# Project PCA

*Chelsey*

Loading in and exploring the data

```
admissionsData <- read.csv("Admission_Predict_Ver1.1.csv", header = TRUE)
#head(admissionsData)
```

The variable we are interested in predicting, Chance.of.Admit, is the 9th variable.

Split the data into test and train data.

```
set.seed(10101)
sample <- sample.int(n = nrow(admissionsData), size = floor(.75*nrow(admissionsData)), replace = FALSE,
train <- admissionsData[sample,]
test <- admissionsData[-sample,]
```

Run PCA on the data and remove the response variable

```
set.seed(43849)
pca.admin <- prcomp(as.matrix(admissionsData[,-c(9)]), scale = TRUE)
summary(pca.admin)
```

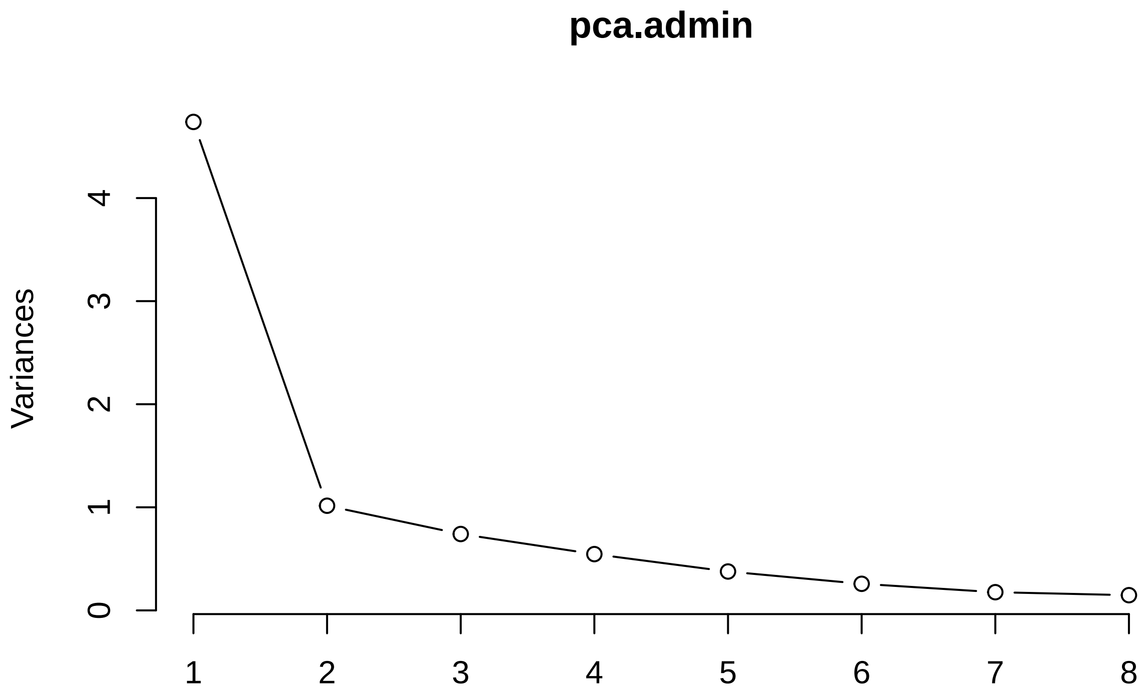
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.1768 1.0076 0.86074 0.73889 0.61428 0.50786
## Proportion of Variance 0.5923 0.1269 0.09261 0.06824 0.04717 0.03224
## Cumulative Proportion  0.5923 0.7192 0.81182 0.88006 0.92723 0.95947
##                            PC7     PC8
## Standard deviation     0.42017 0.38431
## Proportion of Variance 0.02207 0.01846
## Cumulative Proportion  0.98154 1.00000
```

