Assignment 1: Question 3

Hans de Ferrante

February 16, 2018

```r
library(rstudioapi)
library(readr)
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
```

```r
library(missMDA)
knitr::opts_chunk$set(echo = TRUE)
```

```r
current_path <- getActiveDocumentContext()$path
setwd(dirname(current_path ))
```

Data was retrieved from Kaggle. We are interested in house prices.

a-b) Load the data and obtain the needed subset of the data.

```r
datadict <- read_csv("DataDictionary.csv")
```

```
## Parsed with column specification:
## cols(
##   Variable = col_character(),
##   Definition = col_character()
## )
```

```r
c1 <- read_csv("County_time_series.csv", col_types = cols(Date = col_date(), RegionName = col_character
  mutate(Date = as.Date(Date), RegionName = as.factor(RegionName)) %>%
  filter(Date == "2017-11-30")
c1 <- c1[,c(2,6:20)]
indx <- c(2:16)
c1 <- lapply(c1[indx],as.numeric) %>% as.data.frame()
```

c) Preprocess the data by fitting missing observations using PCA.

```r
c2 <- imputePCA(c1, ncp = 6, center=TRUE, scale = TRUE, method = c("Regularized","EM"), row.w = NULL, c
                 nb.init = 1, maxiter = 1000)
```

```
## Warning in impute(X, ncp = ncp, scale = scale, method = method, threshold =
## threshold, : Stopped after 1000 iterations
```

d) Do PCA on the completed observations

We do PCA on the data where observed values are kept and missing values are imputed by the procedure in c). We will do this analysis based on the scaled data. This seems wise as the different observations are measured in different units (e.g. $, $ per square feet). If we would not scale, the PCs would mostly explain the median house prices as these are much larger than median house prices per square feet.

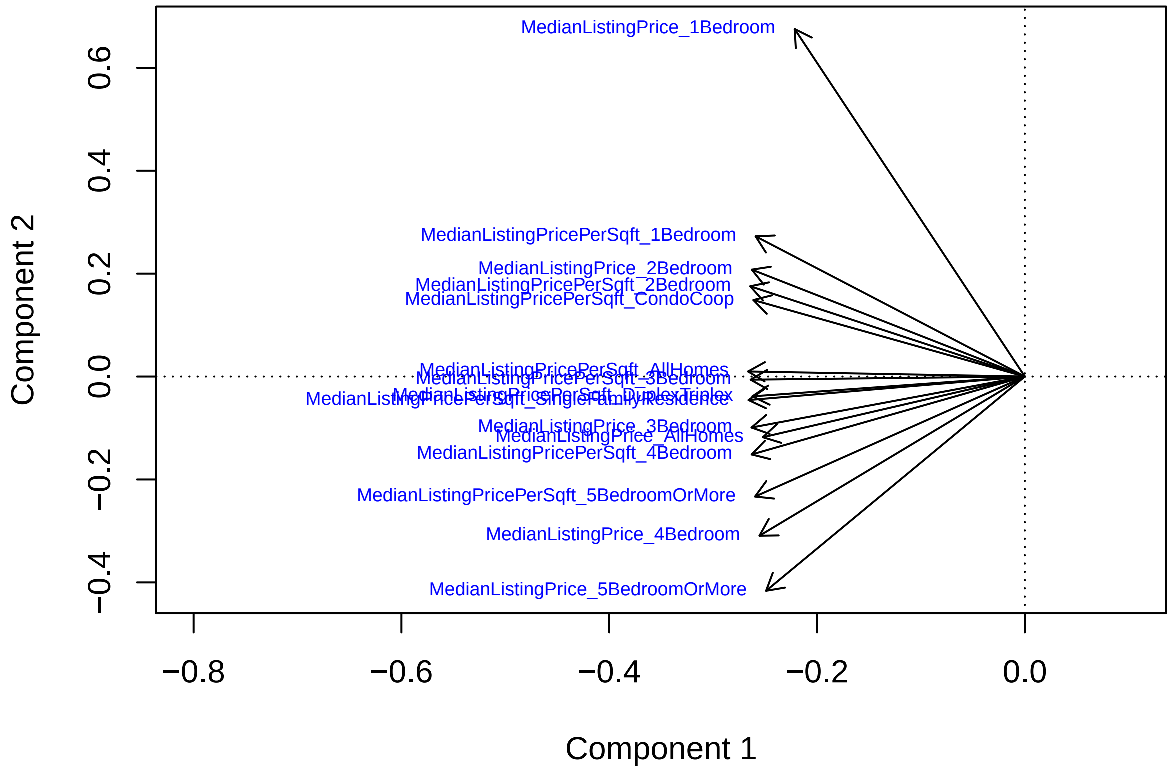
```r
PCA.out <- scale(c2$completeObs, center = TRUE, scale = TRUE) %>%  princomp()
print(PCA.out$loadings[,c(1:2)], digits=2)
```

```
##                                                 Comp.1  Comp.2
## MedianListingPricePerSqft_1Bedroom               -0.26  0.2725
## MedianListingPricePerSqft_2Bedroom               -0.26  0.1754
## MedianListingPricePerSqft_3Bedroom               -0.26 -0.0061
## MedianListingPricePerSqft_4Bedroom               -0.26 -0.1514
## MedianListingPricePerSqft_5BedroomOrMore         -0.26 -0.2332
## MedianListingPricePerSqft_AllHomes               -0.27  0.0099
## MedianListingPricePerSqft_CondoCoop              -0.26  0.1486
## MedianListingPricePerSqft_DuplexTriplex          -0.26 -0.0385
## MedianListingPricePerSqft_SingleFamilyResidence  -0.27 -0.0457
## MedianListingPrice_1Bedroom                      -0.22  0.6753
## MedianListingPrice_2Bedroom                      -0.26  0.2078
## MedianListingPrice_3Bedroom                      -0.26 -0.0991
## MedianListingPrice_4Bedroom                      -0.26 -0.3093
## MedianListingPrice_5BedroomOrMore                -0.25 -0.4163
## MedianListingPrice_AllHomes                      -0.25 -0.1179
```

