PEPERIKSAAN AKHIR STQD6414

P152419

2025-02-22

Contents

Question 1

a) i) Penurunan data merupakan pendekatan untuk menurunkan dimensi ataupun bilangan data. Contoh kaedah penurunan dimensi data adalah Analisis Faktor dan Principal Component Analisis (PCA). Manakalah contoh bagi penurunan bilangan data adalah membuang data yang tidak relevan. Antara tujuan penurunan data adalah bagi mengatasi masalah autokorelasi di antara atribut-atribut dalam sesuatu data tersebut.

ii) Pendiskretan data pula merupakan pendekatan bagi mengubah data daripada bentuk nombor kepada bentuk kategori. Ini bertujuan bagi menyesuaikan data mengikut model pembelajaran mesin yang ingin dilakukan.

b) i)

```
# load data
dm1 = read.csv('./Data/data.DM1.csv')
dm2 = read.table('./Data/data.DM2.txt')
head(dm1)
```

```
##     X state.of.res ID.Customer sex Working       marital.stat ins.health
## 1 734         Ohio     1057778   F      NA      Never Married       TRUE
## 2 480    Minnesota       33651   F    TRUE      Never Married       TRUE
## 3 547   New Jersey     1181596   M    TRUE            Married       TRUE
## 4 539       Nevada      867842   F      NA            Widowed       TRUE
## 5 148   California      863391   M    TRUE            Married       TRUE
## 6 466     Michigan      184686   F    TRUE Divorced/Separated       TRUE
##                   Home.Status recent.move
```

```
## 1                       Rented       FALSE
## 2 Homeowner with mortgage/loan       FALSE
## 3 Homeowner with mortgage/loan       FALSE
## 4     Homeowner free and clear       FALSE
## 5                       Rented       FALSE
## 6     Homeowner free and clear       FALSE
```

```
colnames(dm1)
```

```
## [1] "X"            "state.of.res" "ID.Customer"  "sex"          "Working"
## [6] "marital.stat" "ins.health"   "Home.Status"  "recent.move"
```

