Applied Mathematics: Logic and Statistics

SIW004

Assignment 3

Ismail Sunni
imajimatika@gmail.com
al388270@uji.es

Department de Llenguatges i Sistemes Informatics
Master's Degree in Geospatial Technologies - Erasmus Mundus
Course 2018/2019 – 17 December 2018

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

Assignment 3

Questions:

From the dataset in Table 7.5 (page 154, SIDS data) from Everitt, solve the following exercises:

1. Extract as much as possible information coming out from the data set using graphical tools as described in Chapter 2 of Everitt.

2. Perform a PCA and interpret the results.

3. Find homogeneous clusters amongst the individuals without considering the variable Group (use hierarchical and k-means methods under two distinct choices of distance methods). Compare the results with the existing groups given by variable "group". Then perform LDA and
classify into these groups the following two new observations:
Obs1: (110,3320,0.240,39); Obs2: (120,3310,0.298,37).

Data Loading

Since the data is in the table format, I need to convert it to CSV so that R can read it properly. I did it by copy to text editor then replace space character to a comma. After that, I can just load the data by using something like this:

```
data = read.csv("data.csv", header = TRUE)
```

You will find this line of code at the beginning of all files since I need to load the data first.
The data itself has 5 columns (`Group,HR,BW,Factor68,Gesage`). We can say that the `Group` is the class of each row, and the other four are the variables.

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

Answer 1:

Choosing variables

I need to choose two of them to make a pair that I want to use for my information extraction. I do a correlation matrix calculation for all columns except group. I think it's more interesting to see the variables, than the class. I do it by using this code:

```
#  Show correlation matrix, without group column
cor(data[,2:5]);
```

The result I got is:

```
             HR         BW   Factor68      Gesage
HR       1.00000000 -0.02192954  0.2098967  0.04031584
BW      -0.02192954  1.00000000 -0.0785167  0.42490365
Factor68  0.20989675 -0.07851670  1.0000000 -0.24570910
Gesage    0.04031584  0.42490365 -0.2457091  1.00000000
```

We can see that BW and Gesage has the highest correlation, so I choose them as my pair.

Scatter Plot

From here, I can make a scatter plot for that pair by using this code:

```
##### plotting scatter plot with abline ####
# Divide the plot area to 2x2
par(mfrow=c(2,2))
par(pty="s")
# Scatterplot BW and Gesage
plot(BW,Gesage,pch=1,lwd=2)
title("(a)",lwd=2)
# Scatterplot BW and Gesage with abline / linear regression fit
plot(BW,Gesage,pch=1,lwd=2)
abline(lm(Gesage~BW),lwd=2)
title("(b)",lwd=2)
# Jittered Scatterplot BW and Gesage
data1<-jitter(cbind(BW,Gesage))
plot(BW,Gesage,pch=1,lwd=2)
title("(c)",lwd=2)
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

```
# Plot BW and Gesage with jitter and histogram
plot(BW,Gesage,pch=1,lwd=2)
rug(jitter(BW),side=1)
rug(jitter(Gesage),side=2)
title("(d)",lwd=2)
```

And the result that I got is

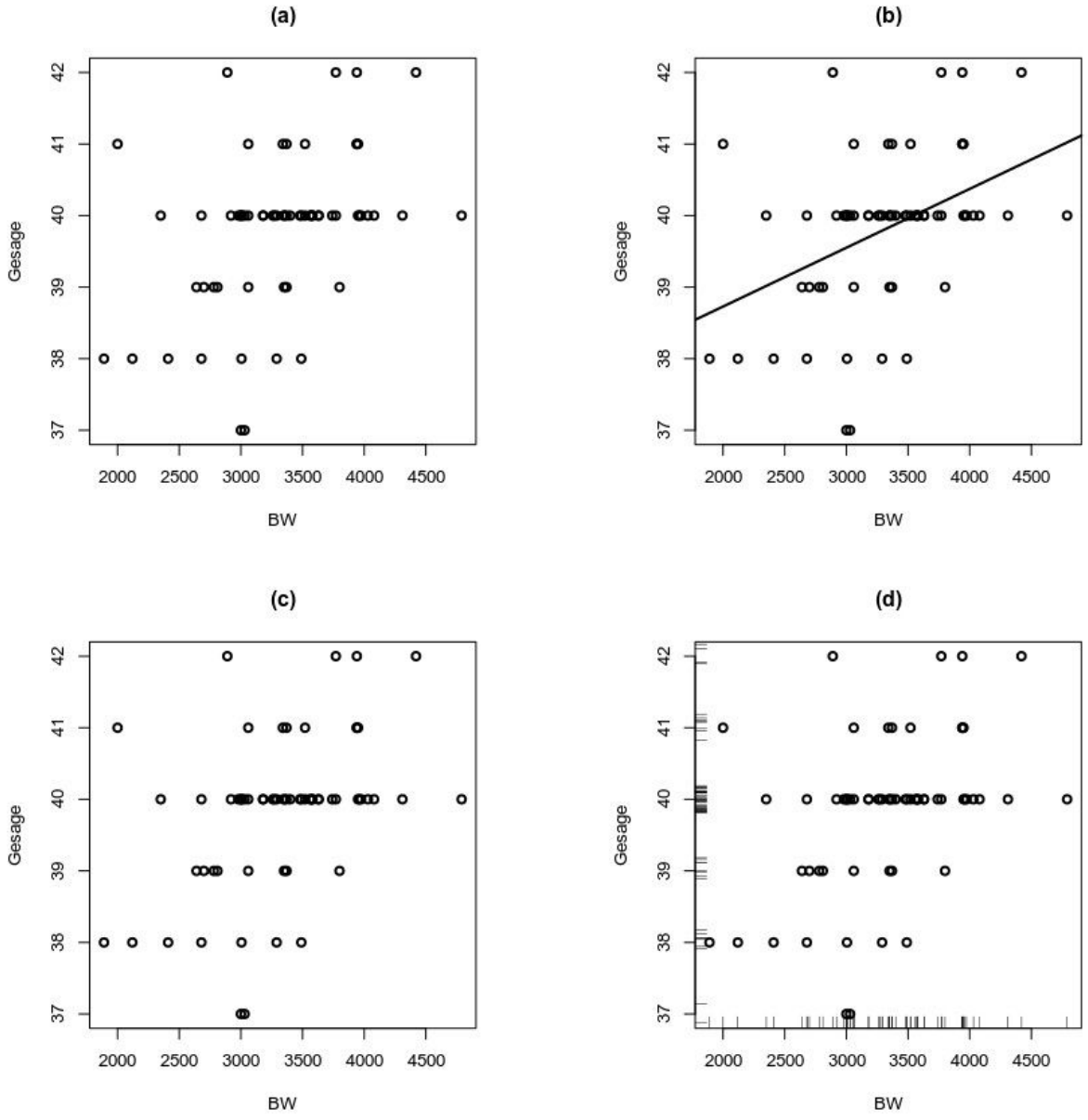

Figure 1.1 *Scatter plot between Gesage and BW*

Description about the plots:

a. The scatter plot between Gesage and BW.
b. The scatter plot between Gesage and BW with linear regression fit.
c. A jittered version of the first scatterplot
d. Adding histogram to the scatterplot C

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

At first, I can't see the pattern in the plot since there are a lot of circle in the Gesage = 40. But after checking the scatterplot B, I can see a pattern, and it's quite right. From the last scatterplot, I understand that there are a dense distribution near Gesage = 40 and BW = 3000-3500.

Scatter plot with label

