Homework 1

Aaron Goodfellow

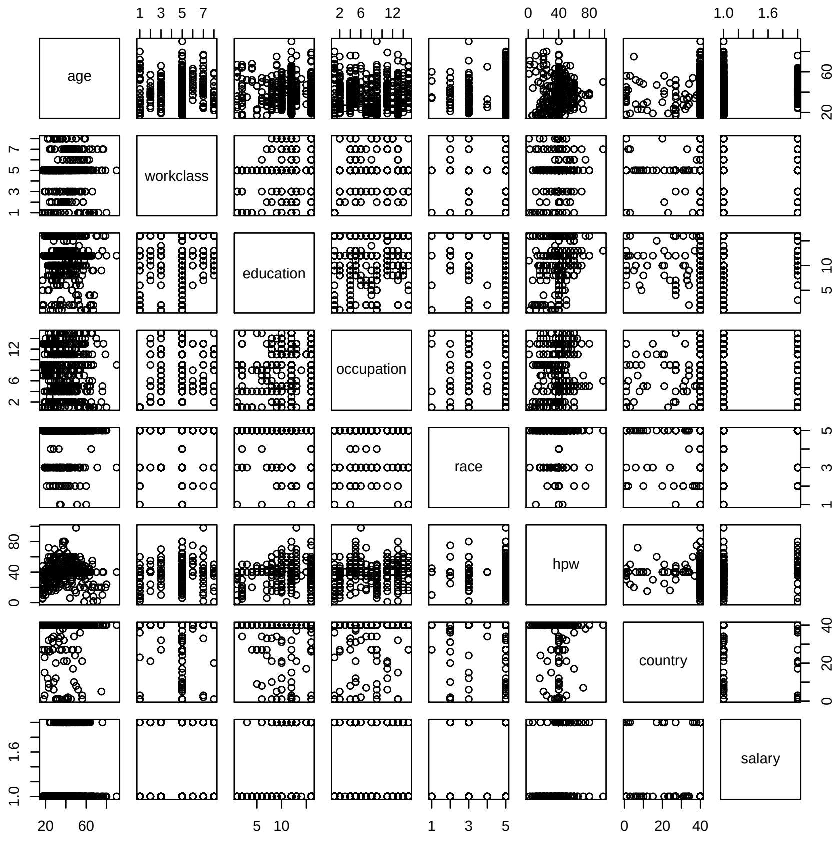
October 1, 2018

1 Census Income

Data Visualization is done via R - the data is read as a csv directly from the website

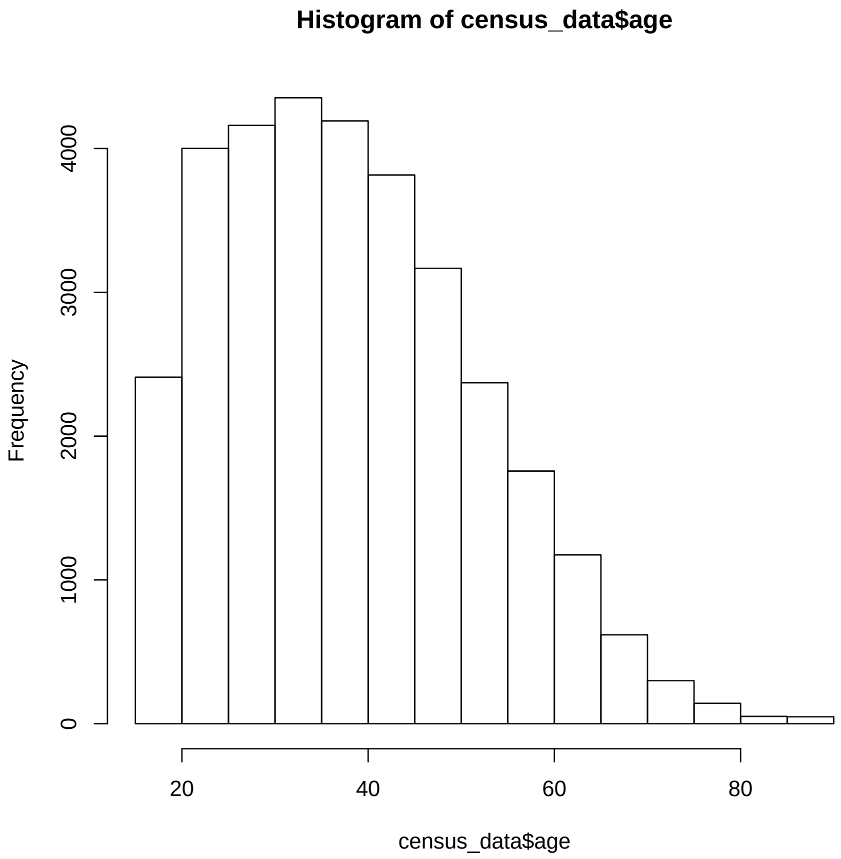
```
# A quick summary of relevant, non-binary
# variables
census_frame <- census_data
census_frame$education.num <- NULL
census_frame$marital <- NULL
census_frame$fnlwgt <- NULL
census_frame$relationship <- NULL
census_frame$sex <- NULL
census_frame$cgain <- NULL
census_frame$closs <- NULL
head(census_data, 5)

##   age         workclass fnlwgt  education education.num
## 1  50  Self-emp-not-inc  83311  Bachelors            13
## 2  38           Private 215646    HS-grad             9
## 3  53           Private 234721       11th             7
## 4  28           Private 338409  Bachelors            13
## 5  37           Private 284582    Masters            14
##               marital         occupation   relationship   race     sex
## 1  Married-civ-spouse    Exec-managerial        Husband  White    Male
## 2            Divorced  Handlers-cleaners  Not-in-family  White    Male
## 3  Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
## 4  Married-civ-spouse     Prof-specialty           Wife  Black  Female
## 5  Married-civ-spouse    Exec-managerial           Wife  White  Female
##   cgain closs hpw        country salary
## 1     0     0  13  United-States  <=50K
## 2     0     0  40  United-States  <=50K
## 3     0     0  40  United-States  <=50K
## 4     0     0  40           Cuba  <=50K
## 5      0      0  40   United-States   <=50K
```

```
plot(head(census_frame, 500))
```



