Yevgeniya Litvinova
al361243@uji.es

SIW004 - Applied Mathematics: Logic and Statistics

Homework 3: Final Homework (individual)

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

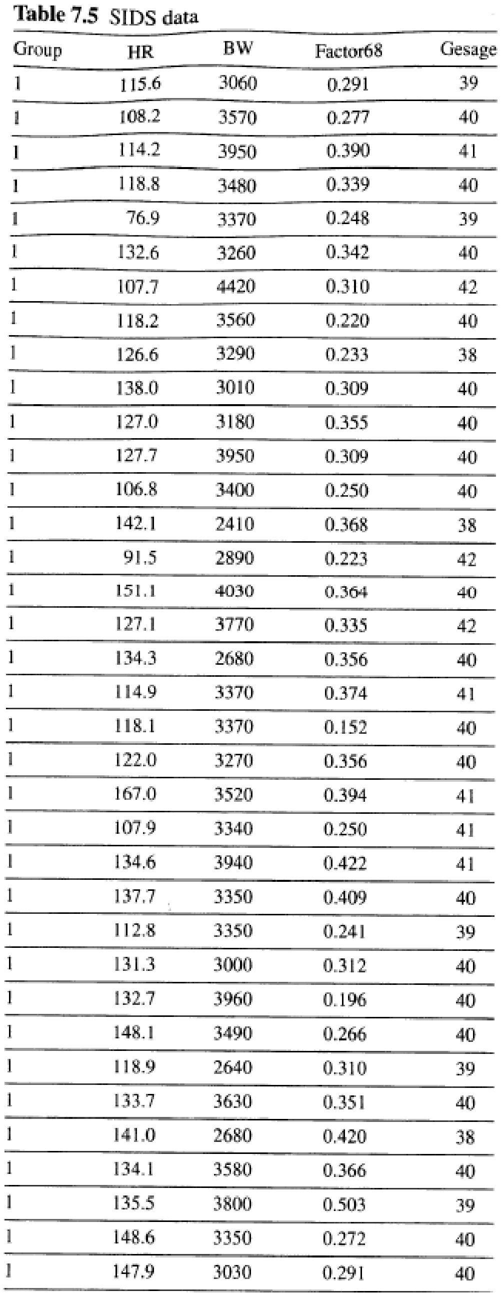


Table 7.5 SIDS data

Group	HR	BW	Factor68	Gesage
1	115.6	3060	0.291	39
1	108.2	3570	0.277	40
1	114.2	3950	0.390	41
1	118.8	3480	0.339	40
1	76.9	3370	0.248	39
1	132.6	3260	0.342	40
1	107.7	4420	0.310	42
1	118.2	3560	0.220	40
1	126.6	3290	0.233	38
1	138.0	3010	0.309	40
1	127.0	3180	0.355	40
1	127.7	3950	0.309	40
1	106.8	3400	0.250	40
1	142.1	2410	0.368	38
1	91.5	2890	0.223	42
1	151.1	4030	0.364	40
1	127.1	3770	0.335	42
1	134.3	2680	0.356	40
1	114.9	3370	0.374	41
1	118.1	3370	0.152	40
1	122.0	3270	0.356	40
1	167.0	3520	0.394	41
1	107.9	3340	0.250	41
1	134.6	3940	0.422	41
1	137.7	3350	0.409	40
1	112.8	3350	0.241	39
1	131.3	3000	0.312	40
1	132.7	3960	0.196	40
1	148.1	3490	0.266	40
1	118.9	2640	0.310	39
1	133.7	3630	0.351	40
1	141.0	2680	0.420	38
1	134.1	3580	0.366	40
1	135.5	3800	0.503	39
1	148.6	3350	0.272	40
1	147.9	3030	0.291	40

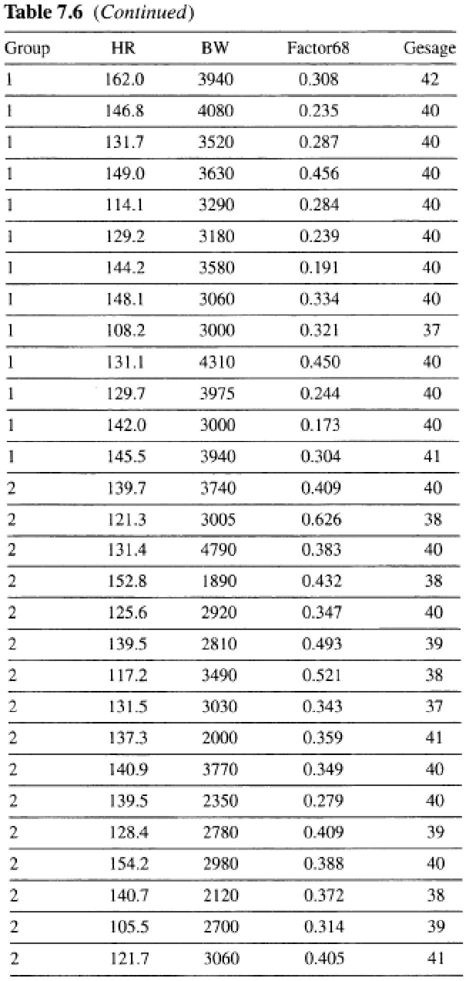


Table 7.6 (*Continued*)

Group	HR	BW	Factor68	Gesage
1	162.0	3940	0.308	42
1	146.8	4080	0.235	40
1	131.7	3520	0.287	40
1	149.0	3630	0.456	40
1	114.1	3290	0.284	40
1	129.2	3180	0.239	40
1	144.2	3580	0.191	40
1	148.1	3060	0.334	40
1	108.2	3000	0.321	37
1	131.1	4310	0.450	40
1	129.7	3975	0.244	40
1	142.0	3000	0.173	40
1	145.5	3940	0.304	41
2	139.7	3740	0.409	40
2	121.3	3005	0.626	38
2	131.4	4790	0.383	40
2	152.8	1890	0.432	38
2	125.6	2920	0.347	40
2	139.5	2810	0.493	39
2	117.2	3490	0.521	38
2	131.5	3030	0.343	37
2	137.3	2000	0.359	41
2	140.9	3770	0.349	40
2	139.5	2350	0.279	40
2	128.4	2780	0.409	39
2	154.2	2980	0.388	40
2	140.7	2120	0.372	38
2	105.5	2700	0.314	39
2	121.7	3060	0.405	41

The data under study were collected in an investigation of sudden infant death syndrome (SIDS). The two groups here consist of 16 SIDS victims and 49 controls. BW represents Birthweight variable. The Factor68 variable arises from spectral analysis of 24 hour recordings of electrocardiograms and respiratory movements made on each child. All the infants have a gestational age of 37 weeks or more and were regarded as full term. Four variables were recorded for each infant as follows:

GROUP - victim of SIDS (2) or control (1),
HR - heart rate in beats per minute,
BW - birthweight in grams,
FACTOR68 - heart and lung function,
GESAGE - gestational age in weeks.

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

In order to identify any existing relationships between the variables in our dataset, we start our analysis by building the matrix of 6 possible scatterplots generated from four variables. The result shown in Figure 1.1 (R code is indicated after the graph) and the correlation matrix below suggest that several pairs of variables in the sudden infant death syndrome data appear to be related in a relatively complex fashion.