To choose the number of principal components to keep, we can either use the Kaiser criterian, cumulative proportion/percent of variance, or a scree plot.

Using the Kaiser criterian, we keep all principal components with a standard deviation greater than 1 (since the data is scaled). Hence the Kaiser criterian is telling us to keep the first two principal components.

I will now compare this with a scree plot.

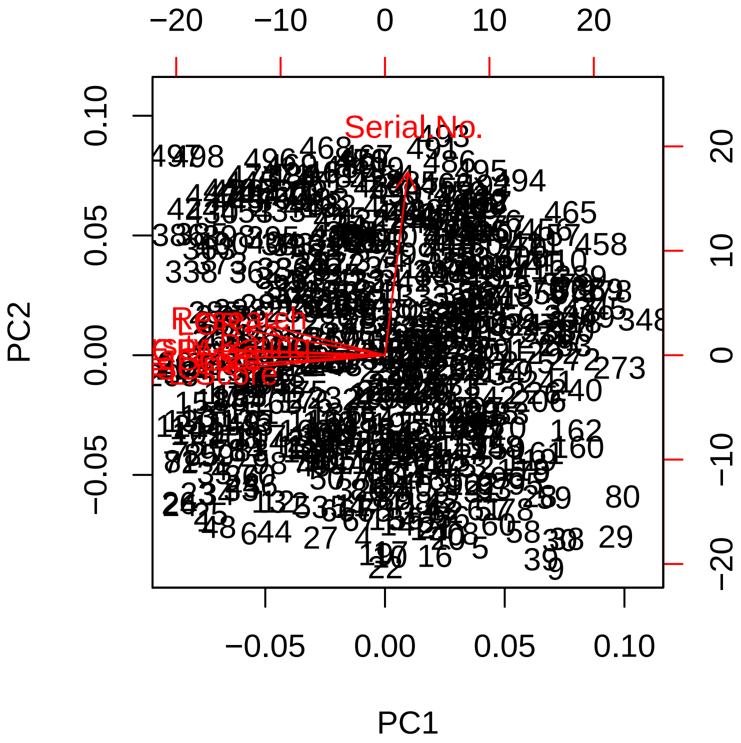
```
plot(pca.admin, type="lines")
```



The above scree plot plots the monotonically decreasing eigenvalues and the location of an 'elbow' or plateau indicates the number of principal components. The scree plot suggests probably 2 principal components, which correlates with the Kaiser criterian.

The first two principal components that will be retained explain 72% of the variation in the data. We can now view the data projected onto the components using a biplot.

