Clustering

Anurag Diwate (ASD190004)

23 March 2023

Data Source: https://www.kaggle.com/datasets/ujjwalchowdhury/walmartcleaned?datasetId=2169207&language=R

Libraries Used: stats, dplyr, ggplot2, ggfortify

```
# Importing Necessary Libraries
library(stats)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggfortify)
library(cluster)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(NbClust)
library(mclust)
```

```
## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
# Importing the csv file
myData <- read.csv("/Users/jaydiwate/Downloads/walmart_cleaned.csv")

# Attaching the data
attach(myData)

# Testing if import and attaching were successful
names(myData)
```

```
##  [1] "X"            "Store"        "Date"         "IsHoliday"    "Dept"
##  [6] "Weekly_Sales" "Temperature"  "Fuel_Price"   "MarkDown1"    "MarkDown2"
## [11] "MarkDown3"    "MarkDown4"    "MarkDown5"    "CPI"          "Unemployment"
## [16] "Type"         "Size"
```

```r
nrow(myData)
```

```
## [1] 421570
```

```r
ncol(myData)
```

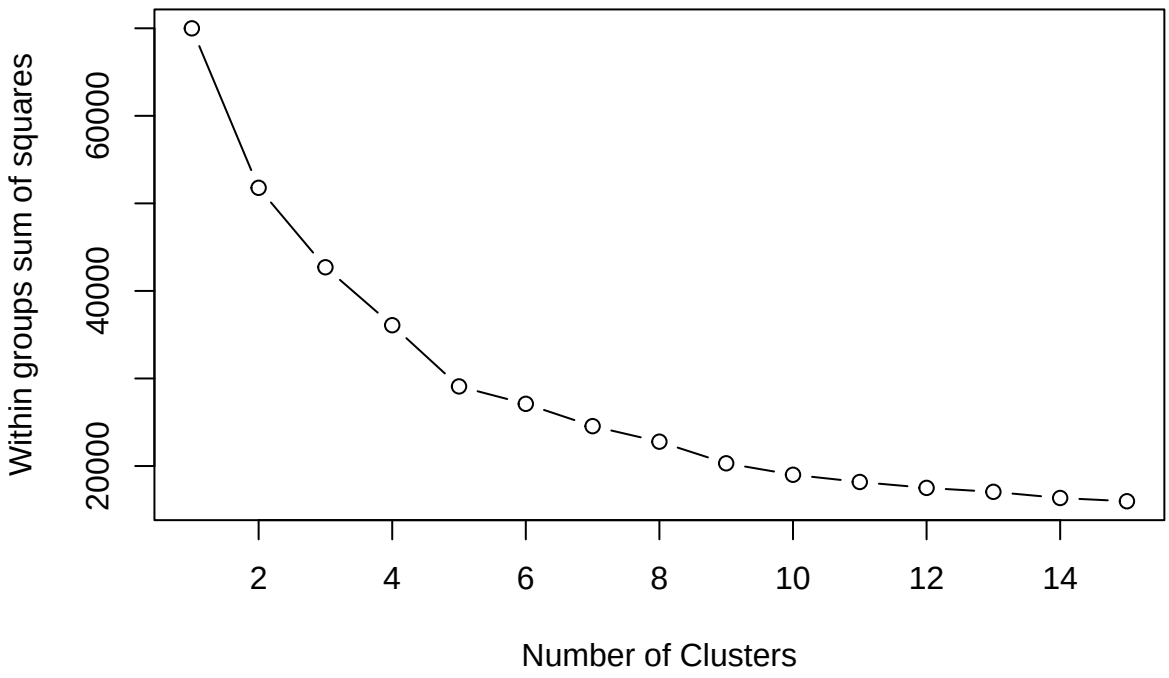
```
## [1] 17
```

```r
summary(myData)
```

```
##        X              Store          Date              IsHoliday
##  Min.   :     0   Min.   : 1.0   Length:421570      Min.   :0.00000
##  1st Qu.:105782   1st Qu.:11.0   Class :character   1st Qu.:0.00000
##  Median :211604   Median :22.0   Mode  :character   Median :0.00000
##  Mean   :211611   Mean   :22.2                      Mean   :0.07036
##  3rd Qu.:317425   3rd Qu.:33.0                      3rd Qu.:0.00000
##  Max.   :423285   Max.   :45.0                      Max.   :1.00000
##       Dept        Weekly_Sales     Temperature       Fuel_Price
##  Min.   : 1.00   Min.   : -4989   Min.   : -2.06   Min.   :2.472
##  1st Qu.:18.00   1st Qu.:  2080   1st Qu.: 46.68   1st Qu.:2.933
##  Median :37.00   Median :  7612   Median : 62.09   Median :3.452
##  Mean   :44.26   Mean   : 15981   Mean   : 60.09   Mean   :3.361
##  3rd Qu.:74.00   3rd Qu.: 20206   3rd Qu.: 74.28   3rd Qu.:3.738
##  Max.   :99.00   Max.   :693099   Max.   :100.14   Max.   :4.468
##    MarkDown1       MarkDown2          MarkDown3          MarkDown4
##  Min.   :    0   Min.   :  -265.8   Min.   :   -29.10   Min.   :    0.0
##  1st Qu.:    0   1st Qu.:     0.0   1st Qu.:     0.00   1st Qu.:    0.0
##  Median :    0   Median :     0.0   Median :     0.00   Median :    0.0
##  Mean   : 2590   Mean   :   880.0   Mean   :   468.09   Mean   : 1083.1
##  3rd Qu.: 2809   3rd Qu.:     2.2   3rd Qu.:     4.54   3rd Qu.:  425.3
##  Max.   :88647   Max.   :104519.5   Max.   :141630.61   Max.   :67474.9
##    MarkDown5           CPI         Unemployment         Type
##  Min.   :     0   Min.   :126.1   Min.   : 3.879   Min.   :1.00
##  1st Qu.:     0   1st Qu.:132.0   1st Qu.: 6.891   1st Qu.:2.00
##  Median :     0   Median :182.3   Median : 7.866   Median :3.00
##  Mean   :  1663   Mean   :171.2   Mean   : 7.960   Mean   :2.41
##  3rd Qu.:  2168   3rd Qu.:212.4   3rd Qu.: 8.572   3rd Qu.:3.00
```