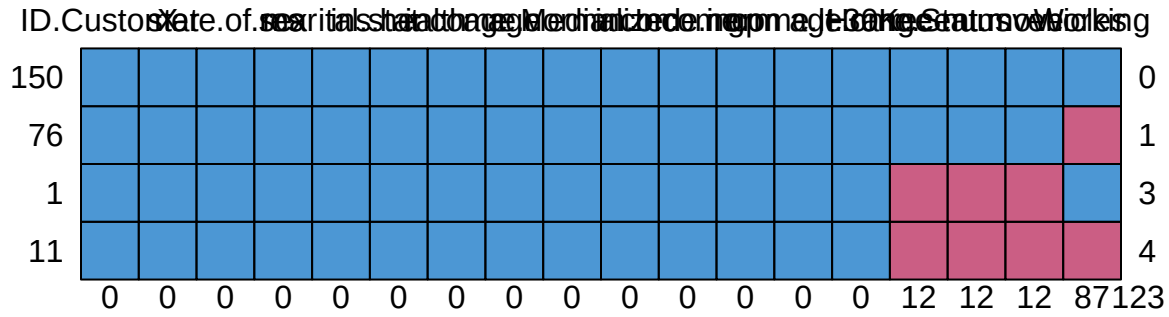
```
colnames(dm2)
```

```
##  [1] "cust.id"        "income"         "num.vehicles"   "age"
##  [5] "age.normalized" "Median.Income"  "income.norm"    "gp"
##  [9] "income.lt.30K"  "age.range"
```

```
data = merge(dm1, dm2, by.x = 'ID.Customer', by.y = 'cust.id')
head(data)
```

```
##   ID.Customer   X   state.of.res sex Working       marital.stat ins.health
## 1        2068 248       Illinois   F      NA            Married       TRUE
## 2        5641 635       New York   M    TRUE      Never Married      FALSE
## 3        6369 704 North Carolina   F    TRUE      Never Married       TRUE
## 4        8322  85     California   F    TRUE      Never Married       TRUE
## 5       14989 793   Pennsylvania   M      NA            Married       TRUE
## 6       17946 245          Idaho   F    TRUE Divorced/Separated       TRUE
##                   Home.Status recent.move income num.vehicles age
## 1     Homeowner free and clear       FALSE  11300            2  49
## 2        Occupied with no rent       FALSE  20000            0  22
## 3                       Rented        TRUE  12000            1  31
## 4 Homeowner with mortgage/loan       FALSE 180000            1  40
## 5                       Rented       FALSE   9400            2  44
## 6                       Rented       FALSE  85000            1  51
##   age.normalized Median.Income income.norm         gp income.lt.30K age.range
## 1     -0.1431242         49293   0.2292415 0.52462028          TRUE   (25,65]
## 2     -1.5744649         44819   0.4462393 0.49471258          TRUE    [0,25]
## 3     -1.0973514         52683   0.2277775 0.06606553          TRUE   (25,65]
## 4     -0.6202378         39832   4.5189797 0.78161393         FALSE   (25,65]
## 5     -0.4081873         52758   0.1781720 0.12305510          TRUE   (25,65]
## 6     -0.0370990         61308   1.3864422 0.53752766         FALSE   (25,65]
```

```
library(mice)
md.pattern(data)
```




```
##     ID.Customer X state.of.res sex marital.stat ins.health income age
## 150           1 1            1   1            1          1      1   1
## 76            1 1            1   1            1          1      1   1
## 1             1 1            1   1            1          1      1   1
## 11            1 1            1   1            1          1      1   1
##               0 0            0   0            0          0      0   0
##     age.normalized Median.Income income.norm gp income.lt.30K age.range
## 150              1             1           1  1             1         1
## 76               1             1           1  1             1         1
## 1                1             1           1  1             1         1
## 11               1             1           1  1             1         1
##                  0             0           0  0             0         0
##     Home.Status recent.move num.vehicles Working
## 150           1           1            1       1   0
## 76            1           1            1       0   1
## 1             0           0            0       1   3
## 11            0           0            0       0   4
##              12          12           12      87 123
```

```
table(data$Working)
```

```
##
## FALSE  TRUE
##    14   137
```

```r
table(data$Home.Status)
```

```
##
##     Homeowner free and clear Homeowner with mortgage/loan
##                           48                           94
##        Occupied with no rent                       Rented
##                            5                           79
```

```r
table(data$recent.move)
```

```
##
## FALSE  TRUE
##   196    30
```

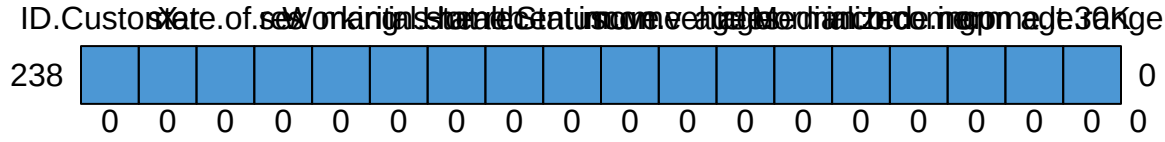
```r
vehicle = median(data$num.vehicles, na.rm = T)
vehicle
```

```
## [1] 2
```

```r
data$Working = ifelse(is.na(data$Working), TRUE ,data$Working) #logical
data$Home.Status = ifelse(is.na(data$Home.Status), 'Homeowner with mortgage/loan',
                          data$Home.Status) # category
data$recent.move = ifelse(is.na(data$recent.move), FALSE,
                          data$recent.move) # logical
data$num.vehicles = ifelse(is.na(data$num.vehicles), vehicle,
                           data$num.vehicles) # integer
```

```r
md.pattern(data)
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##     ID.Customer X state.of.res sex Working marital.stat ins.health Home.Status
## 238           1 1            1   1       1            1          1           1
##               0 0            0   0       0            0          0           0
##     recent.move income num.vehicles age age.normalized Median.Income
## 238           1      1            1   1              1             1
##               0      0            0   0              0             0
##     income.norm gp income.lt.30K age.range
## 238           1  1             1         1 0
##               0  0             0         0 0
```

ii.

```
library(dplyr)

# maklumat individu bekerja
data1 = data %>%
  filter(Working == TRUE) %>%
  select('state.of.res', 'age', 'Median.Income', 'marital.stat', 'Home.Status', 'sex')

# maklumat individu tidak bekerja
data2 = data %>%
  filter(Working == FALSE) %>%
  select('state.of.res', 'age', 'Median.Income', 'marital.stat', 'Home.Status', 'sex')
```

