Exercise 9

Nicole Navarro

October 3, 2016

1. Read the data into memory as a csv file

```
#get urls
data_url<-"https://archive.ics.uci.edu/ml/machine-learning-databases/hepatitis/hepatitis.data"
names_url<-"https://archive.ics.uci.edu/ml/machine-learning-databases/hepatitis/hepatitis.names"

#download files
#download.file(url,"~/school/fall 16/EDA/data/hepatitis_data.csv")
#download.file(names_url,"~/school/fall 16/EDA/data/hepatitis_names.txt")

#read into R
hepatitis_data<-read.csv("~/school/fall 16/EDA/data/hepatitis_data.csv", header=FALSE)
```

2. Name the features as described in the names file

```
colnames(hepatitis_data)<- c("class","age","sex","steroid","antivirals","fatigue","malaise","anorexia",
```

3. Write the result to memory as a csv file

```
write.csv(hepatitis_data,"~/school/fall 16/EDA/data/hepatitis_data_headers.csv",row.names=FALSE)
```

4. How many complete cases are there?

```
hepatitis_data[hepatitis_data == "?"] <- NA
#another way to do this is by levels(x)[levels(x)=="?"]<-na
#for each column that needs it
#this actually gets rid of ? as a factor completely so NA is just missing data

complete_cases<-(complete.cases(hepatitis_data))
#number of complete cases
sum(complete_cases)
```

```
## [1] 80
```

```
#percent of complete cases
sum(complete_cases)/nrow(hepatitis_data)
```

```
## [1] 0.516129
```

5. Subsetting the data on Age, Sex, Bilirubin, ALK, SGOT and Albumin, compute the number of missing values for the Bilirubin feature. Convert the last four features to numeric values. How many complete cases are there for the subsetted frame?

```
subset1<-hepatitis_data[c("age","sex","bilirubin","alk phosphate","sgot","albumin")]

#number of missing values for bilirubin
length(which(is.na(subset1[c("bilirubin")])))
```

```
## [1] 6
```

```
#convert factor variables to numeric
subset1[,3]<-as.numeric((as.character(subset1[,3])))
subset1[,4]<-as.numeric((as.character(subset1[,4])))
subset1[,5]<-as.numeric((as.character(subset1[,5])))
subset1[,6]<-as.numeric((as.character(subset1[,6])))

#number of complete cases
sum(complete.cases(subset1))
```

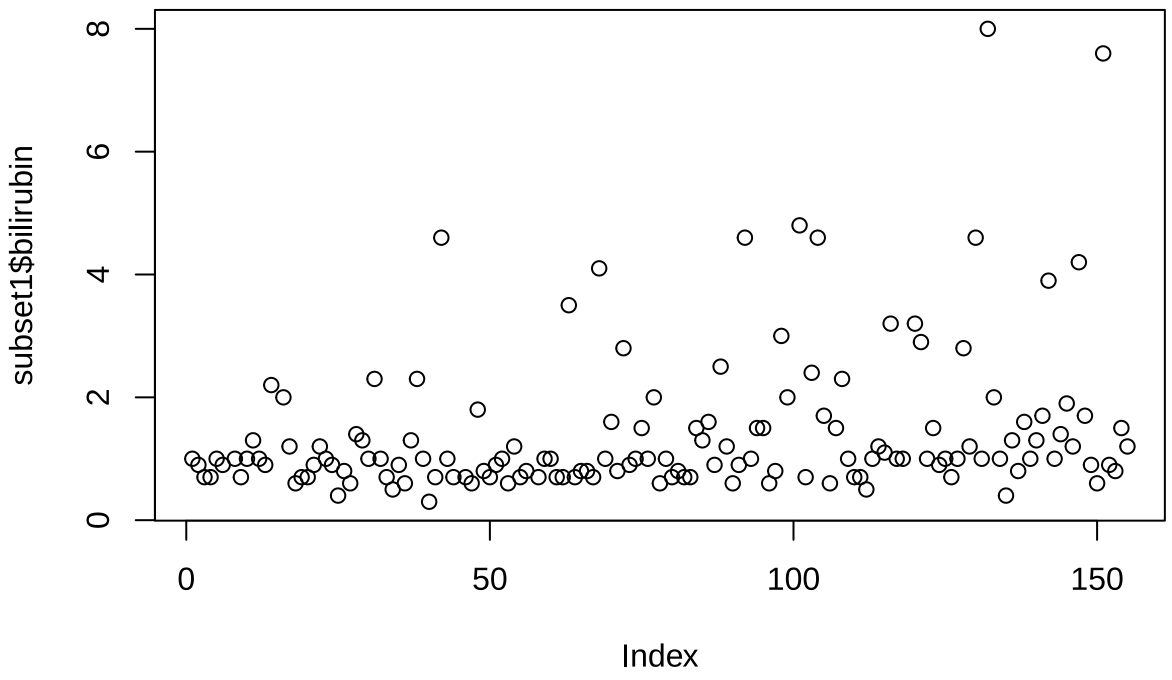
```
## [1] 120
```

```
#percent of complete cases
sum(complete.cases(subset1))/nrow(subset1)
```