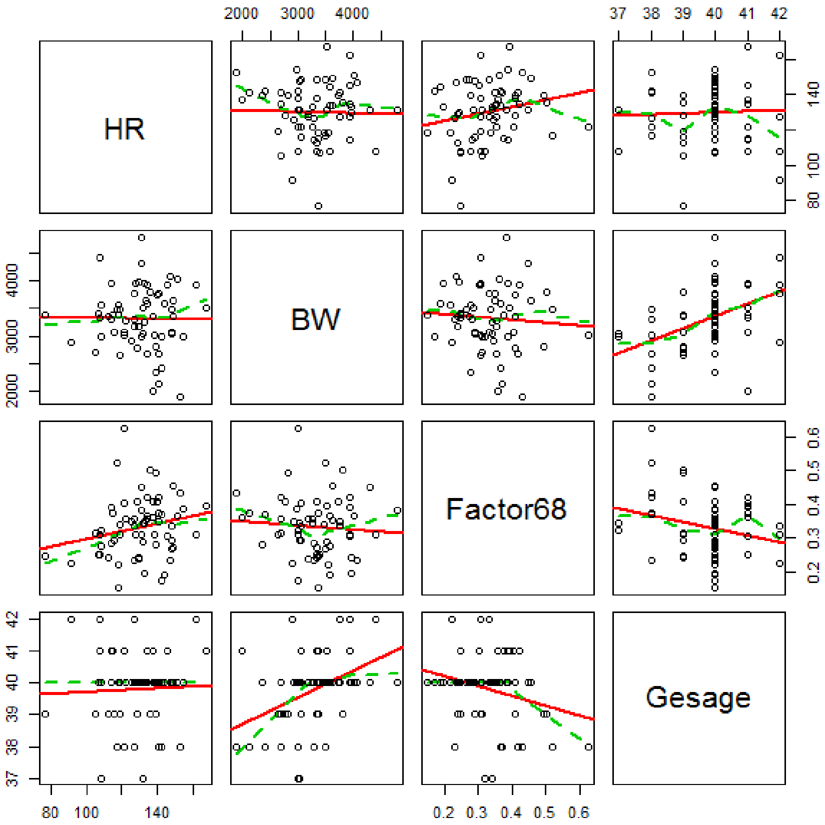
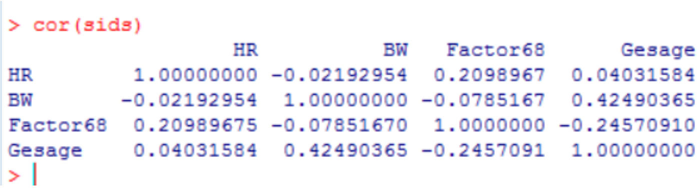


Figure 1.1. Scatterplot matrix of sudden infant death syndrome data showing linear and locally weighted regression on each panel.

The scatterplot above was produced using the following **R code:**

```
sids<- data.frame(read.csv("C:\\Users\\Yevgeniya Litvinova\\Documents\\SIW 004 Math and Stats\\final individual\\final_data.csv"))
sids <- sids[,-1]
attach(sids)
pairs(sids,panel=function(x,y) {abline(lsfit(x,y)$coef,lwd=2, col=2)
                         lines(lowess(x,y),lty=2,lwd=2, col=3)
                         points(x,y)})
```



```
> cor(sids)
                  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
> |
```

Let us hypothesize that birth weight depends on gestational age so we can proceed with data exploration in this exercise. The first step in trying to prove our assumption is to examine a scatterplot of the aforementioned variables (Figure 1.2). Plots (a) and (b) suggest a possible link between increasing gestational age and birth weight. The dataset under study does not consist of a large number of observations, however in this analysis we still undertake jittering and the amount of added noise becomes easily noticeable from (c) .

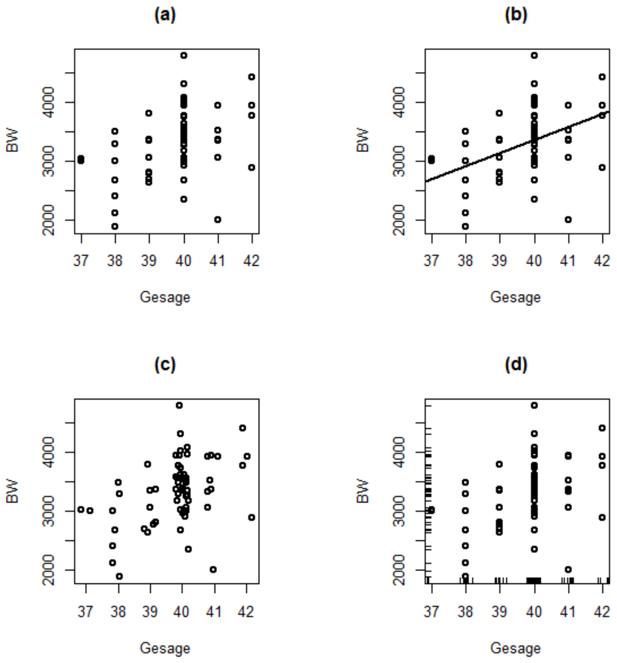


Figure 1.2. (a) Scatterplot for Birthweight against Gestational age; **(b)** scatterplot for Birthweight against Gestational age with added linear regression; **(c)** jittered scatterplot for Birthweight against Gestational age; **(d)** scatterplot for Birthweight against Gestational age with information about marginal distribution of the two variables added.

The scatterplot above was produced using the following **R code:**

```
par(mfrow=c(2,2))
par(pty="s")
plot(Gesage,BW,pch=1,lwd=2)
title("(a)",lwd=2)
plot(Gesage,BW,pch=1,lwd=2)
abline(lm(BW~Gesage),lwd=2)
title("(b)",lwd=2)
airpoll1<-jitter(cbind(Gesage,BW))
plot(airpoll1[,1],airpoll1[,2],xlab="Gesage",ylab="BW",pch=1,lwd=2)
title("(c)",lwd=2)
plot(Gesage,BW,pch=1,lwd=2)
rug(jitter(Gesage),side=1)
rug(jitter(BW),side=2)
title("(d)",lwd=2)
```

Creating a scatterplot where data is labelled according to its name/ordinal number (Figure 1.3) helps understanding which of the observations exhibit an unusual behavior. Thus, in this case those could be Infant 52 with the maximum value of birth weight uncommon for its gestational age group or, in contrast, Infant 58 with a dramatically low weight value.

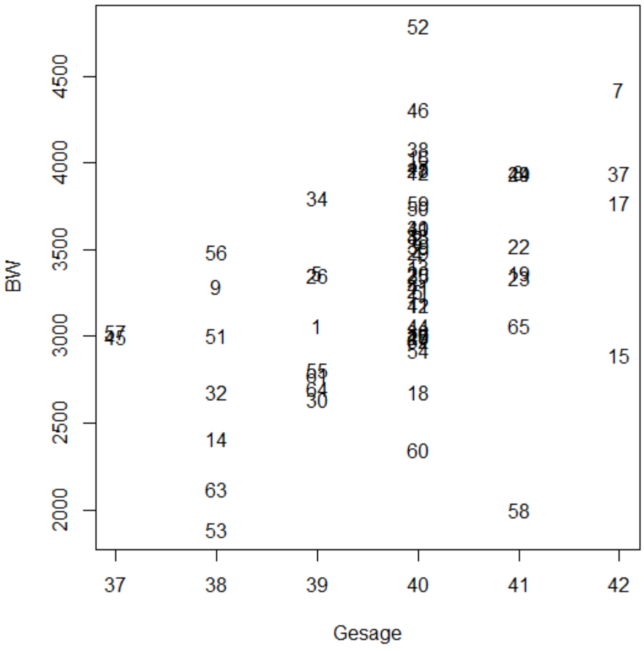


Figure 1.3. Scatterplot for Birthweight against Gestational age with points labeled by observation number.

The scatterplot above was produced using the following **R code:**

```
names<-abbreviate(row.names(sids))
par(mfrow=c(1,1))
```

```
plot(Gesage,BW,lwd=2,type="n")
text(Gesage,BW,labels=names,lwd=2)
```

In Figure 1.4, a simple linear and locally weighted regression fits were added to the scatterplot. Such fits are designed to use the data themselves to suggest the type of fit needed. The resulting graphs are shown in Figure 1.4 where apart from a small wobble for gestational age values 38 to 40, the linear fit and the locally weighted fit are rather similar.

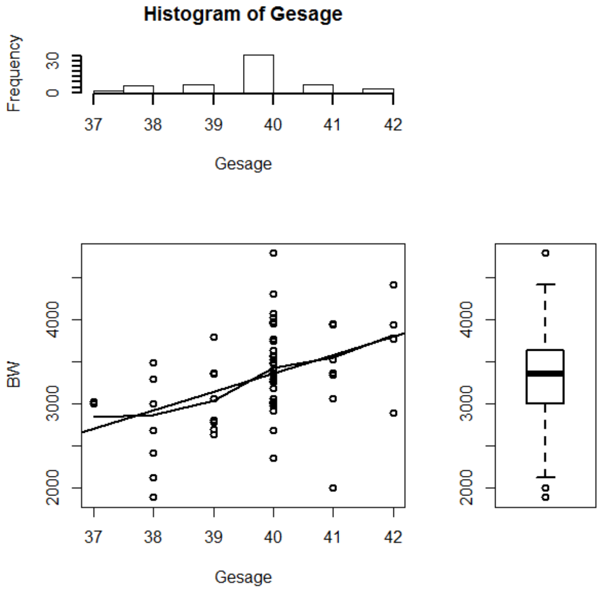


Figure 1.4. Scatterplot for Birthweight against Gestational age with added linear regression and locally weighted regression fits and marginal distribution information.