Question 2 - Perlombongan Aturan Sekutuan

a.

algoritma apriori digunakan bagi melihat pola dan aturan dalam data yang mempunyai konsep support, confidence dan juga lift

- Support : peratusan transaksi yang mempunyai barang yang tertentu
- Confidence : peratusan transaksi yang mempunyai barang A juga mempunya barang B
- Lift : Kebarangkalian barang A dibeli apabila barang B dibeli

b.

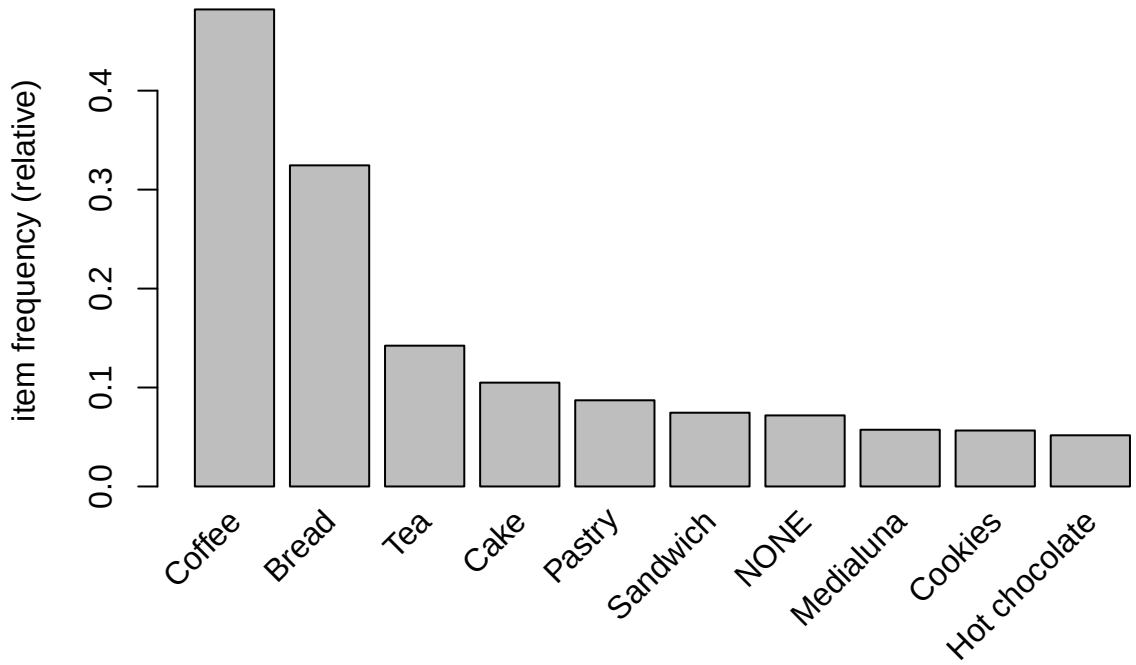
i.

```
library(arules)
bread = read.transactions('./Data/BreadBasket.csv', format = 'single', header = T,
                          sep = ',', cols = c('Transaction','Item'))

arules::inspect(arules::unique(head(bread)))
```

```
##     items                           transactionID
## [1] {Bread}                         1
## [2] {Medialuna, Scandinavian}       10
## [3] {Chimichurri Oil, Scandinavian} 1000
## [4] {Bread, Truffles}               1001
## [5] {Brownie, Focaccia}             1002
## [6] {Bread, Coffee}                 1003
```

