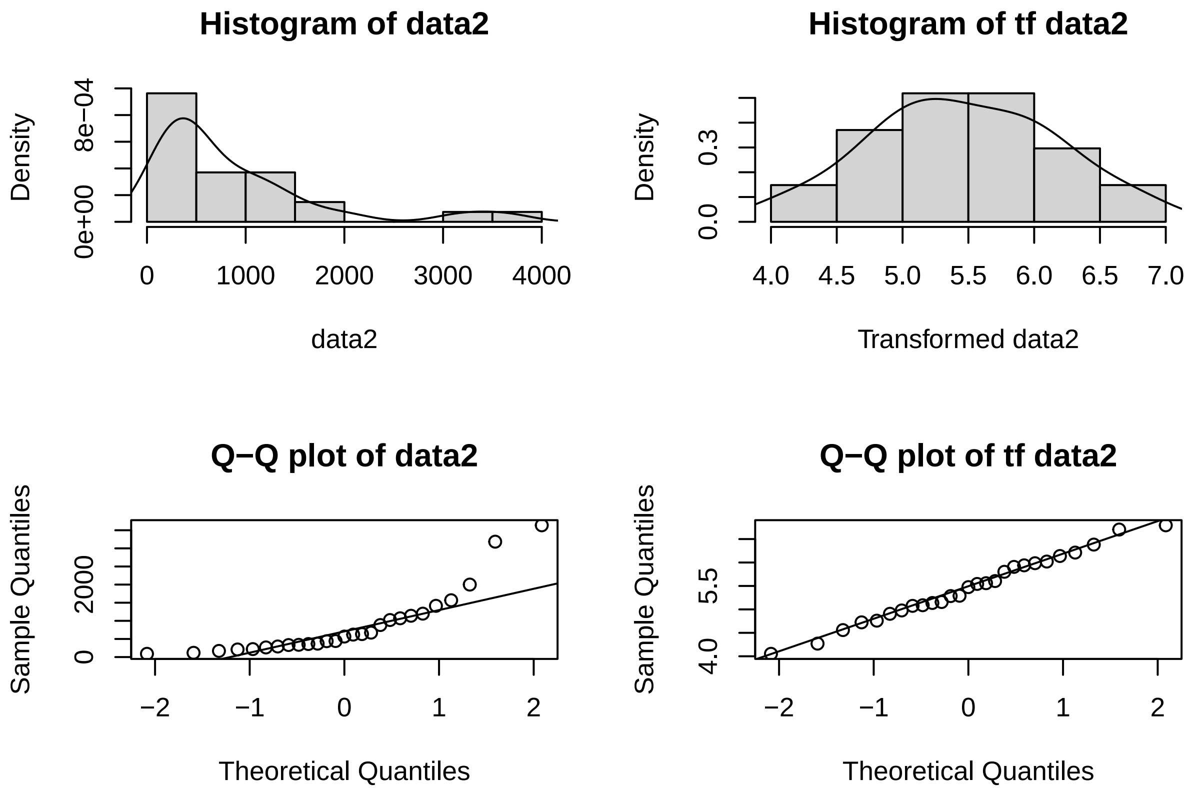
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
data(textile)
str(textile)
```

```
## 'data.frame':    27 obs. of  1 variable:
##  $ textile: int  674 370 292 338 266 210 170 118 90 1414 ...
```