```
biplot(pca.admin)
```



The PCA plot above is suggesting that serial number has no relation to the other factors. In fact, this shouldn't affect the data at all. I will remove this column (column 1) as well and reperform the PCA.

```r
set.seed(43849)
pca.admin2 <- prcomp(as.matrix(admissionsData[,-c(1,9)]), scale = TRUE)
summary(pca.admin)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.1768 1.0076 0.86074 0.73889 0.61428 0.50786
## Proportion of Variance 0.5923 0.1269 0.09261 0.06824 0.04717 0.03224
## Cumulative Proportion  0.5923 0.7192 0.81182 0.88006 0.92723 0.95947
##                            PC7     PC8
## Standard deviation     0.42017 0.38431
## Proportion of Variance 0.02207 0.01846
## Cumulative Proportion  0.98154 1.00000
```

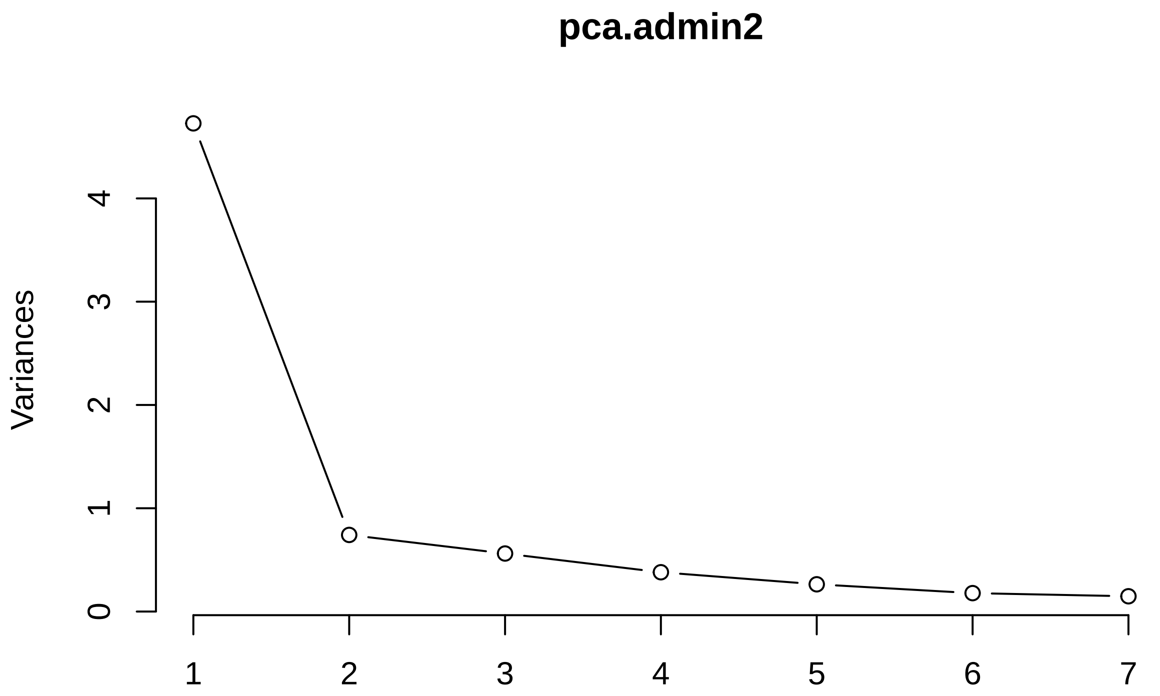