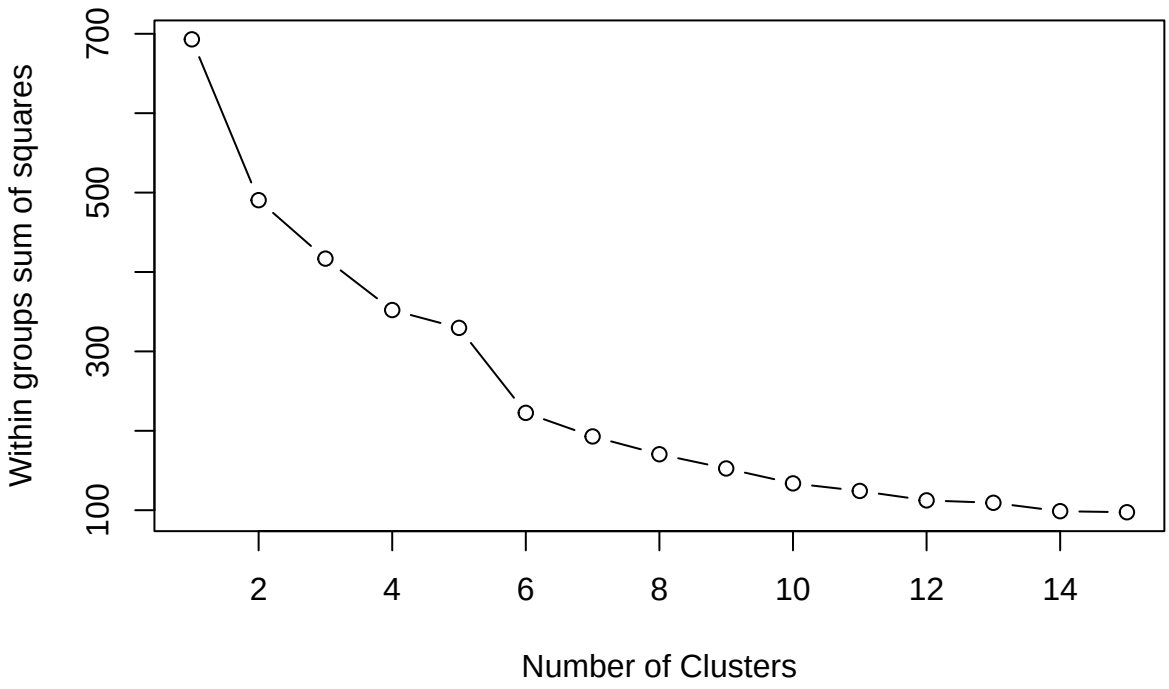
```
##  Max.   :108519   Max.   :227.2   Max.   :14.313   Max.   :3.00
##       Size
##  Min.   : 34875
##  1st Qu.: 93638
##  Median :140167
##  Mean   :136728
##  3rd Qu.:202505
##  Max.   :219622
```

```
# Taking a subset of relevant columns and 10K rows
myData = select(myData, c(4, 5, 6, 7, 8, 14, 15))
myData = head(myData, 10000)
myData <- na.omit(myData)
# Making a smaller sample of size 100.
mySample <- sample_n(myData, 100, replace = FALSE, prob = NULL)

# Preparing the data for clustering
df <- scale(myData)
df2 <- scale(mySample)

# Using wss function
wssplot <- function(data, nc=15, seed=1234){
                  wss <- (nrow(data)-1)*sum(apply(data,2,var))
                      for (i in 2:nc){
                set.seed(seed)
                    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
              plot(1:nc, wss, type="b", xlab="Number of Clusters",
                           ylab="Within groups sum of squares")
              wss
       }

# Determine number of clusters
wssplot(df)
```



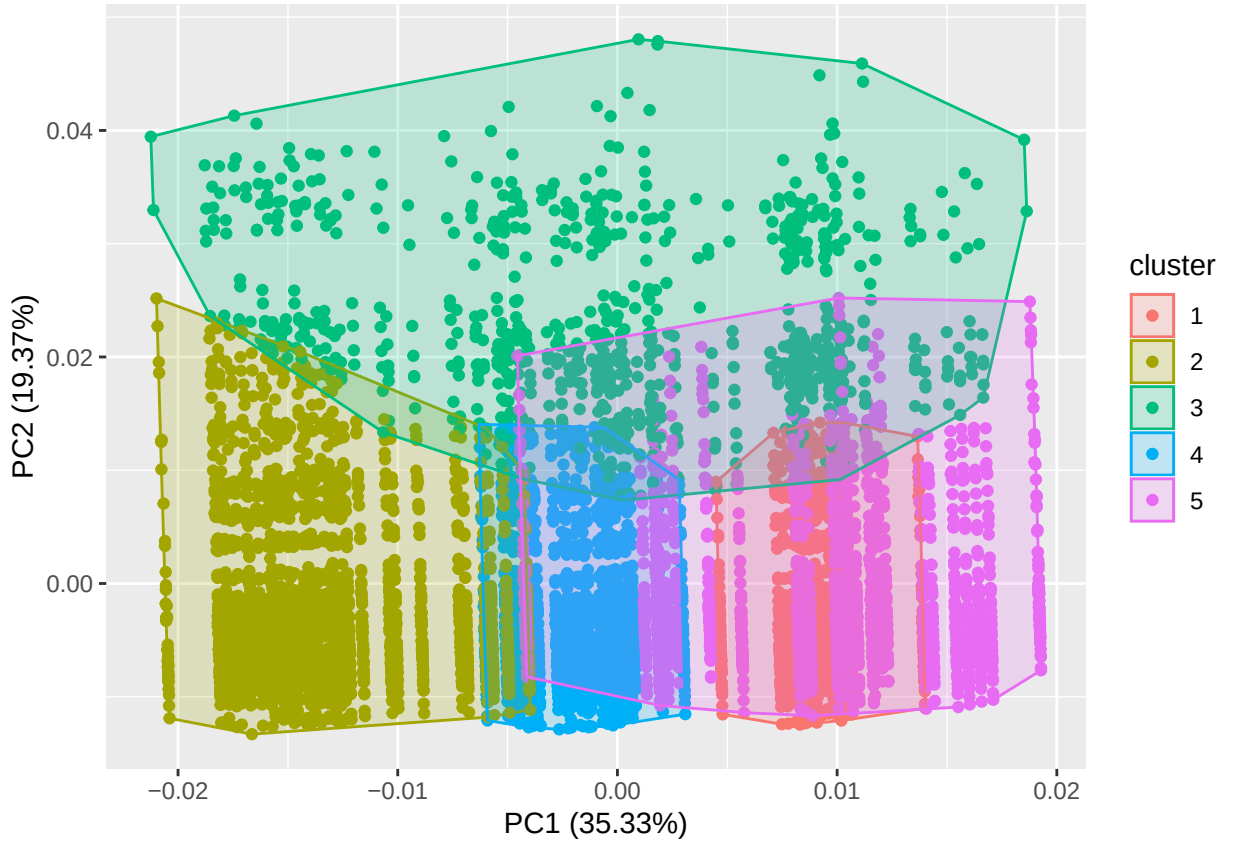

```
##  [1] 69993.00 51778.76 42701.24 36084.21 29093.77 27102.18 24553.05 22781.78
##  [9] 20311.86 19002.02 18178.47 17501.93 17051.09 16347.24 15977.54
```

```
wssplot(df2)
```