The scatterplot above was produced using the following **R code:**

```
par(fig=c(0,0.7,0,0.7))
plot(Gesage,BW,lwd=2)
abline(lm(BW~Gesage),lwd=2)
lines(lowess(Gesage,BW),lwd=2)     #local weighted regression
par(fig=c(0,0.7,0.65,1),new=TRUE)
#
hist(Gesage,lwd=2)
par(fig=c(0.65,1,0,0.7),new=TRUE)
boxplot(BW,lwd=2)
```

Building convex hull scatterplot helps identify the outlying observations, which can often considerably distort the value of a correlation coefficient and might be excluded from the calculation without disturbing the general shape of the bivariate distribution.

The convex hull in Figure 1.5 was built on the scatterplot of two variables: birth weight and gestational age. Initial correlation value between these variables was 0.424 and the removal of outliers made it rise to 0.504. As one can observe, the change in the correlation after removal of points giving the shape to the convex hull is small but distinguishable.

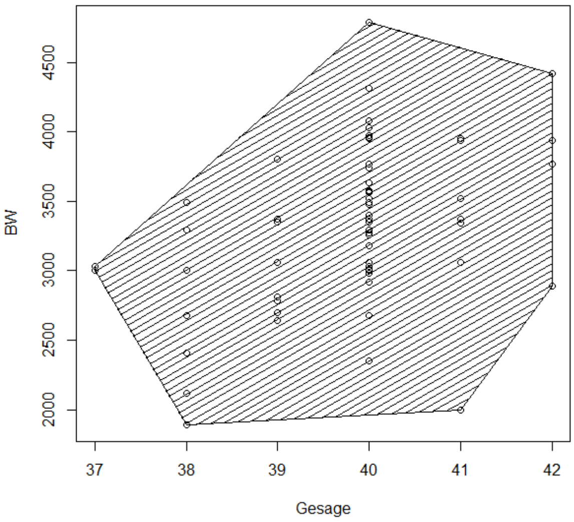


Figure 1.5. Scatterplot for Birthweight against Gestational age showing convex hull of the data.

The scatterplot above was produced using the following **R code:**

```
hull<-chull(Gesage,BW) #to draw a polygon with boundaries for point
plot(Gesage,BW,pch=1)
polygon(Gesage[hull],BW[hull],density=15,angle=30)
cor(Gesage,BW)
cor(Gesage[-hull],BW[-hull])
```

Chi-plot allows making more careful judgements on whether or not the variables are independent. As shown in Figure 1.6., there is only a moderate dependence (previously shown by the correlation value of 0.5) between the birth weight and gestational age suggested by the fact that about 1/2 of the observations are located in the horizontal band indicated on the plot.

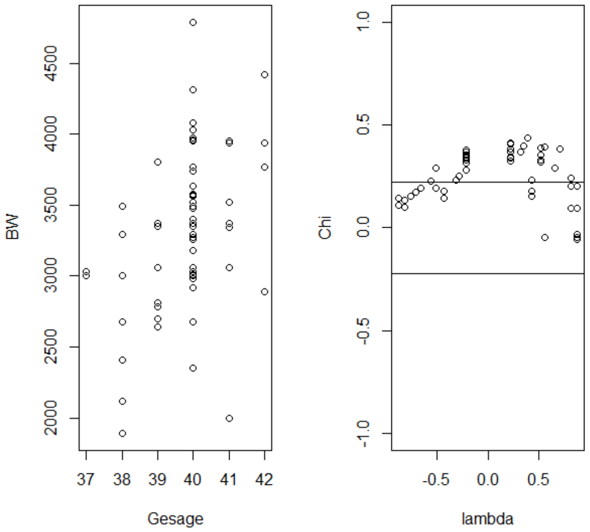


Figure 1.6. Chi-plot of Birthweight and Gestational age.

The scatterplot above was produced using the following **R code:**

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

Robust estimators of location, scale, and correlation were used to build the bivariate boxplot illustrated in Figure 1.7. The inner ellipse the "hinge" which contains 50 percent of the data. The outer is the "fence" and only one observation outside of it could possible indicate that it is a troublesome outlier.

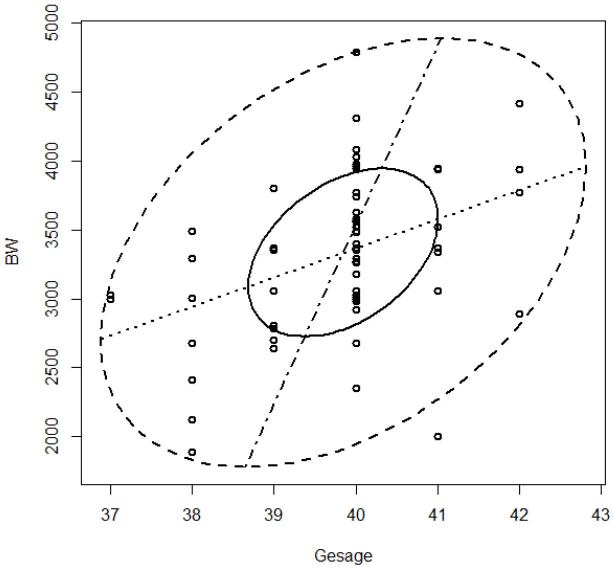


Figure 1.7. Bivariate boxplot of Birthweight and Gestational age (robust estimator of location, scale, and correlation).

The scatterplot above was produced using the following **R code:**

```
bvbox(cbind(Gesage,BW),xlab="Gesage",ylab="BW")
```

In the code above, two variables Gesage and BW make up the bivariate distribution and the default robust estimators are used to create a bivariate boxplot.

Bivariate density scatterplots are helpful to identify regions where there are high and low densities/ “clusters” of observations. Figure 1.8 (the peak) and Figure 1.9 (small contour interval) display the increase in density for the medium values of gestational age and birth weight.

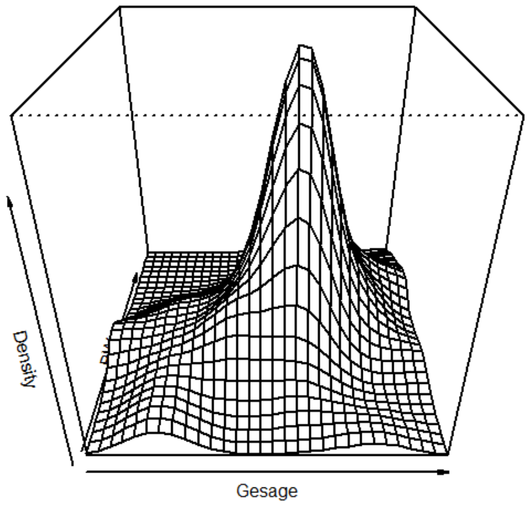


Figure 1.8. Perspective plot of estimated bivariate density of Birthweight and Gestational age.

The scatterplot above was produced using the following **R code:**

```
den1<-bivden(Gesage,BW)
persp(den1$seqx,den1$seqy,den1$den,xlab="Gesage",ylab="BW",
zlab="Density",lwd=2, theta = 0, phi = 30) #theta (azimuthal angle) and phi (vision angle)
give different viewing perspectives
#
```

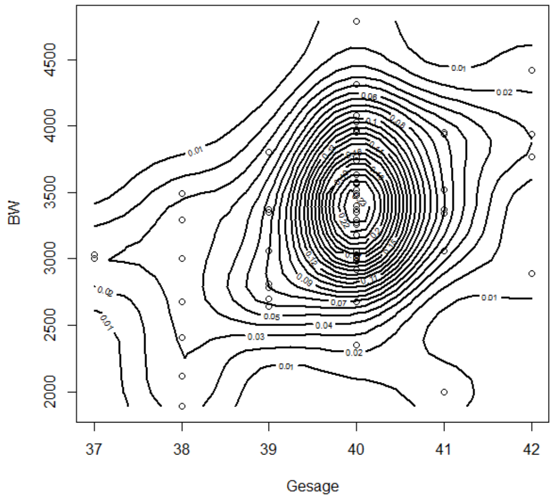


Figure 1.9. Contour plot of estimated bivariate density of Birthweight and Gestational age.

The scatterplot above was produced using the following **R code:**

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

Using a bubbleplot makes it possible to display Factor68 on our scatterplot by adding circles with radii proportional to these values and centered on the appropriate point in the scatterplot. Here in Figure 1.10, it is hardly feasible to note any observation exhibiting an extraordinary behavior.