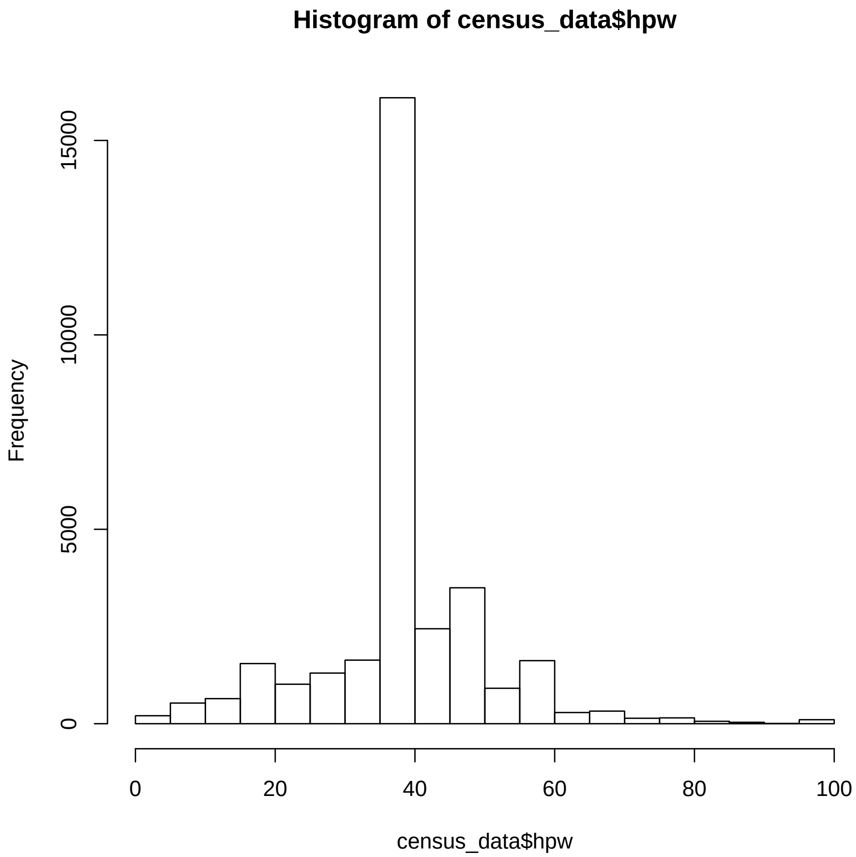
The above plot shows a scatter plot matrix of the first 500 points of data in this set. One interesting relationship to be seen off the bat is the one between age and hourse per week worked. It seems there is a large concentration of young to middle aged people working 40 hours a week. Let's observe the frequencies of both age and hours-per-week in histograms, and then look at them ploted together in a smooth scatterplot

```
# Histogram of age
hist(census_data$age)
```



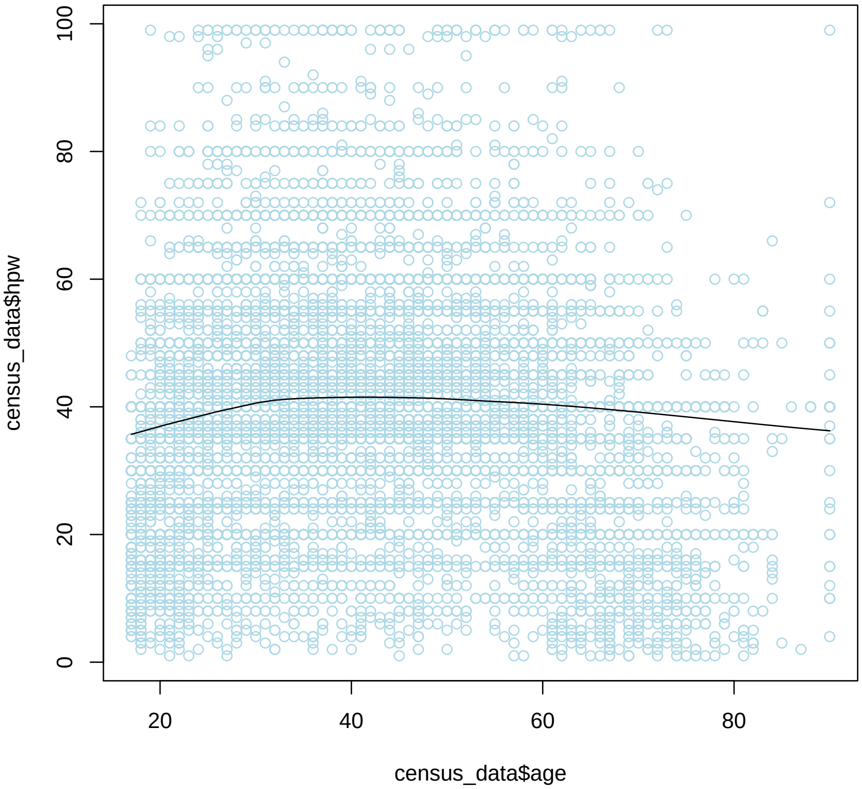
A frequency histogram of the ages of people surveyed. It's a standard distribution, skewed right slightly. This isn't very surprising - I think most people would expect there to be more 40 year olds than 80 year olds...

```
# Histogram of hpw
hist(census_data$hpw)
```