On the first principal component all variables have loadings with magnitudes above .01. On the first component, they all have the same signs and similar magnitudes, meaning that within regions the median listing prices of houses are significantly positively correlated. On the second principal component, the median listing price of all homes per square feet and of houses with 3 bedrooms per square feet have magnitudes below .01. Thus, these variables are not important for the second principal component. The most important variables for these components relate to big houses and small houses (1 vs 5 bedrooms) and they correlate negatively. Hence, this second component seems to have to do with the relation between house size and price.

e) Make a loading plot and interpret.

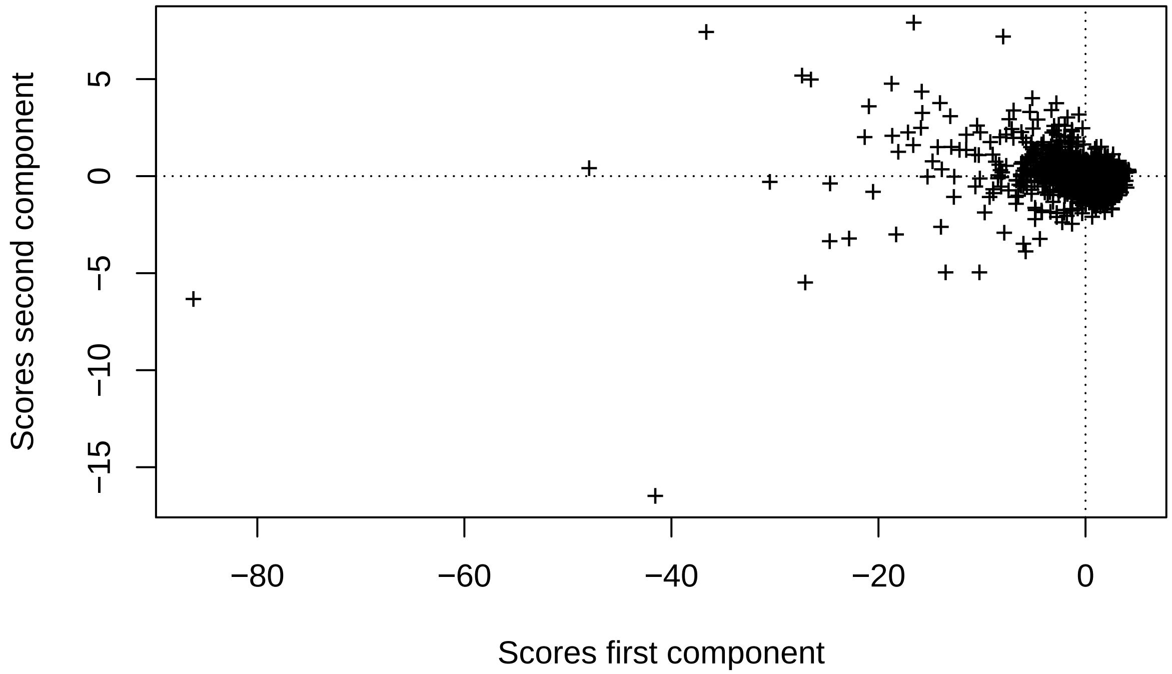
```r
plot(PCA.out$loadings[,1], PCA.out$loadings[,2], xlim=c(-.8,.1), type="n",
     xlab="Component 1", ylab="Component 2")

abline(v=0, lty=3)
abline(h=0, lty=3)
for (i in 1:length(PCA.out$loadings[,1]))  {
    arrows(0,0,PCA.out$loadings[i,1],PCA.out$loadings[i,2], length=.1)
  }
text(PCA.out$loadings[,1], PCA.out$loadings[,2], row.names(PCA.out$loadings),
     cex=0.6, pos=2, col="blue")
```