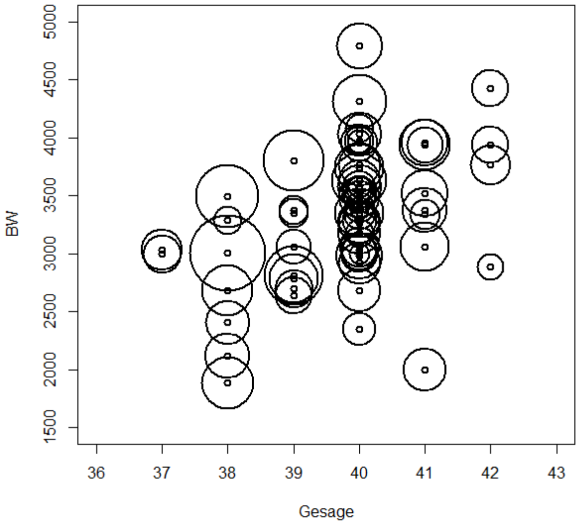


Figure 1.10. Bubbleplot of Birthweight and Gestational age with Factor68 represented by radii of circles.

The scatterplot above was produced using the following **R code:**

```
plot(Gesage,BW,pch=1,lwd=2,ylim=c(1500,5000),xlim=c(36,43))
symbols(Gesage,BW,circles=Factor68,inches=0.4,add=TRUE,lwd=2)
```

The conditioning plot or coplot is a potentially powerful visualization tool for studying the bivariate relationship of a pair of variables conditional on the value of one more other variables. Such a plot can often highlight the presence of interactions between the variables.

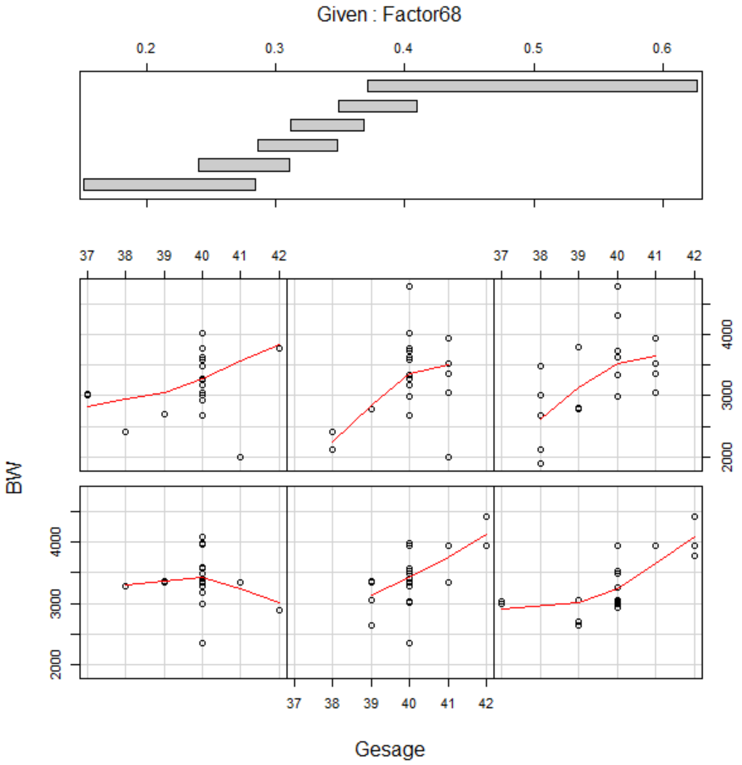


Figure 1.11. Coplot of Birthweight and Gestational age conditional on Factor68.

The scatterplot above was produced using the following **R code:**

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

The result is shown in Figure 1.11. The relationship between gestational age and birth weight is highly complex at all levels of Factor 68.

Based on the information extracted from the initial data we could conclude that birth weight is moderately dependent on gestational age of an infant, however further analysis could possibly provide more details about this relationship.

2. Perform a PCA and interpret the results.

To begin our principal component analysis, we shall remove the variable Group, since it will disturb the general image of relationships with other variables. In addition, we should ignore the HR variable, since we want to investigate the determinants of heart rate by regressing the rest of the variables against it.

The pairs function generates a scatterplot shown below in Figure 2.1 where one can note that all three variables are measured on different scales. Therefore, it seems necessary to extract the principal components from the correlation rather than the covariance matrix.

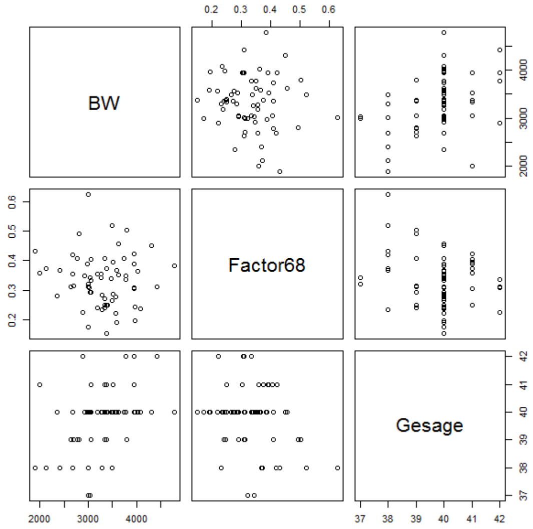


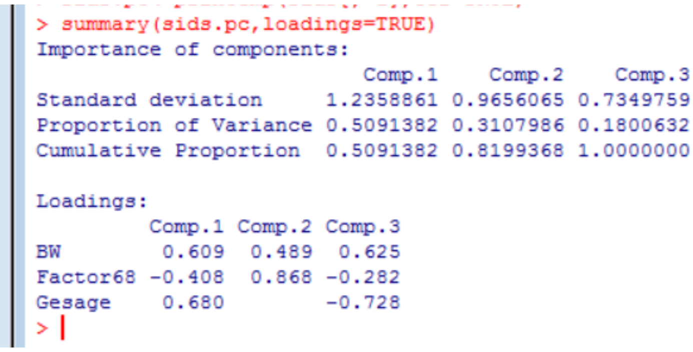
Figure 2.1. Scatterplot matrix of SIDS data dataset

R-code:

```
sids<- data.frame(read.csv("C:\\Users\\Yevgeniya Litvinova\\Documents\\SIW 004 Math and Stats\\final individual\\final_data.csv"))
#
sids <- sids[,-1]
attach(sids)
sids#
pairs(sids[,-1])  # remove Heart rate variable from the data to make regression over the rest 3 variables.
```

The resulting output of the PCA is shown in Table 2.1 where we can see that the first two components account for 82% of the variance of the original variables. Scores on these components help us to summarize the whole data in further analysis.
Loadings show us the coefficients for individual components, and variables which define them, i.e. those with a high value.

The first component we can label as “Development stage” of an infant because it is characterized by high values of Birth weight and Gestational Age, and the second component as “Factor68” which represent heart and lung function.



```
> summary(sids.pc,loadings=TRUE)
Importance of components:
                          Comp.1    Comp.2    Comp.3
Standard deviation     1.2358861 0.9656065 0.7349759
Proportion of Variance 0.5091382 0.3107986 0.1800632
Cumulative Proportion  0.5091382 0.8199368 1.0000000

Loadings:
         Comp.1 Comp.2 Comp.3
BW        0.609  0.489  0.625
Factor68 -0.408  0.868 -0.282
Gesage    0.680        -0.728
> |
```

Table 2.1. Results from the PCA of the SIDS data

R-code:
cor(sids[,-1])
sids.pc<-princomp(sids[,-1],cor=TRUE)
summary(sids.pc,loadings=TRUE)

Besides labeling the obtained components ,we want to use them as the basis for several graphical displays of the observations under study. Figure 2.2 shows that observation 51 is suspected to have the highest Factor68 value and observation 7 the highest birth weight and gestational age respectively.

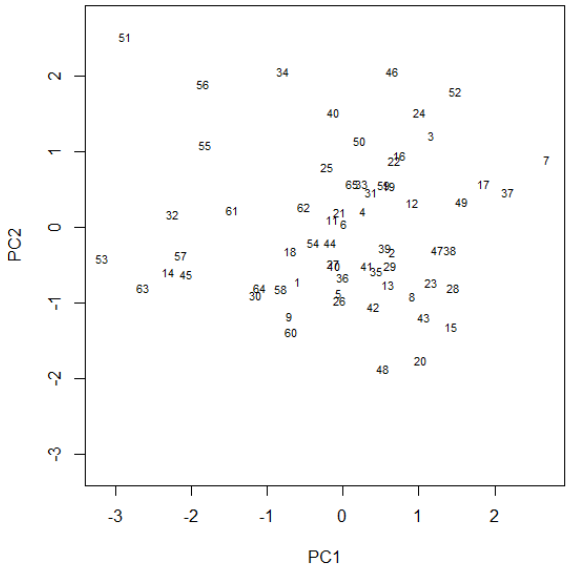


Figure 2.2. Scatterplot of the principal components of SIDS data observations

R-code:

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

Figure 2.3 shows the heart rate variable seems to be more related to the second principal component score, but not perhaps the first one. We will prove this formally by running the linear regression below.