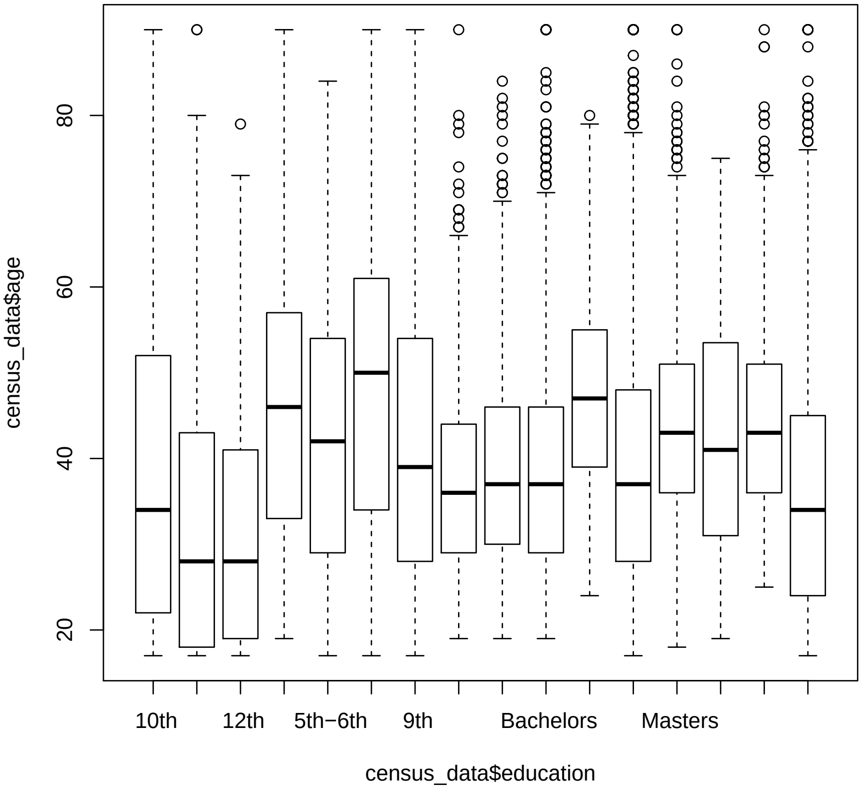
This is an interesting histogram - apparently the huge majority of people work 40 hours per week. This is pretty standard across the United States for full-time employees, so I guess it's not too surprising.

```
# Relationship between age and hpw
scatter.smooth(census_data$hpw ~ census_data$age, col='lightblue',)
```



As we can see from the above data, the average working time for a majority of people is around 40 hours a week. It tapers off slightly at the end of one's life, but not by much, surprisingly. Also, I find that presumably 18 year olds are starting at almost 40 hours per week themselves! I would have expected a more standard distribution.

```r
# Plot of age vs education
plot(census_data$age ~ census_data$education)
```



I threw this plot in, because I thought it was fairly interesting. I would assume that as one gets higher and higher degrees, the median age would tend to increase, but that doesn't seem to exactly be the case. It's important to remember though, this data isn't showing the age at which people graduated, merely the age of the people now and what sort of degree those people have

2 Multivariate Normal Distributions

a)

This problem requires 100 three-dimensional vectors from a normal distribution with a mean vector of [1, 2, 1] and a covariance matrix of the following form:

$$\begin{bmatrix} 5 & 0.8 & -0.3 \\ 0.8 & 3 & 0.6 \\ -0.3 & 0.6 & 4 \end{bmatrix}$$

To generate data meeting these requirements I used the following code snippet:

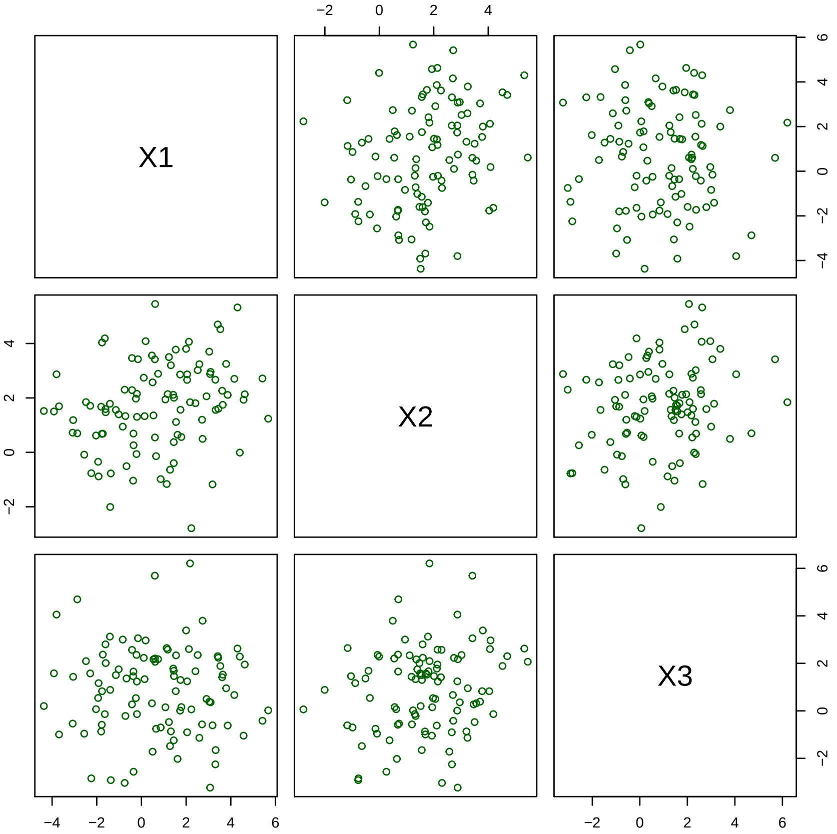
```
library("MASS", lib.loc="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
# Create a covariance matrix
covariance <- matrix(
        c(5, 0.8, -0.3, 0.8, 3, 0.6, -0.3, 0.6, 4),
        nrow=3,
        ncol=3)

# Create all 100 vectors from the given mean and covariance matrix
norm_dist_sample <- mvrnorm(100, c(1, 2, 1), covariance)

# I then put those vectors into a dataframe so they would be easier to work with
df <- data.frame(norm_dist_sample)

# Here are the first five entries as an example:
head(df, 5)
```

```
##           X1        X2          X3
## 1 -0.7182553 1.3339378 -0.21481182
## 2  4.6251870 2.1336970  1.95087207
## 3  4.2987714 5.3252363  2.62371274
## 4 -2.0327488 0.6195676  0.06915152
## 5  2.9132804 2.0595466  0.50148857
```

b)

R is really great when it comes to making scatter plots that show the relationship between different elements. All we need to do is call the plot function on the dataframe from above, and R is smart enough to take care of everything else.

```
# Plot the relationships between x1, x2, and x3 as a scatter plot
plot(df, col='darkgreen')
```



c)

Again, I rely on R's builtin libraries to calculate both the Euclidean distance from each point to another, and to compute the Mahalanobis distance from the mean:

```
# Calculate the euclidean distances point to point
stats::dist(head(df, 5), method = "euclidean")

##          1        2        3        4
## 2 5.820840
## 3 7.011294 3.277985
## 4 1.522778 7.082483 8.291999
## 5 3.771955 2.244289 4.133782 5.169493

# Calculate the Mahalanobis distances from point to mean
```

```
mahalanobis(head(df, 5), c(1, 2, 1), covariance)

##         1         2         3         4         5
## 1.0379456 3.1176429 5.2706483 2.3348249 0.7813596
```

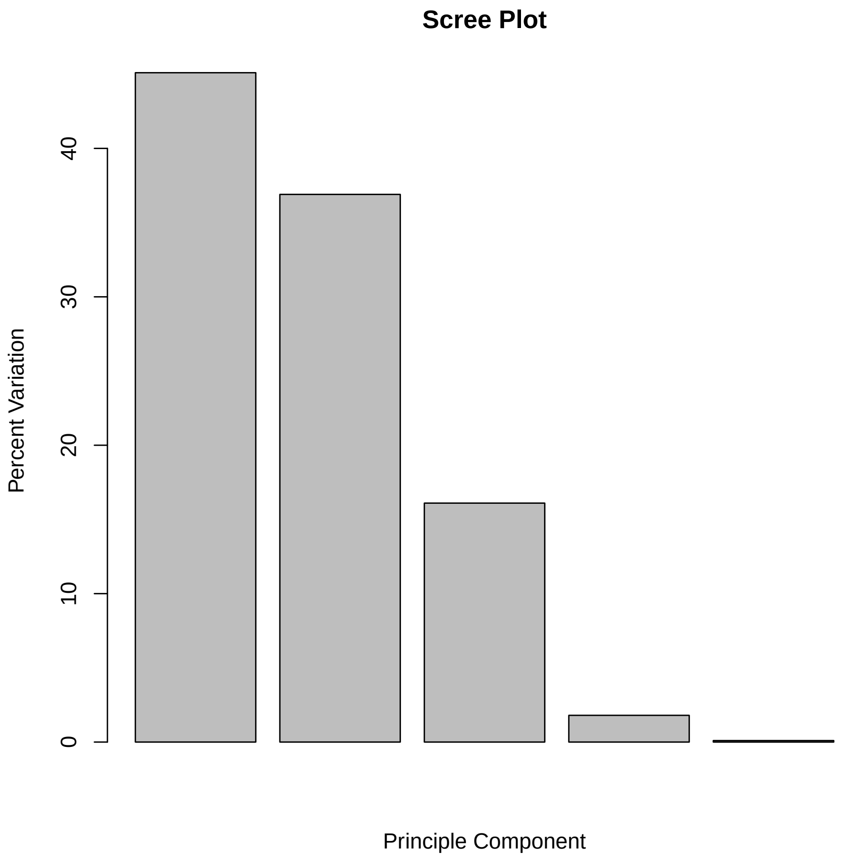
3 Principal Component Analysis

Given data of the following form:

```
## Warning in matrix(+c(5700, 12.8, 2500, 270, 25000, 1000, 10.9, 600,
10, :  data length [59] is not a sub-multiple or multiple of the number
of rows [12]

##       [,1] [,2] [,3]  [,4]  [,5]
##  [1,] 5700 12.8 2500   270 25000
##  [2,] 1000 10.9  600    10 10000
##  [3,] 3400  8.8 1000    10  9000
##  [4,] 3800 13.6 1700   140 25000
##  [5,] 4000 12.8 1600   140 25000
##  [6,] 8200  8.3 2600    60 12000
##  [7,] 1200 11.4  400    10 16000
##  [8,] 9100 11.5 3300    60 14000
##  [9,] 9900 12.5 3400   180 18000
## [10,] 9600 13.7 3600   390 25000
## [11,] 9600  9.6 3300    80 12000
## [12,] 9400 11.4 4000 13000  5700
```

