# Problem 4.3

Chen Bo Calvin Zhang

26/10/2020

In this problem, we will conduct a standard PCA using the PCA function available in R and also with code implemented using basic functions. Moreover, we will compare PCA with PCA whitening. The iris flower data set will be used.

First, we need to load the data set. Then we need to centre and standardize it. Lastly we will store the last column (the spcies) as a categorical varible.

```
# load iris data set
data(iris)
# centre and stadardise data by considering the first 4 columns as the 5th is categorical
X = scale(iris[, 1:4], center = TRUE, scale = TRUE)
# save the 5th column as a categorical variable for the species
species = iris[, 5]
```

Now, we perform the PC analysis and print out the results.

```
pca = prcomp(X)
print(pca)
```

```
## Standard deviations (1, .., p=4):
## [1] 1.7083611 0.9560494 0.3830886 0.1439265
##
## Rotation (n x k) = (4 x 4):
##                     PC1         PC2        PC3        PC4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

We need to verify that the principal components are uncorrelated and that the variances are not equal to 1 (as they have not been whitened).

```
print(zapsmall(var(pca$x)))
```

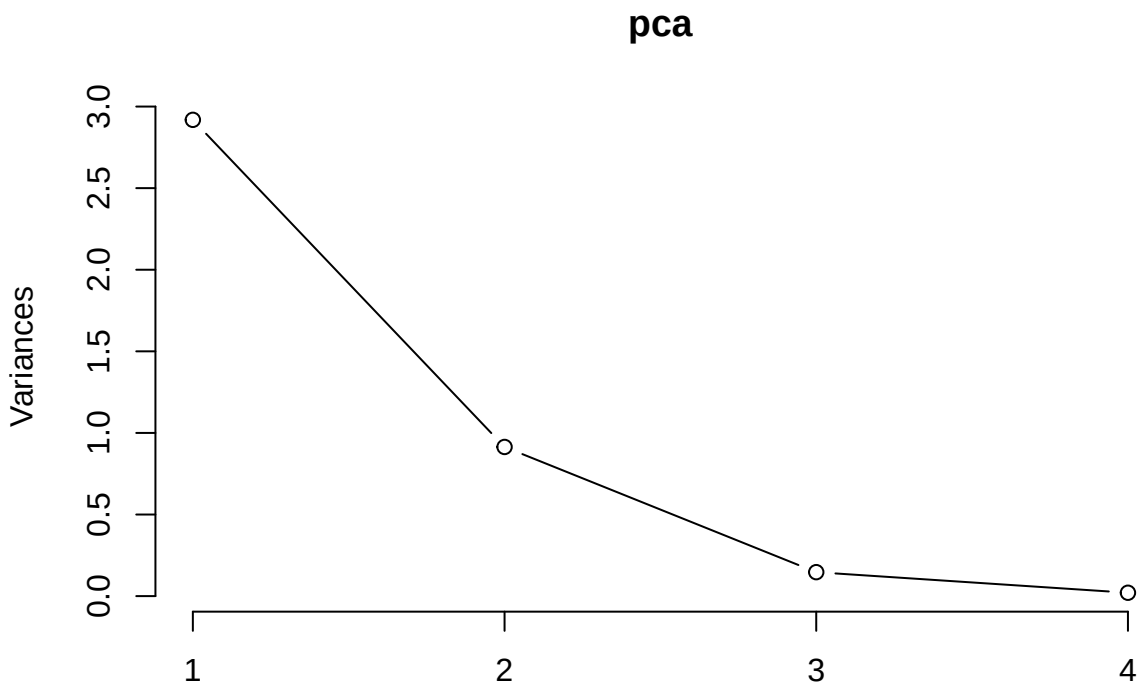
```
##          PC1       PC2       PC3       PC4
## PC1 2.918498 0.0000000 0.0000000 0.0000000
## PC2 0.000000 0.9140305 0.0000000 0.0000000
## PC3 0.000000 0.0000000 0.1467569 0.0000000
## PC4 0.000000 0.0000000 0.0000000 0.0207148
```

We then use summary() to get the (cumulative) proportions of variation for each PC and display a graphical visualisation.