```
##### Scatter plot with abbreviation ####
names<-abbreviate(row.names(data))
par(mfrow=c(1,1))
plot(BW,Gesage,lwd=2,type="n")
text(BW,Gesage,labels=names,lwd=2)
```

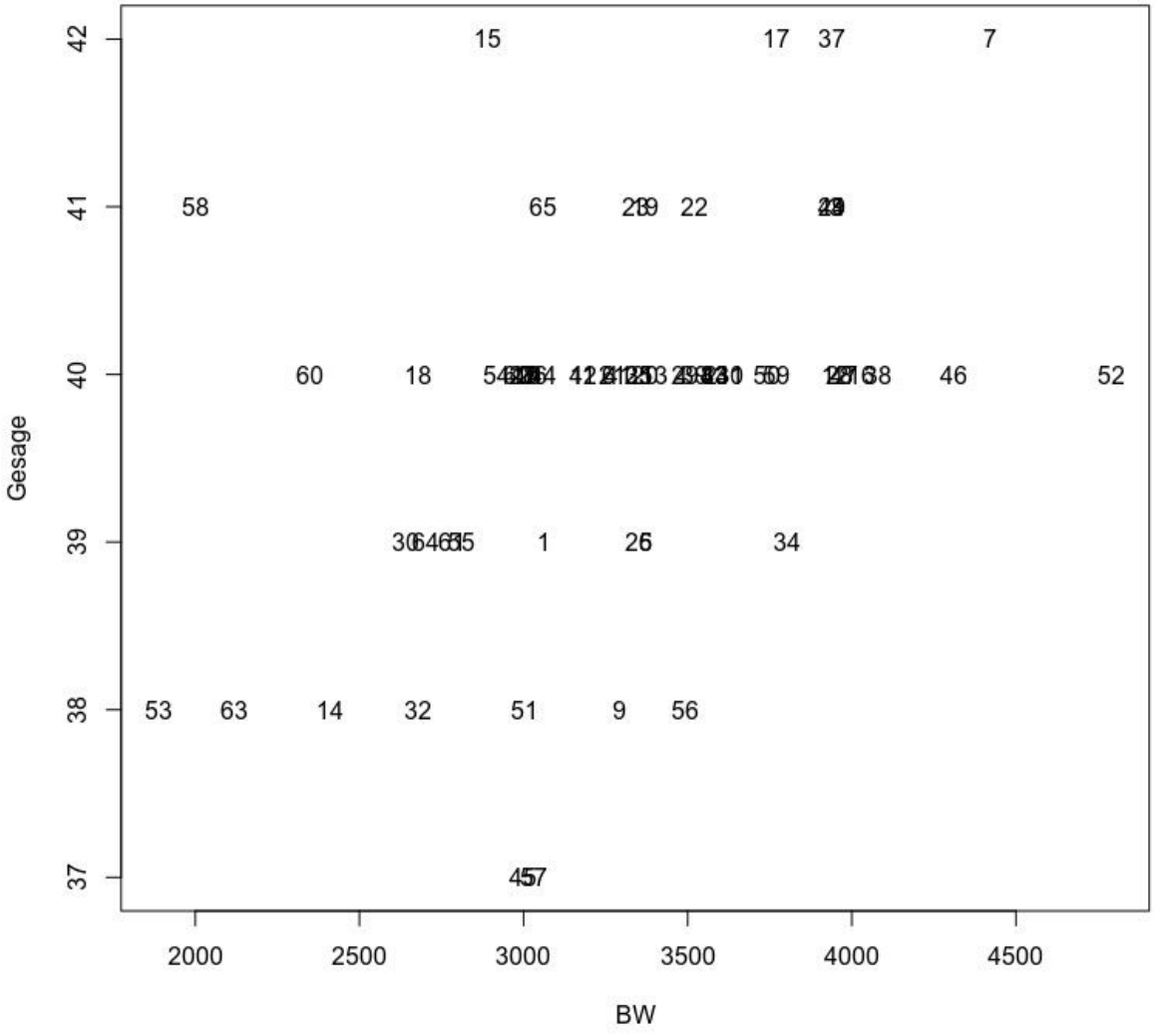

Figure 1.2 *Scatter plot with label*

From the Figure 1.2, we can identify which data that is not in the main line pattern. But since the label is the row number, it's not quite clear. For example, the Gasage = 37, it seems the data are 45 and 57. Another interesting out of the pattern is 58 and 15.

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

Linear regression and locally weighted fit

```
##### Scatterplot BW against Gesage with linear regression and
locally weighted fit, and marginal distribution ####
# set up plotting area for scatterplot
par(fig=c(0,0.7,0,0.7))
plot(BW,Gesage,lwd=2)
# add regression line
abline(lm(Gesage~BW),lwd=2)
# add locally weighted regression fit
lines(lowess(BW,Gesage),lwd=2)
# set up plotting area for histogram
par(fig=c(0,0.7,0.65,1),new=TRUE)
hist(BW,lwd=2)
# set up plotting area for boxplot
par(fig=c(0.65,1,0,0.7),new=TRUE)
boxplot(Gesage,lwd=2)
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

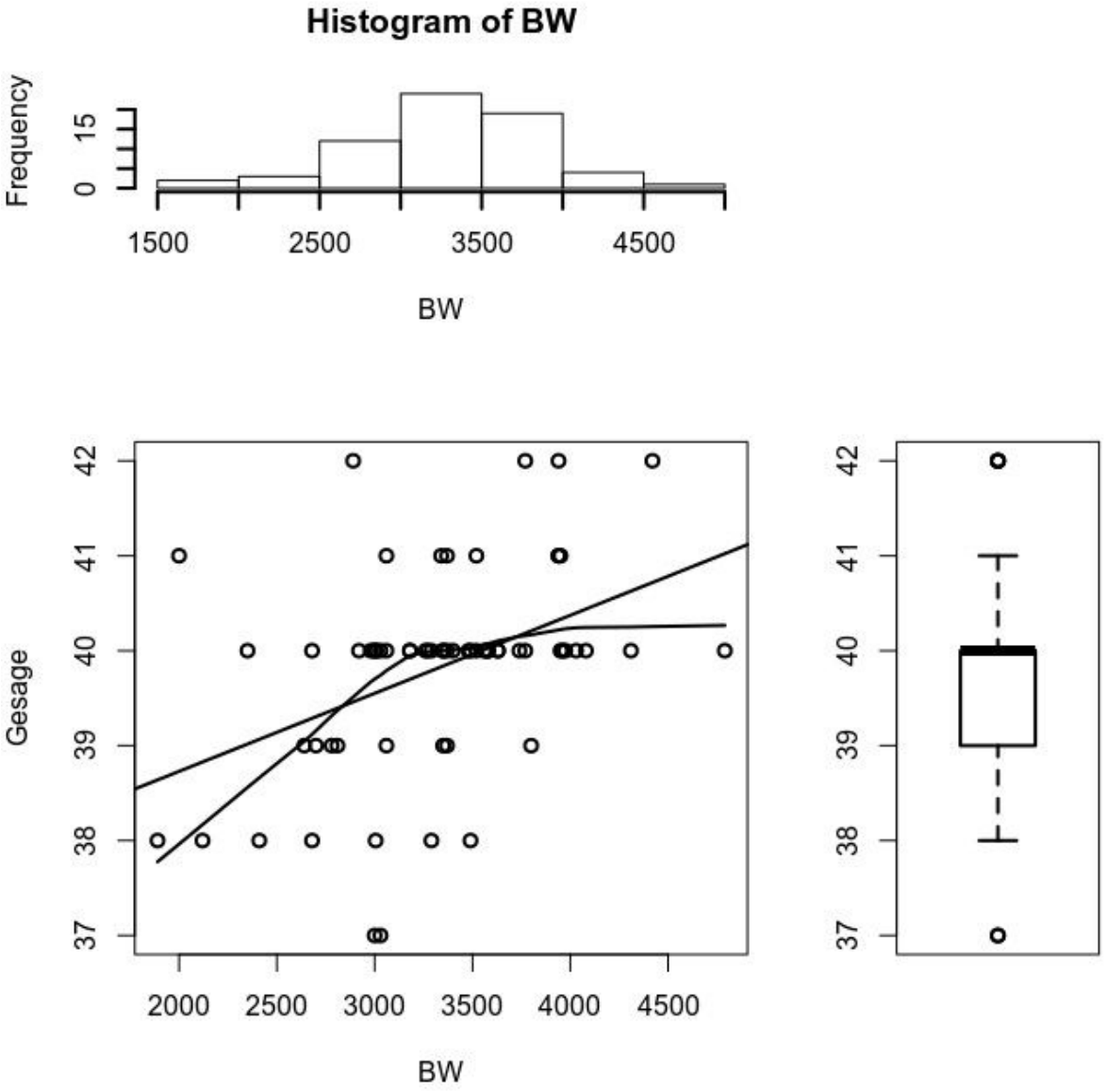

Figure 1.3 *Scatter plot with label*

From the figure 1.3, we can see that the linear regression and locally weighted fit are not really in the same direction. In this case, we can say that the dependency is not that strong.