The loading plot clearly shows that all variables on the first principal component are positively correlated. Hence, median listing prices for different types of homes tend to be correlated within areas (both in terms of $ per square feet as well as in terms of median prices per se). We see that the second principal component indicates that median listing prices (per square feet) for a low number of bedroom apartments are negatively correlated with median listing prices (per square feet) of houses with a high number of bedrooms. Hence, the second principal component seems to grasp variation relating to the sizes of houses.

f) Make a scores plot and interpret.

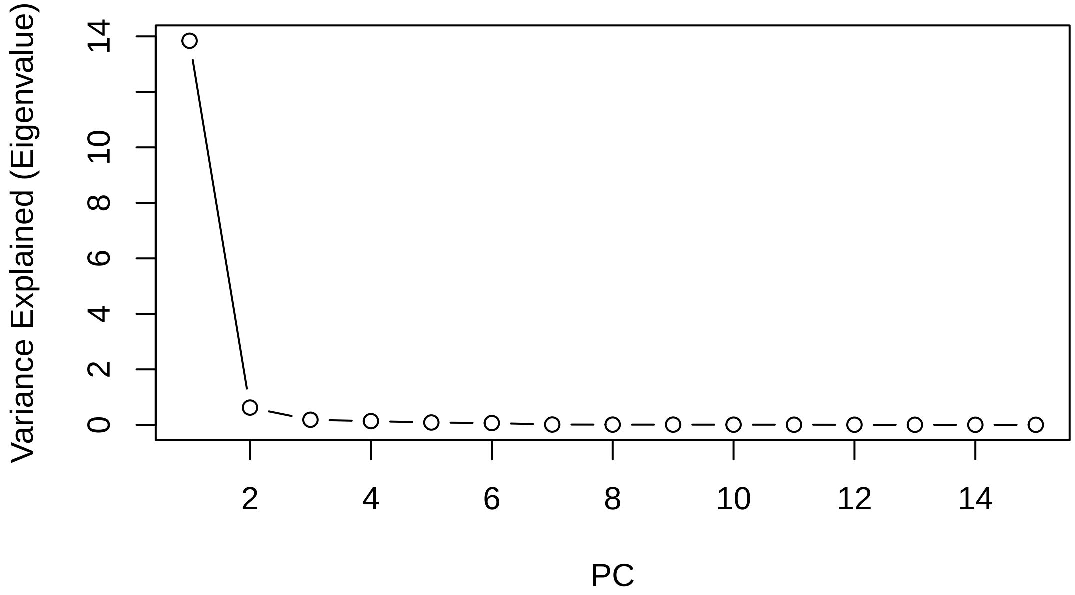
```
plot(PCA.out$scores[,1], PCA.out$scores[,2], pch="+",
     xlab="Scores first component", ylab="Scores second component")
abline(v=0, lty=3)
abline(h=0, lty=3)
```



From the scores plot, we see that there is much variation in the first principal component; scores range from just above 0 to below -80. From our interpretation of the loadings of the first principal component, this indicates that there is much variation in house prices per region. In comparison, there is less variation based on the sizes of houses measured in the number of bedrooms.

g) Make a scree plot of the eigenvalues ordered from the largest to smallest. Explain how many PCs you would choose.

