R **for Principal Components Analysis**
Dolores Romero Morales

The aim of this workshop is to work on building Principal Components using a cars dataset that is familiar to you (mtcars), which can be found in R, and an email dataset that is also familiar to you (spam.txt), which can be found on CANVAS.

Note: To plot in 3D we will use the R package 'scatterplot3D'. Follow the two steps below to complete the installation and uploading of the package:

```
install.packages('scatterplot3d',dependencies=TRUE)
library('scatterplot3d')
```

Step 1. Load the ‘mtcars’ dataset and get the dimension of the dataset.

Answer:

```
data(mtcars)
dim(mtcars)
```

Step 2. To get familiar with the mtcars dataset, calculate the covariance matrix and print it with two decimal places.

Answer:

```
covmatmtcars <- cov(mtcars)
print(round(covmatmtcars,2))
```

Note

Throughout the course, we have found examples of predictor variables in a very different scale to the rest. The covariance matrix sheds some light on this. This is again the reasonable for the need of standardization (or, alternatively, normalization).

Step 3. Perform a Principal Components Analysis.

Answer:

```
myPCA <- prcomp(mtcars,center=T,scale.=T)
```

Note1

In Principal Components Analysis we need the data to be centered around the mean. This is achieved with ‘center=T’.

Note2

Instead of standardizing the data ourselves prior to performing Principal Components Analysis, we can simply use the options ‘center=T,scale.=T’ within the ‘prcomp()’ function.

Step 4. Plot the first two Principal Components.

Answer:

```
x <- myPCA$x[,1]
y <- myPCA$x[,2]
plot(x, y, xlab="PC1", ylab="PC2", main="Principal Component Analysis")
text(x, y, labels=row.names(mtcars), cex = 0.7)
```

Step 5. Repeat Step 4 for the first three Principal Components. Use the scatterplot3d package, and the function ‘scatterplot3d()’.

Answer:

```
x <- myPCA$x[,1]
y <- myPCA$x[,2]
```

```
z <- myPCA$x[,3]
scatterplot3d(x, y, z, xlab="PC1", ylab="PC2", zlab="PC3", main="Principal Component Analysis")
text(x, y, z, labels=row.names(mtcars), cex = 0.7)
```

Note1

This plot is more difficult to visualize, but one could use some of the options to improve it. For instance, ‘angle’, yielding

```
scatterplot3d(x, y, z, xlab="PC1", ylab="PC2", zlab="PC3", angle = 15, main="Principal Component Analysis")
text(x, y, z, labels=row.names(mtcars), cex = 0.7)
```

Note2

Some graphical packages would allow you to interact with the plot and see different slices of the data.

Step 6. Print the covariance matrix of the Principal Components, using two decimal places, and get the total of the diagonal.

Answer:

```
covmatPCA <- cov(myPCA$x)
sum(diag(covmatPCA))
```

Note1

We have standardized the data, and therefore, the covariance matrix of the standardized ‘mtcars’ data is equal to the correlation matrix of ‘mtcars’.

Note2

The sum of the diagonal of the Principal Components is equal to 11, which is equal to the sum of the diagonal of the correlation matrix of ‘mtcars’.

Step 7. Print information on the proportion of total variance explained by the Principal Components.

Answer:

```
summary(myPCA)
```

Note

This function reports the standard deviation of the Principal Components in the first row, while the second and third rows refer to variance figures.

Step 8. Perform Steps 3-5 to the spam dataset.

Answer:

```
myspam <- read.table(file.choose(),header=TRUE)
myspamwithoutlabel <- myspam[,-58]
myPCA <- prcomp(myspamwithoutlabel,center=T,scale.=T)
x <- myPCA$x[,1]
y <- myPCA$x[,2]
plot(x, y, xlab="PC1", ylab="PC2", main="Principal Component Analysis")
summary(myPCA)
```