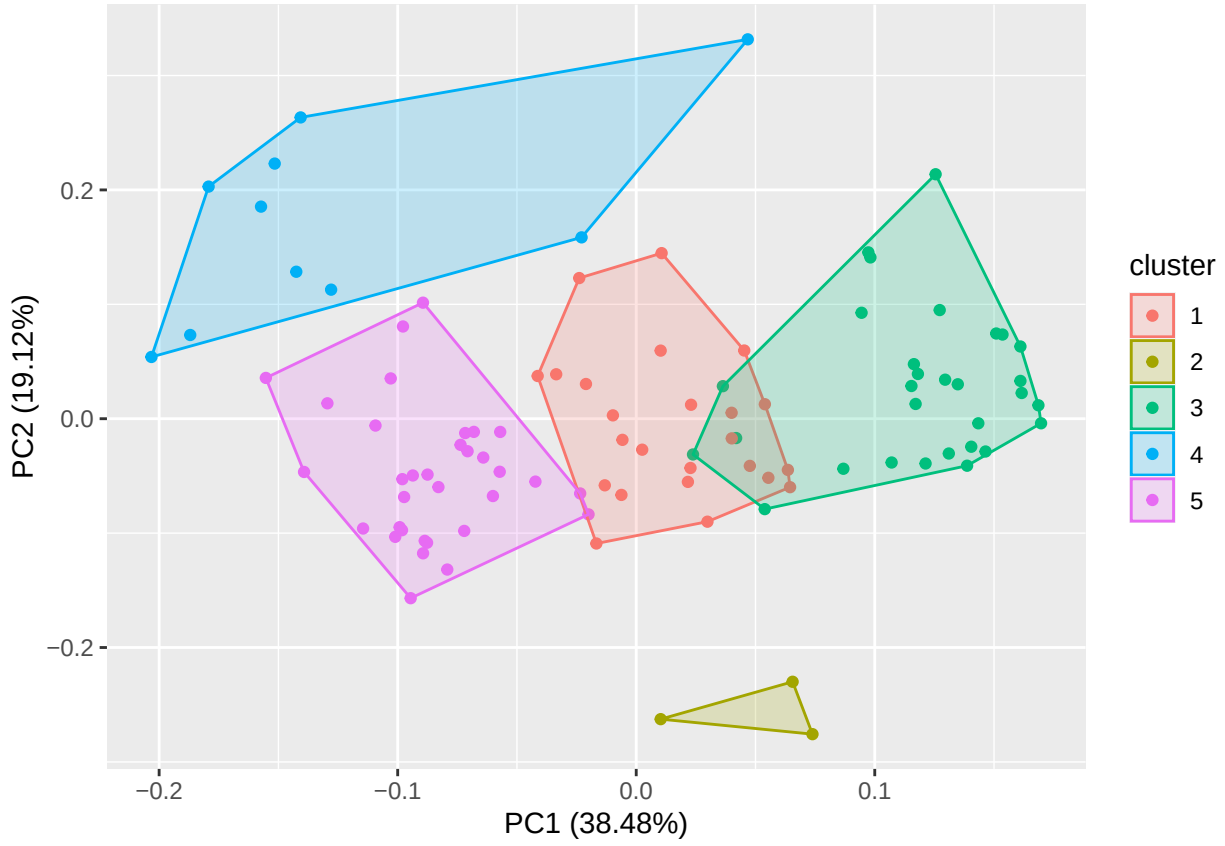



```
##  [1] 693.00000 490.45029 416.82858 352.04769 329.61130 222.59272 192.89521
##  [8] 170.47751 152.56538 133.79399 124.24860 112.36728 109.33342  98.74803
## [15]  97.43065
```

```
# Plot k-means Data
KM = kmeans(df, 5)
autoplot(KM, df, frame=TRUE)
```



