NMAI061-22-EX4

Matej Nemec

Visualization of multi-dimensional data and Principal Component Analysis

1)Select data

We decided to go with the classic mtcars data used for many ML tutorials. This datasets characterizes 32 different car models based mainly on their power, weight and engine characteristics.

```
data=mtcars
```

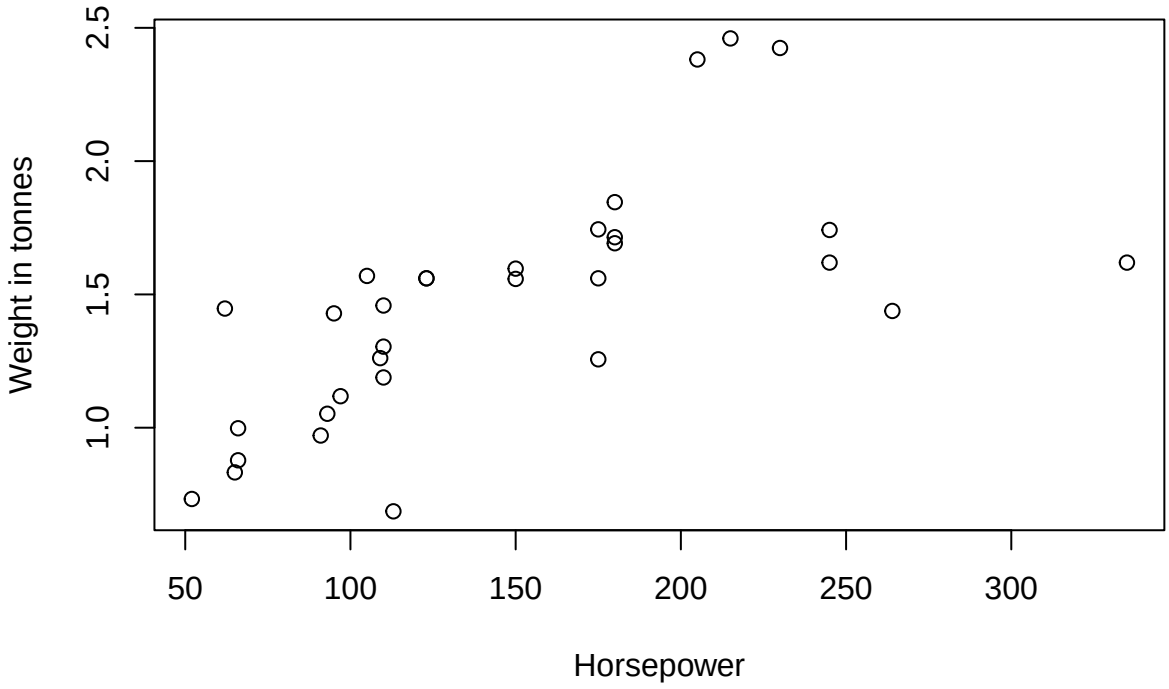
2)Describe and visualize data

We should first explore the nature of individual variables.

1. **mpg** - Miles/(US) gallon
2. **cyl** - Number of cylinders
3. **disp** - Displacement (cu.in.)
4. **hp** - Gross horsepower
5. **drat** - Rear axle ratio
6. **wt** - Weight (1000 lbs)
7. **qsec** - 1/4 mile time
8. **vs** - Engine (0 = V-shaped, 1 = straight)
9. **am** - Transmission (0 = automatic, 1 = manual)
10. **gear** - Number of forward gears
11. **carb** - Number of carburetors

We can see that there are 4-5 variables with very low number of unique values. However in case of cyl, gear and carb these are actually counts of cylinders, forward gears and carburators respectively and as such there is a quantitave relationship between their values. So we can keep these for dimensional reduction. However the vs and am values are categorical representing engine shape and automatic/manual transmission. Therefore there is no quantitative relationship between their values and we should probably not use them for PCA so we will remove them. Nest let us visualize two selected variables. We decided to go with with weight and horsepower as we expect correlation between the two.

```r
data=data[,-c(8,9)] #throw out categoricals
data$wt=data$wt*0.453592 #we prefer actual units over multiples of 1000lbs
plot(data$hp,data$wt,xlab = 'Horsepower',ylab = 'Weight in tonnes')
```