Convex hull

```
##### Convex hull ####
hull<-chull(BW,Gesage)
plot(BW,Gesage,pch=1)
polygon(BW[hull],Gesage[hull],density=15,angle=30)
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

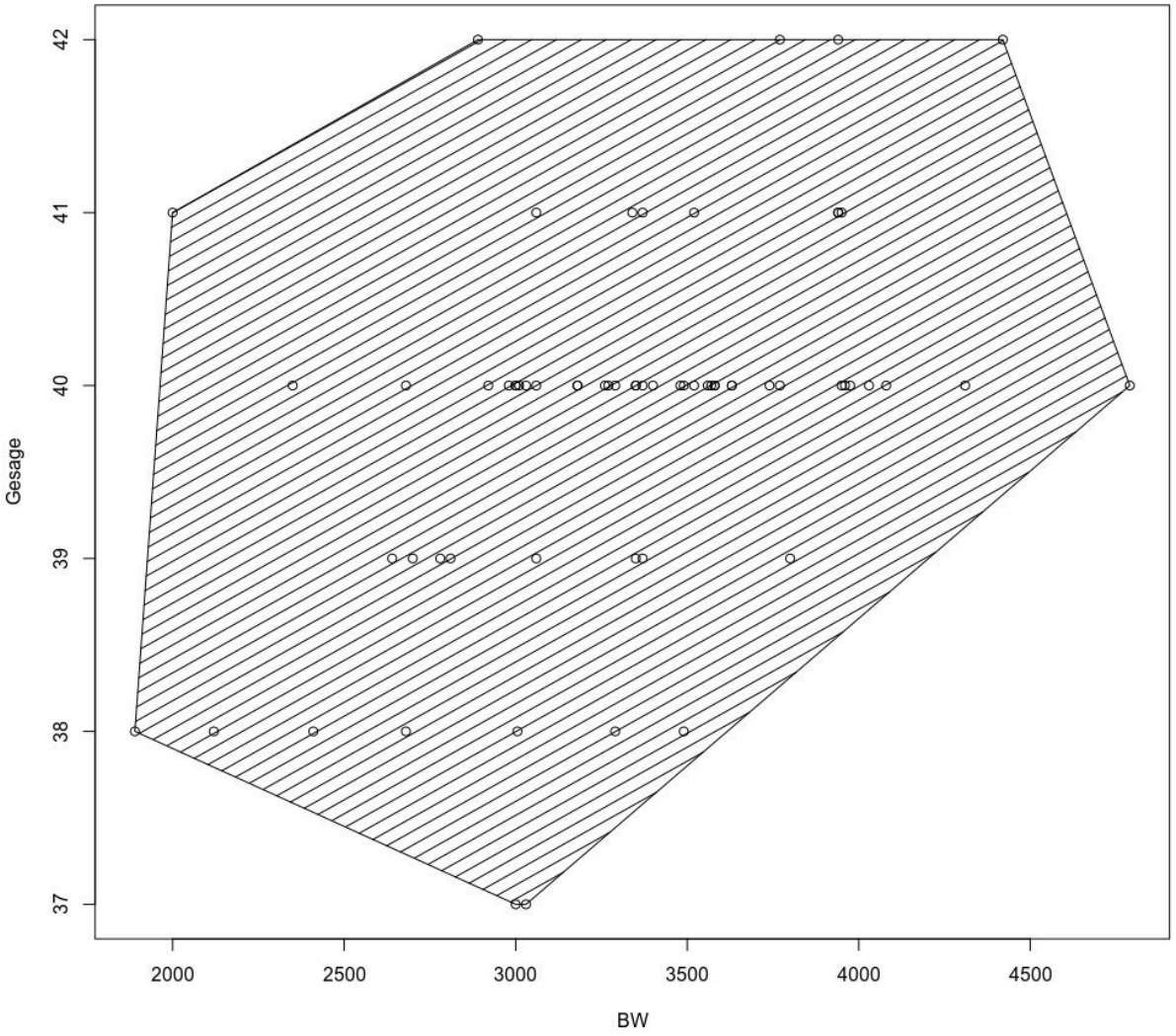

Figure 1.4 *Scatter plot with convex hull*

The convex hull in the figure 1.4 shows us about the minimum area of polygon that contains all of the data. We can use it to see the outlier (the one that if removed will make the convex hull smaller significantly). In this case we can say the left-most with Gesage = 41 for example is an outlier.

Chi plots

```
##### Chi plots ####
chiplot(BW,Gesage,vlabs=c("BW","Gesage"))
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

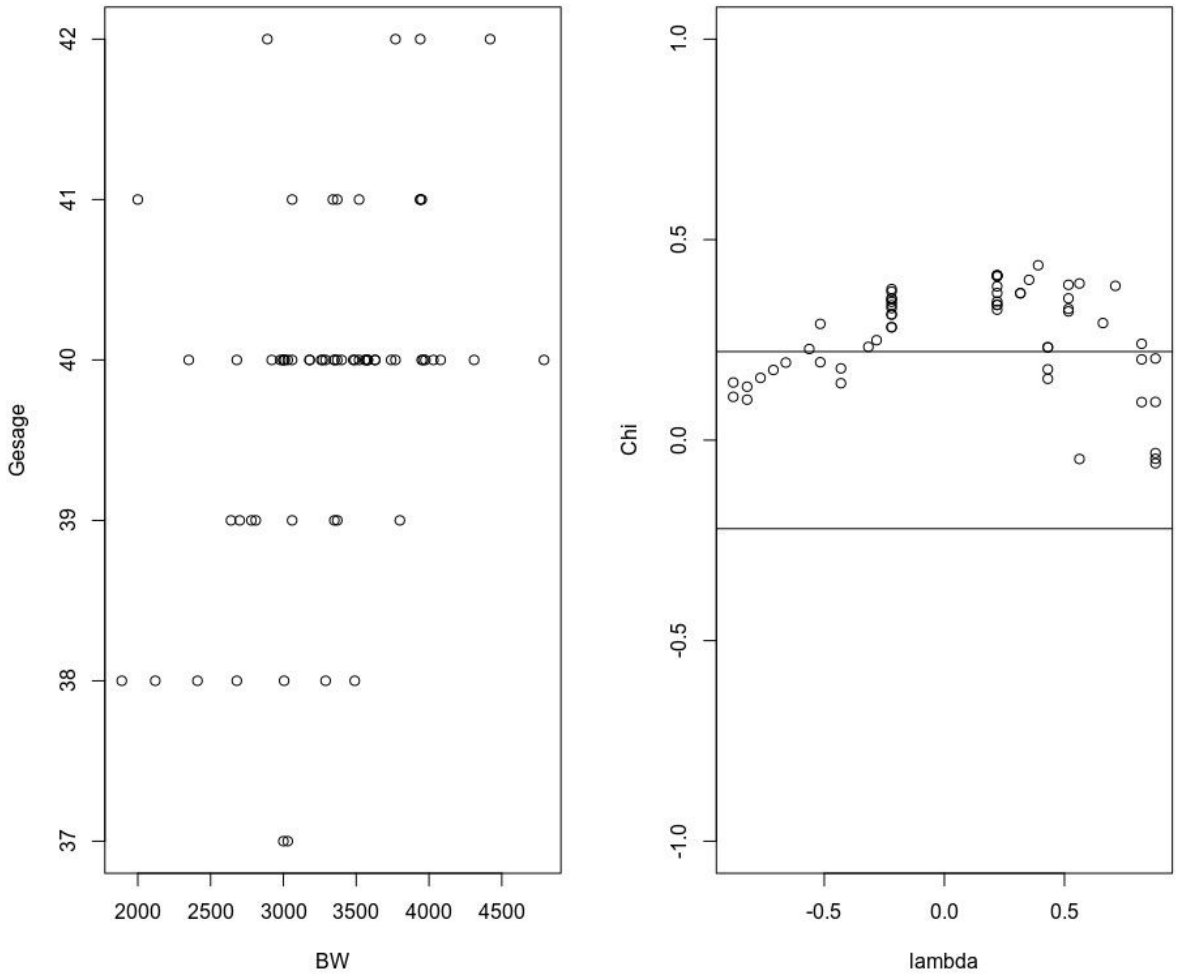

Figure 1.5 *Chiplot with interval band*

In the figure 1.5, we can see that not all of the data are outside the confidence interval band. Although, it's clear that there are more data outside the interval band. This mean, the Gesage and BW have dependency but not a high one. This result is consistent with the previous result.