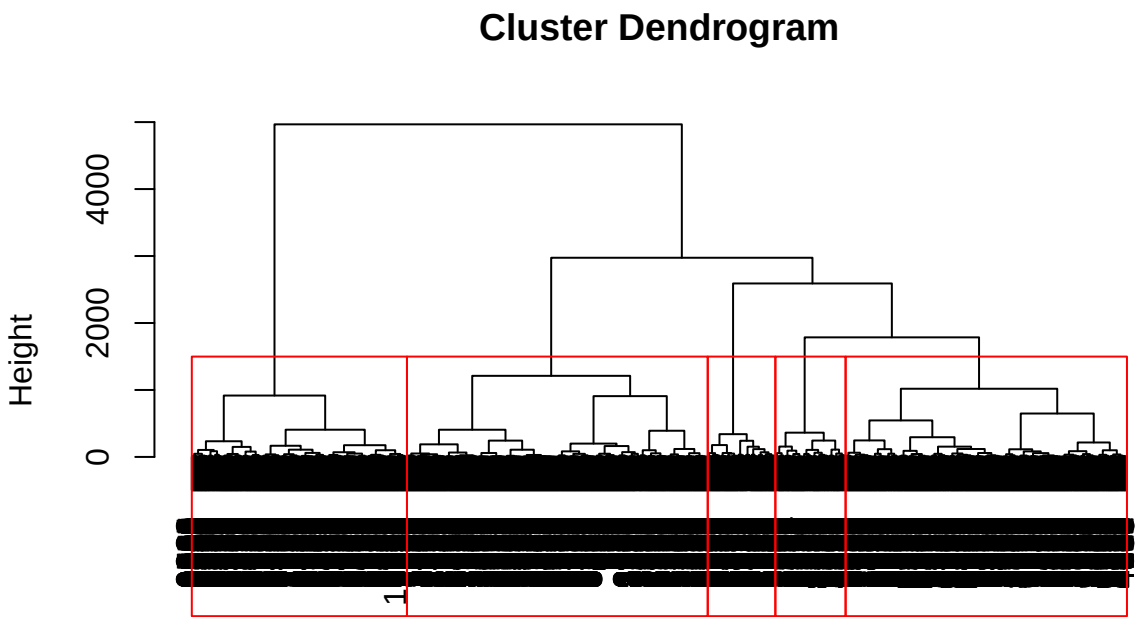

```
# Plot k-means Data for the sample
KM2 = kmeans(df2, 5)
autoplot(KM2, df2, frame=TRUE)
```




```
# Ward Hierarchical Clustering
d <- dist(df, method = "euclidean") # distance matrix
fit <- hclust(d, method="ward")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

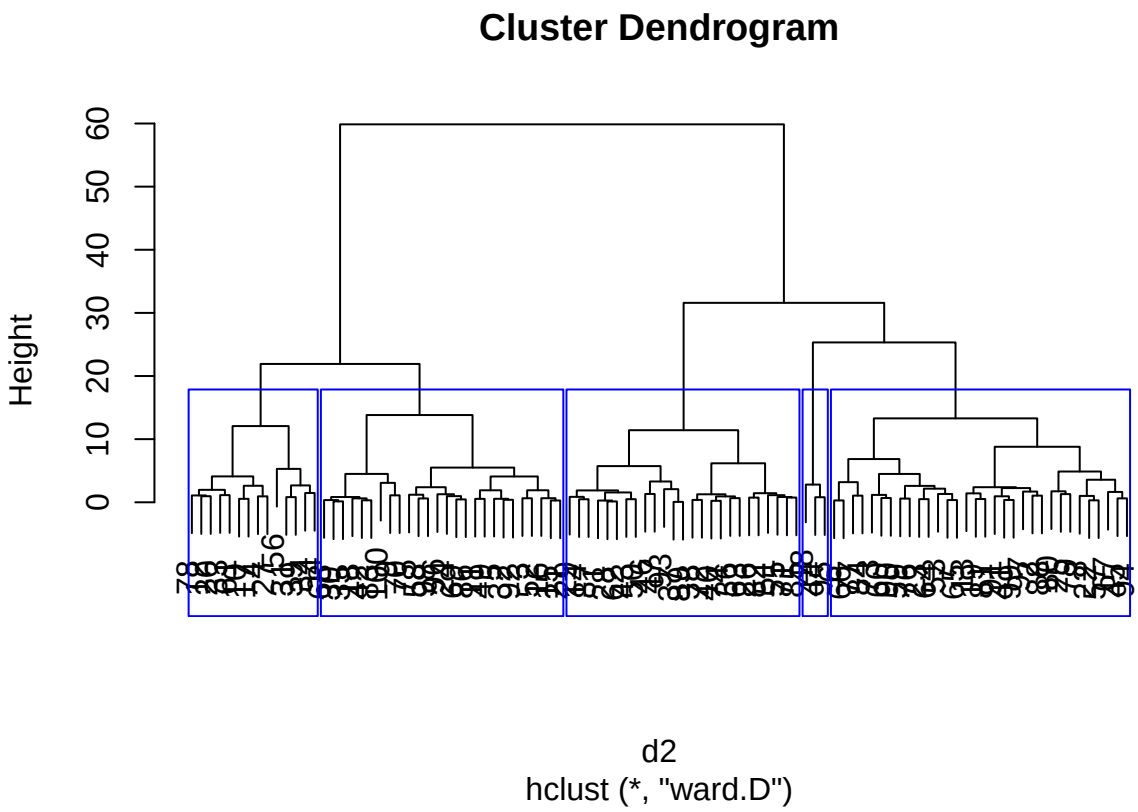
```
plot(fit) # display dendogram
groups <- cutree(fit, k=5) # cut tree into 5 clusters
# draw dendogram with red borders around the 5 clusters
rect.hclust(fit, k=5, border="red")
```



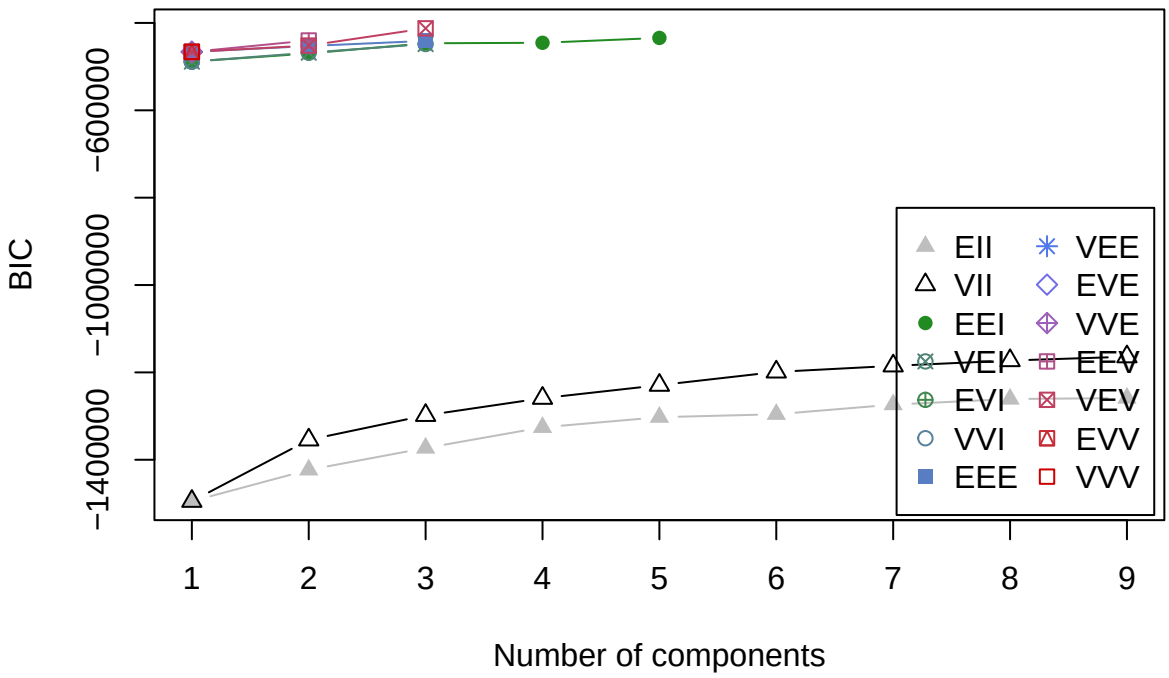