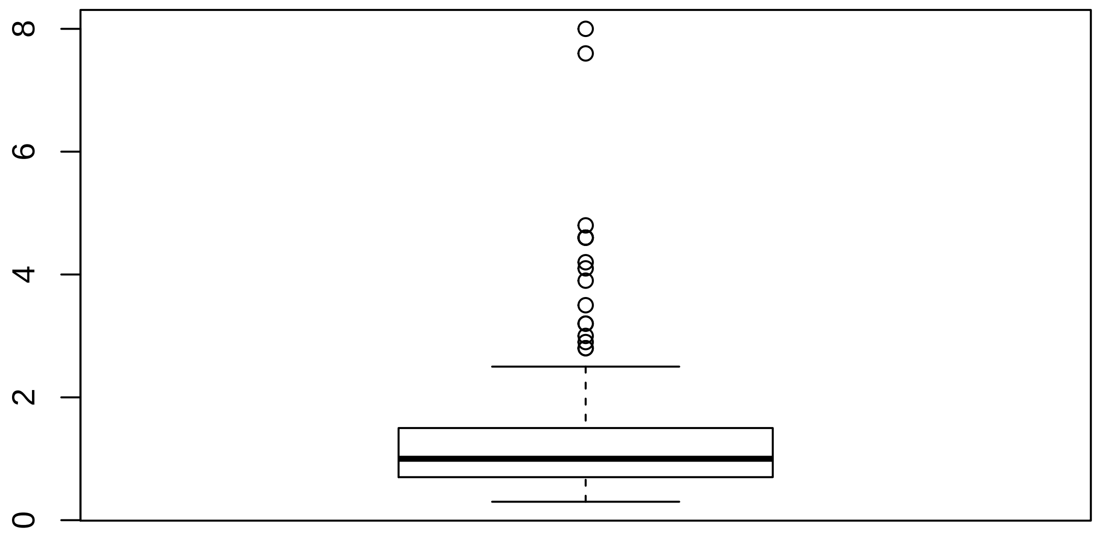
```
## [1] 0.7741935
```

6. Are there any outliers in the Bilirubin and Albumin entries?

```
plot(subset1$bilirubin)
```

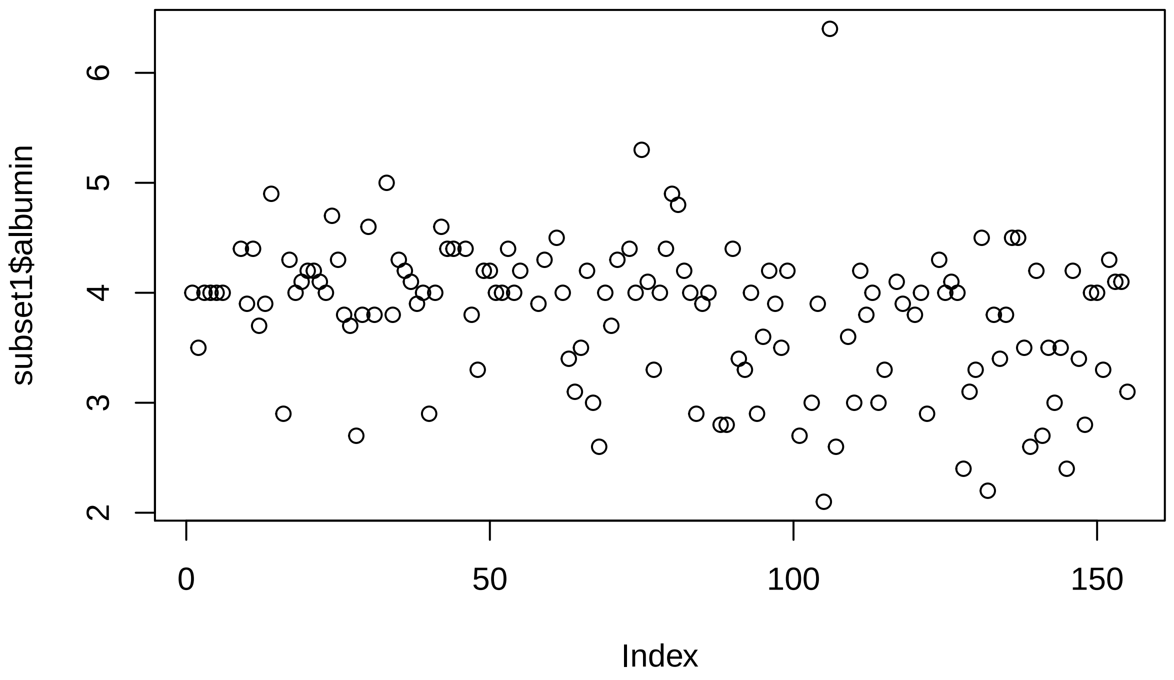


```r
boxplot(subset1$bilirubin)
```

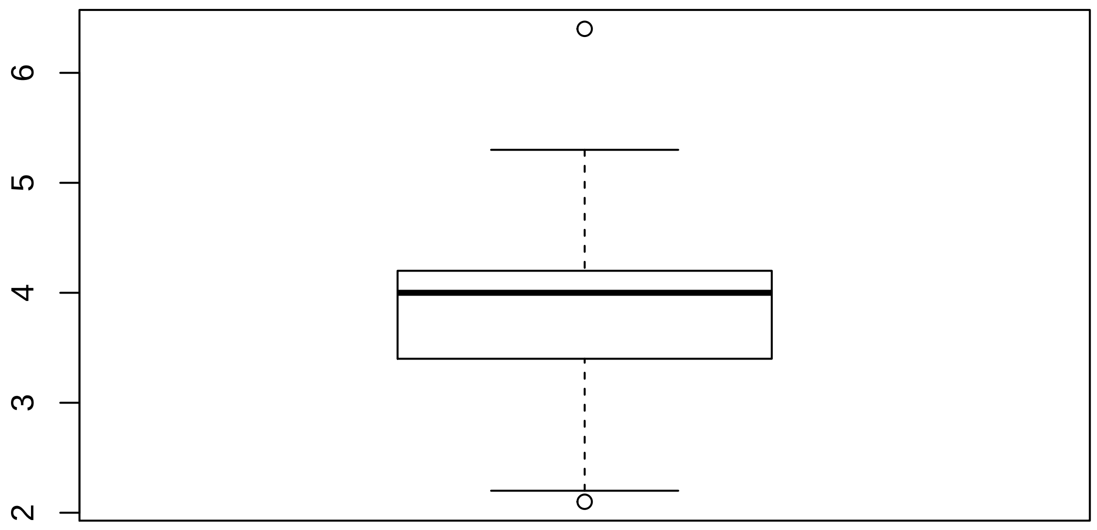


It appears there may be some outliers on the higher end of the data. After doing some research into Bilirubin, I believe that this variable is giving us total bilirubin levels. While normal total bilirubin levels are between 0 and 1.9, it is possible for levels to reach much higher than the highest results shown in our data. Bilirubin levels of close to 8 like the points that exist in our data are certainly possible, so despite looking like outliers, I am unable to truly determine if the high values shown in our data are outliers or relevant pieces of data without more information.

```r
plot(subset1$albumin)
```



```r
boxplot(subset1$albumin)
```



Normal Albumin levels are typically between 3.5 and 5.5. Based on the plots, I would guess that the high point of 6.4 is most likely an outlier. The low point of 2.1 may not necessarily be an outlier because there exist many data points below the typical levels.