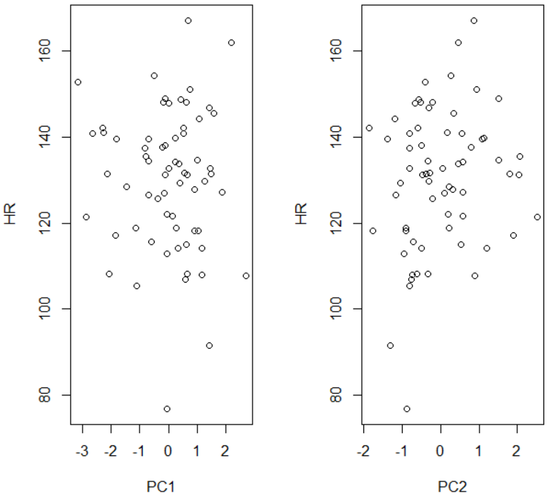
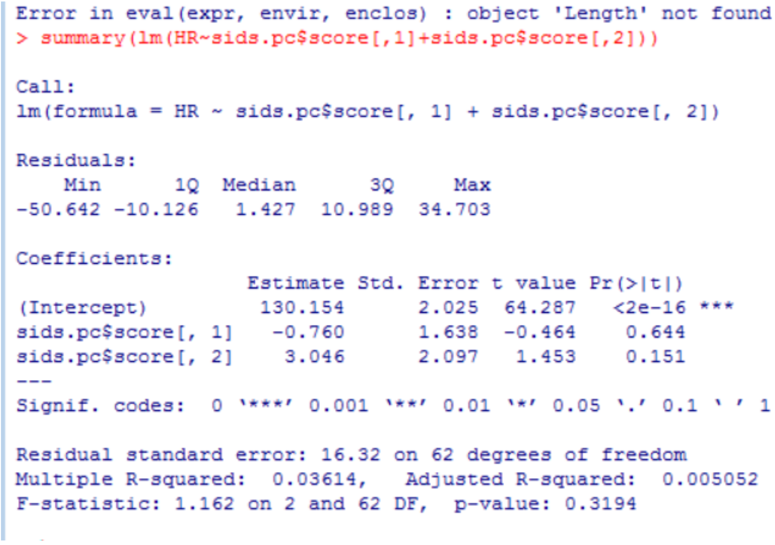


Figure 2.3. Plots of HR against first two components.

R-code:

```
sids.pc$scores[,1:2]
#
par(mfrow=c(1,2))
plot(sids.pc$scores[,1],HR,xlab="PC1")
plot(sids.pc$scores[,2],HR,xlab="PC2")
```

Undertaking a formal regression analysis of the data allows us to see how big each component's influence is on the heart rate variable. Table 2.2 demonstrates that it is predicted by the second, rather than the first component, which is logically true because the heart rate of an infant clearly depends on its heart and lung function (Factor68) abilities.



```
Error in eval(expr, envir, enclos) : object 'Length' not found
> summary(lm(HR~sids.pc$score[,1]+sids.pc$score[,2]))

Call:
lm(formula = HR ~ sids.pc$score[, 1] + sids.pc$score[, 2])

Residuals:
    Min      1Q  Median      3Q     Max 
-50.642 -10.126   1.427  10.989  34.703 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)         130.154      2.025  64.287   <2e-16 ***
sids.pc$score[, 1]   -0.760      1.638  -0.464    0.644    
sids.pc$score[, 2]    3.046      2.097   1.453    0.151    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.32 on 62 degrees of freedom
Multiple R-squared:  0.03614,	Adjusted R-squared:  0.005052 
F-statistic: 1.162 on 2 and 62 DF,  p-value: 0.3194
```

Table 2.2. Output of linear regression analysis of Heart Rate against first two principal component scores.

R-code:
summary(lm(HR~sids.pc$score[,1]+sids.pc$score[,2]))

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".

We start this exercise with performing a cluster analysis using k-means clustering technique. The dataset will be partitioned into a specified number of groups based on the value of the within-group sum of squares. Here, the initial number of clusters is specified as 6 as can be seen in Figure 3.1. The “sharp” change in the graph indicates that the best solution for our dataset is just two clusters.

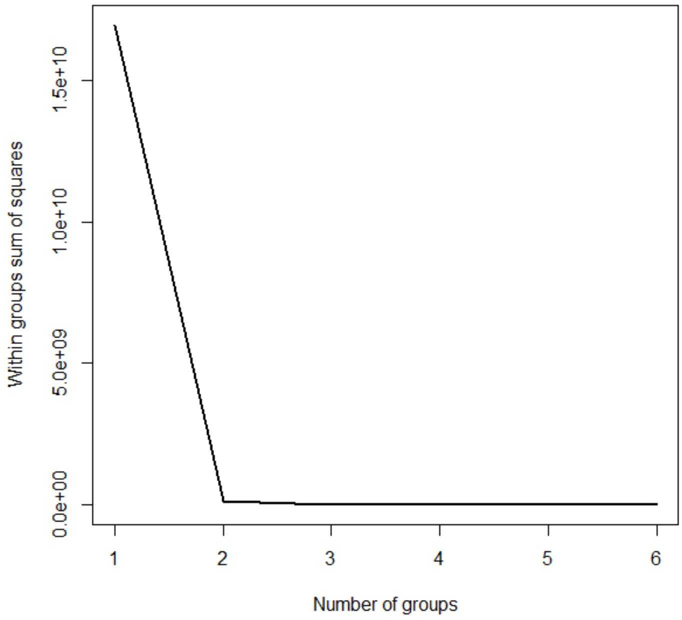


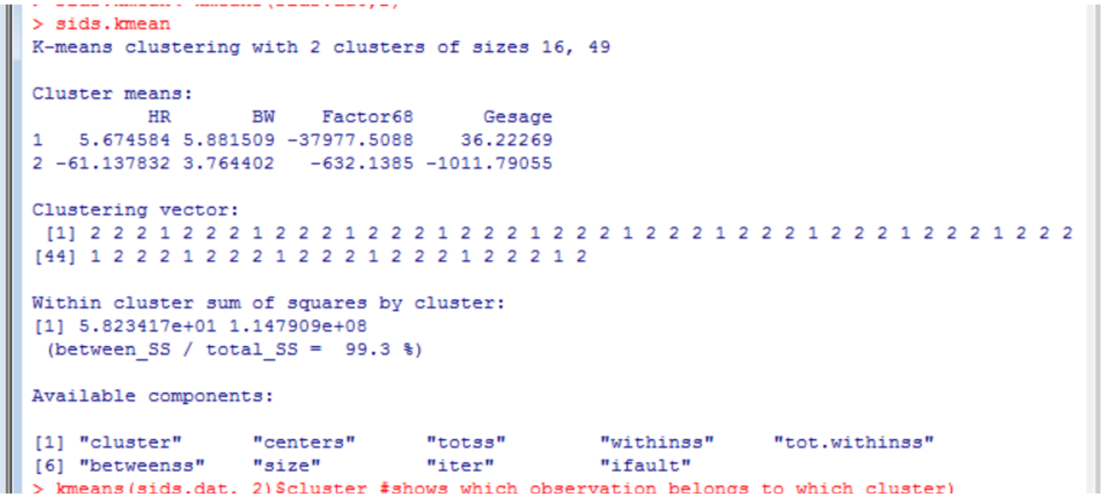
Figure 3.1. Plot of within-cluster sum of squares against number of clusters.