```
# Ward Hierarchical Clustering for the sample data
d2 <- dist(df2, method = "euclidean") # distance matrix
fit2 <- hclust(d2, method="ward")
```

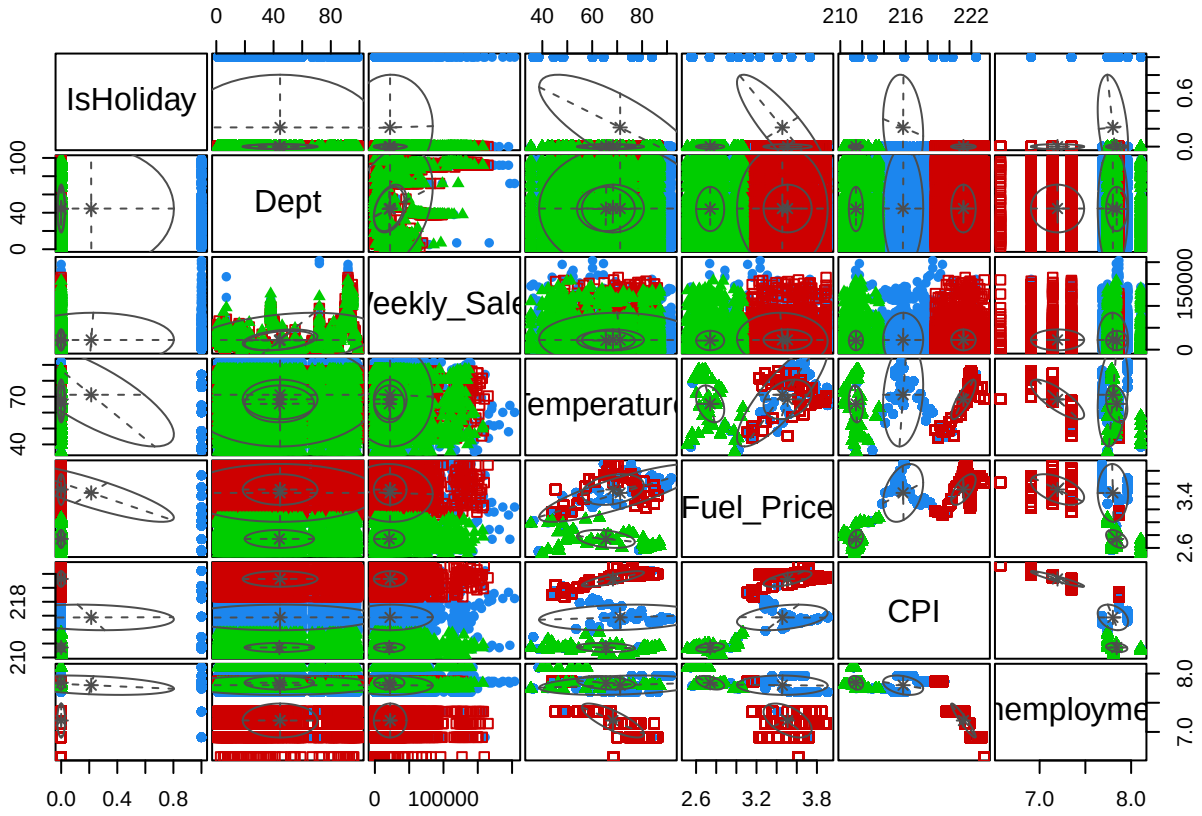
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

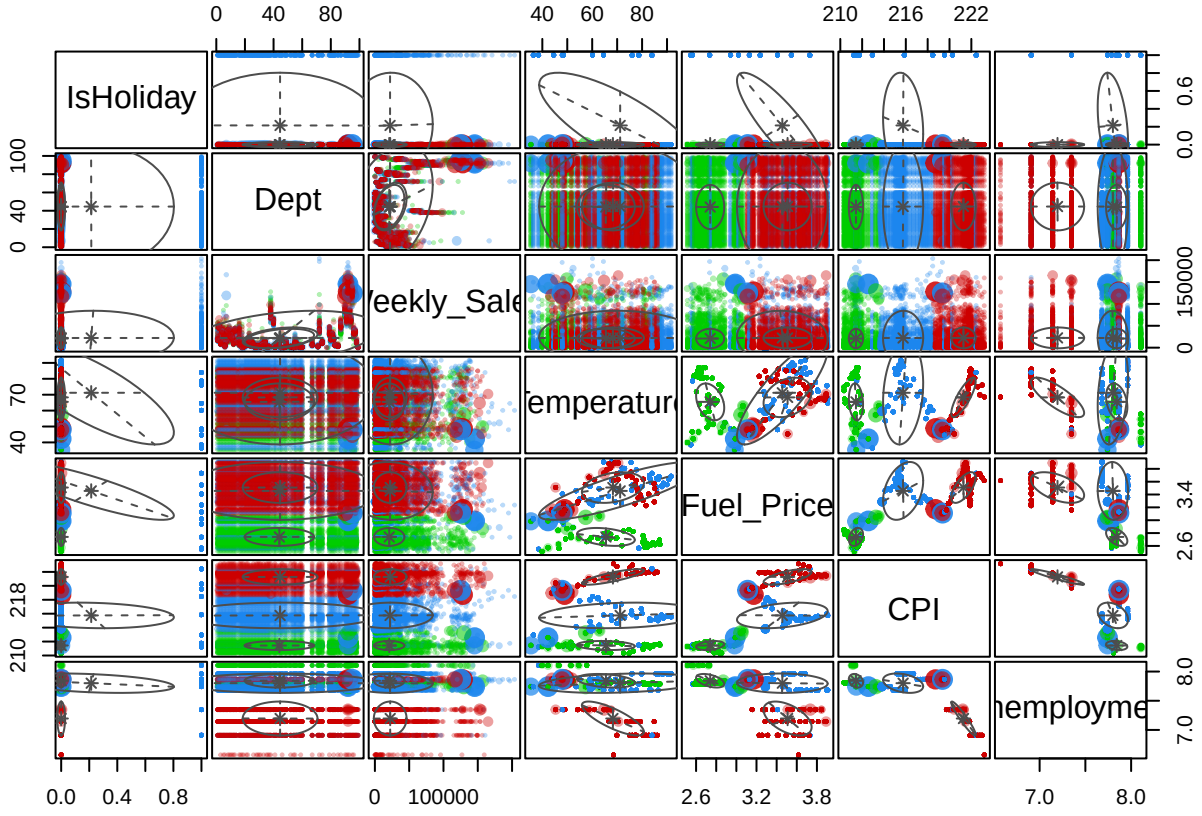
```
plot(fit2) # display dendogram
groups2 <- cutree(fit2, k=5) # cut tree into 5 clusters
# draw dendogram with blue borders around the 5 clusters
rect.hclust(fit2, k=5, border="blue")
```

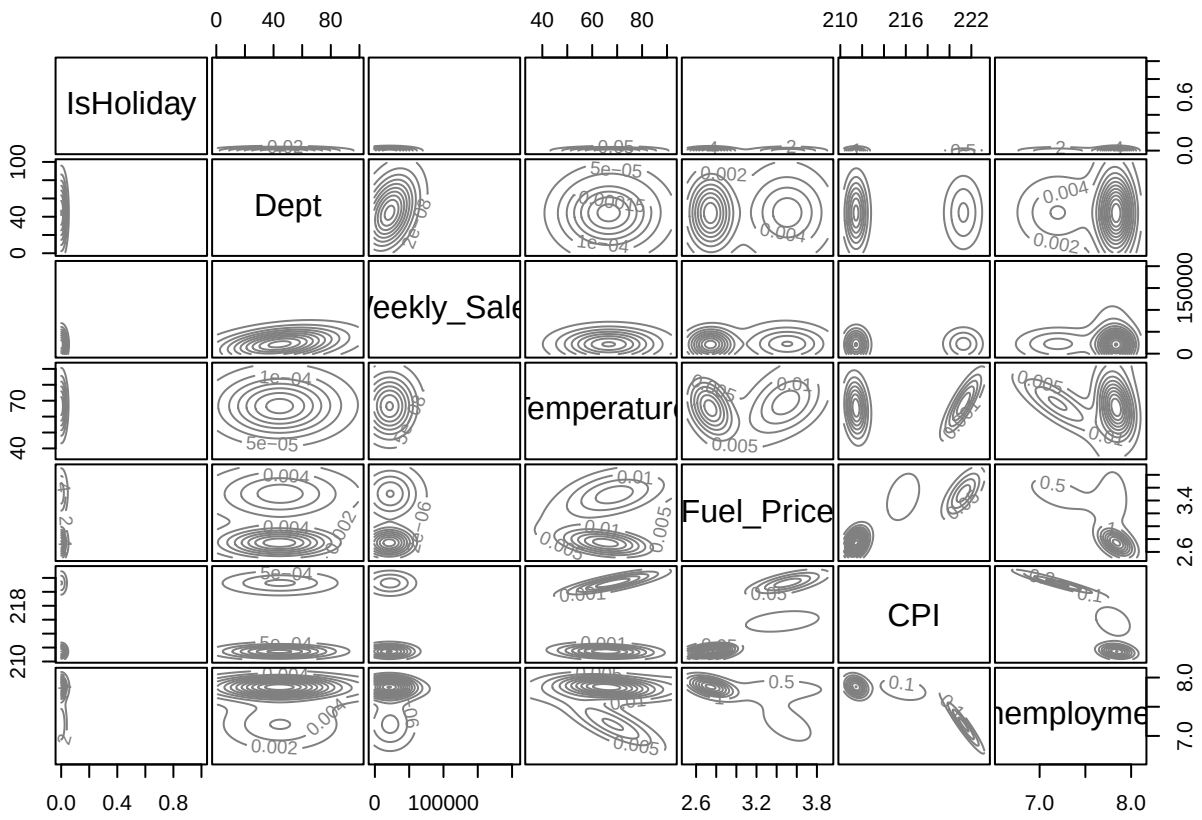