3)Perform and comment on Principal Component Analysis

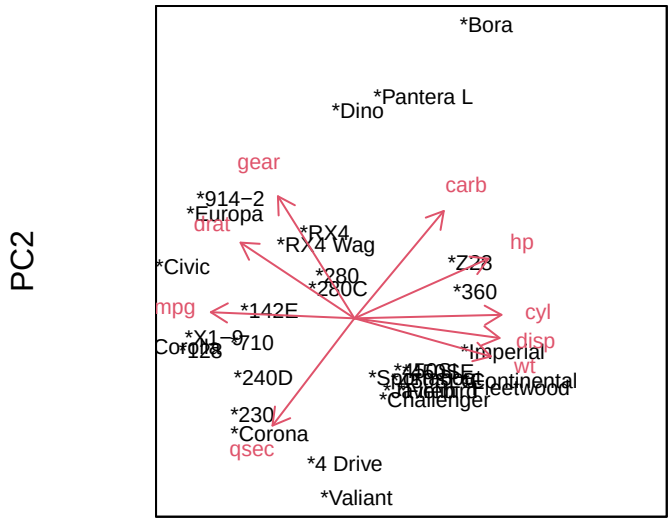
We need to realize that our variables are on completely different scales so we should use the **center** and **scale** arguments when fitting PCA.

```r
pca=prcomp(data, center = TRUE,scale. = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.3782 1.4429 0.71008 0.51481 0.42797 0.35184 0.32413
## Proportion of Variance 0.6284 0.2313 0.05602 0.02945 0.02035 0.01375 0.01167
## Cumulative Proportion  0.6284 0.8598 0.91581 0.94525 0.96560 0.97936 0.99103
##                           PC8     PC9
## Standard deviation     0.2419 0.14896
## Proportion of Variance 0.0065 0.00247
## Cumulative Proportion  0.9975 1.00000
```

We can see that first two components explain almost 90% of the variance which is reasonable and allows us to represent the data in 2D without losing too much information. But first we will create a biplot showing us the mapping of original variables onto the pca plot.

```
temp=strsplit(rownames(data), ' ', fixed = T, perl = F, useBytes = F)
labs=c()
for(i in 1:nrow(data)){
  if (length(temp[[i]])==1){labs=c(labs,paste0('*',temp[[i]][[1]]))
  }else if(length(temp[[i]])==2){labs=c(labs,paste0('*',temp[[i]][2]))
  }else{
    labs=c(labs,paste0('*',temp[[i]][2],' ' ,temp[[i]][3]))
  }
}
biplot(pca,xlabs=labs,yaxt='n', xaxt='n',cex=0.7)
```



4)For two selected variables compute and visualize confidence interval for the means (including Bonferroni correction)

```
plot(data$hp,data$wt,xlab = 'Horsepower',ylab = 'Weight in tonnes')
points(mean(data$hp), mean(data$wt), col="red")
tt1=t.test(data$hp)$conf.int
tt2=t.test(data$wt)$conf.int
lines(c(tt1[1],tt1[1],tt1[2],tt1[2],tt1[1]), c(tt2[1],tt2[2],tt2[2],tt2[1],tt2[1]))

S=var(cbind(data$hp,data$wt))
ev=eigen(S)
S12=ev$vectors%*%diag(1/sqrt(ev$values))%*%t(ev$vectors)
n=nrow(data)

m1=mean(data$hp)
```

```r
m2=mean(data$wt)
mriz=seq(-0.5,1.5,by=0.005)
contour(mriz,mriz,outer(mriz,mriz,function(x,y){n*apply((cbind(x-m1,y-m2)%*%S12)^2,1,sum)<=qchisq(0.95,1


tt1b=t.test(data$hp,conf.level=0.975)$conf.int
tt2b=t.test(data$wt,conf.level=0.975)$conf.int

lines(x=c(tt1b[1],tt1b[1],tt1b[2],tt1b[2],tt1b[1]), y=c(tt2b[1],tt2b[2],tt2b[2],tt2b[1],tt2b[1]),col="bl
```