To choose the number of principal components to keep, we can either use the Kaiser criterian, cumulative proportion/percent of variance, or a scree plot.

Using the Kaiser criterian, we keep all principal components with a standard deviation greater than 1 (since the data is scaled). Hence the Kaiser criterian is telling us to keep the first principal component.

However, the cumulative proportion of variance explained from the first principal component is only 67.5%. The first two components might be better as these explain 78.1% of the data.

I will now compare this with a scree plot.

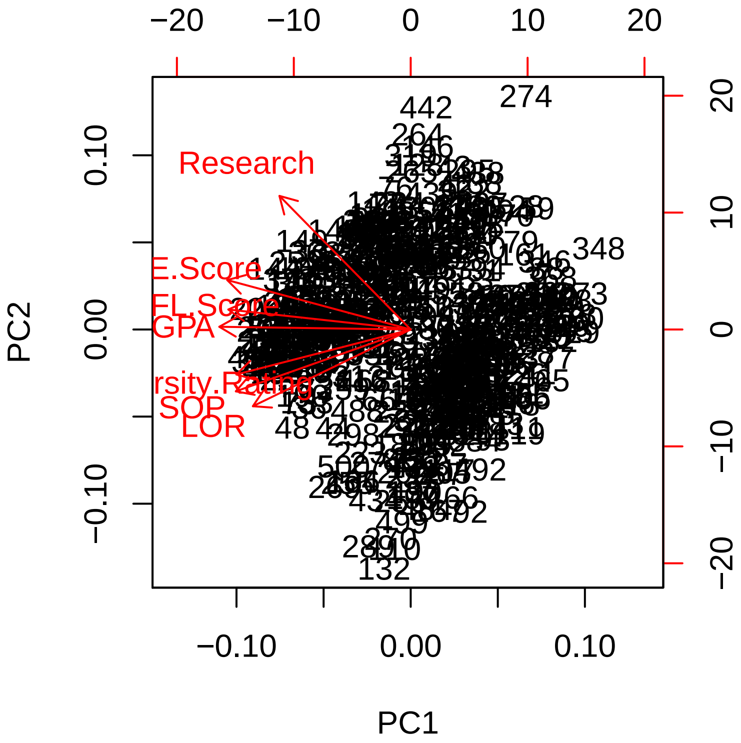
```r
plot(pca.admin2, type="lines")
```



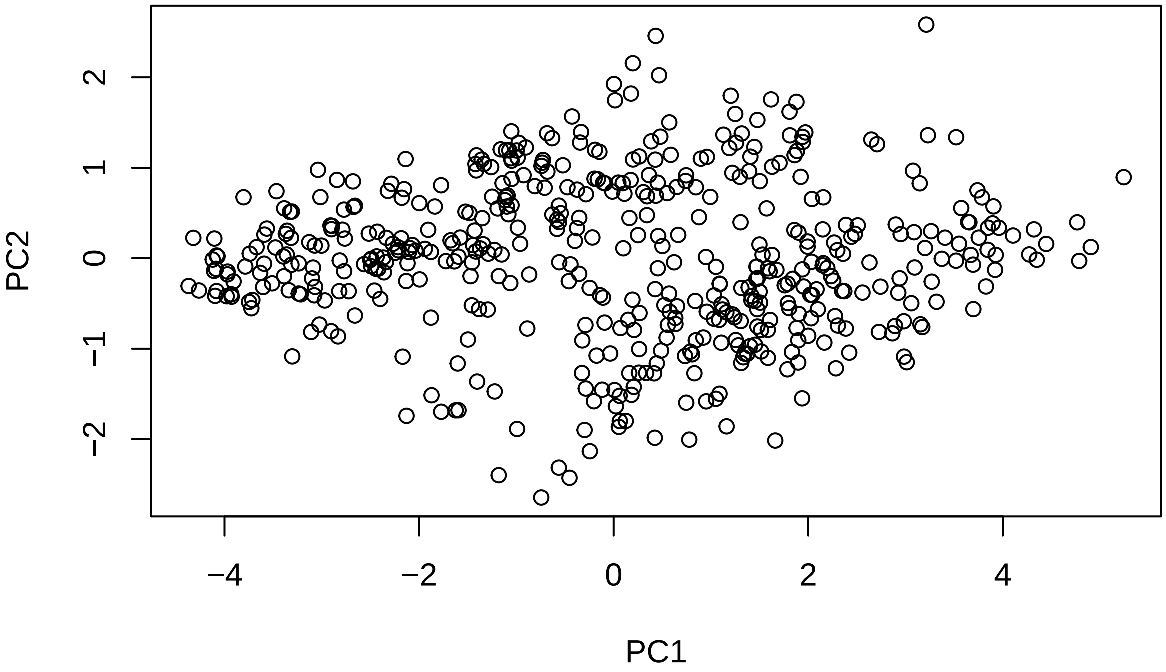
The scree plot suggests 2 principal components.

We can now view those with a biplot.