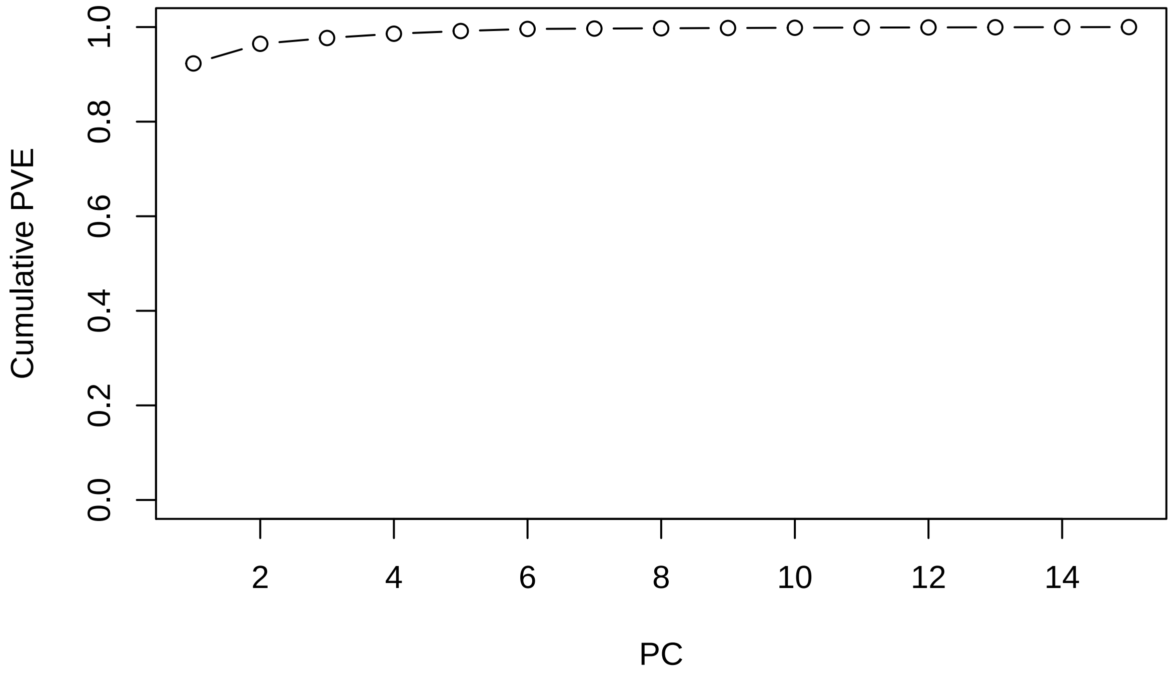
```r
pr.var=PCA.out$sdev^2
plot(pr.var,xlab="PC",ylab="Variance Explained (Eigenvalue)",type='b')
```



From the scree plot, it is not immediately clear how to choose the number of PCs. There is an elbow at the second/third component. The eigenvalues of the principal component seem to flatten of from the third component onwards. Therefore, it seems reasonable to choose the first two principal components to describe these data. This idea is further supported by a clear interpretation to the first two components (namely PC1: price differences across neighborhoods, PC2: price differents across different types of houses), and no clear interpretation to the third component (and onwards).

h) Make a plot of the cumulative percent of variance explained.