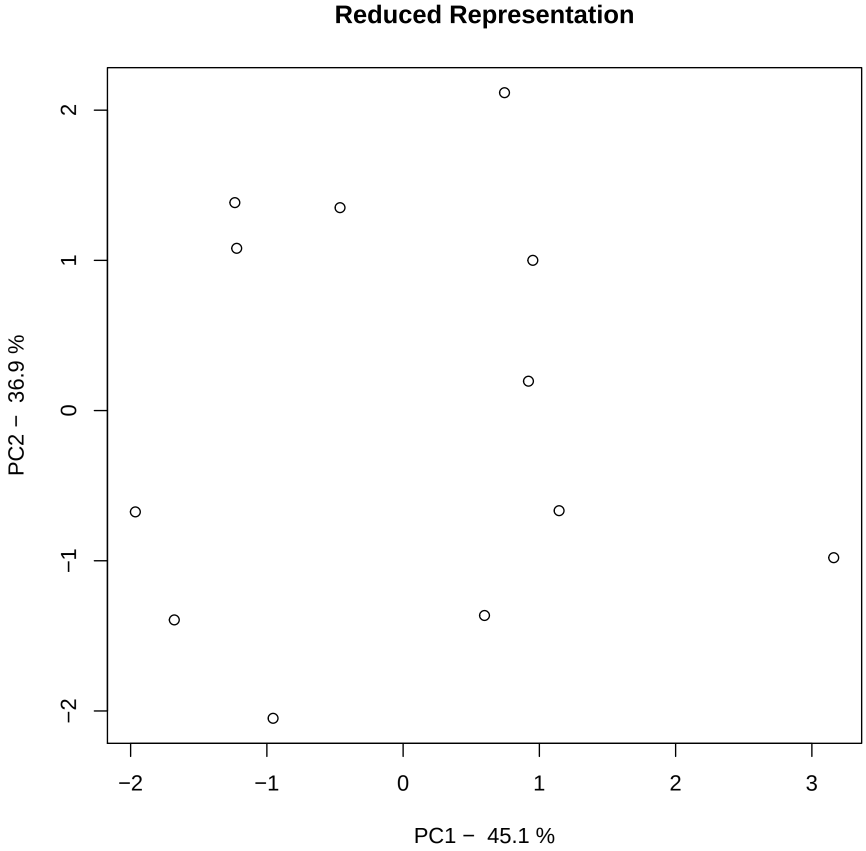
We are to reduce this five dimensional matrix to two dimensions using PCA. I will also list the eigenvalues and eigenvectors obtained via PCA.
Using a builtin R funciton, prcomp, we can pass our matrix, standardize the values, and git a 5x5 matrix of the eigenvectors:

```
# Save the eigenvectors as a matrix called pca_five.
# Center and standardize the values
pca_five <- prcomp(five_dim_matrix,
              center = TRUE,
              scale. = TRUE)
pca_five

## Standard deviations (1, .., p=5):
## [1] 1.50109168 1.35802019 0.89805977 0.30057262 0.07516434
##
## Rotation (n x k) = (5 x 5):
##              PC1        PC2         PC3         PC4         PC5
## [1,]  0.59854138  0.1908131 -0.39038773  0.04052256 -0.67178413
## [2,] -0.06358739  0.6591606  0.44142262  0.59878285 -0.08982789
## [3,]  0.61641955  0.2503448 -0.17673210  0.02888658  0.72476598
## [4,]  0.44708876 -0.1482336  0.78386461 -0.38558892 -0.12253933
## [5,] -0.24050664  0.6666722 -0.08405667 -0.70021694 -0.01831432

# Display eigenvalues
pca_five$sdev

## [1] 1.50109168 1.35802019 0.89805977 0.30057262 0.07516434

# Manipulate data to show a percentage variance graph:
pca_five.percentage <-
  round(pca_five$sdev^2/sum(pca_five$sdev^2)*100, 1)

# Scree plot with the relative variances for each principle component:
barplot(pca_five.percentage, main='Scree Plot',
        xlab='Principle Component', ylab='Percent Variation')
```



As we can see from the above plot, the first and second principle components are the ones that most influence this dataset. Plotting a two dimensional scatter plot with just this data will yeild:

```
plot(pca_five$x[,1], pca_five$x[,2], main='Reduced Representation',
     xlab= paste('PC1 - ', pca_five.percentage[1], '%'),
     ylab = paste('PC2 - ', pca_five.percentage[2], '%'))
```