```r
biplot(pca.admin2)
```

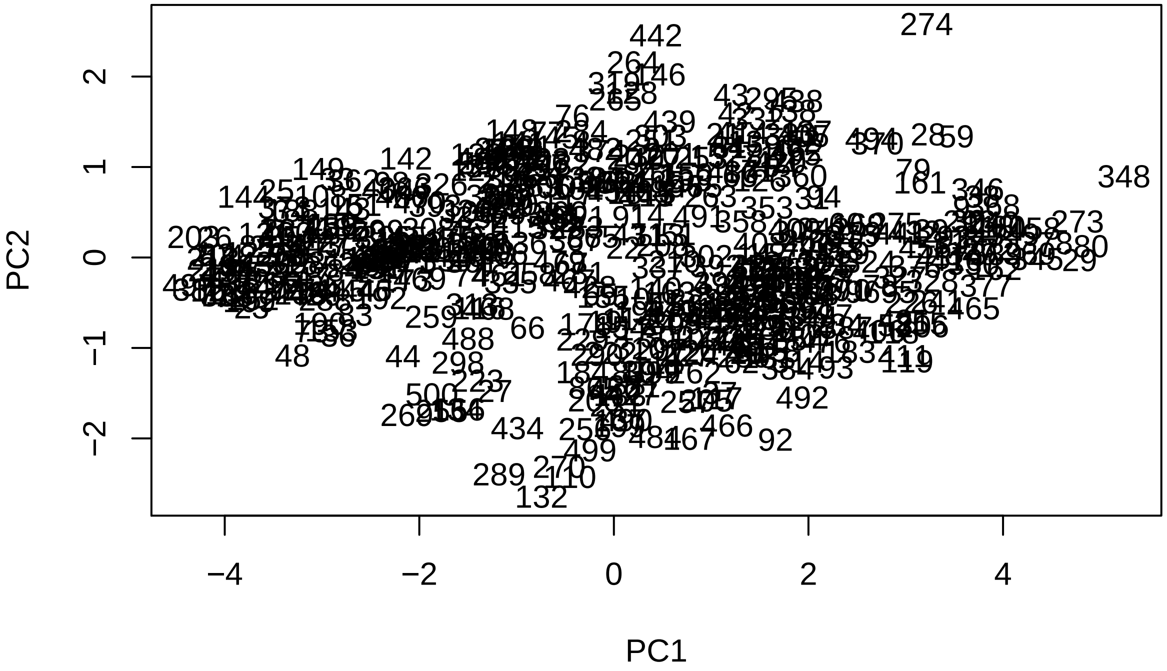


```
plot(pca.admin2$x[,1:2])
```



We can put data labels on the biplot by observation number

```
plot(pca.admin2$x[,1:2], type = "n")
text(pca.admin2$x[,1:2], labels = 1:nrow(admissionsData))
```



Take a look at the component loadings (eigenvectors) which provide the coefficients of the original variables, rounded to 2 decimal places.

```
round(pca.admin2$rotation[,1:2], 2)
```

```
##                     PC1   PC2
## GRE.Score         -0.40  0.27
## TOEFL.Score       -0.40  0.11
## University.Rating -0.38 -0.25
## SOP               -0.38 -0.34
## LOR               -0.35 -0.43
## CGPA              -0.42  0.02
## Research          -0.29  0.74
```

These are the coefficients of the original variables. The magnitudwes are pretty similar for the first component, perhaps with the exception of research. They are also all containing the same sign. WHAT DOES THIS MEAN?

The second component is a little less clear. The highest magnitude is the research aspect, alog with the letter of recommendation. Perhaps this component indicates previous experience you have. A reference letter most likely comes from someone you have worked with, conducted research with, volunteered with, or TA'd for. Therefore this could be indicative of research and other activities in both academic and non-academic settings.

To explore the second principal component closer, let's look at the larger magnitudes.

```
round(pca.admin2$rotation[,2],2)[abs(pca.admin2$rotation[,2]) > .2]
```

```
##         GRE.Score University.Rating               SOP               LOR
##              0.27             -0.25             -0.34             -0.43
##          Research
##              0.74
```

We have a very large positive magnitude for research and a positive magnitude for GRE score. The loading of university rating, SOP, and LOR is negative. HOW DO I INTERPRET THIS?

IS PC1 PEOPLE WHO ARE NOT AS LIKELY TO GET IN AND PC2 IS PEOPLE WHO ARE MORE LIKELY TO GET IN? NO!

We can now look at the four students who scored highest on PC1:

```
admissionsData[order(pca.admin2$x[,1], decreasing = TRUE)[1:4],1:9]
```

```
##     Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA
## 348        348       299          94                 1 1.0 1.0 7.34
## 80          80       294          93                 1 1.5 2.0 7.36
## 29          29       295          93                 1 2.0 2.0 7.20
## 273        273       294          95                 1 1.5 1.5 7.64
##     Research Chance.of.Admit
## 348        0            0.42
## 80         0            0.46
## 29         0            0.46
## 273        0            0.49
```

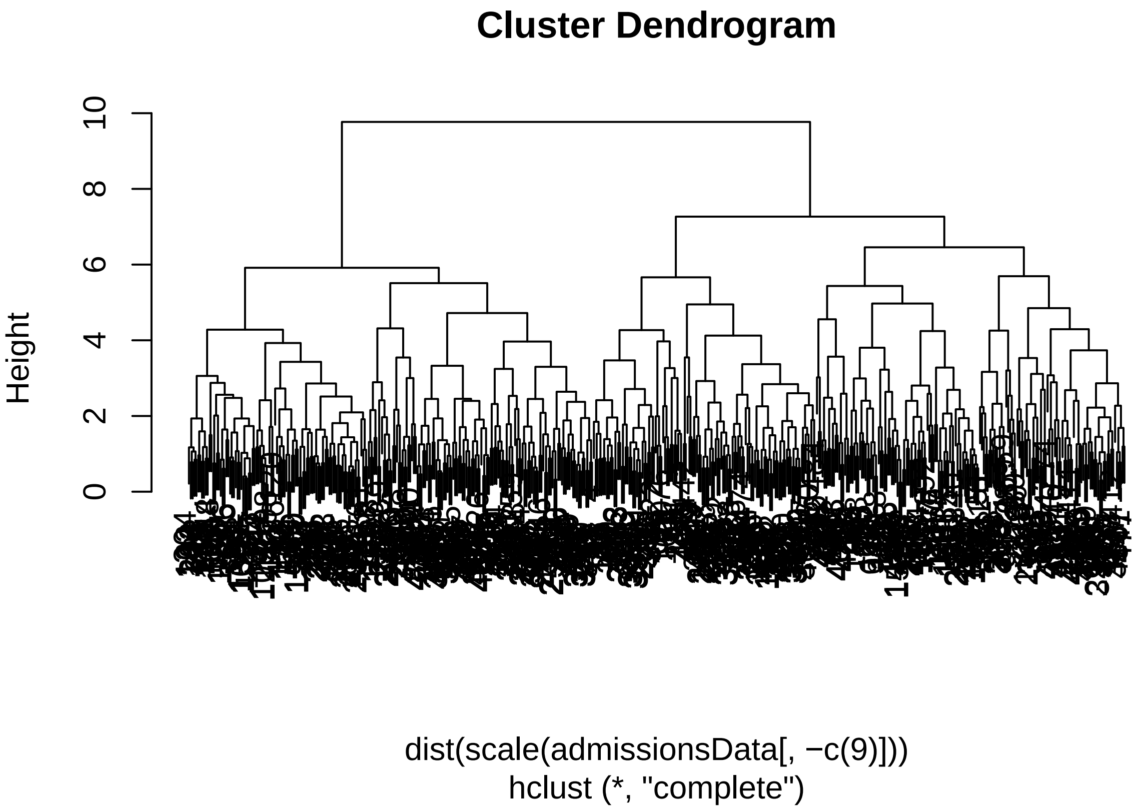
```
admissionsData[order(pca.admin2$x[,2], decreasing = TRUE)[1:4], 1:9]
```