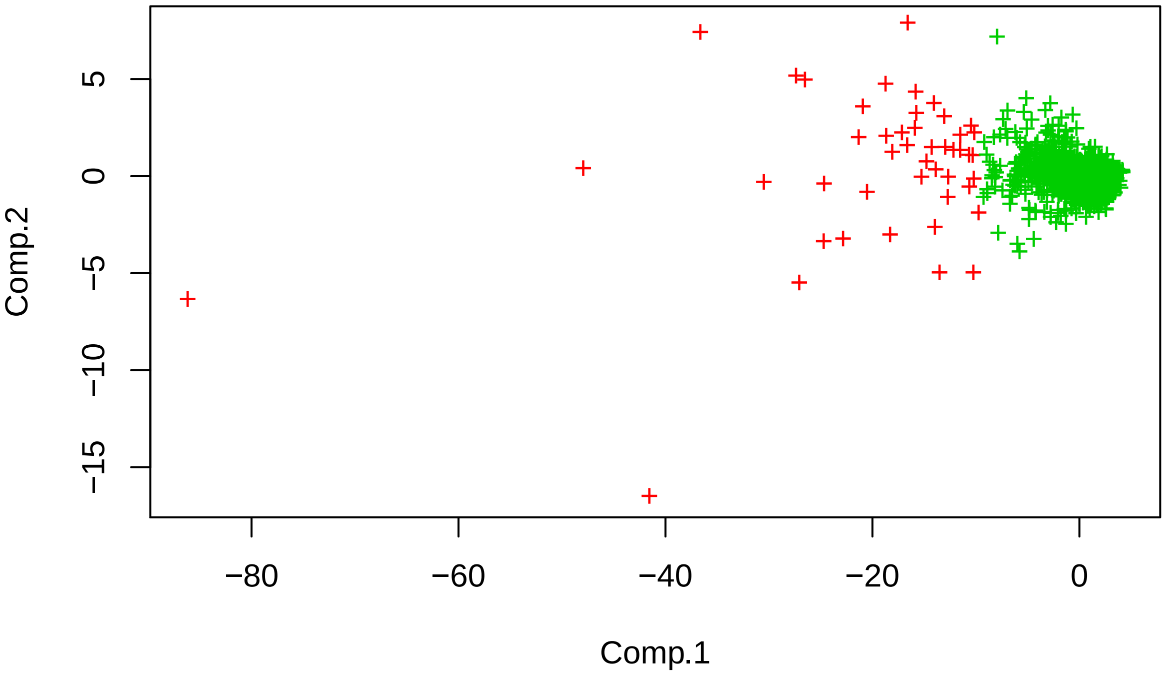
```
pve = pr.var/sum(pr.var)
plot(cumsum(pve),xlab="PC",ylab="Cumulative PVE",ylim=c(0,1),type='b')
```



Since the first two principal components already capture >96% of variation, and the additional percentages of variance per component flatten off too after the third component, we would arrive at the same conclusion as in (g); use the first two components to descibe the data.

i) Do K-means clustering based on the first two principal component scores. Choose k=2 with 20 starting positions.

```
set.seed(224920)
km.out <- kmeans(PCA.out$scores[,1:2], 2, nstart = 20)
plot(PCA.out$scores[,1:2], col = (km.out$cluster+1), pch = '+')
```



The clustering shows that the observations are most strongly separated based on scores on the first principal component. Based on our interpretation of the first principal component loadings, the group in red may correspond to regions with high median prices for its homes (loadings negative & scores highly negative implies prices highly positive). The houses in green correspond to regions with average and low median home prices.

j) An index of home values

A time index of the home values in all available regions could be constructed by using the scores on the first principal component, as we can interpret the first principal component as the influence of area on the median house prices. Two problems that arise from computation of such a time index are that: (i) The loadings will depend on time, such that the Z1-score is not really an exogenous index/measure of home value in an area. (ii) Median house prices are not observed for all categories for each date and region. We have dealt with this by predicting missing median house prices based on the median prices we did observe. This could be very sensible to outliers, especially if the number of observations used to compute a median is low. E.g. a single 20 million US dollar 2 bedroom apartment in Made, Noord-Brabant would obviously be an outlier but our current methodology would predict based on it that Made is a very expensive region, as the methodology will infer for other types of homes in Made that they are also very expensive (Note: current imputation of median prices is not pursued in the time dimension, only based on prices for 2017-11-30).

Problem (i) could possibly be resolved by constructing a common basis of loading vectors across time indices. Problem (ii) is more severe as it is inherent to our data; we only have median prices available and a median is much more informative if we know on how many listings it is based.