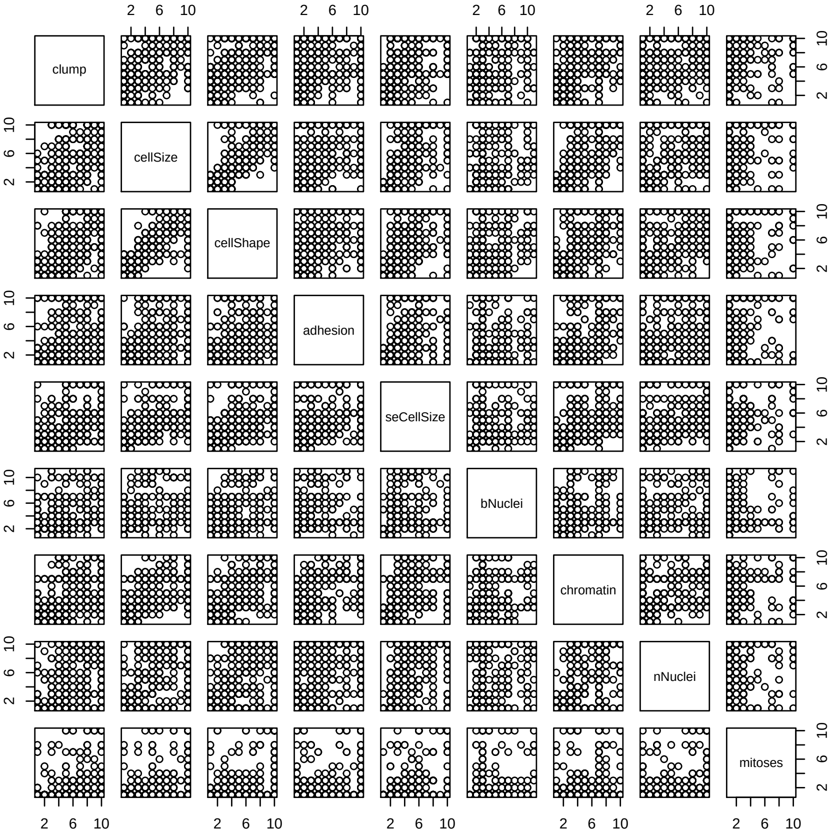


Although the variables in this instance are not labeled, for the above graph, I've labeled the axes as PC1 and PC2 as primary component 1 and 2 respectivley, along with the percentage variance.

4 Breast Cancer Dataset

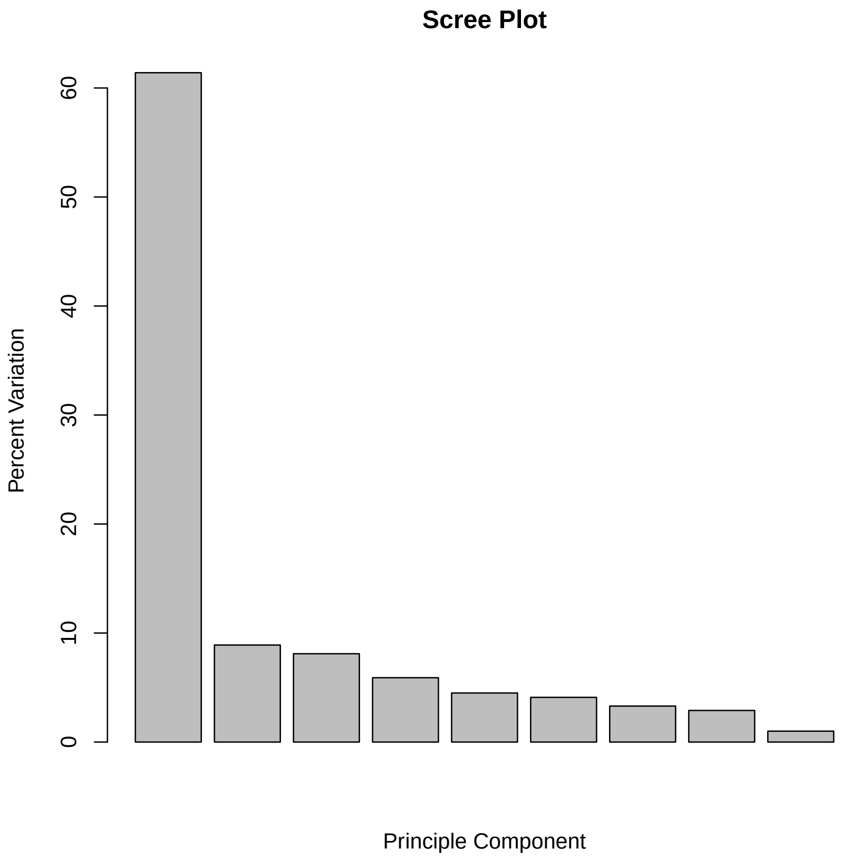
The following matrix shows the head of the data which is being read directly from the url with R's 'read.csv' function. I've also included a preliminary scatterplot matrix (omitting codeNum and class) to see if any relationships jump out right away:

```
##   codeNum clump cellSize cellShape adhesion seCellSize bNuclei chromatin
## 1 1002945     5        4         4        5          7      10         3
## 2 1015425     3        1         1        1          2       2         3
## 3 1016277     6        8         8        1          3       4         3
## 4 1017023     4        1         1        3          2       1         3
## 5 1017122     8       10         10        8          7      10          9
## 6 1018099     1        1          1        1          2      10          3
##   nNuclei mitoses class
## 1       2       1     2
## 2       1       1     2
## 3       7       1     2
## 4       1       1     2
## 5       7       1     4
## 6       1       1     2
```