R-code:

```
sids<- data.frame(read.csv("C:\\Users\\Yevgeniya Litvinova\\Documents\\SIW 004 Math and Stats\\final individual\\final_data.csv"))
#
group = sids[,1]
sids <- sids[,-1]
attach(sids)
#
matrix.sids <- matrix(c(sids[,1],sids[,2],sids[,3],sids[,4]), ncol = 4)



#typification
sd.vector = apply (sids,2,sd)
mean.vector = apply(sids, 2, mean)
```

```
top = (apply(sids,2, function(x) x) -apply(sids, 2, mean))
sd = apply (sids,2,sd)
sids.dat<-sweep(top,2,sd,FUN="/")

n<-length(sids.dat[,1])
wss1<-(n-1)*sum(apply(sids.dat,2,var))
wss<-numeric(0)
for(i in 2:6) {
        W<-sum(kmeans(sids.dat,i)$withinss)
        wss<-c(wss,W)
}
#
wss<-c(wss1,wss)
plot(1:6,wss,type="l",xlab="Number of groups",ylab="Within groups sum of squares",lwd=2)
#
```

Next, the k-mean clustering is ran and the details are displayed in Figure 3.2, where each observation is assigned a specific cluster based on its properties.



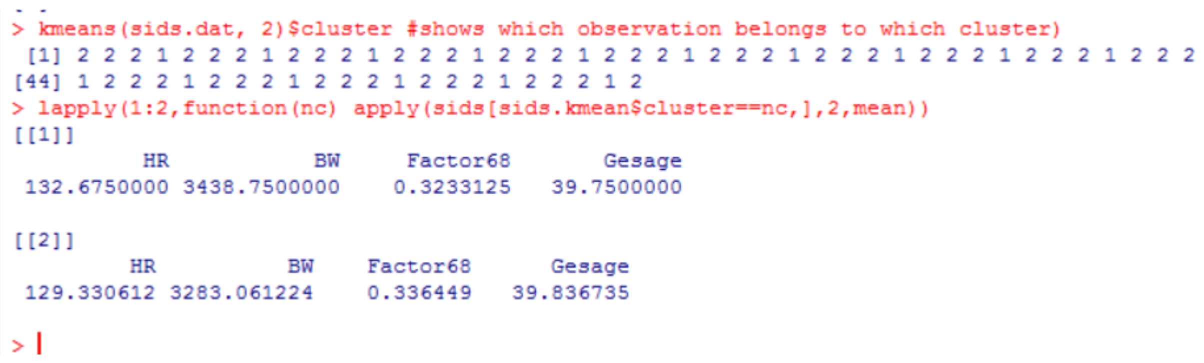
```
> sids.kmean
K-means clustering with 2 clusters of sizes 16, 49

Cluster means:
          HR       BW    Factor68      Gesage
1   5.674584 5.881509 -37977.5088    36.22269
2 -61.137832 3.764402   -632.1385 -1011.79055

Clustering vector:
 [1] 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2
[44] 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2

Within cluster sum of squares by cluster:
[1] 5.823417e+01 1.147909e+08
 (between_SS / total_SS =  99.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
> kmeans(sids.dat, 2)$cluster #shows which observation belongs to which cluster)
```

Figure 3.2. Details of Two-Group Solution for SIDS Dataset.

R-code:

```
sids.kmean<-kmeans(sids.dat,2)
sids.kmean
```

Calculating cluster means (Figure 3.3) for the raw data could tell us about what dinstibguishes each cluster one from another, i.e. in our case Cluster One is characterized by a high heart rate and birth weight values, whereas Cluster Two has higher Factor68 and Gesage values.



```
> kmeans(sids.dat, 2)$cluster #shows which observation belongs to which cluster)
 [1] 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2
[44] 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2
> lapply(1:2,function(nc) apply(sids[sids.kmean$cluster==nc,],2,mean))
[[1]]
         HR          BW    Factor68      Gesage 
 132.6750000 3438.7500000   0.3233125  39.7500000 

[[2]]
        HR         BW   Factor68     Gesage 
 129.330612 3283.061224   0.336449  39.836735 

> 
```

Figure 3.3. Cluster means for SIDS Data.

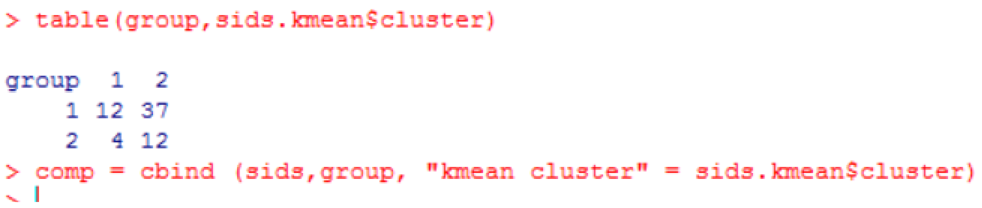
R-code:

kmeans(sids.dat, 2)$cluster

lapply(1:2,function(nc) apply(sids[sids.kmean$cluster==nc,],2,mean))

#

Now we would like to check if there is any association between the Group variable (manually assigned control/victim categories) and distinct compositional groups found by the cluster analysis. The resulting cross-classification is shown in Figure 3.4. It can be noticed that Cluster 1 contains 12 observations from Group 1 and 4 observations from Group 2, whereas Cluster 2 - 37 from Group 1 and 12 from Group 2 respectively. Both clusters seem to be homogeneous.



```
> table(group,sids.kmean$cluster)
     
group  1  2
    1 12 37
    2  4 12
> comp = cbind (sids,group, "kmean cluster" = sids.kmean$cluster)
> 
```

Figure 3.4. Cross-tabulation of Cluster Label and Group.

table(group,sids.kmean$cluster)

comp = cbind (sids,group, "kmean cluster" = sids.kmean$cluster)

#

Applying another clustering technique may lead to obtaining different results. Thus, we implement the agglomerative hierarchical clustering procedure. In Figure 3.5. two methods of measuring intercluster dissimilarity are displayed and both dendrograms seem to be entirely different from each other.

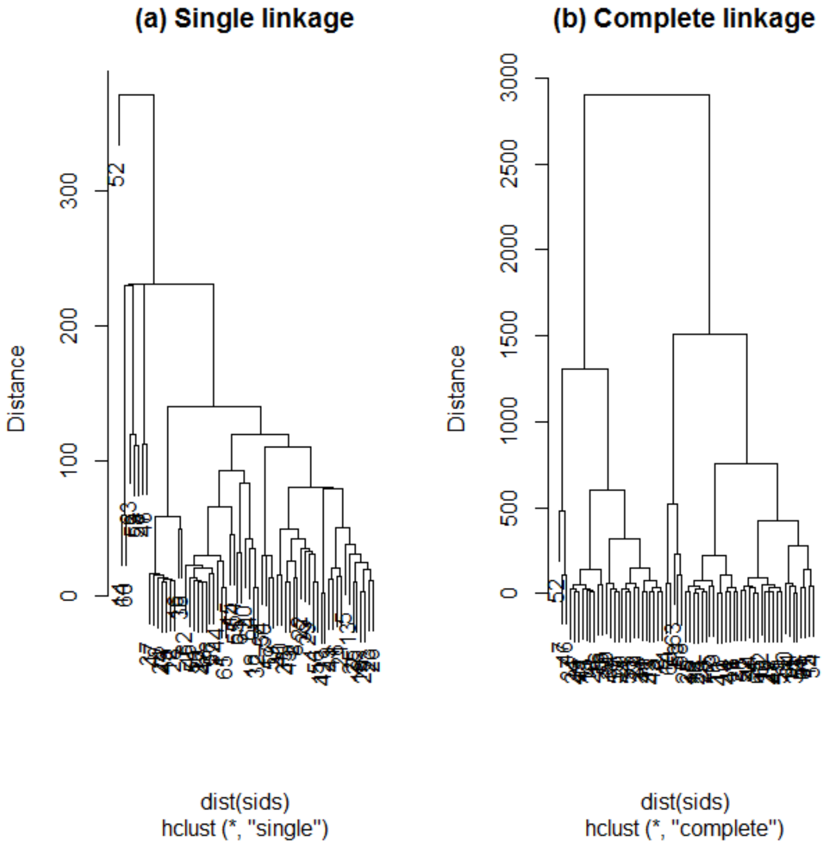


Figure 3.5. Single linkage and complete linkage dendrograms for the sudden infant death syndrome data.

R-code:

```
par(mfrow=c(1,2))
plclust(hclust(dist(sids),method="single"),labels=row.names(sids),ylab="Distance")
title("(a) Single linkage")
plclust(hclust(dist(sids),method="complete"),labels=row.names(sids),ylab="Distance")
title("(b) Complete linkage")
```

Firstly, we will concentrate on complete linkage and examine the clustering found by “cutting” the complete linkage dendrogram at height 1600 (in order to obtain only two clusters i.e. to match the number of existing Groups).