Bivariate boxplot

```
#### Bivariate Boxplot####
# Create bvbox with robust estimator method (the default one)
bvbox(cbind(BW,Gesage),xlab="BW",ylab="Gesage")
# Create bvbox with other (non robust) method
bvbox(cbind(BW,Gesage),xlab="BW",ylab="Gesage",method="o")
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

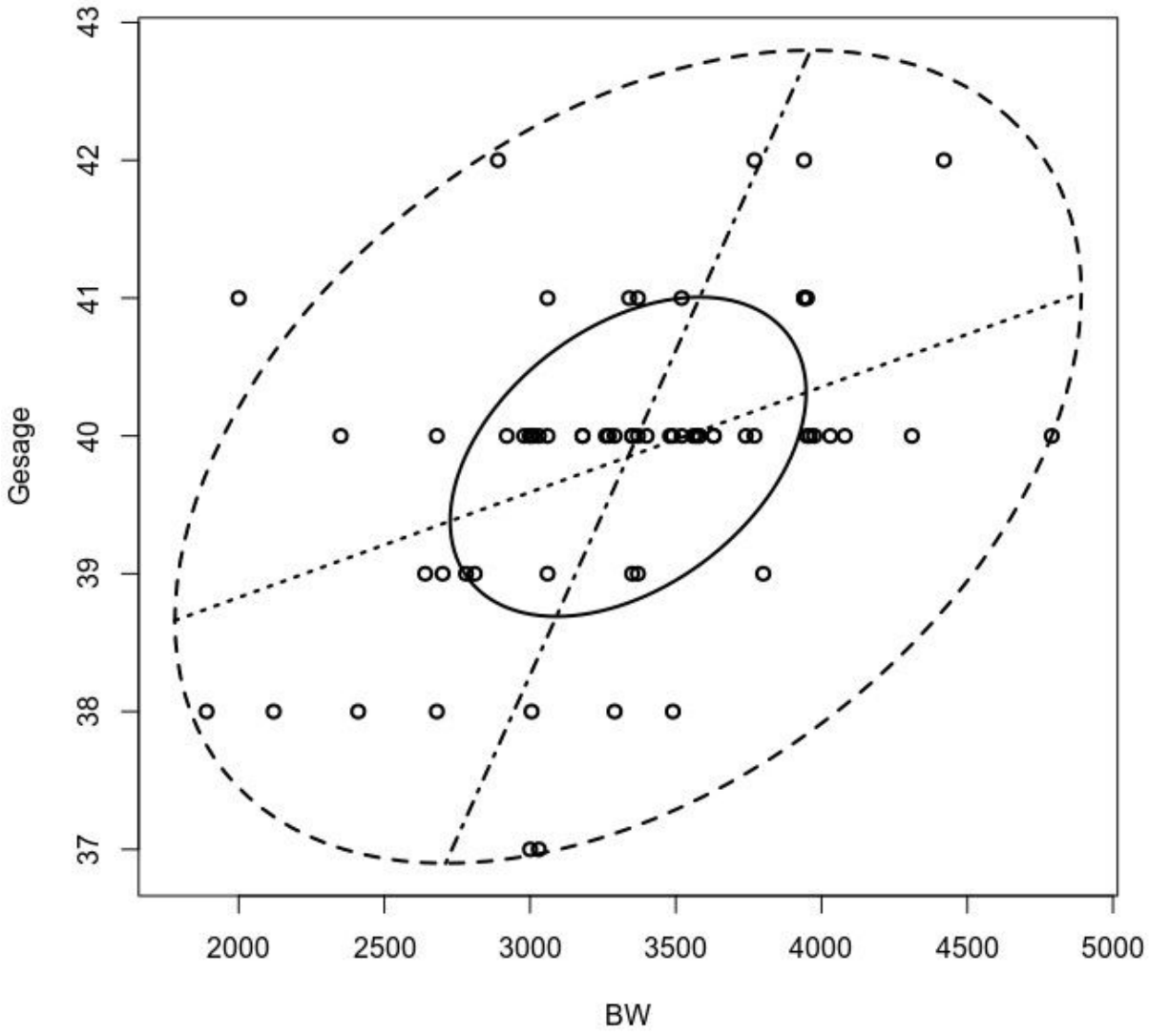

Figure 1.6 *Bivariate boxplot with robust method*

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

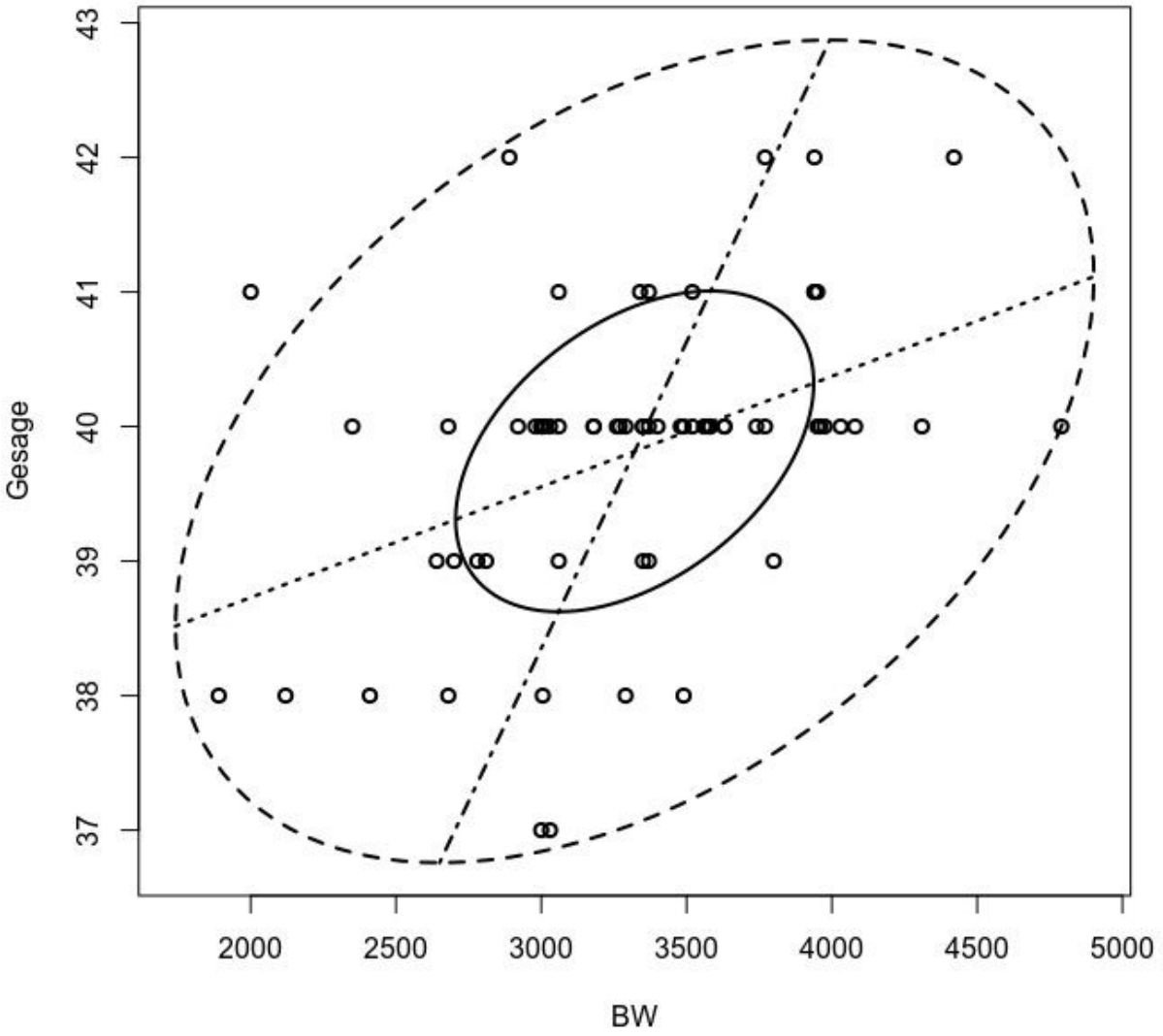

Figure 1.7 *Bivariate boxplot with other method*

Bivariate boxplot can be used to see the outlier. We can use either robust or other method to run the bivariate boxplot. This time, those two method give the similar result and we can see that only one outlier (Gesage=41, the most left one).

Bivariate density perspective

```
##### perspective  plot of Bivariate Density ####
den1<-bivden(BW,Gesage)
persp(den1$seqx,den1$seqy,den1$den,xlab="BW",ylab="Gesage",
      zlab="Density",lwd=2)
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

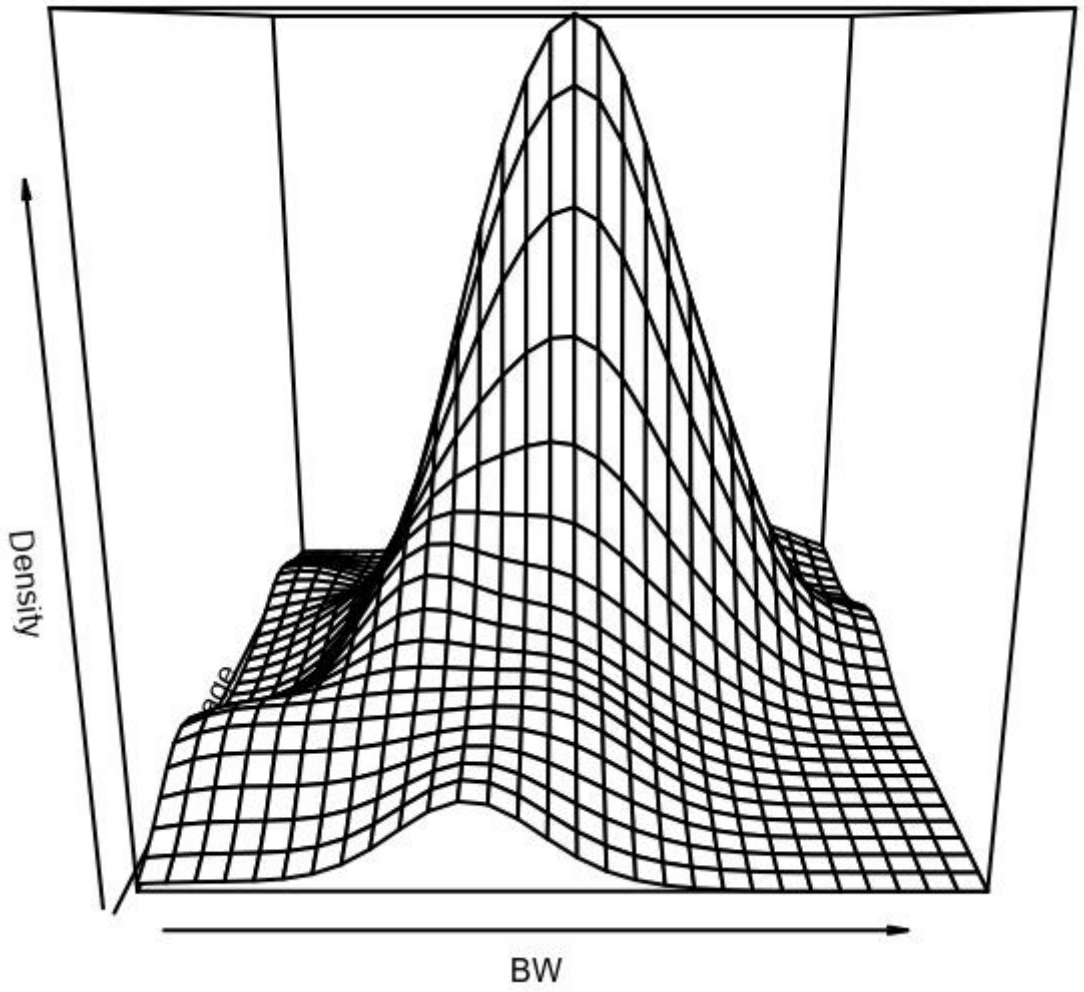

Figure 1.8 *Perspective plot of the estimated bivariate density*

Another way to see bivariate boxplot is in 3D image. Unfortunately, in this case we can see much about the outlier.