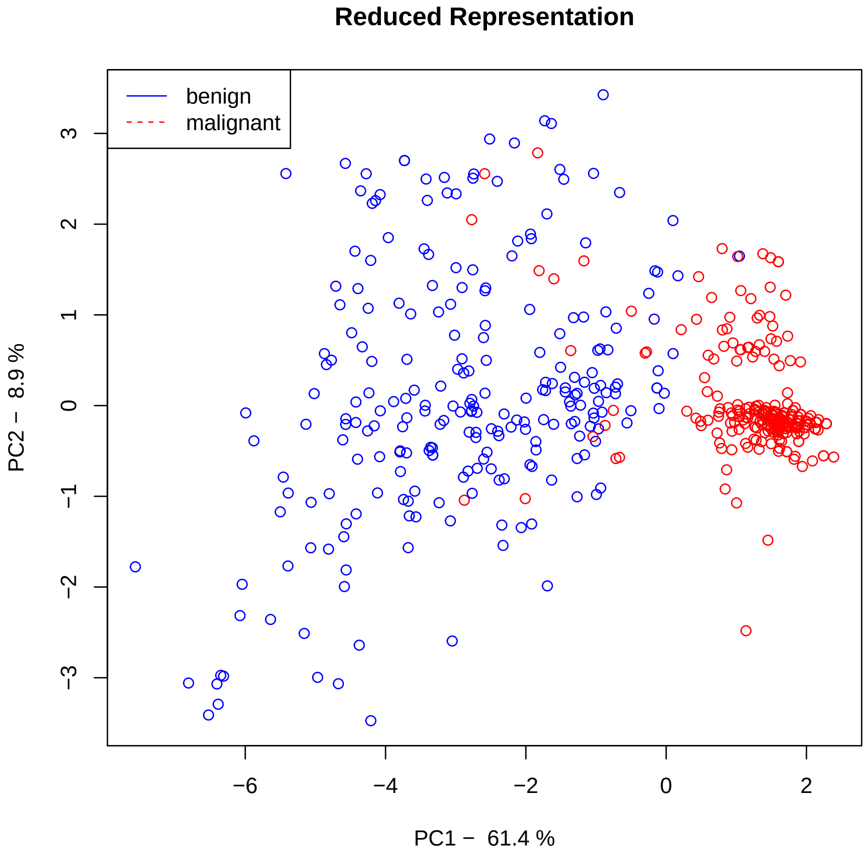


Other than a linear relationship between cell shape and cell size, I don't see any other scatterplots that immediatley jump out at me. Continuing on to perform principle component analysis of this data, we run the following code:

```r
# Simply remove rows with '?' in them
cancer_data.numeric$bNuclei <- as.numeric(cancer_data.numeric$bNuclei)
# Perform pca on the data
cancer_pca <- prcomp(cancer_data.numeric,
              center = TRUE,
              scale. = TRUE)
# Prints out eigenvalues and eigenvectors
cancer_pca

## Standard deviations (1, .., p=9):
## [1] 2.3500384 0.8928532 0.8523119 0.7277885 0.6332209 0.6102180 0.5460986
## [8] 0.5118452 0.3007939
##
## Rotation (n x k) = (9 x 9):
##                   PC1          PC2         PC3         PC4        PC5
## clump      -0.3118196  0.123669719 -0.09320064  0.88457534 -0.1686176
## cellSize   -0.3952446 -0.001734926 -0.12143457 -0.00951957  0.1538036
## cellShape  -0.3899699  0.027217683 -0.13194673  0.04073577  0.1451925
## adhesion   -0.3368386 -0.200260293 -0.20282099 -0.33486419 -0.6004279
## seCellSize -0.3498276 -0.125299901  0.05728384 -0.11679758  0.6720307
## bNuclei    -0.2263319  0.793062624  0.50773008 -0.17792083 -0.1340930
## chromatin  -0.3542529  0.034464059 -0.29959152 -0.16258740 -0.2157494
## nNuclei    -0.3512024  0.009090677 -0.03986792 -0.15935602  0.1440764
## mitoses    -0.2420119 -0.545837889  0.75206728  0.08072126 -0.1714821
##                    PC6         PC7         PC8           PC9
## clump       0.03573302 -0.14696822 -0.21150408  0.0164936623
## cellSize    0.10761908  0.10945066  0.47314622  0.7468986072
## cellShape   0.06110760  0.08933507  0.60043278 -0.6593922038
## adhesion    0.34339185 -0.45576982 -0.08012116 -0.0242503320
## seCellSize  0.40455077 -0.12247633 -0.45867960 -0.0657531887
## bNuclei     0.10444370 -0.01478183 -0.03566343 -0.0003247543
## chromatin  -0.15074458  0.73834232 -0.37537548 -0.0424204924
## nNuclei    -0.81356982 -0.39526239 -0.10223215  0.0158993811
## mitoses    -0.07936992  0.18469790  0.03798345 -0.0104113788

# Manipulate data to show a percentage variance graph:
cancer_pca.percentage <-
  round(cancer_pca$sdev^2/sum(cancer_pca$sdev^2)*100, 1)

# Scree plot with the relative variances for each principle component:
barplot(cancer_pca.percentage, main='Scree Plot',
        xlab='Principle Component', ylab='Percent Variation')
```



