**CSE 635, Spring 2021, Homework 8**
**Sima Shafaei**

Run a multinom model to predict the second variable, cylwith 3 levels: “4”, “6”, and “8”, as a function of the 10 variables c(1,3,4,5,6,7,8,9,10,11) and compute the accuracy of this model

Code:

```
library(nnet)
head(mtcars)
str(mtcars)
m1=multinom(mtcars$cyl~mtcars$mpg+mtcars$disp+mtcars$hp+mtcars$drat+mt
cars$wt+mtcars$qsec+mtcars$vs)
p1=predict(m1)
t=table(mtcars$cyl,p1)
t
acc=sum(diag(t))/sum(t)
acc
```

Results:

```
>m1=multinom(mtcars$cyl~mtcars$mpg+mtcars$disp+mtcars$hp+mtcars$drat+mtcars$wt+mtcars
$qsec+mtcars$vs)
# weights:  27 (16 variable)
initial  value 35.155593
iter  10 value 10.627593
iter  20 value 0.072489
iter  30 value 0.002017
final  value 0.000000
converged
```

**confusion matrix:**

| | **4** | **6** | **8** |
|---|---|---|---|
| **4** | 11 | 0 | 0 |
| **6** | 0 | 7 | 0 |
| **8** | 0 | 0 | 14 |

**accuracy: 100%**

Perform PCA to decorrelateor compress the 10 variables: c(1,3,4,5,6,7,8,9,10,11)

Code:

```
mtcars_scale=scale(mtcars[,-2])

head(mtcars_scale)

summary(mtcars[,-2])
summary(mtcars_scale)
```

```
cov(mtcars[,-2])
cov(mtcars_scale)  #diag will become 1

c=cor(mtcars_scale)
c

s=svd(c)
PC=mtcars_scale%*%s$u
head(PC)
summary(PC)
```

Results:

**From the summary of PC we can see that the first column has the largest range and the range get smaller and smaller for following columns**

List the variances of the 10 principle components

Code:

```
cov(PC)
s$d
```

Results:

**Variances are:**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 5.6978 | 2.6441 | 0.6067 | 0.2695 | 0.2229 | 0.2061 | 0.13500 | 0.1178 | 0.0769 | 0.0228 |

**As we can see variances are sorted from largest to smallest**

Run a multinom model to predict the second variable cyl as a function of the first principle component.

Code:

```
m2=multinom(mtcars$cyl~PC[,1])
p2=predict(m2)

t=table(mtcars$cyl,p2)
t
acc=sum(diag(t))/sum(t)
acc
```

Results:

```
> m2=multinom(mtcars$cyl~PC[,1])
# weights:  9 (4 variable)
initial  value 35.155593
iter  10 value 0.684501
iter  20 value 0.028064
iter  30 value 0.017481
```

iter  40 value 0.012504
iter  50 value 0.009071
iter  60 value 0.003600
iter  70 value 0.001523
iter  80 value 0.000751
iter  90 value 0.000687
iter 100 value 0.000555
final  value 0.000555
stopped after 100 iterations

**confusion matrix:**

| | **4** | **6** | **8** |
|---|---|---|---|
| **4** | 11 | 0 | 0 |
| **6** | 0 | 7 | 0 |
| **8** | 0 | 0 | 14 |

**accuracy: 100%**