Bivariate density plot contour

```
##### contour plot of Bivariate Density ####
plot(BW,Gesage)
contour(den1$seqx,den1$seqy,den1$den,lwd=2,nlevels=20,add=TRUE)
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

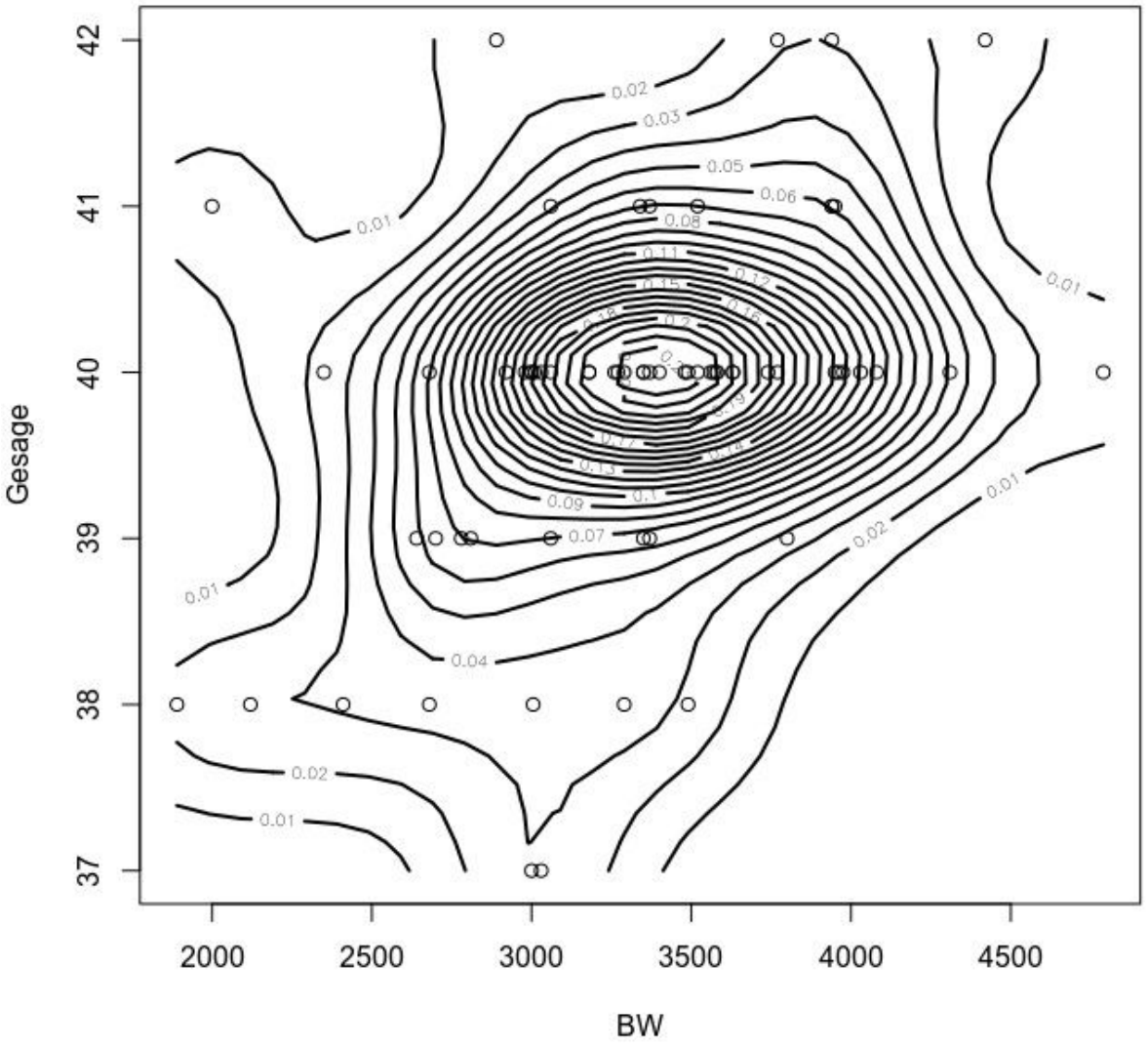

Figure 1.9 *Contour plot of the estimated bivariate density*

Another way to see bivariate density is using contour. Here, we can see more clearly about the outlier that we suspect before, (Gasage=41, the left most).

```
##### matrix plots  ####
pairs(data,panel=function(x,y) {abline(lsfit(x,y)$coef,lwd=2)
 lines(lowess(x,y),lty=2,lwd=2)
 points(x,y)})
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

Scatter plot all pairs

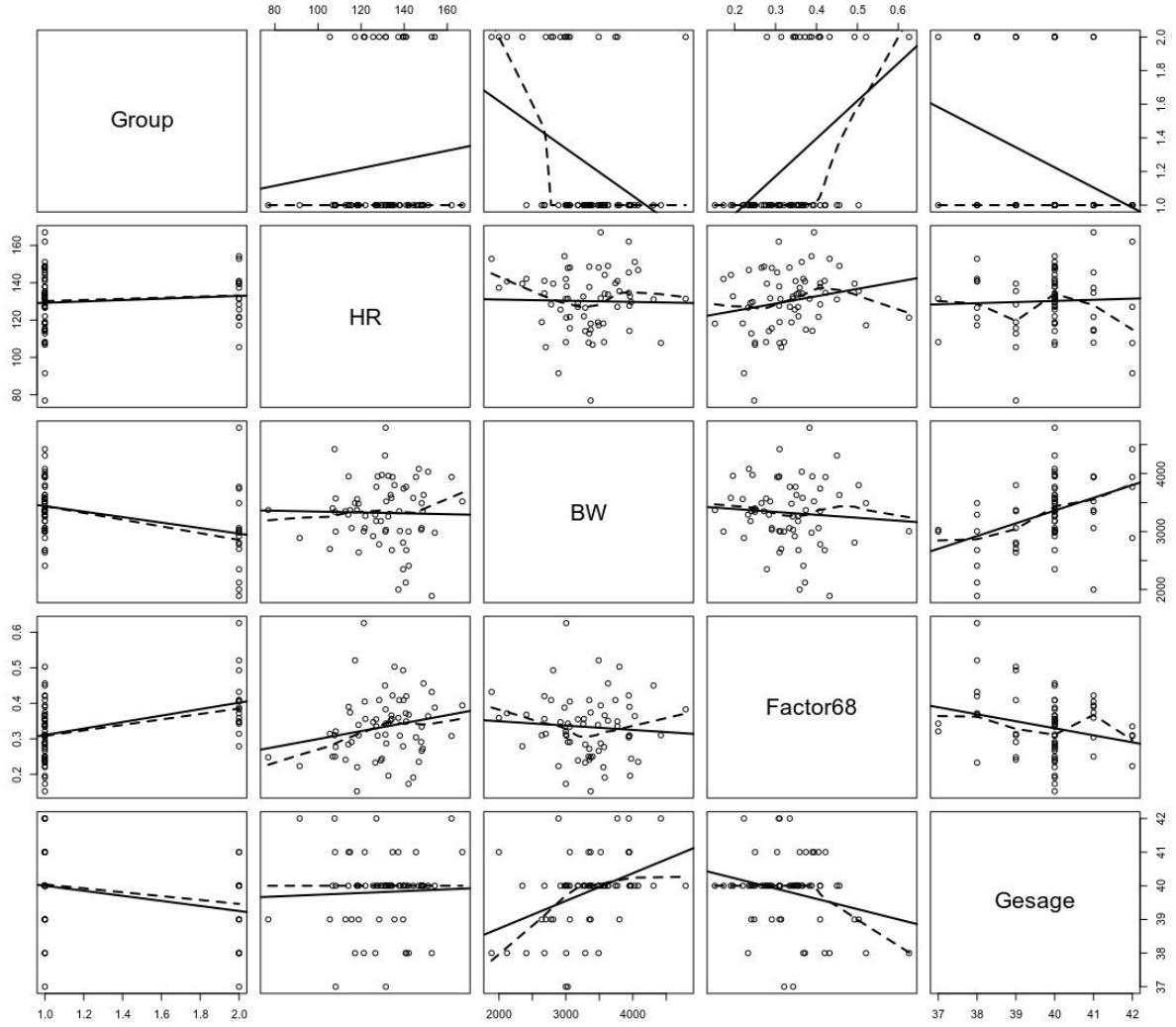

Figure 1.10 *Scatter plot of all pairs with linear and local weight regression fits.*

In the figure 1.10, the regressions doesn't have a good fit among all of the possible pair. Even our best pair (BW-Gesage) also doesn't have high dependency.