R-code:

```
two<-cutree(hclust(dist(sids),method="complete"),h=1600)
```

```
#
```

The resulting cluster labels for observations can be found from:

R-code:

```
sids.clus<-lapply(1:2,function(nc) row.names(sids)[two==nc])sids.clus
clus =
c(1,2,2,2,1,1,2,2,1,1,1,2,1,1,1,2,2,1,1,1,1,2,1,2,1,1,1,2,2,1,2,1,2,2,1,1,2,2,2,2,1,1,2,1,1,2,2,1,
2,2,1,2,1,1,1,2,1,1,2,1,1,1,1,1,1)
```

The means for the observations in each cluster can be found as follows:

```
sids.mean<-lapply(1:2,function(nc) apply(sids[two==nc,],2,mean))
sids.mean
```

The clusters can be shown on a scatterplot matrix of the data, however the diagram obtained suggests that the evidence for two distinct clusters in the data is not convincing.

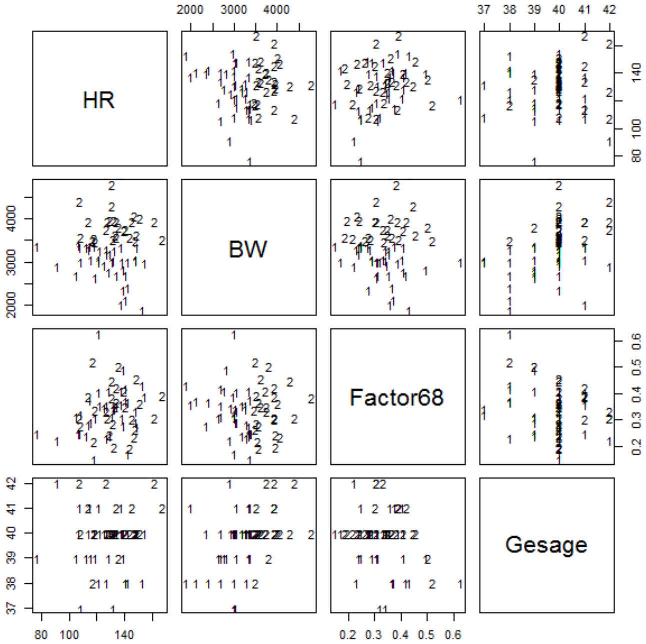
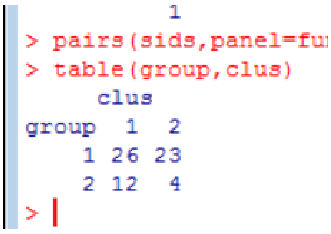


Figure 3.6. Scatterplot of sudden infant death syndrome data showing two cluster solution from complete linkage.

R-code:

```
dev.off()
#
pairs(sids,panel=function(x,y) text(x,y,two))
```

Now we would like to check if there is any association between the Group variable (manually assigned control/victim categories) and distinct compositional groups found by the complete linkage cluster analysis. The resulting cross-classification is shown in Figure 3.7. It can be noticed that Cluster 1 contains 26 observations from Group 1 and 12 observations from Group 2, whereas Cluster 2 - 23 from Group 1 and 4 from Group 2 respectively. Both clusters seem to be homogeneous.



```
             1
> pairs(sids,panel=fu
> table(group,clus)
     clus
group  1  2
    1 26 23
    2 12  4
> 
```

Figure 3.7. Cross-tabulation of Cluster Label and Group.

R-code:

```
table(group,clus)
```

Secondly, we will also try the single linkage method and examine the clustering found by “cutting” the dendrogram at height 200 because if we wanted to match the number of Groups we would have to cut the diagram at the height of 300 (see Figure 3.5) and would result in only one observation (#52) belonging to Cluster 1 and the rest of them to Cluster 2 what seems like an incorrect solution.

R-code:

```
five<-cutree(hclust(dist(sids),method="single"),h=200) #
sids.clus<-lapply(1:5,function(nc) row.names(sids)[five==nc])
sids.mean<-lapply(1:5,function(nc) apply(sids[five==nc,],2,mean))
sids.mean
sids.clus
clus =
c(1,1,1,1,1,1,2,1,1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,
1,1,1,4,5,1,1,1,1,5,1,3,1,1,5,1,1)
#
```

The clusters can be shown on a scatterplot matrix of the data, however the diagram obtained suggests that the evidence for five distinct clusters in the data is not convincing.

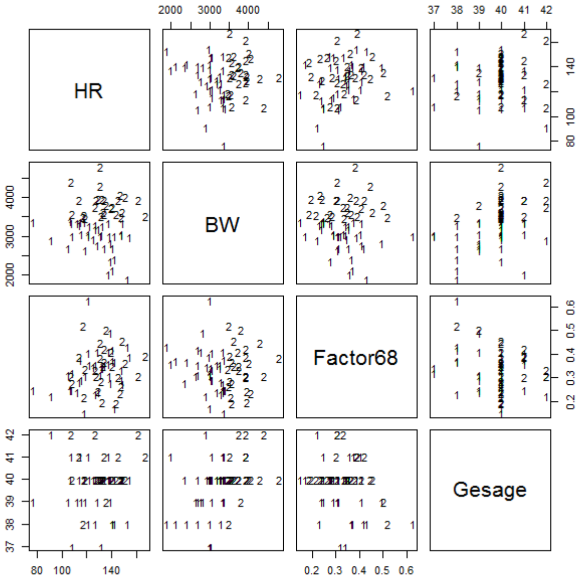
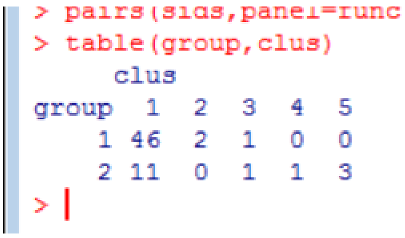


Figure 3.8. Scatterplot of sudden infant death syndrome data showing five cluster solution from single linkage.

R-code:
pairs(sids,panel=function(x,y) text(x,y,two))

To check whether there is any association between the Group variable (manually assigned control/victim categories) and distinct compositional groups found by the single linkage cluster analysis we cross-classify them as shown in Figure 3.9. It can be noticed that Cluster 1 contains 46 observations from Group 1 and 11 observations from Group 2, Cluster 2 - 2 from Group 1, Cluster 3 has 1 observation from each of Groups, Cluster 4 - 1 observation from the Group 2, and finally Cluster 5 has 3 observations from Group 2. Two out of five clusters seem to be homogeneous.



```
> pairs(sids,panel=func
> table(group,clus)
     clus
group  1  2  3  4  5
    1 46  2  1  0  0
    2 11  0  1  1  3
>
```

Figure 3.9. Cross-tabulation of Cluster Label and Group.

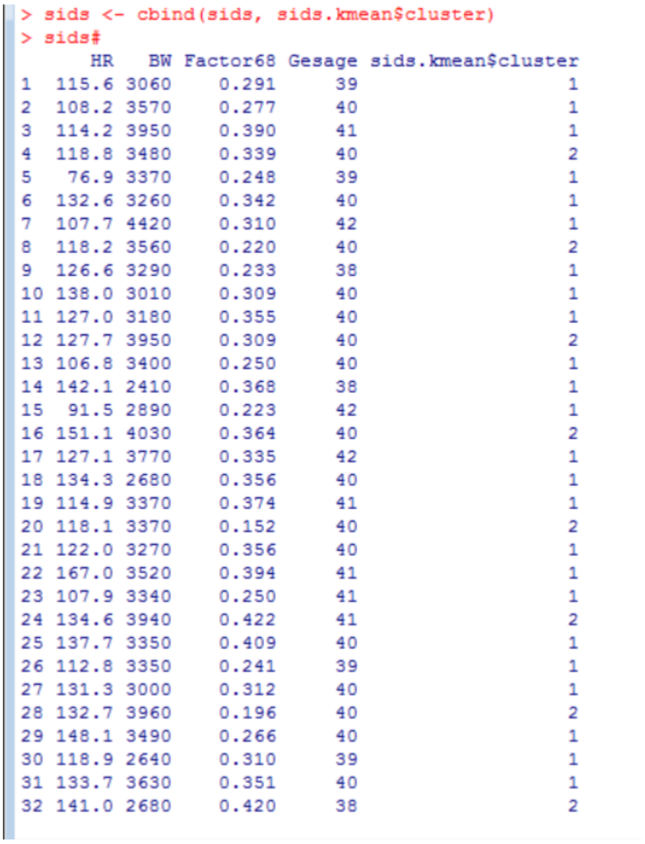
R-code:
table(group,clus)

Overall, it can be concluded that k-means method appears to be a more accurate clustering procedure when compared to hierarchical clustering and the challenges related to its application, i.e. identifying the height of dendrogram.