```r
hist(textile[,1])
```

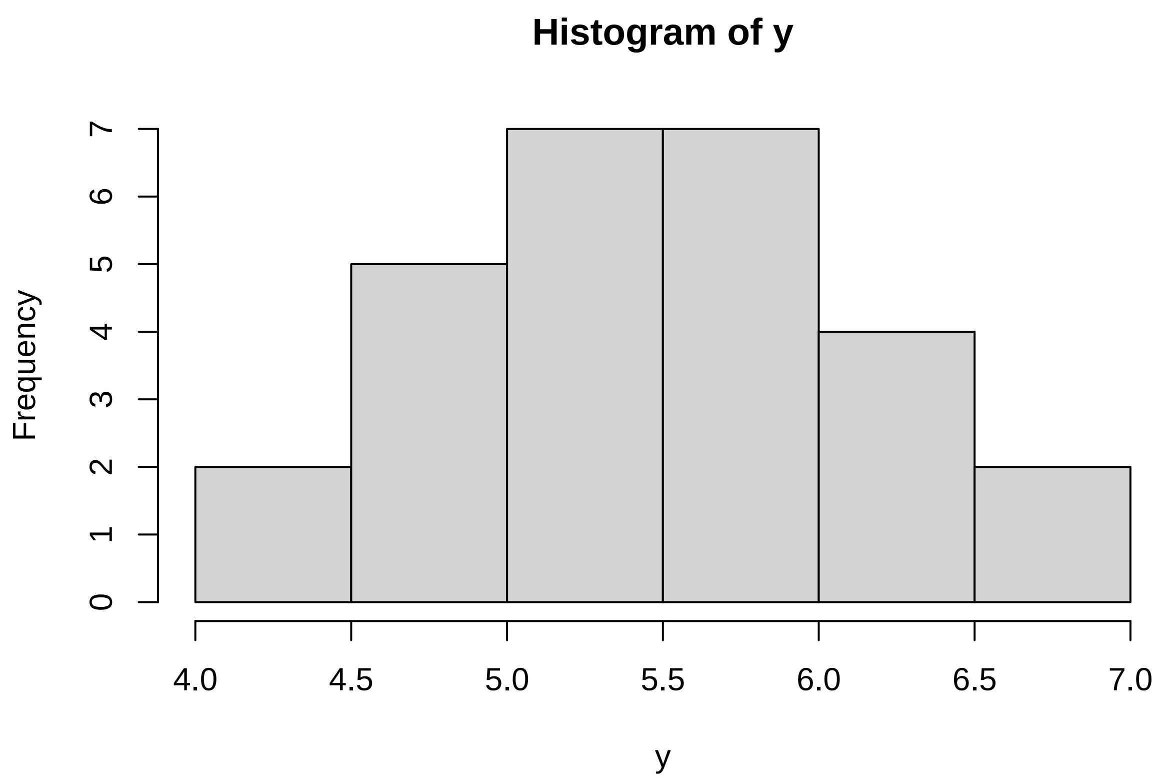

```
data2=textile[,1]
out = boxcoxnc(data2, method='mle',lambda = seq(-2,2,0.0001))
```




```
##
##   Box-Cox power transformation
## -------------------------------------------------------------------
##
##   lambda.hat : -0.0474
##
##
##   Shapiro-Wilk normality test for transformed data (alpha = 0.05)
## -------------------------------------------------------------------
##
##   statistic  : 0.988
##   p.value    : 0.982
##
##   Result     : Transformed data are normal.
## -------------------------------------------------------------------
```

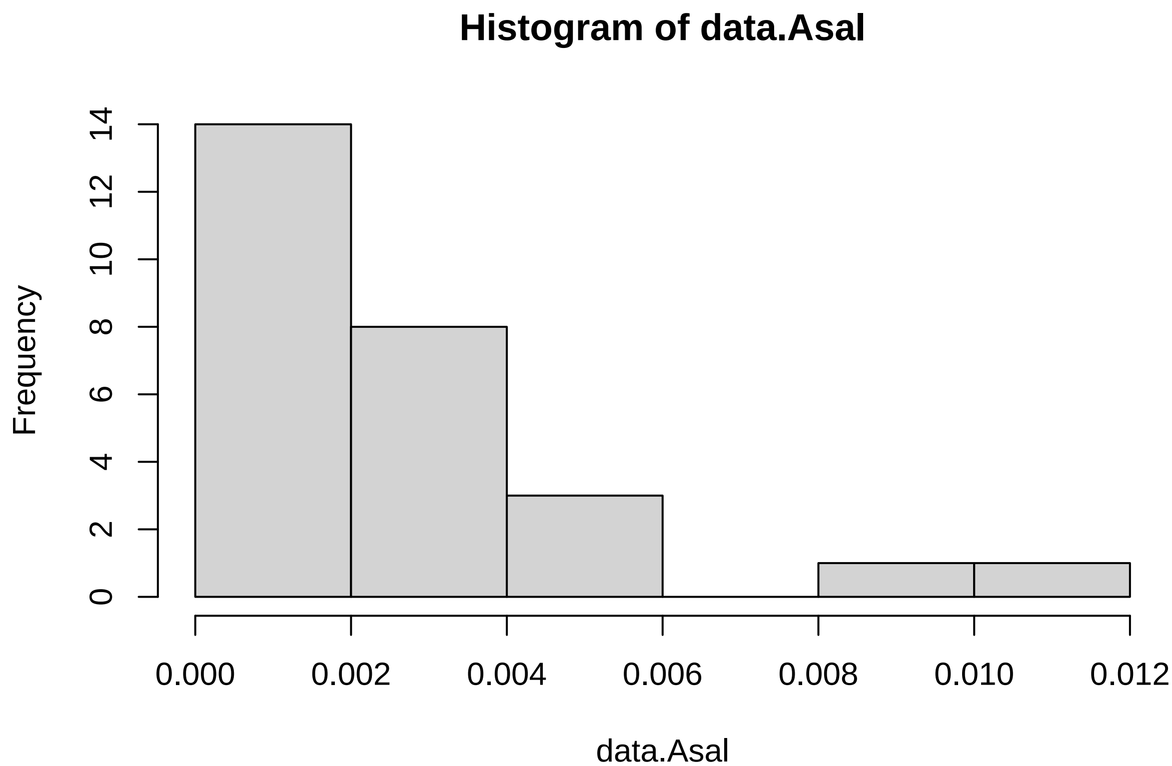
Data yang dijelmakan kepada taburan normal

```
y = (data2^(-0.0474)-1)/(-0.0474)
hist(y)
```



Selepas analisis perlombongan data dijalankan, keputusan analisis perlu dijelmakan kepada bentuk data yang asal

```
data.Asal = (y*(-0.0474)+1)^(1/0.0474)
hist(data.Asal)
```


Histogram of data.Asal
Frequency
0
2
4
6
8
10
12
14
0.000
0.002
0.004
0.006
0.008
0.010
0.012
data.Asal