Conditioning plot

```
#### Conditioning Plot  ######
coplot(BW~Gesage|Factor68)
coplot(BW~Gesage|Factor68,panel=function(x,y,col,pch)
 panel.smooth(x,y,span=1))
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

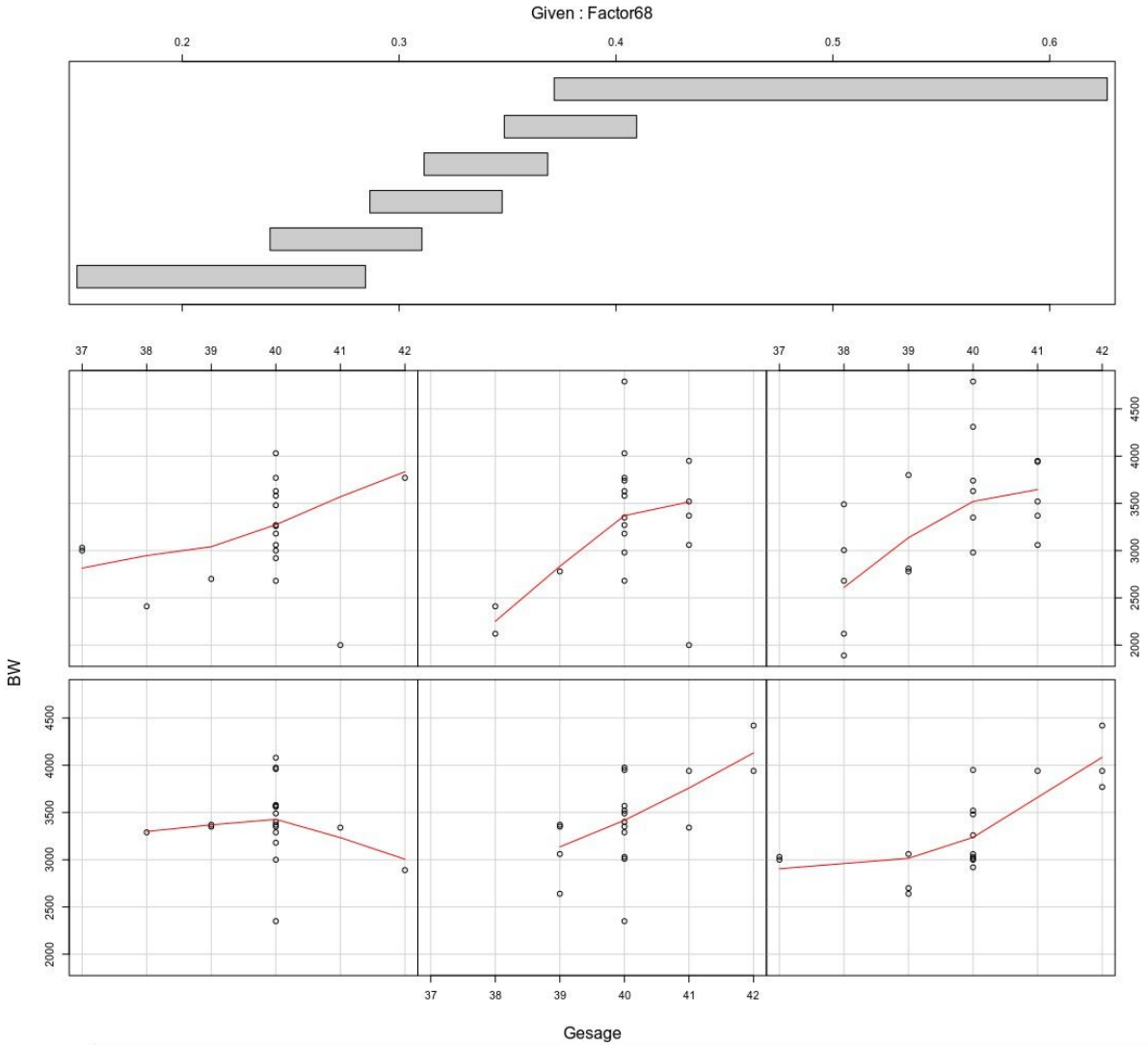

Figure 1.11 *Coplot BW and Gesage conditional on Factor68 with added locally weighted regression fit*

In the figure 1.11, we can see the connection between BW and Gesage for each slice base on Factor68. It's not quite clear, but in the 5th slice, we can see a better fit regression for BW and Gesage.

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

Answer 2

Calculating the principal components by using correlation matrix can be done by using this code:

```
# Calculating Principal Component Value.
# Calculating the Correlation Matrix of attribute 2 to 5 (HR to
Gesage)
# cor=TRUE implies using Correlation Matrix
data.pc<-princomp(data[,2:5],cor=TRUE)
summary(data.pc,loadings=TRUE)
```

I exclude “Group column in that code by slicing the data. The result is shown below:

```
> summary(data.pc,loadings=TRUE)
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4
Standard deviation     1.2402115 1.0707983 0.8948664 0.7172730
Proportion of Variance 0.3845312 0.2866522 0.2001965 0.1286201
Cumulative Proportion  0.3845312 0.6711834 0.8713799 1.0000000

Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4
HR       -0.148  0.760  0.584  0.242
BW        0.590  0.310 -0.488  0.563
Factor68 -0.444  0.517 -0.633 -0.367
Gesage    0.658  0.242  0.142 -0.699
```

Figure 2.1 *Summary of principal component analysis*

In the figure 2.1, we can see the components that has standard deviation more than 1 are component 1 and component 2. But, their cumulative proportion is only 0.67. This number is smaller than usual threshold (70%). Even though, I will choose these first two component only as principal component since it’s quite close to 70 and the third component has standard deviation that is less than 1. So the component 3 and component 4 are discarded from this step.

In the next step, I want to see the plot for each principal component (PC) to the Group, perhaps we can see a pattern here. I do it by using this code:

```
### Create Graph for Group against PC1 and Pc2
par(mfrow=c(1,2))
plot(data.pc$scores[,1],Group,xlab="PC1")
plot(data.pc$scores[,2],Group,xlab="PC2")
```

And the scatter plot are below:

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

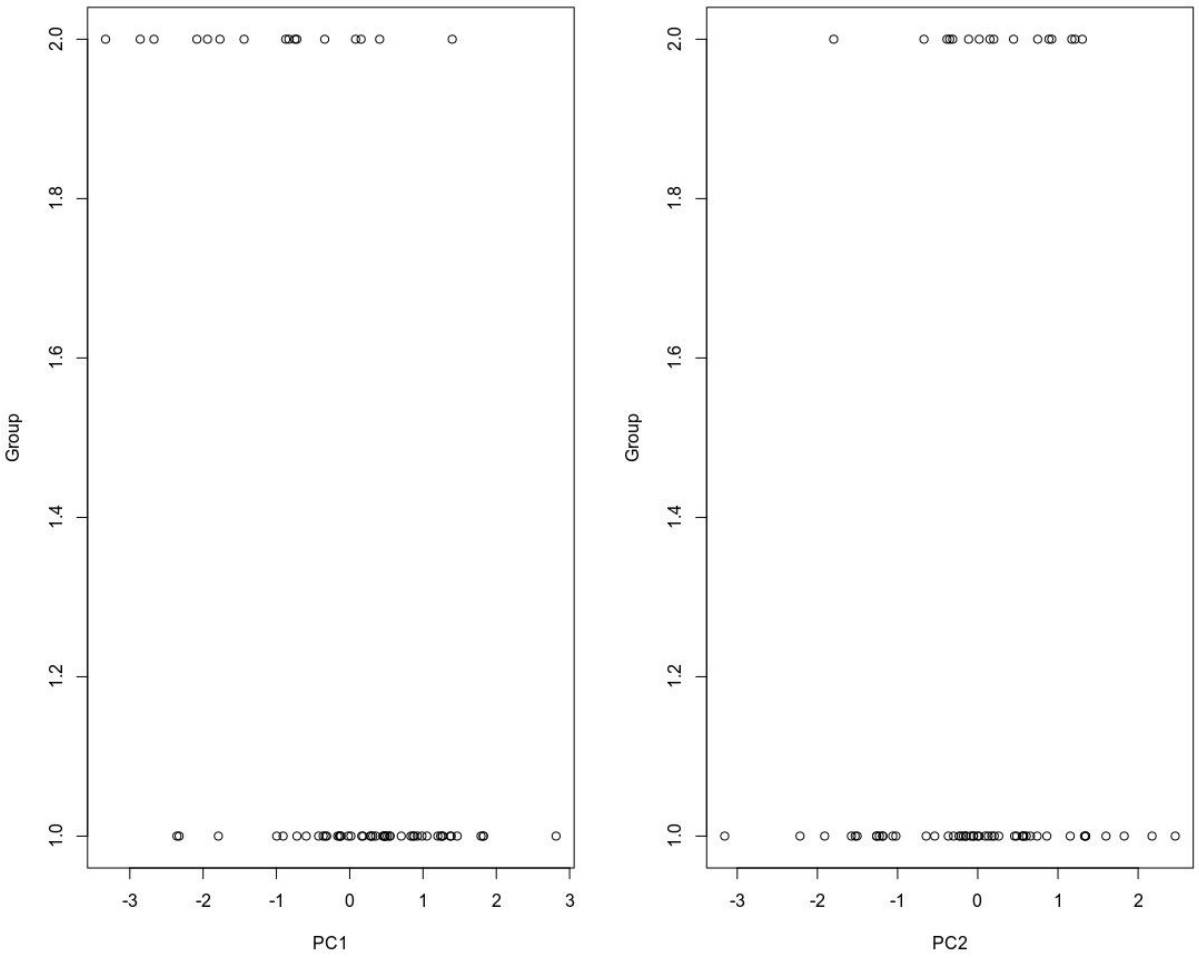

Figure 2.2 *Scatter plot between PC1 and PC2 to Group*

But here, we can't say many thing since Group is a class and only has 2 values. In the next step, we can see the scatter plot for PC1 vs PC2 by using this code:

```
par(pty="s")
plot(data.pc$scores[,1],data.pc$scores[,2],
ylim=range(data.pc$scores[,1]), xlab="PC1",ylab="PC2",type="n",lwd=2)
text(data.pc$scores[,1],data.pc$scores[,2],
labels=abbreviate(row.names(data)),cex=0.7,lwd=2)
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

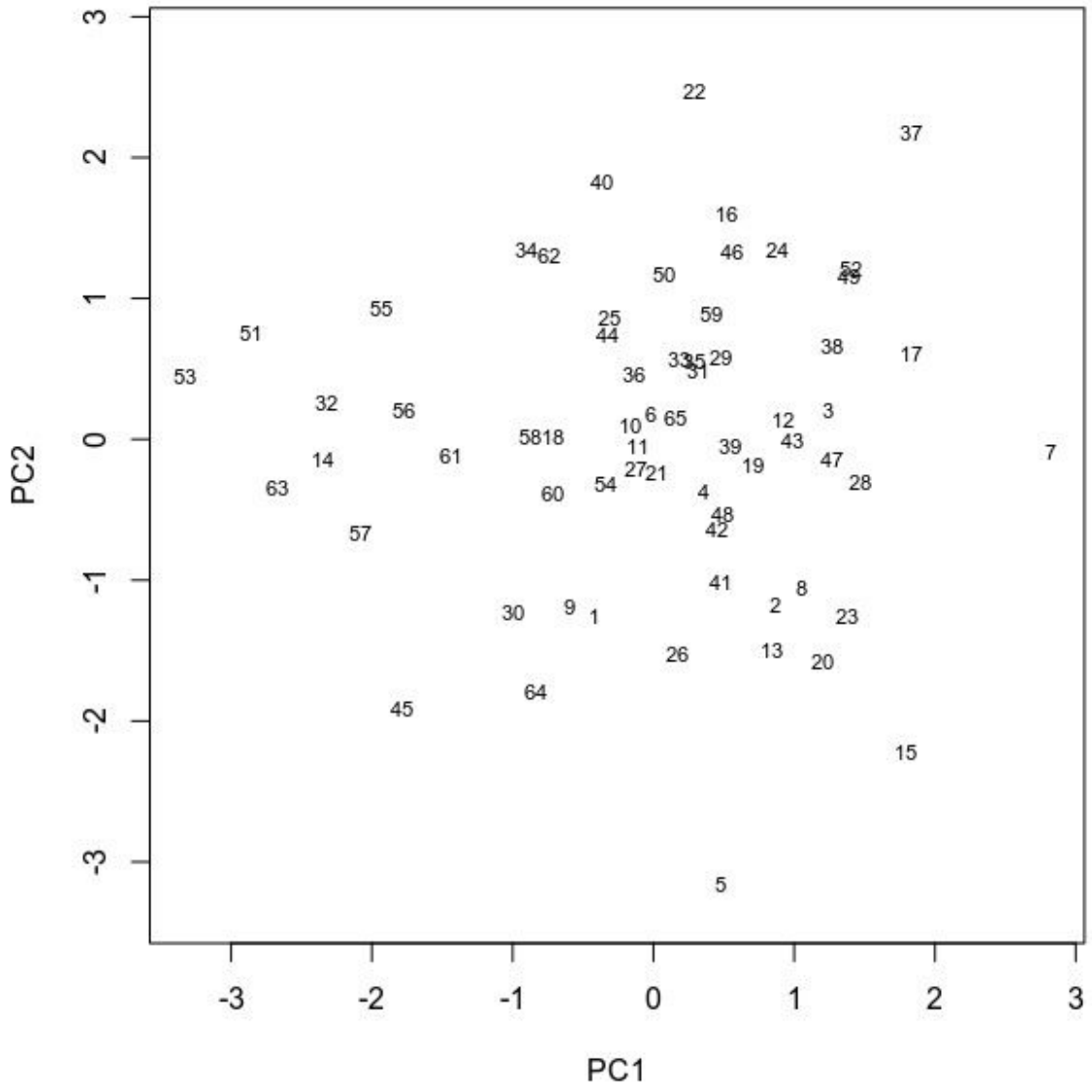

Figure 2.3 *Scatter plot between PC1 and PC2*

Since there is only one pair of principal component, we only need to check one scatter plot. It shouldn't have outlier or special feature. By checking the figure 2.3, we can see that there is almost no outlier. The only suspect is number 5. But this is much better than the scatter plot for any pair in the question 1 (there are several far / outlier).

After performing this PCA we can conclude that component 1 and component 2 are the principal component for this data set.

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

Answer 3

Dendrogram

First, we will create the whole dendrogram from the clustering using 3 methods. I exclude the group column for an obvious reason. Here is the code:

```
# Ignore the group column
data.components = data[,2:5]

# Create dendrogram for the clustering using 3 methods
par(mfrow=c(1,3))
plclust(hclust(dist(data.components),method="single"),labels=row.name
s(data.components),ylab="Distance")
title("(a) Single linkage")
plclust(hclust(dist(data.components),method="complete"),labels=row.na
mes(data.components),ylab="Distance")
title("(b) Complete linkage")
plclust(hclust(dist(data.components),method="average"),labels=row.nam
es(data.components),ylab="Distance")
title("(c) Average linkage")
```

Here is the result from the code above:

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

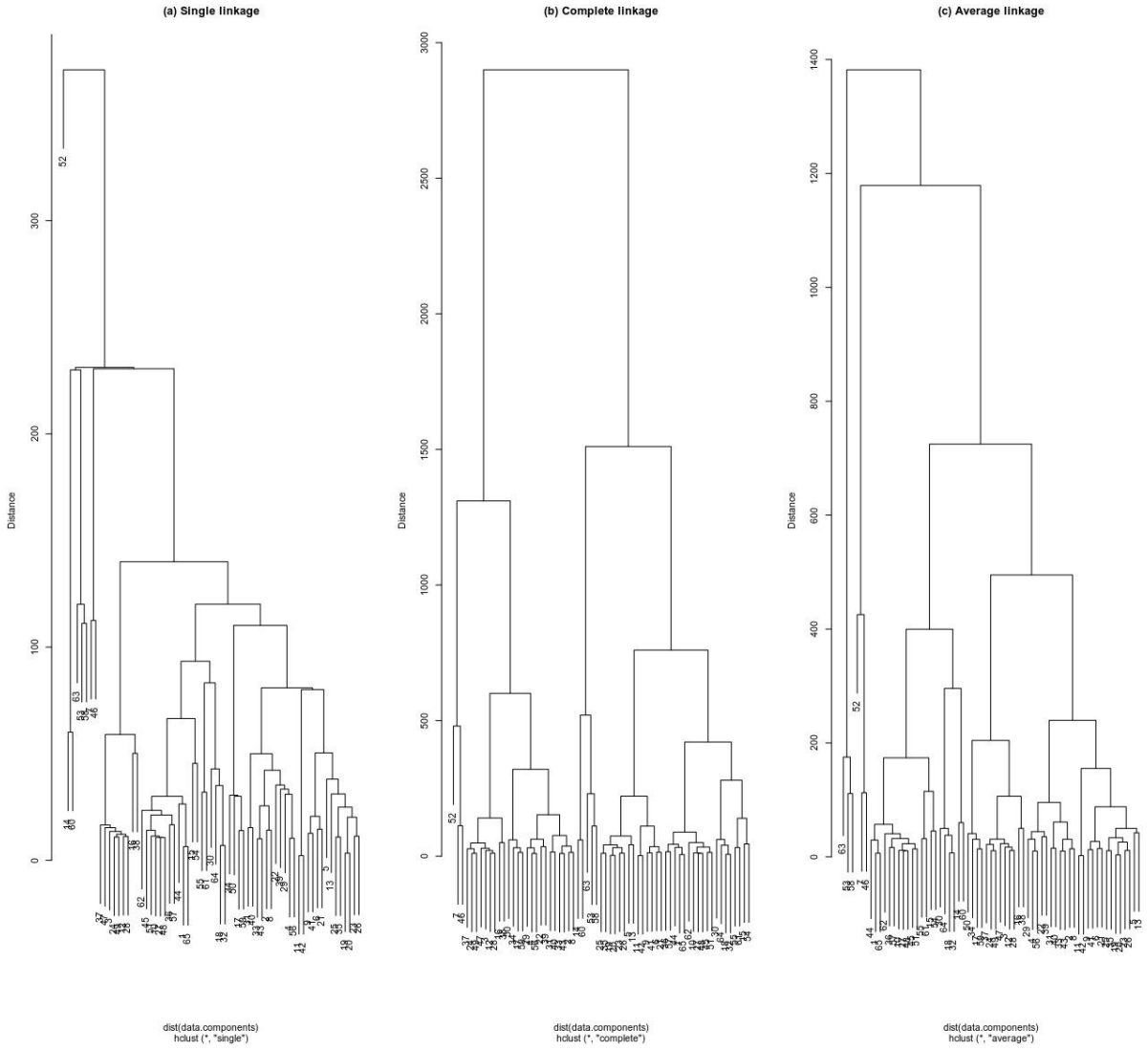

Figure 3.1 *Three dendrogram from the clustering using three different method*

From the figure 3.1 we can see the dendrogram for each method. SInce we know that the original data has only two group, we can see that single linkage is not a good representation for the clustering. By using single linkage, we will get two clusters and only one element in one of the cluster.