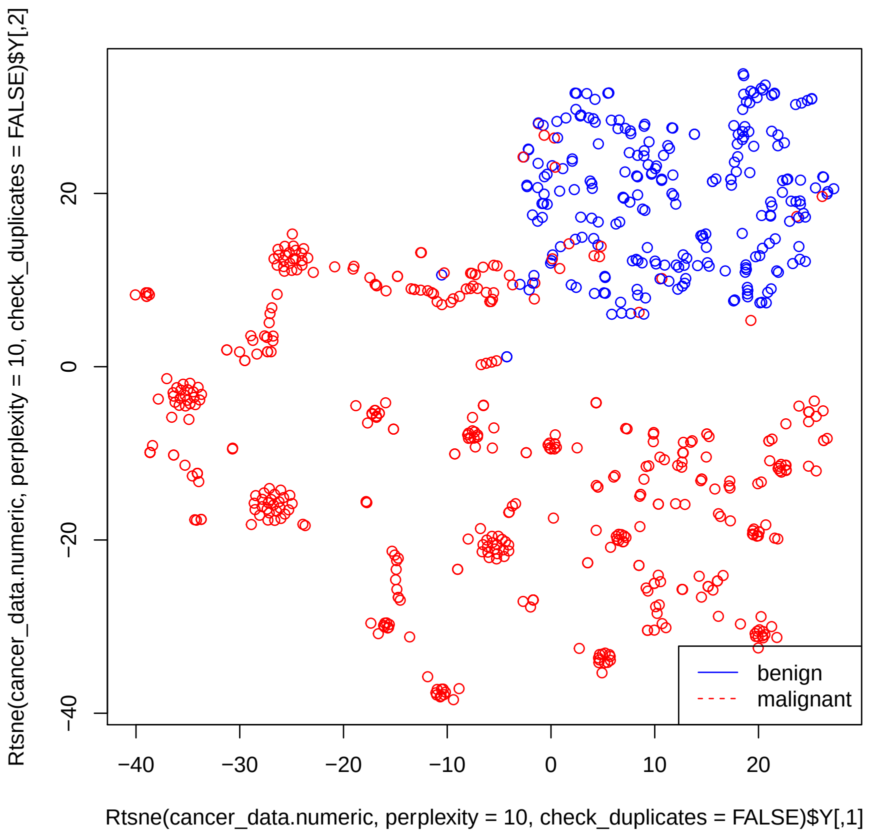
```r
# Add class info back into the data
cancer_pca$class <- cancer_data$class

# Plot data with color-coded groups
plot(cancer_pca$x[,1], cancer_pca$x[,2], main='Reduced Representation',
     xlab= paste('PC1 - ', cancer_pca.percentage[1], '%'),
     ylab = paste('PC2 - ', cancer_pca.percentage[2], '%'),
     col=cancer_pca$class)
legend('topleft', legend = c('benign', 'malignant'),
       col = c('blue', 'red'), lty = c(1, 2, 1))
```



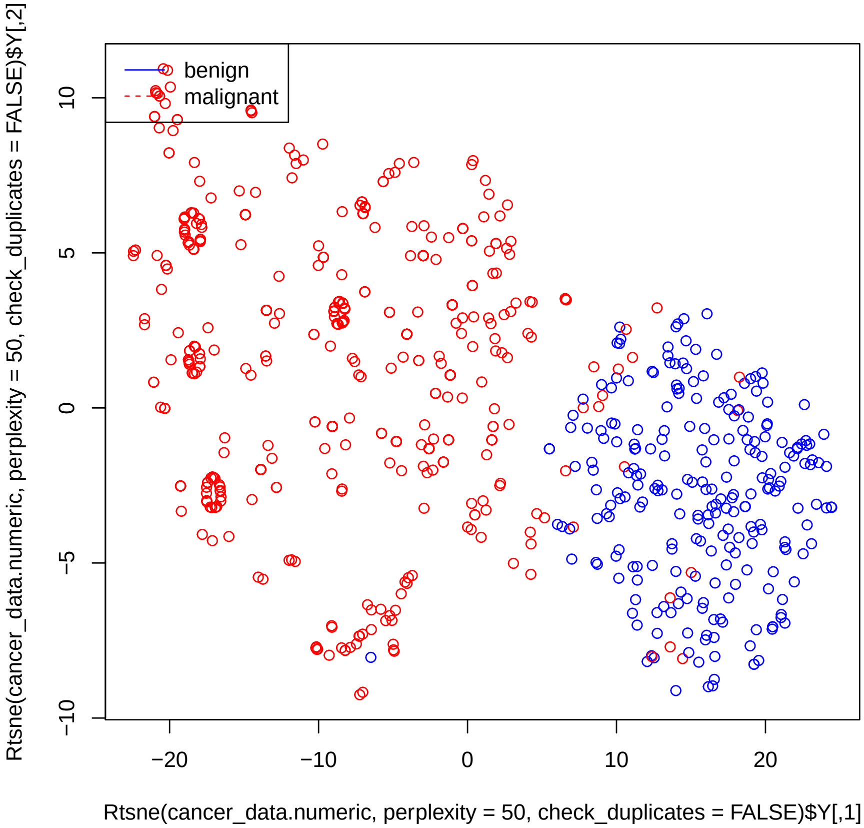
5 t-SNE Visualization

```
library("Rtsne", lib.loc="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
```

```
    # Plot data with color-coded groups
    # tSNE perplexity 10
    plot(Rtsne(cancer_data.numeric, perplexity=10,
          check_duplicates = FALSE)$Y, col=cancer_data$class)
    legend('bottomright', legend = c('benign', 'malignant'),
      col = c('blue', 'red'), lty = c(1, 2, 1))
```



```
# Plot data with color-coded groups
# tSNE perplexity 50
plot(Rtsne(cancer_data.numeric, perplexity=50,
      check_duplicates = FALSE)$Y, col=cancer_data$class)
legend('topleft', legend = c('benign', 'malignant'),
  col = c('blue', 'red'), lty = c(1, 2, 1))
```



These results are both so different from each other, and so different from the PCA results obtained in problem number 4 that I'm slightly concerned I'm doing something incorrectly! However, in all three examples, you can see that there are two clear groups - that malignant and benign tumors are clearly differentiable from each other.