ii.

```
#
itemFrequencyPlot(bread, topN = 10)
```



iii.

```
rule = apriori(bread, parameter=list(supp=0.01, conf=0.2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 66
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[103 item(s), 6613 transaction(s)] done [0.00s].
## sorting and recoding items ... [30 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [36 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

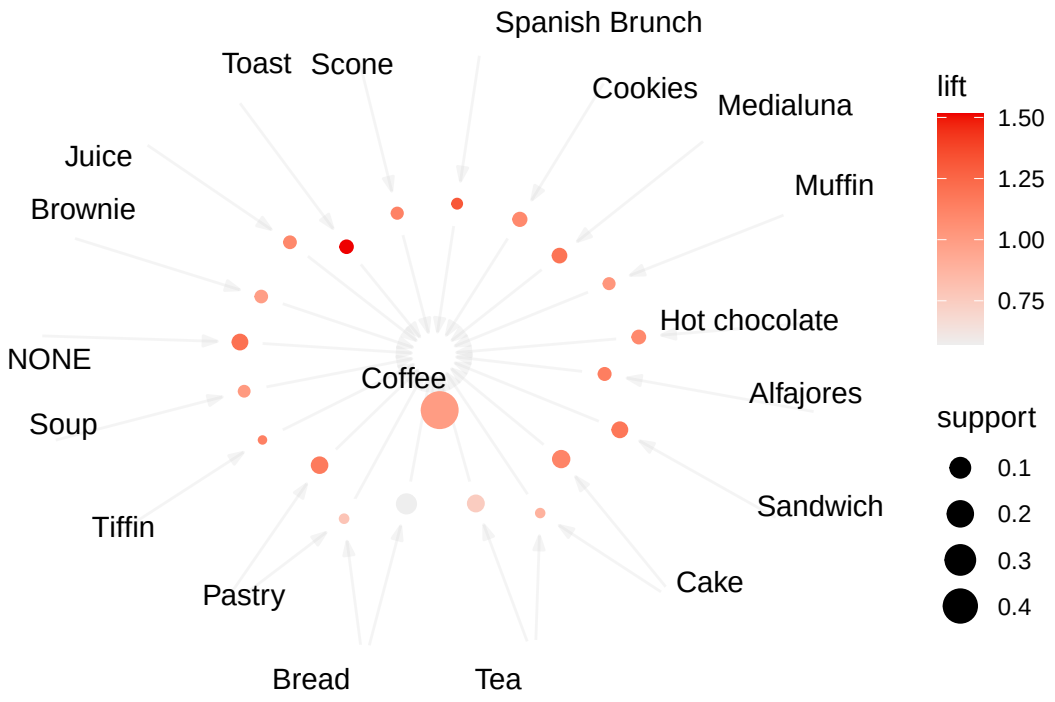
iv.