The other two dendrograms have a better representation for the data. Since the question ask for two distinct method, I will use the complete and average method.

Optimum number of cluster (kNN)

Next we want to know what's the optimum number of cluster for this data. I use this code to generate the graph that compare number of groups and within group sum of squares:

```
# knn analysis to find optimum number of cluster
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

```
rge<-apply(data.components,2,max)-apply(data.components,2,min) #
Range = max - min
data.dat<-sweep(data.components,2,rge,FUN="/")  # naive normalization
: dividing by the range
#
n<-length(data.dat[,1])  # number of observation
wss1<-(n-1)*sum(apply(data.dat,2,var))  # get variance from the
column
wss<-numeric(0)
for(i in 2:5) {
 W<-sum(kmeans(data.dat,i)$withinss)
 wss<-c(wss,W)
}
wss<-c(wss1,wss)
plot(1:5,wss,type="l",xlab="Number of groups",ylab="Within groups sum
of squares",lwd=2)
```

And below is the result:

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

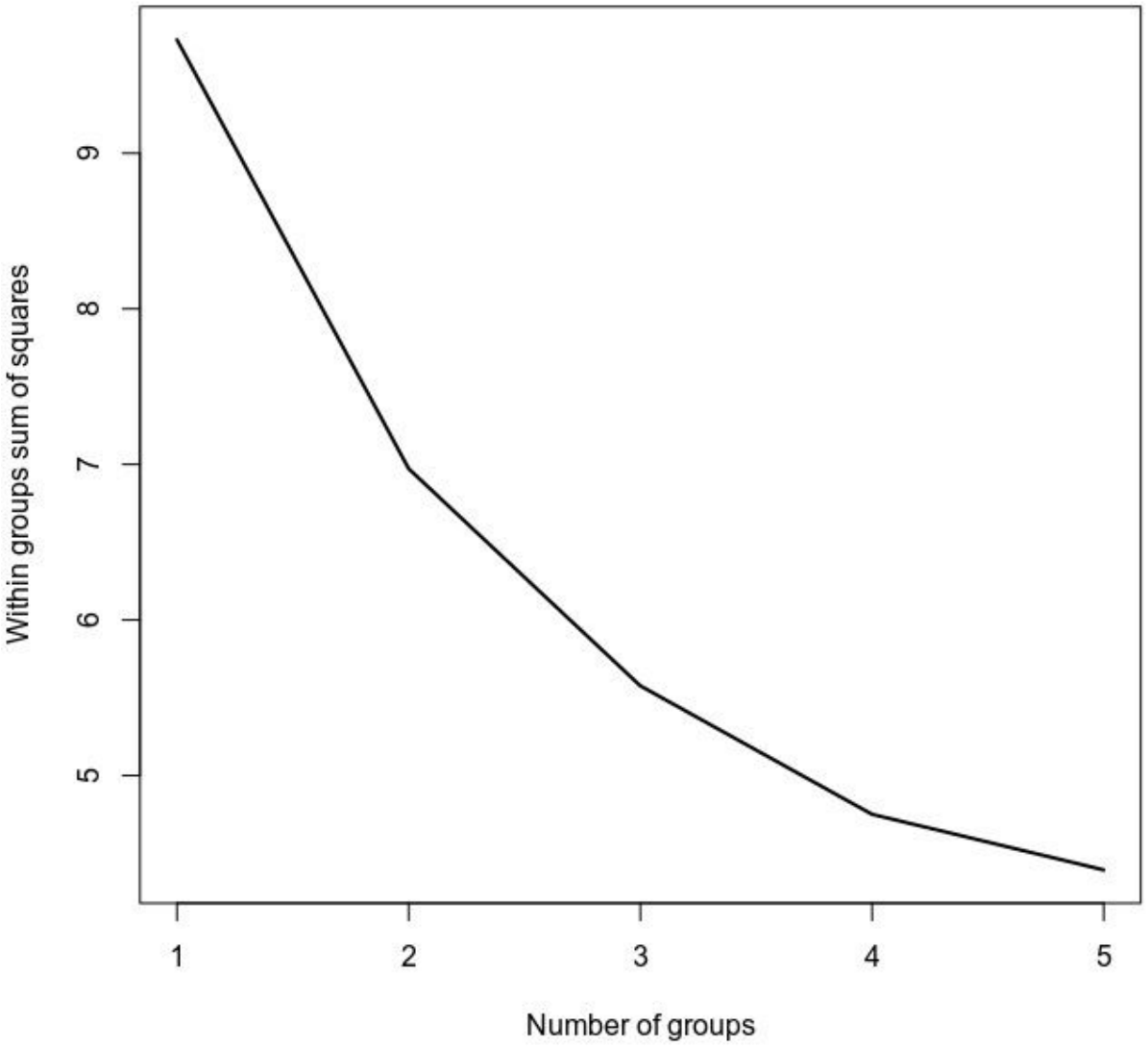

Figure 3.2 *Plot of number of groups to within group sum of squares*

Based on the figure 3.2, it's not easy to choose the optimum number of clusters. But I will choose 2 as the number of groups since the slope is much higher than 3 (the rest is even smaller).

Accuracy compared to original data

Now we can compare the clustering result with the original group. Since in clustering we can specify which cluster is which group, I will just compare the result from the clustering to the original group and do it reversely also (i.e. cluster 1 = group 1 and cluster 1 group 2). Then we choose the higher accuracy. I did it by using this code:

```
# Complete method
# Use h=2000 to cut the dendrogram (example) and get 2 clusters
twoComplete<-cutree(hclust(dist(data.components),method="complete"),h
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

```
=2000)
as.data.frame(twoComplete) # show the content vertically

data.clusComplete<-lapply(1:2,function(nc)
row.names(data)[twoComplete==nc])
data.meanComplete<-lapply(1:2,function(nc)
apply(data[twoComplete==nc,],2,mean))
data.meanComplete
data.clusComplete

# Compare to original data
accuracyComplete1<-sum(abs(twoComplete-Group)) / nrow(data)
accuracyComplete1
# Inverse the value of twoComplete
twoComplete2 <- (3 - twoComplete)
twoComplete2
accuracyComplete2<-sum(abs(twoComplete2-Group)) / nrow(data)
accuracyComplete2
```

At first we cut the tree so that we only have 2 cluster, then we compare to the original data to get the first accuracy, then we inverse the comparison. Finally we get two number of accuracy: 0.538 and 0.461. So, we can say that the clustering using the Complete method has accuracy **0.538**.

By doing the same thing but with Average method, using this code:

```
# Average method
# Use h=1200 to cut the dendeogram (example) and get 2 clusters
twoAverage<-cutree(hclust(dist(data.components),method="average"),h=1
200)
as.data.frame(twoAverage) # show the content vertically

data.clusAverage<-lapply(1:2,function(nc)
row.names(data)[twoAverage==nc])
data.meanAverage<-lapply(1:2,function(nc)
apply(data[twoAverage==nc,],2,mean))
data.meanAverage
data.clusAverage
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

```
# Compare to original data
accuracyAverage1<-sum(abs(twoAverage-Group)) / nrow(data)
accuracyAverage1
# Inverse the value of twoAverage
twoAverage2 <- (3 - twoAverage)
accuracyAverage2<-sum(abs(twoAverage2-Group)) / nrow(data)
accuracyAverage2
```

From this method we got accuracy number 0.8 and 0.2. So, the accuracy for the Average method is **0.8.**

Classify new observations

By using the two method above we can classify the new observations. First, we need to load the new observation by using this code:

```
# New data
# Declare new data
newdata<-rbind(c(110,3320,0.240,39), c(120,3310,0.298,37))
colnames(newdata) <- colnames(data.components)
newdata <- data.frame(newdata)
```

After that by using LDA we can predict the group for the new observations. Below is the code to do it for both method:

```
## Complete method
# linear discriminant analysis
disComplete<-lda(twoComplete~HR+BW+Factor68+Gesage,data=data.componen
ts,prior=c(0.5,0.5))

# Predict the cluster for the new data
predict(disComplete,newdata = newdata)

## Average method
# linear discriminant analysis
disAverage<-lda(twoAverage~HR+BW+Factor68+Gesage,data=data.components
,prior=c(0.5,0.5))
```

Name: Ismail Sunni
Email: imajimatika@gmail.com
UJI: al388270@uji.es

```
# Predict the cluster for the new data
predict(disAverage,newdata = newdata)
```

The result is below:

Method	**Complete method**		**Average method**	
Observation	**Group**	**Confidence**	**Group**	**Confidence**
1	1	0.945	1	0.998
2	1	0.900	1	0.998

From the result, it can be concluded that both of the new observations are belongs to group 1 with high confidence with both methods.