```
# Model Based Clustering
modelClust <- Mclust(myData)
plot(modelClust) # plot results
```


BIC
−1400000
−1000000
−600000
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
VEE
EVE
VVE
EEV
VEV
EVV
VVV


IsHoliday
Dept
Weekly_Sale
Temperatur
Fuel_Price
CPI
nemployme

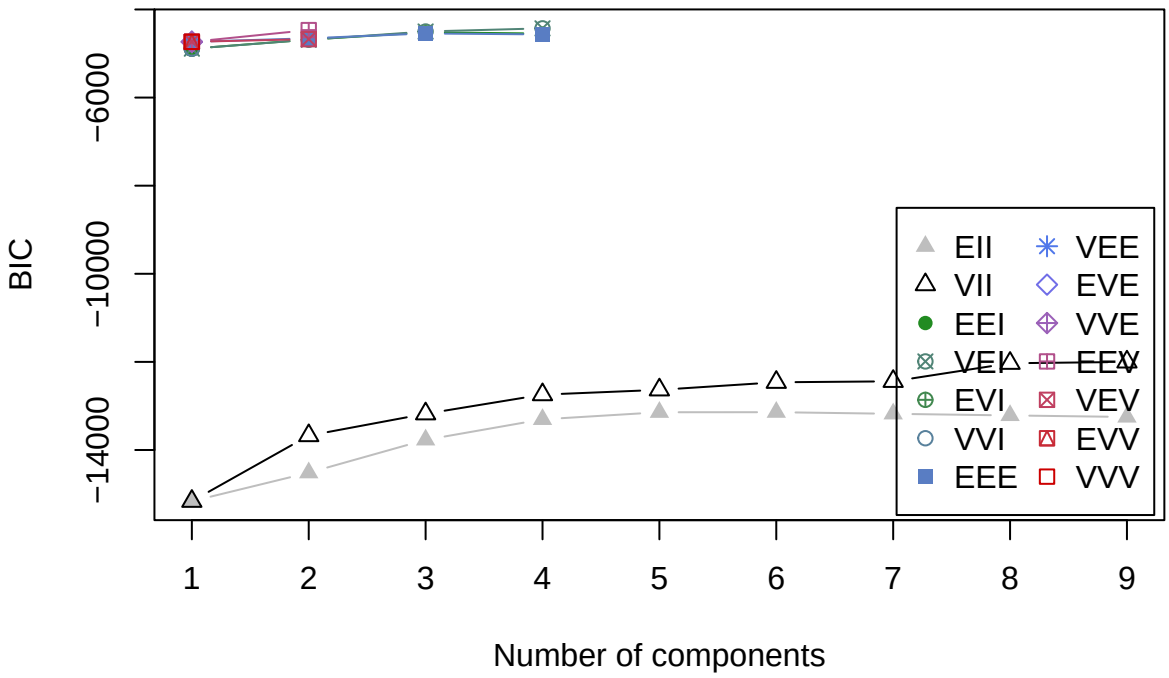

IsHoliday
Dept
Weekly_Sale
Temperatur
Fuel_Price
CPI
Unemployme

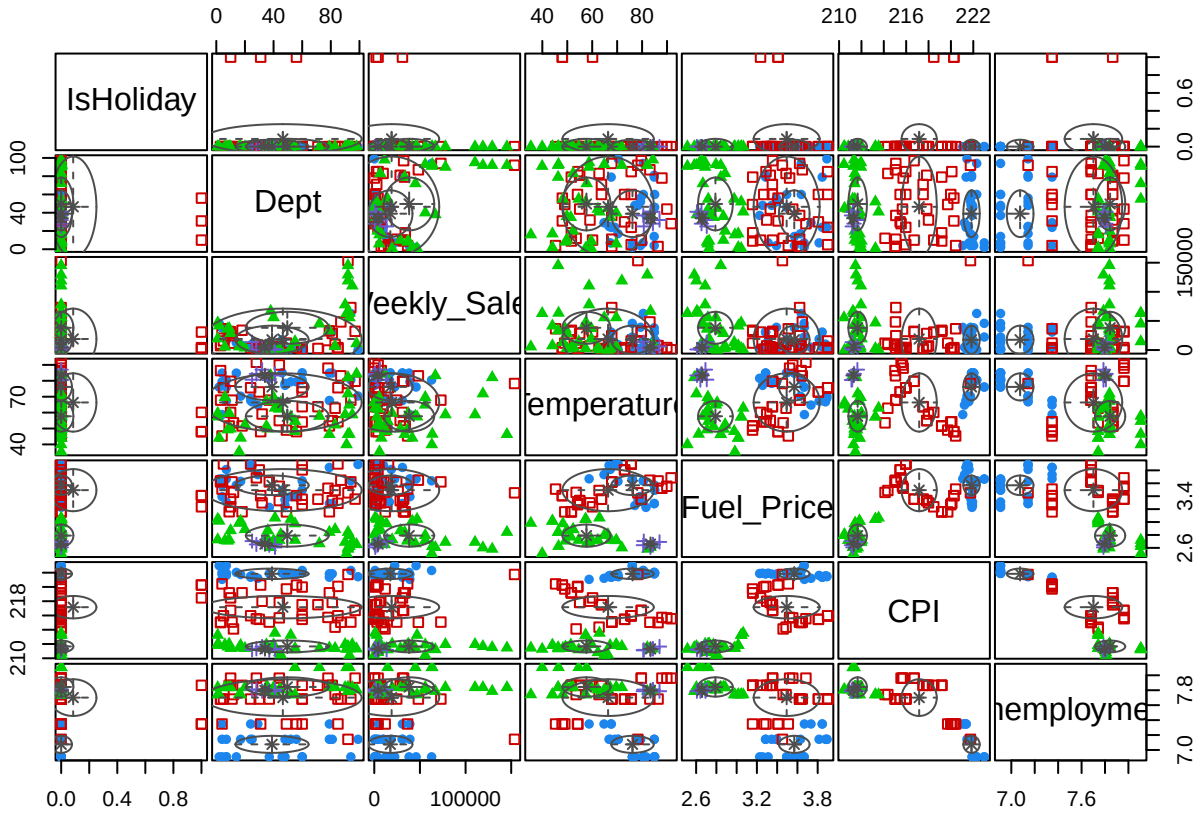


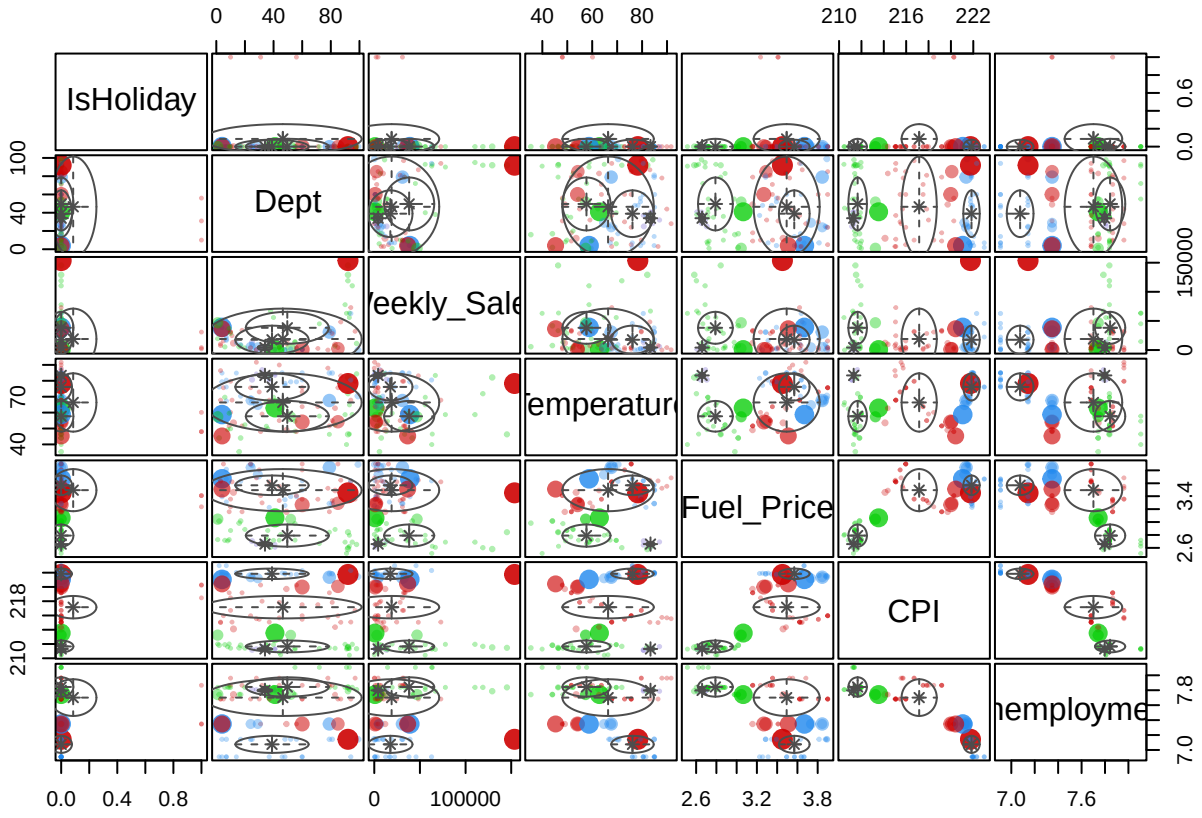