```
##     Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA
## 274        274       312          99                 1 1.0 1.5 8.01
## 442        442       332         112                 1 1.5 3.0 8.66
## 264        264       324         111                 3 2.5 1.5 8.79
## 146        146       320         113                 2 2.0 2.5 8.64
##     Research Chance.of.Admit
## 274        1            0.52
## 442        1            0.79
## 264        1            0.70
## 146        1            0.81
```

## Clustering on PCA

We can prove that if we do NOT reduce the components, then the distance between observations are retained.

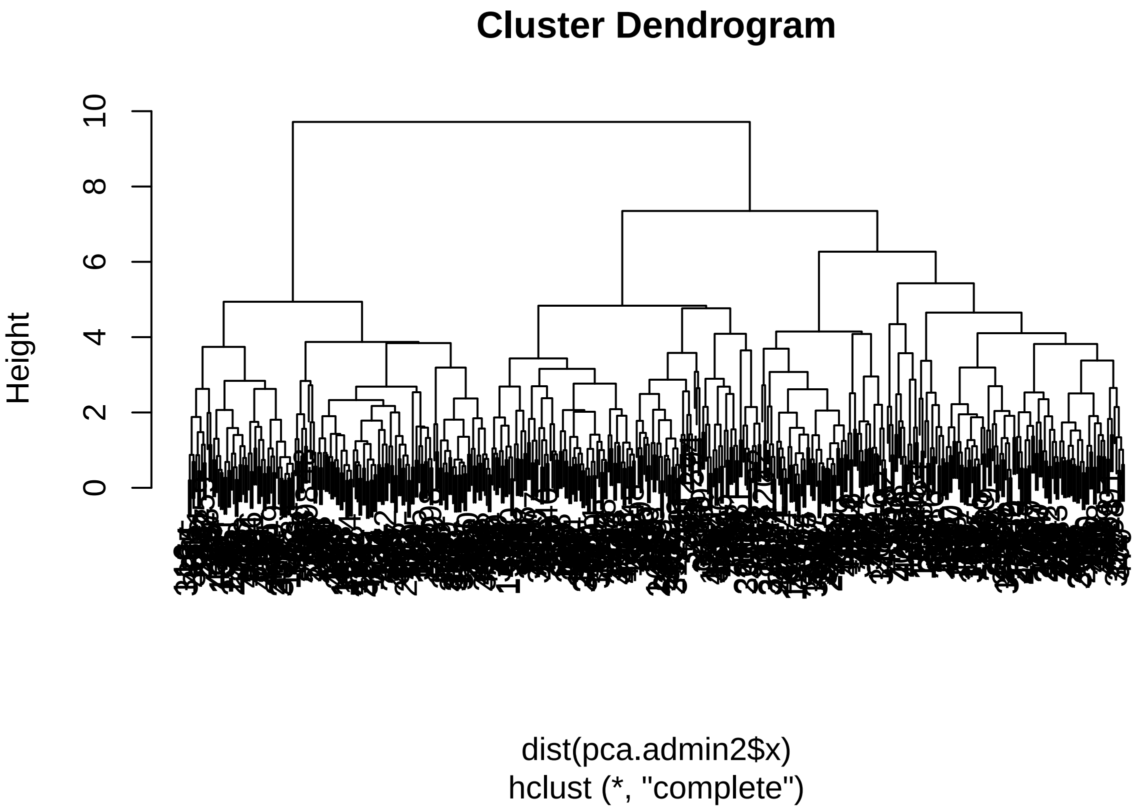
```
test1 <- hclust(dist(scale(admissionsData[,-c(9)])))
plot(test1)
```



IF I AM CORRECT, THIS SHOULD MATCH UP WITH WHAT SOMEONE GETS FROM DOING HIERARCHICAL CLUSTERING ON THE DATA WITHOUT THE PCA.

Now we can do hierarchical clustering with respect to the principal components.

```r
test2 <- hclust(dist(pca.admin2$x))
plot(test2)
```



It is difficult to see, but these should be the same dendrograms. We can use R to compare the distance matrices to compare the pairwise distances contained in the objects. Note that here I still need to remove the response column of chance of admission (9) and the unique identifier of the serial number (1).