7. Bin the age variables in units of decades

```
age_decades<-with(subset1,cut(age,breaks = c(0,10,20,30,40,50,60,70,80),include.lowest = TRUE))
subset1$decades<-age_decades
```

8. Aggregate the data to obtain mean readings for the last 4 variables as a function of sex and age, with age as a binned factor.

For sex: 1=male 2=female

```
aggregate(bilirubin ~ sex + decades,subset1,mean)
```

```
##   sex decades bilirubin
## 1   1  [0,10] 0.7000000
## 2   1 (10,20] 0.9500000
## 3   2 (10,20] 2.3000000
## 4   1 (20,30] 1.2458333
## 5   2 (20,30] 0.9200000
## 6   1 (30,40] 1.2086957
## 7    2 (30,40] 0.6500000
## 8    1 (40,50] 1.8580645
## 9    2 (40,50] 0.8666667
## 10   1 (50,60] 1.9150000
## 11   2 (50,60] 1.4500000
## 12   1 (60,70] 1.0428571
## 13   2 (60,70] 2.0000000
## 14   1 (70,80] 0.8500000
```

```
aggregate(subset1$`alk phosphate` ~ sex + decades,subset1,mean)
```

```
##    sex decades subset1$`alk phosphate`
## 1    1  [0,10]                256.0000
## 2    1 (10,20]                124.5000
## 3    2 (10,20]                150.0000
## 4    1 (20,30]                105.1875
## 5    2 (20,30]                100.8000
## 6    1 (30,40]                100.1000
## 7    2 (30,40]                 50.0000
## 8    1 (40,50]                102.2593
## 9    2 (40,50]                132.0000
## 10   1 (50,60]                102.3750
## 11   2 (50,60]                128.0000
## 12   1 (60,70]                104.0000
## 13   2 (60,70]                146.3333
## 14   1 (70,80]                105.5000
```

```
aggregate(sgot ~ sex + decades,subset1,mean)
```

```
##    sex decades      sgot
## 1    1  [0,10]  25.00000
## 2    1 (10,20] 135.00000
## 3    2 (10,20]  68.00000
## 4    1 (20,30]  77.73913
## 5    2 (20,30] 101.00000
## 6    1 (30,40]  77.48936
## 7    2 (30,40]  24.00000
## 8    1 (40,50]  97.78125
## 9    2 (40,50]  81.66667
## 10   1 (50,60]  93.04762
## 11   2 (50,60]  37.00000
## 12   1 (60,70] 111.00000
## 13   2 (60,70] 120.33333
## 14   1 (70,80]  42.00000
```

```
aggregate(albumin ~ sex + decades,subset1,mean)
```

```
##    sex decades  albumin
## 1    1  [0,10] 4.200000
## 2    1 (10,20] 3.450000
## 3    2 (10,20] 3.900000
## 4    1 (20,30] 4.218182
## 5    2 (20,30] 3.920000
## 6    1 (30,40] 3.847727
## 7    2 (30,40] 4.050000
## 8    1 (40,50] 3.578571
## 9    2 (40,50] 4.200000
## 10   1 (50,60] 3.700000
## 11   2 (50,60] 3.400000
## 12   1 (60,70] 3.700000
## 13   2 (60,70] 3.400000
## 14   1 (70,80] 3.700000
```