```
rule = apriori(bread, parameter=list(supp=0.01, conf=0.2),
               appearance=list(default='lhs', rhs='Coffee'))
```

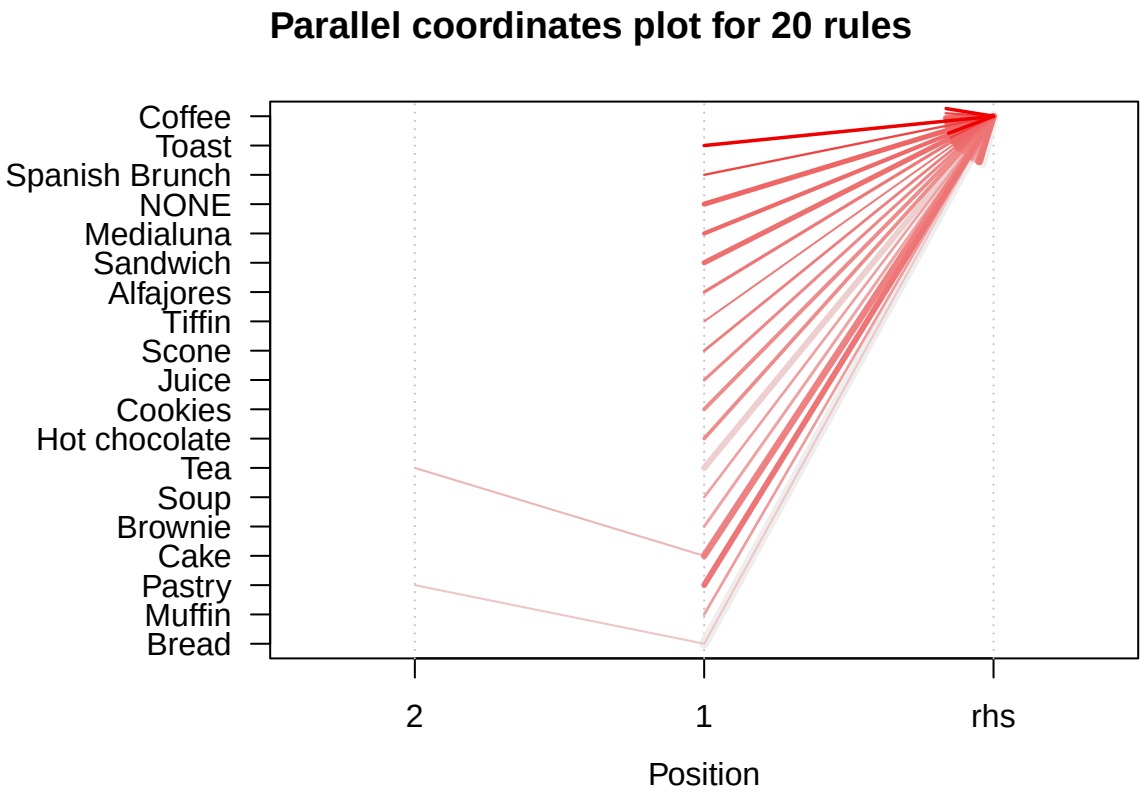
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 66
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[103 item(s), 6613 transaction(s)] done [0.00s].
## sorting and recoding items ... [30 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [21 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

v.

```
library(arulesViz)
plot(rule, method = 'graph')
```




```
plot(rule, method = 'paracoord')
```



Question 3

a) a) Spatial_Lines merupakan bentuk data reruang di mana setiap titik kedudukan akan disambungkan kepada titik kedudukan yang seterusnya menggunakan garis

b) Saptial_Polygon pula merupakan bentuk data reruang di mana ianya seperti bentuk spatial line tetapi tidak mempunyai titik yang tidak bersambung. Ini kerana, titik terakhir akan disambungkan kepada titik pertama.

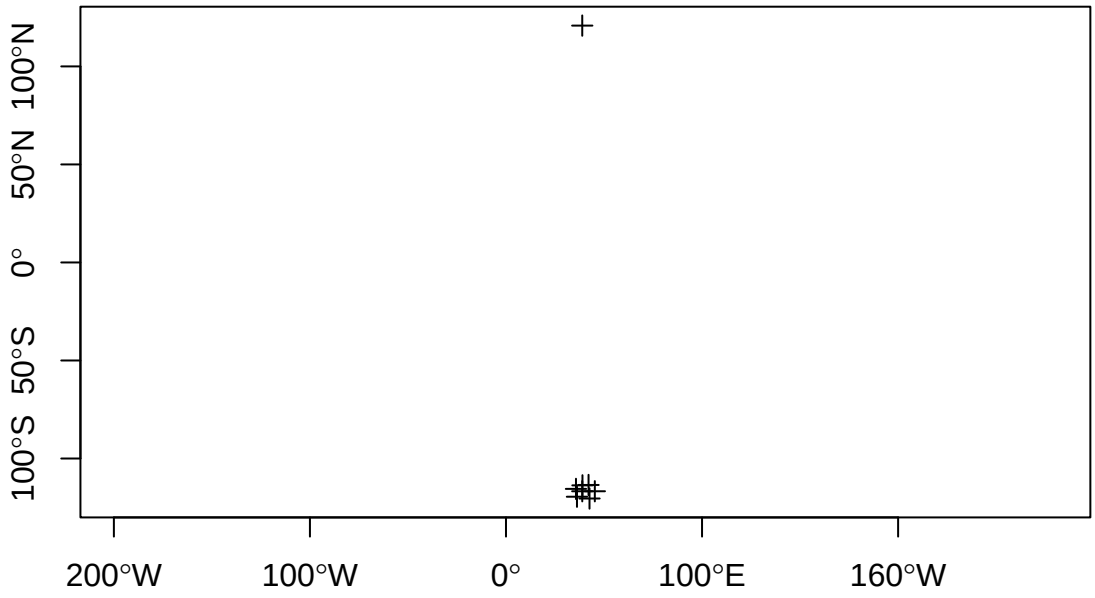
b) a)