```r
all.equal(dist(scale(admissionsData[,-c(1,9)])), dist(pca.admin2$x), check.attributes = FALSE)
```

```
## [1] TRUE
```

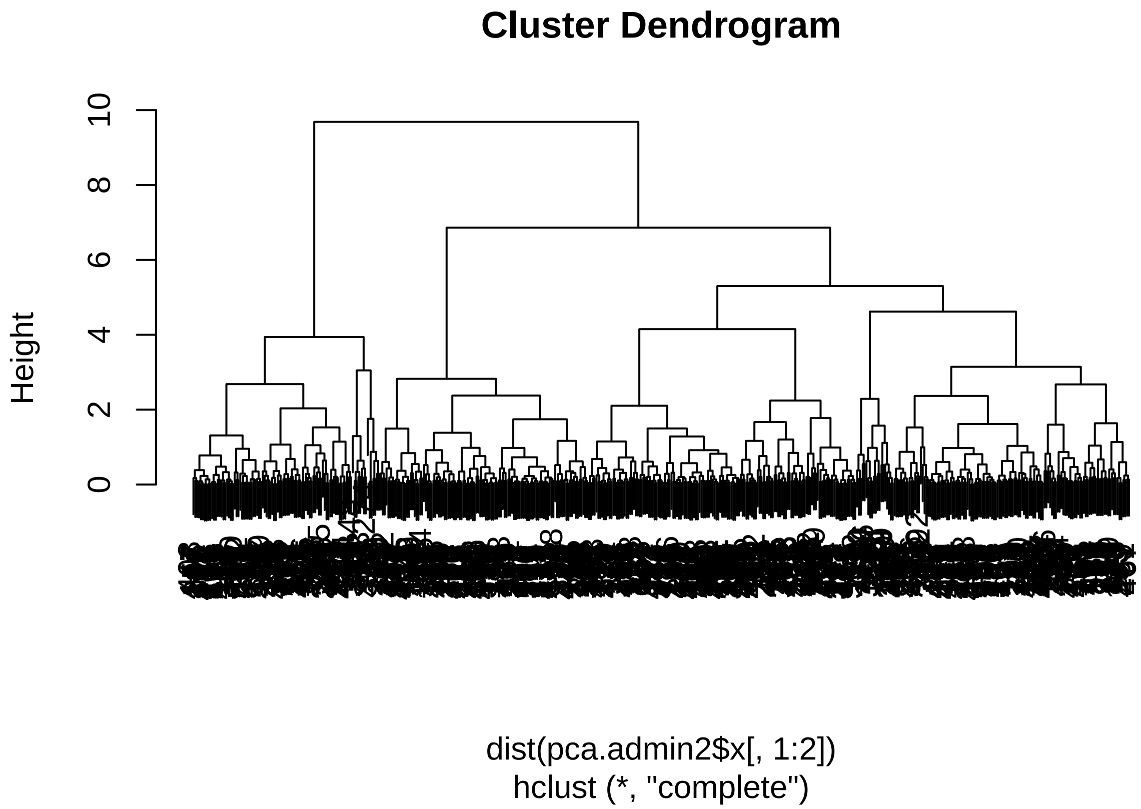
Since this is true, it is saying that these distance matrices are the exact same. On the other hand, if we reduce the components (say to the two we used earlier), then we should see some loss of information. Let's see how much.

```r
all.equal(dist(scale(admissionsData[,-c(1,9)])), dist(pca.admin$x[,1:2]), check.attributes = FALSE)
```

```
## [1] "Mean relative difference: 0.1878109"
```

Clustering applied to the scores on the first two components.

```r
pcclust <- hclust(dist(pca.admin2$x[,1:2]))
plot(pcclust)
```



NOTE: I DO NOT REALLY KNOW WHAT I AM TRYING TO CLUSTER ON HERE .... MAYBE I CAN MEET WITH THE PERSON WHO IS DOING HC TO COMBINE THESE IDEAS?