```
summary(modelClust) # display the best model
```

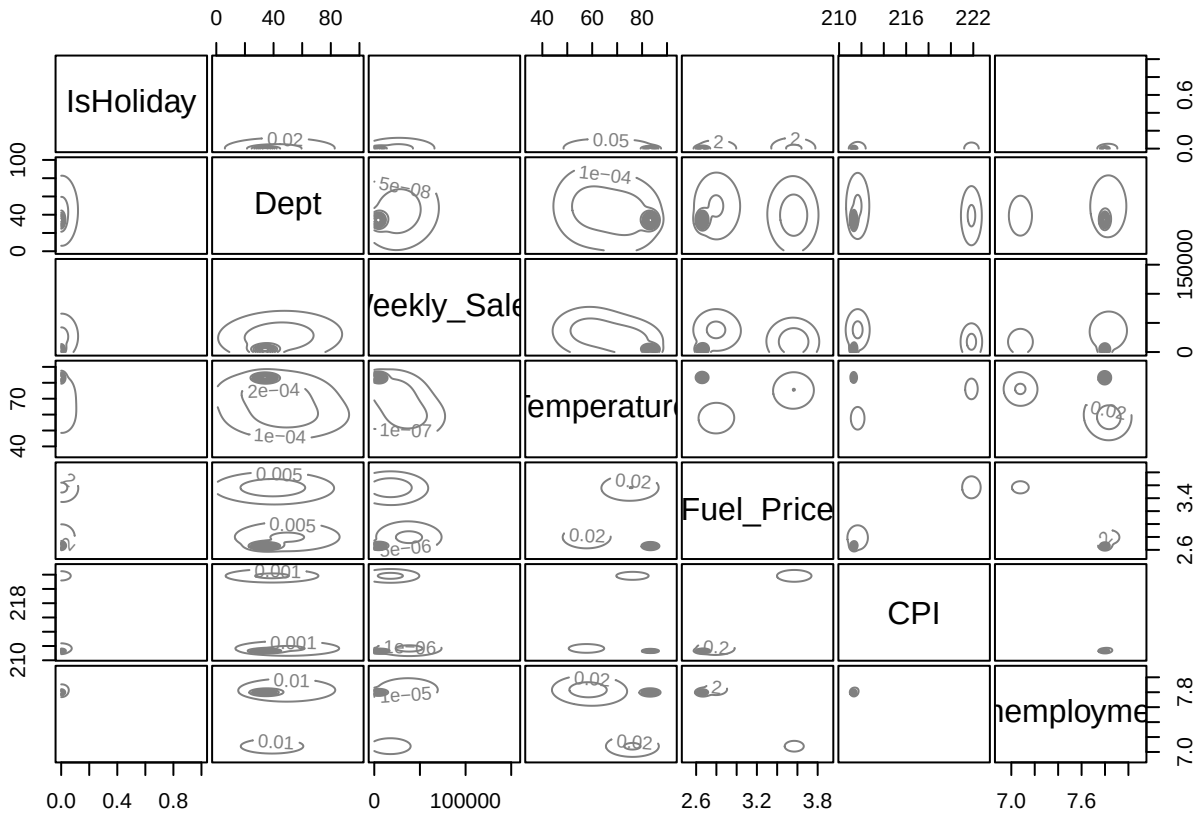
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 3 components:
##
##  log-likelihood     n df       BIC     ICL
##       -205758.4 10000 95 -412391.9 -412403
##
## Clustering table:
##    1    2    3
## 3375 2972 3653
```

```
# Model Based Clustering for sample data
modelClust2 <- Mclust(mySample)
plot(modelClust2) # plot results
```