```
stesen = c('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H')
Latitud = c(45.3, 42.6, 38.9, 42.1, 35.7, 38.9, 36.2, 39)
Longitud = c(-116.7, -120.4, -116.7, -113.5, -115.5, 120.8, -119.5, -113.7)
Suhu = c(40.5, 32.1, 14.4, 40.1, 33.2, 27.4, 27.8, 31.3)
Jumlah.Hujan.Mingguan = c(184.85, 300.11, 3.53, 405.67, 94.78, 794.84, 154.67,
                          594.85)

reruang = data.frame(stesen, Latitud, Longitud, Suhu, Jumlah.Hujan.Mingguan)

head(reruang)
```

```
##   stesen Latitud Longitud Suhu Jumlah.Hujan.Mingguan
## 1      A    45.3   -116.7 40.5                184.85
## 2      B    42.6   -120.4 32.1                300.11
```

```
## 3      C    38.9   -116.7 14.4                3.53
## 4      D    42.1   -113.5 40.1              405.67
## 5      E    35.7   -115.5 33.2               94.78
## 6      F    38.9    120.8 27.4              794.84
```

```
library(sp)
library(rspat)
crdref = CRS('+proj=longlat +datum=WGS84')
lonlat = cbind(reruang$Latitud, reruang$Longitud)
pts = SpatialPoints(lonlat, proj4string = crdref)
plot(pts, axes = T)
```



ii.

```
model = glm(Suhu~Jumlah.Hujan.Mingguan, data = reruang)
summary(model)
```

```
##
## Call:
## glm(formula = Suhu ~ Jumlah.Hujan.Mingguan, data = reruang)
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           28.887986   4.971473   5.811  0.00114 **
## Jumlah.Hujan.Mingguan  0.006196   0.012298   0.504  0.63235
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 76.39801)
##
##     Null deviance: 477.78  on 7  degrees of freedom
## Residual deviance: 458.39  on 6  degrees of freedom
## AIC: 61.089
##
## Number of Fisher Scoring iterations: 2
```

Question 4 - Perlombongan Data Graf

a) Kemodularan merupakan ukuran struktur rangkaian dengan nod-nod mempamerkan pengelompokan jika terdapat ketumpata yang lebih besar dalam kelompok atau kurang ketumpatan di antara mereka.

b)

c)

```
library(UserNetR)
data("DHHS")
class(DHHS)
```

```
## [1] "network"
```

Analisis Nod Prominen

```
library(statnet)
degree(DHHS)
```

```
##  [1]  2 14 28 14 46 42 42 48 76 14 44 66 38 22 42 28 68 88 14  8 12 32 28 14 14
## [26] 22 30 58 26 36 60 48 50 38 30 42 52 40 30 38 36 30 10 10 20 10  8 20 24 34
## [51] 90 28  6 18
```

```
closeness(DHHS)
```

```
##  [1] 0.3486842 0.5300000 0.5698925 0.5247525 0.6309524 0.6162791 0.6162791
##  [8] 0.6385542 0.7681159 0.5300000 0.6235294 0.7260274 0.6022727 0.5408163
## [15] 0.6091954 0.5578947 0.7260274 0.8412698 0.5300000 0.4953271 0.5196078
## [22] 0.5824176 0.5638298 0.5247525 0.5196078 0.5463918 0.5698925 0.6794872
## [29] 0.5638298 0.5955056 0.6883117 0.6385542 0.6463415 0.6022727 0.5760870
## [36] 0.6162791 0.6543210 0.6091954 0.5760870 0.6022727 0.5955056 0.5760870
## [43] 0.4491525 0.5047619 0.5408163 0.5145631 0.5000000 0.5408163 0.5578947
## [50] 0.5955056 0.8688525 0.5698925 0.4732143 0.5353535
```