9. Sort the data on the Bilirubin columns (ascending)

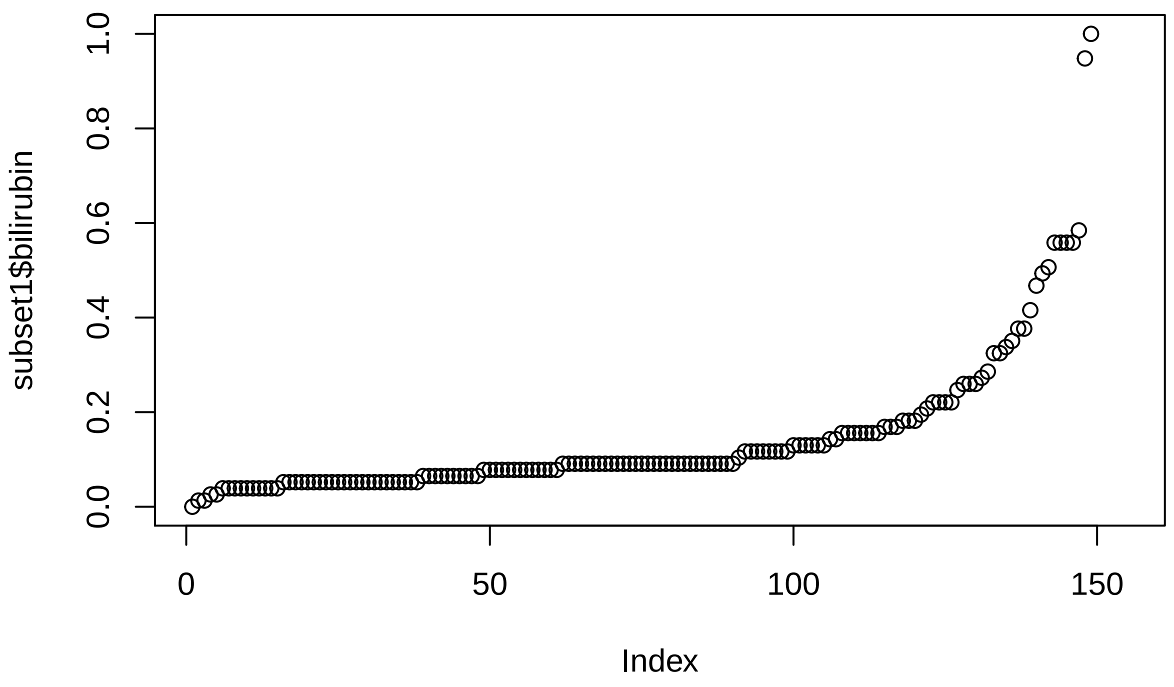
```
subset1<-subset1[order(subset1$bilirubin,decreasing=FALSE),]
```