BIC
−6000
−10000
−14000
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
VEE
EVE
VVE
EEV
VEV
EVV
VVV


IsHoliday
Dept
Weekly_Sale
Temperatur
Fuel_Price
CPI
Unemployme


IsHoliday
Dept
Weekly_Sale
Temperatur
Fuel_Price
CPI
nemployme
0 40 80
40 60 80
210 216 222
0.0 0.6
0 40 100
0 150000
40 70
2.6 3.4
210 218
7.0 7.8
0.0 0.4 0.8
0 100000
2.6 3.2 3.8
7.0 7.6




```
summary(modelClust2) # display the best model
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEI (diagonal, equal shape) model with 4 components:
##
##  log-likelihood   n df       BIC       ICL
##       -2120.041 100 41 -4428.893 -4429.539
##
## Clustering table:
##  1  2  3  4
## 26 35 33  6
```

Takeaways: The insight we get from K-means clustering is what groupings exist within our data that is not explicitly mentioned or labeled. This can be very useful to identify what types of groups exist in a complex set of data such as this one. It can also help identify any unknown groups that we didn't know existed within the dataset. Hierarchical clustering kind of builds on this, as it identifies how similar or different the readingsdata points are. Hierarchical clustering can also help us find smaller clusters of data as it narrows down the dendogram. Finally, model based clustering assumes that the data generated came from the model, and back tracks to work out what the original model might be. This can be useful to us since we can find the model of best fit for the data, and use it to better define clusters within our data.