```
betweenness(DHHS)
```

```
##  [1]   0.0000000 106.4528257  11.3707189   2.6059524  51.3802870  30.1855141
##  [7]  17.0707973  34.4422443 174.4299595   1.2852878  21.4819972 167.5568684
```

```
## [13]  20.0207363   7.6950660  47.5979001   5.5017039 118.7149057 307.8897502
## [19]   3.9213868   0.4000000   1.0943639  59.6929055  17.8814138   1.7023810
## [25]   4.4953620   1.4249184  10.9751848  40.3420442   0.7916667   5.2271451
## [31] 112.3095927  18.9046515  21.0505787   6.8874891  11.0608050   7.8341722
## [37]  30.4186671  28.3216278   4.7907232  11.6730818  19.2547899   8.9753299
## [43]   0.5833333   1.0333333   9.4411236   1.0583333   0.3809524   2.7732026
## [49]  28.1754690  37.9170256 469.8340280   8.6162706   0.0000000   1.0741323
```

iii)

b)

Question 5 - Perlombongan Data Teks

a) lexicon merupakan senarai perkataan yang berperanan untuk menentukan sesuatu perkataan tersebut merupakan perkataan yang positif ataupun perkataan yang negatif.

b) Aplikasi analisis sentiment dapat mengklasifikasikan perkataan kepada 8 emosi asas iaitu *Anger, Fear, Anticipation, Trust, Surprise, Sadness, Joy, Disgust.*

c)

```
text = readLines('./Data/data.txt')

class(text)
```

```
## [1] "character"
```

```
library(tm)

docs=Corpus(VectorSource(text))
inspect(head(docs))
```

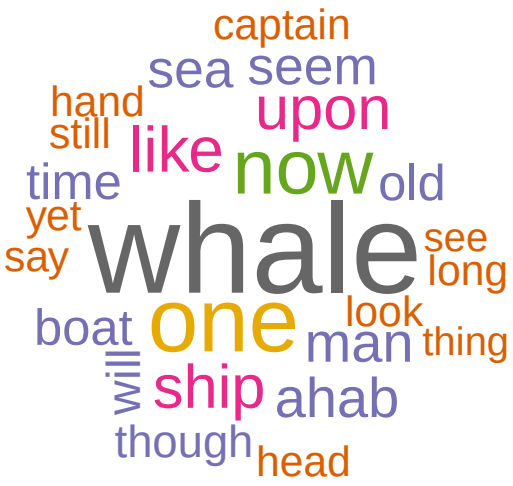
```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 6
##
## [1] MOBY DICK; OR THE WHALE                         By Herman Melville
## [4]                         CHAPTER 1. Loomings.
```

```
tospace = content_transformer(function(x, pattern) gsub(pattern, '', x))

doc2 = tm_map(docs, tospace, '!')
doc3 = tm_map(doc2, tospace, ':')
doc4 = tm_map(doc3, tospace, ',')
doc5 = tm_map(doc4, content_transformer(tolower))
doc6 = tm_map(doc5, removeNumbers)
doc7 = tm_map(doc6, removeWords, stopwords('english'))
doc8 = tm_map(doc7, removePunctuation)
doc9 = tm_map(doc8, stripWhitespace)
doc10 = tm_map(doc9, stemDocument)
```

```
dtm = TermDocumentMatrix(doc10)
m = as.matrix(dtm)

dim(m)
```

```
## [1] 11664 21212
```

```
v = sort(rowSums(m), decreasing = T)
d = data.frame(word = names(v), freq = v)
library(wordcloud)
wordcloud(words = d$word, freq = d$freq, min.freq = 50, max.words = 25,
          random.order = F, colors = brewer.pal(8, 'Dark2'))
```



Berdasarkan maklumat yang diperoleh daripada awan perkataan, whale merupakan perkataan yang paling kerap diulang di dalam teks tersebut.