10. Standardize Bilirubin and Albumin and plot the outcome as a scatterplot

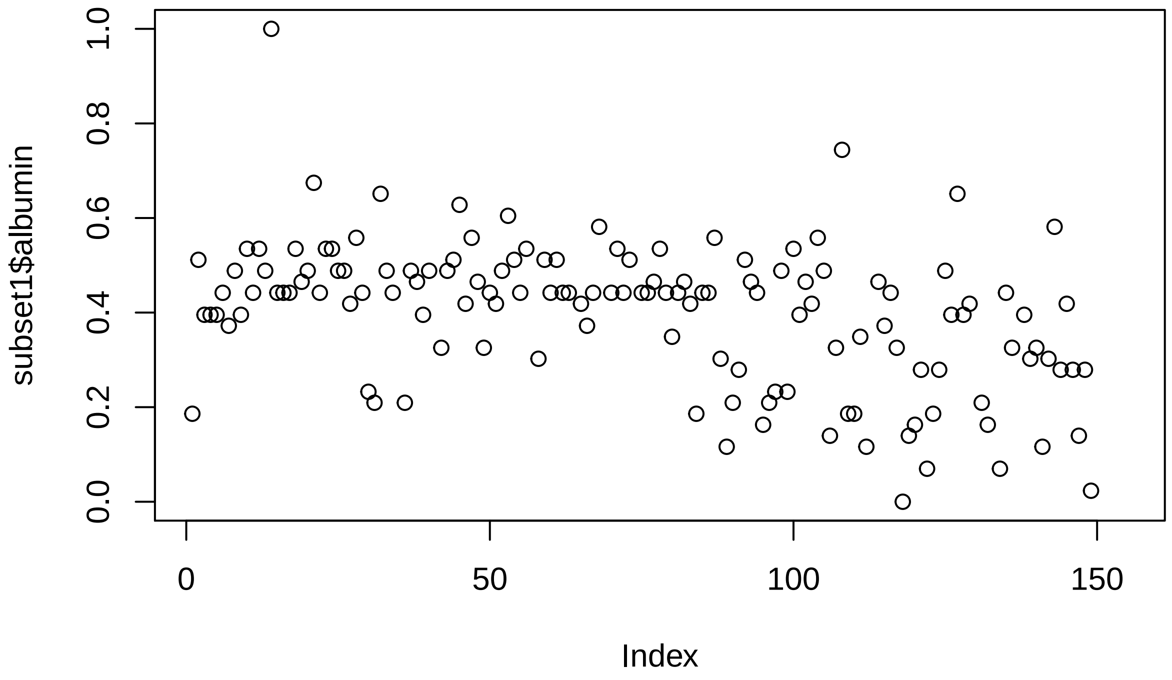
```
standard_trans<-function(x) {
  M <- max(x, na.rm=TRUE)
  m <- min(x, na.rm=TRUE)
  Y <- (x-m)/(M-m)
  return(Y)
}

#standardize the data
subset1$bilirubin <- standard_trans(subset1$bilirubin)
subset1$albumin <- standard_trans(subset1$albumin)

#plot
plot(subset1$bilirubin)
```



```r
plot(subset1$albumin)
```



11. Consider the data frame consisting of the complete cases for the variables Bilirubin, ALK, SGOT and Albumin. What fraction of the variance does the first principal component account for?

```
clean_subset <- subset1[which(complete.cases(subset1[,3:6])),c("age","sex","bilirubin","alk phosphate",

test<- subset1[complete.cases(subset1[,3:6]),3:6]

#PCA = principal component analysis
#may need to also normalize data???
#log transform gets rid of skew -- (can't have 0s) helps with PCA and other machine learning algorithms

#stats package -- function prcomp
hep_pca <- prcomp(test)
summary(hep_pca)
```

```
## Importance of components:
##                            PC1     PC2   PC3    PC4
## Standard deviation     82.2130 50.1399 0.134 0.1079
## Proportion of Variance  0.7289  0.2711 0.000 0.0000
## Cumulative Proportion   0.7289  1.0000 1.000 1.0000
```

12. Subsetting the data on Age, Sex, Steroid and Antivirals columns and join the resulting data frame with the data frame of complete cases for Age, Sex, Bilirubin, ALK, SGOT and Albumin. What are the dimensions of the resulting frame?

```
clean_data <- merge(clean_subset[,1:6],hepatitis_data[,c("age","sex","steroid","antivirals","bilirubin"
dim(clean_data)
```

```
## [1] 120   8
```