```
print(summary(pca))
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

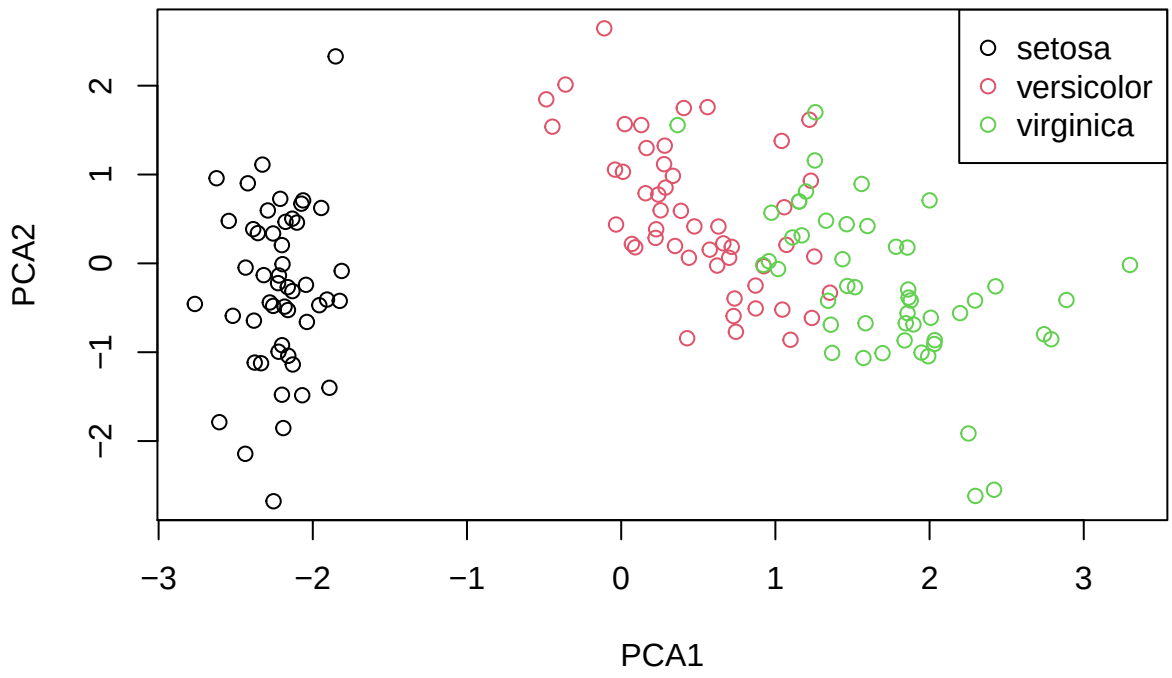
```
plot(pca, type = "l")
```




We can see that the first two components capture 95% of the variance.

We can now plot the first two components.

```
plot(pca$x[, 1], pca$x[, 2], col = species, xlab="PCA1", ylab="PCA2")
legend("topright", levels(species), col=1:3, pch=1)
```




Now, we want to perform the same analysis without the help of the prcomp() function.

```
# compute the PCs without the prcomp() function
eigen = eigen(cov(X))
U = eigen$vectors
lambda = eigen$values
X.PCA = X %*% U
```

Let us check the results.

```
print(zapsmall(var(X.PCA))) # PCs
```

```
##          [,1]      [,2]      [,3]      [,4]
## [1,] 2.918498 0.0000000 0.0000000 0.0000000
## [2,] 0.000000 0.9140305 0.0000000 0.0000000
## [3,] 0.000000 0.0000000 0.1467569 0.0000000
## [4,] 0.000000 0.0000000 0.0000000 0.0207148
```

```
print(lambda) # variance
```

```
## [1] 2.91849782 0.91403047 0.14675688 0.02071484
```

```
# compute the proportions of variation
print(lambda / sum(lambda))
```

```
## [1] 0.729624454 0.228507618 0.036689219 0.005178709
```

```
# compute the cumulative proportions of variation
print(cumsum(lambda) / sum(lambda))
```

