Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering
IDP Institut für Datenanalyse
und Prozessdesign

HS 2016

Statistisches Data Mining (StDM)

Woche 4

Aufgabe 1

Lab

Read and do the excersises of chapter 10.5.1 in ILSR

Aufgabe 2

K-Means und PCA

This is taken from ILSR.

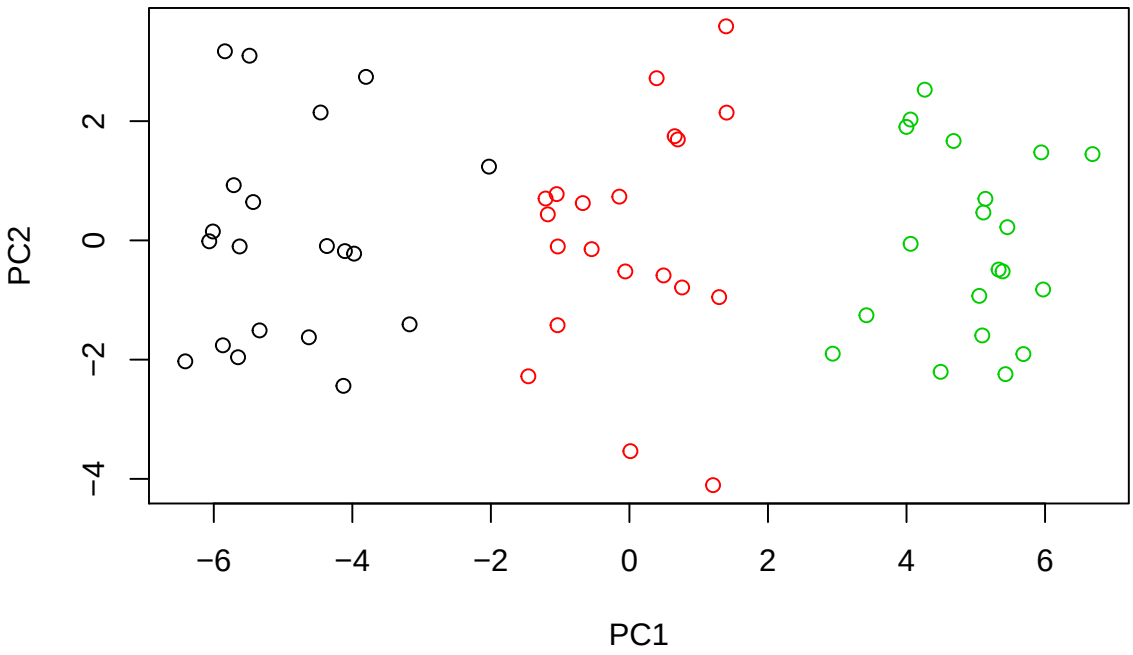
In this problem, you will generate simulated data, and then perform PCA and K-means clustering on the data.

a) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables. Hint: There are a number of functions in R that you can use to generate data. One example is the rnorm() function; runif() is another option. Be sure to add a mean shift to the observations in each class so that there are three distinct classes.

```
X <- rbind(matrix(rnorm(20*50, mean = 0), nrow = 20),
           matrix(rnorm(20*50, mean=0.7), nrow = 20),
           matrix(rnorm(20*50, mean=1.4), nrow = 20))
```

b) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then continue on to part (c). If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes. Do not continue to part (c) until the three classes show at least some separation in the first two principal component score vectors.

```
X.pca = prcomp(X)$x
plot(X.pca[,1:2], col=c(rep(1,20), rep(2,20), rep(3,20)))
```



c) Perform K-means clustering of the observations with K = 3. How well do the clusters that you obtained in K-means cluster- ing compare to the true class labels? Hint: You can use the table() function in R to compare the true class labels to the class labels obtained by clustering. Be careful how you interpret the results: K-means clustering will arbitrarily number the clusters, so you cannot simply check whether the true class labels and clustering labels are the same.

```
res = kmeans(X, centers = 3)
true_class = c(rep(1,20), rep(2,20), rep(3,20))
table(res$cluster, true_class)
```

```
##    true_class
##      1  2  3
##   1  0  0 20
##   2 19  0  0
##   3  1 20  0
```

All are perfectly clustered

d) Perform K-means clustering with K = 2. Describe your results.

```
res = kmeans(X, centers = 2)
true = c(rep(1,20), rep(2,20), rep(3,20))
table(res$cluster, true_class)
```

```
##    true_class
##      1  2  3
##   1 20 10  0
##   2  0 10 20
```