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).

Two new observations will be classified considering the clusters previously found using K-means clustering method.



```
> sids <- cbind(sids, sids.kmean$cluster)
> sids#
      HR   BW Factor68 Gesage sids.kmean$cluster
1  115.6 3060    0.291     39                  1
2  108.2 3570    0.277     40                  1
3  114.2 3950    0.390     41                  1
4  118.8 3480    0.339     40                  2
5   76.9 3370    0.248     39                  1
6  132.6 3260    0.342     40                  1
7  107.7 4420    0.310     42                  1
8  118.2 3560    0.220     40                  2
9  126.6 3290    0.233     38                  1
10 138.0 3010    0.309     40                  1
11 127.0 3180    0.355     40                  1
12 127.7 3950    0.309     40                  2
13 106.8 3400    0.250     40                  1
14 142.1 2410    0.368     38                  1
15  91.5 2890    0.223     42                  1
16 151.1 4030    0.364     40                  2
17 127.1 3770    0.335     42                  1
18 134.3 2680    0.356     40                  1
19 114.9 3370    0.374     41                  1
20 118.1 3370    0.152     40                  2
21 122.0 3270    0.356     40                  1
22 167.0 3520    0.394     41                  1
23 107.9 3340    0.250     41                  1
24 134.6 3940    0.422     41                  2
25 137.7 3350    0.409     40                  1
26 112.8 3350    0.241     39                  1
27 131.3 3000    0.312     40                  1
28 132.7 3960    0.196     40                  2
29 148.1 3490    0.266     40                  1
30 118.9 2640    0.310     39                  1
31 133.7 3630    0.351     40                  1
32 141.0 2680    0.420     38                  2
```

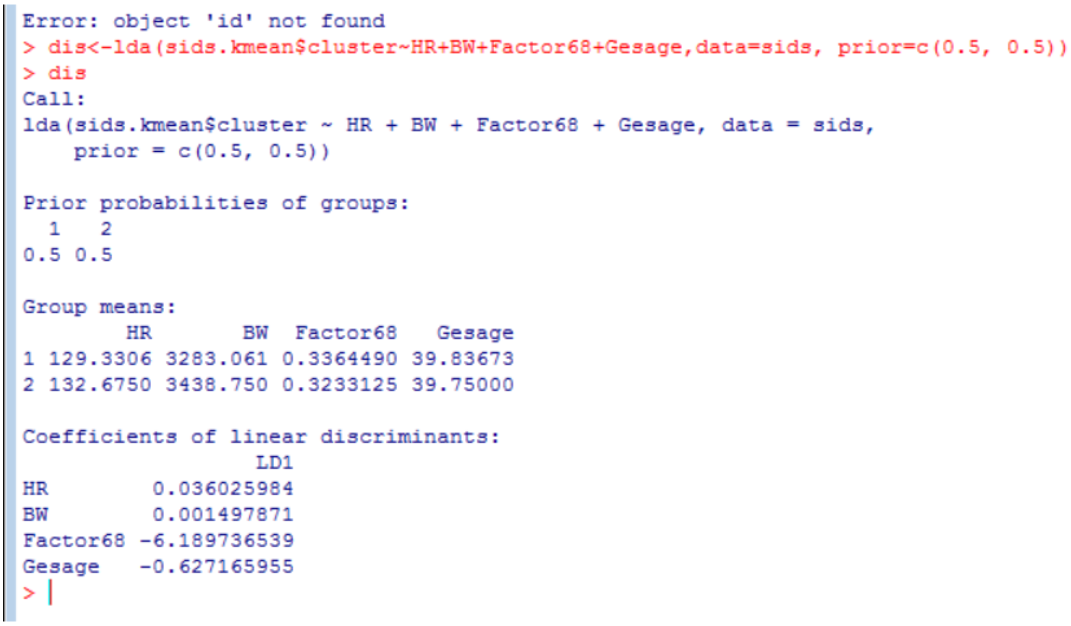
Figure 3.10. Sudden Instant Death Syndrome dataset with added cluster column.

R-code:

```
sids<- data.frame(read.csv("C:\\Users\\Yevgeniya Litvinova\\Documents\\SIW 004 Math and Stats\\final individual\\final_data.csv"))
#
group = sids[,1]
sids <- sids[,-1]
attach(sids)
#
```

```
matrix.sids <- matrix(c(sids[,1],sids[,2],sids[,3],sids[,4]), ncol = 4)

#typification
sd.vector = apply (sids,2,sd)
mean.vector = apply(sids, 2, mean)

top = (apply(sids,2, function(x) x) -apply(sids, 2, mean))
sd = apply (sids,2,sd)
sids.dat<-sweep(top,2,sd,FUN="/")
sids.kmean<-kmeans(sids.dat,2)
sids.kmean$cluster
sids <- cbind(sids, sids.kmean$cluster)
sids#
```

To classify new observations, Fisher’s linear discriminant function needs to be applied assuming that the prior probabilities of our observations being in each group are the same. This gives results shown in the Figure 3.11.



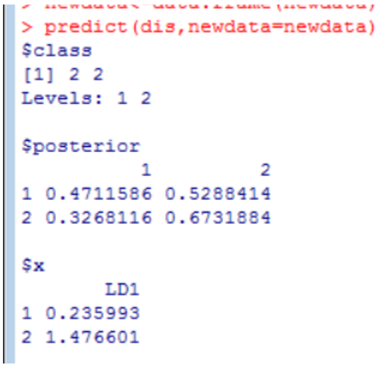
```
Error: object 'id' not found
> dis<-lda(sids.kmean$cluster~HR+BW+Factor68+Gesage,data=sids, prior=c(0.5, 0.5))
> dis
Call:
lda(sids.kmean$cluster ~ HR + BW + Factor68 + Gesage, data = sids,
    prior = c(0.5, 0.5))

Prior probabilities of groups:
  1   2
0.5 0.5

Group means:
        HR       BW  Factor68   Gesage
1 129.3306 3283.061 0.3364490 39.83673
2 132.6750 3438.750 0.3233125 39.75000

Coefficients of linear discriminants:
                  LD1
HR        0.036025984
BW        0.001497871
Factor68 -6.189736539
Gesage   -0.627165955
> 
```

Figure 3.11. Results from discriminant function on SIDS Data.

R-code:

```
library(MASS)
dis<-lda(sids.kmean$cluster~HR+BW+Factor68+Gesage,data=sids, prior=c(0.5, 0.5))
#
```

Next the *predict* function is applied to give the classification probabilities of belonging to one of the given groups. Figure 3.12 demonstrates that both Obs1 and Obs2 belong to the second cluster with moderate probability score equal to 0.53 and 0.67 respectively.



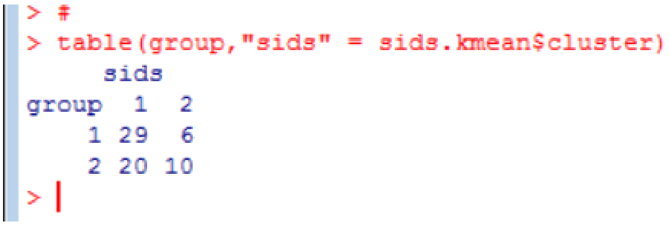
```
> newdata<-data.frame(newdata)
> predict(dis,newdata=newdata)
$class
[1] 2 2
Levels: 1 2

$posterior
          1         2
1 0.4711586 0.5288414
2 0.3268116 0.6731884

$x
       LD1
1 0.235993
2 1.476601
```

Figure 3.12. Results from predict on Obs1 and Obs2.

R-code:

```
newdata<-rbind(c(110,3320,0.240,39),c(120,3310,0.298,37))
colnames(newdata)<-colnames(sids[,-5])
newdata<-data.frame(newdata)
predict(dis,newdata=newdata)
```

Performance of the discriminant function should be evaluated by calculating the misclassification rate (the approach known as the "plug-in" estimate). This can be achieved by applying the *predict* function as follows:



```
> #
> table(group,"sids" = sids.kmean$cluster)
     sids
group  1  2
    1 29  6
    2 20 10
> 
```

Figure 3.13. Results from the misclassification rate *predict* function.

R-code:

```
group<-predict(dis,method="plug-in")$class
#
table(group,"sids" = sids.kmean$cluster)
```

Counts of correct and incorrect classification can be seen on Figure 3.13. The misclassification rate in this case is very high and equals to 38%. It is possible, that a better result can be obtained using a different normalization technique for k-means method.