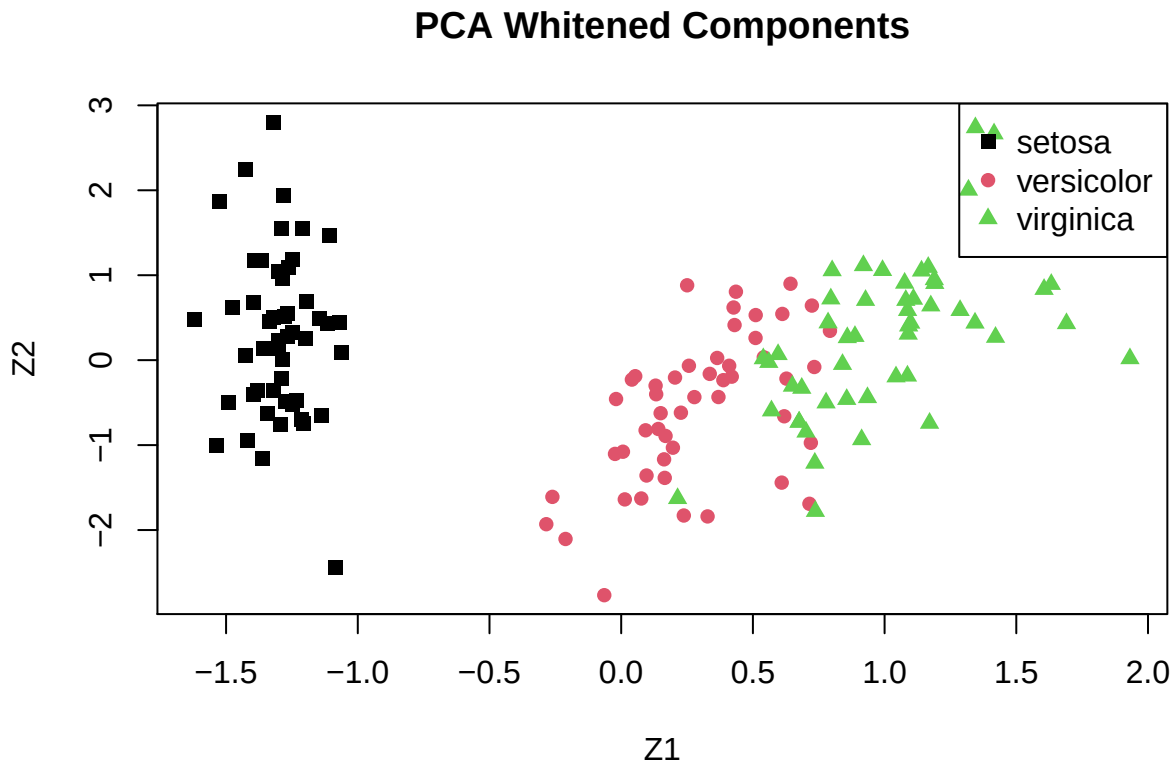
```
## [1] 0.7296245 0.9581321 0.9948213 1.0000000
```

Lastly, let us compare the results with PCA whitening.

```
library("corpcor")
library("whitening")
Z = whiten(X, method="PCA")
print(zapsmall(var(Z)))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

```
plot(Z[,1], Z[,2], xlab="Z1", ylab="Z2", main="PCA Whitened Components",
     col=as.integer(species), pch=as.integer(species)+14)
legend("topright", levels(species), col=1:3, pch=(1:3)+14 )
```



```r
imp = rowSums(cor(Z, X)^2)
print(imp/ sum(imp))
```

```
## [1] 0.729624454 0.228507618 0.036689219 0.005178709
```