The middle class is forced to a wrong class. The extreme classes are classified correctly

e) Now perform K-means clustering with K = 4, and describe your results.

```
res = kmeans(X, centers = 4)
true = c(rep(1,20), rep(2,20), rep(3,20))
table(res$cluster, true_class)
```

```
##    true_class
##       1  2  3
##   1   0  0  9
##   2   1 20  0
##   3 19  0  0
##   4  0  0 11
```

One of the classes is splitt into 2 classes

f) Now perform K-means clustering with K = 3 on the first two principal component score vectors, rather than on the raw data. That is, perform K-means clustering on the 60 × 2 matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.

```
res = kmeans(X.pca[,1:2], centers = 3)
true = c(rep(1,20), rep(2,20), rep(3,20))
table(res$cluster, true_class)
```

```
##    true_class
##       1  2  3
##   1   0  0 20
##   2   1 20  0
##   3 19  0  0
```

Same result as above, the PCA carries enough information

g) Using the scale() function, perform K-means clustering with K = 3 on the data after scaling each variable to have standard deviation one. How do these results compare to those obtained in (b)? Explain.

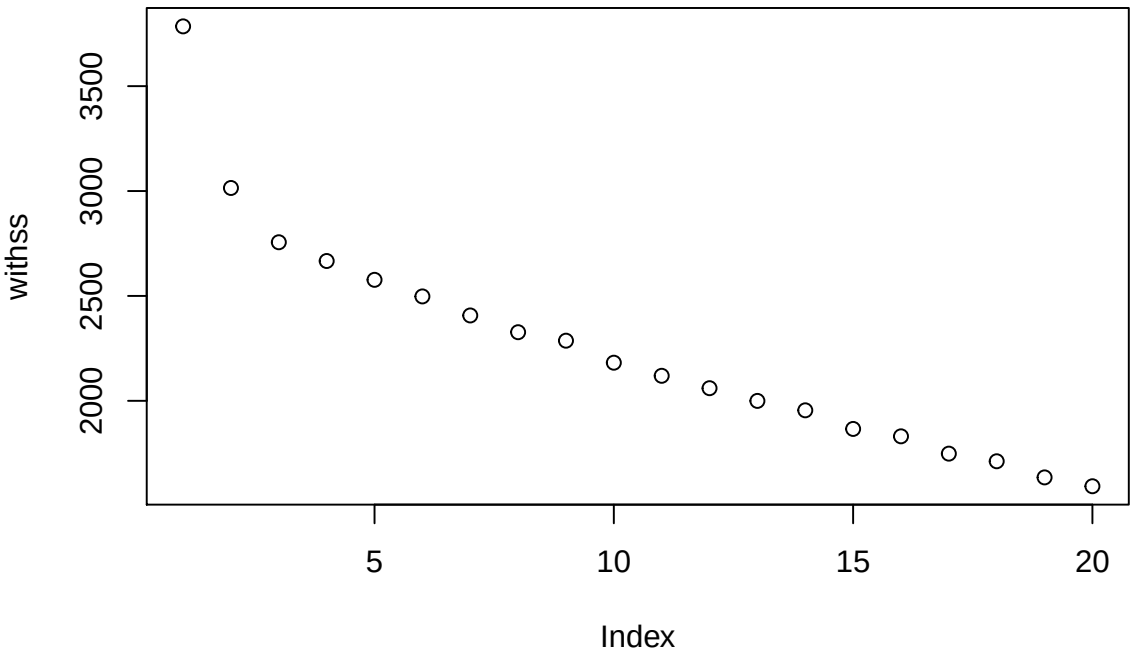
```
res = kmeans(scale(X), centers = 3)
true = c(rep(1,20), rep(2,20), rep(3,20))
table(res$cluster, true_class)
```

```
##    true_class
##       1  2  3
##   1 19  0  0
##   2  0  0 20
##   3  1 20  0
```

Same result as above, scaling does not change the results

h) Look at the total within sum of squares of the clusters for varying number of k's. Which is the best number of k?

```
withss = rep(NA,20)
for (k in 1:length(withss)) {
  withss[k] = sum(kmeans(X,k)$withinss)
}
plot(withss)
```


withss
2000
2500
3000
3500
5
10
15
20
Index