STAT158_Midterm

Oliver Shanklin

March 11, 2019

1)

a)

Read the data into R. State the names of the columns in this dataset, and report what "class" R assigned to each one.

```
stat_class_data <- read.csv("Stat_class_data.csv", header=TRUE)

colnames(stat_class_data)
```

```
## [1] "Sex"    "Height" "Color"  "Number"
```

The classes for each column are:

```
class(stat_class_data$Sex)
```

```
## [1] "factor"
```

```
class(stat_class_data$Height)
```

```
## [1] "integer"
```

```
class(stat_class_data$Color)
```

```
## [1] "factor"
```

```
class(stat_class_data$Number)
```

```
## [1] "numeric"
```

b)

How many females are in this data set? How many males?

```
sum(stat_class_data$Sex == "F")
```

```
## [1] 73
```

```
sum(stat_class_data$Sex == "M")
```

```
## [1] 60
```

There are 73 females and 60 males in this data set.

c)

Get R to tell you all the values that favorite color takes on. What are the three most popular favorite colors?

```
levels(stat_class_data$Color)
```

```
##  [1] "Black"  "Blue"   "Brown"  "Green"  "Orange" "Purple" "Red"
##  [8] "Teal"   "White"  "Yellow"
```

```
colors_data <- levels(stat_class_data$Color)

n <- length(stat_class_data$Color)
color_proportion <- numeric()


for(i in seq(1:length(colors_data))){
  color_proportion[i] <- sum(stat_class_data$Color == colors_data[i])
}
color_proportion <- color_proportion / n
color_proportion
```

```
##  [1] 0.015037594 0.390977444 0.022556391 0.203007519 0.060150376
##  [6] 0.142857143 0.060150376 0.052631579 0.007518797 0.045112782
```

So the 3 most popular colors are Blue, Green, and Purple.

d)

Have R give you summary statistics for height. What is the value for median height?

```
summary(stat_class_data$Height)
```

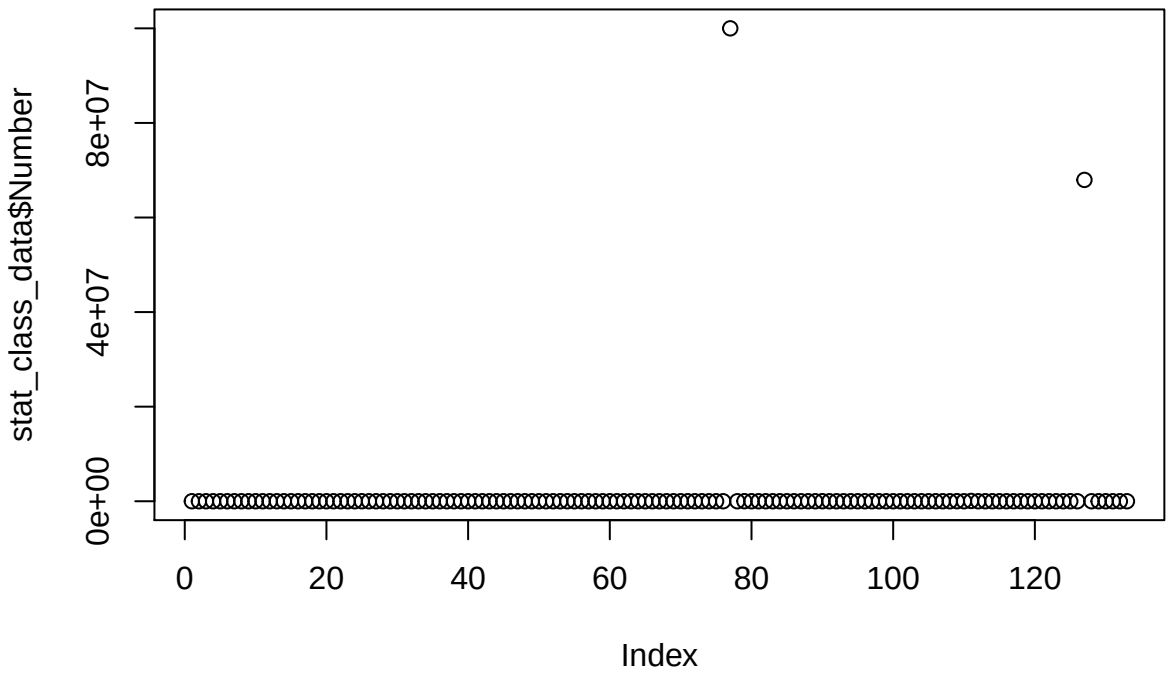
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.00   66.00   68.00   68.35   71.00   79.00
```

The median height is 68. And I will assume that is inches.

e)

Make a plot of "number". Describe what this plot is showing you.

```
plot(stat_class_data$Number)
```



This plot is showing the number on the y-axis and the position in the vector on the x-axis. So, since there are 2 very large numbers in the data set, it is hard to see all the smaller values on a plot.

f)

Get R to calculate how many values for "number" are greater than 5000. What proportion of students is this?

```
sum(stat_class_data$Number > 5000)
```

```
## [1] 7
```

```
sum(stat_class_data$Number > 5000) / length(stat_class_data$Number)
```

```
## [1] 0.05263158
```

There are 7 numbers greater than 5000, so the proportion is 0.052631.

g)

Rows 77 and 127 contain extreme outliers for the "number" variable. Use R to create a new variable set that excludes these rows but contains the values of "number" for all the other rows. Report the mean of number with and without these two extreme outliers included.

```
numbers_removed <- stat_class_data$Number[-c(77,127)]

mean(stat_class_data$Number)
```

```
## [1] 1263612
```

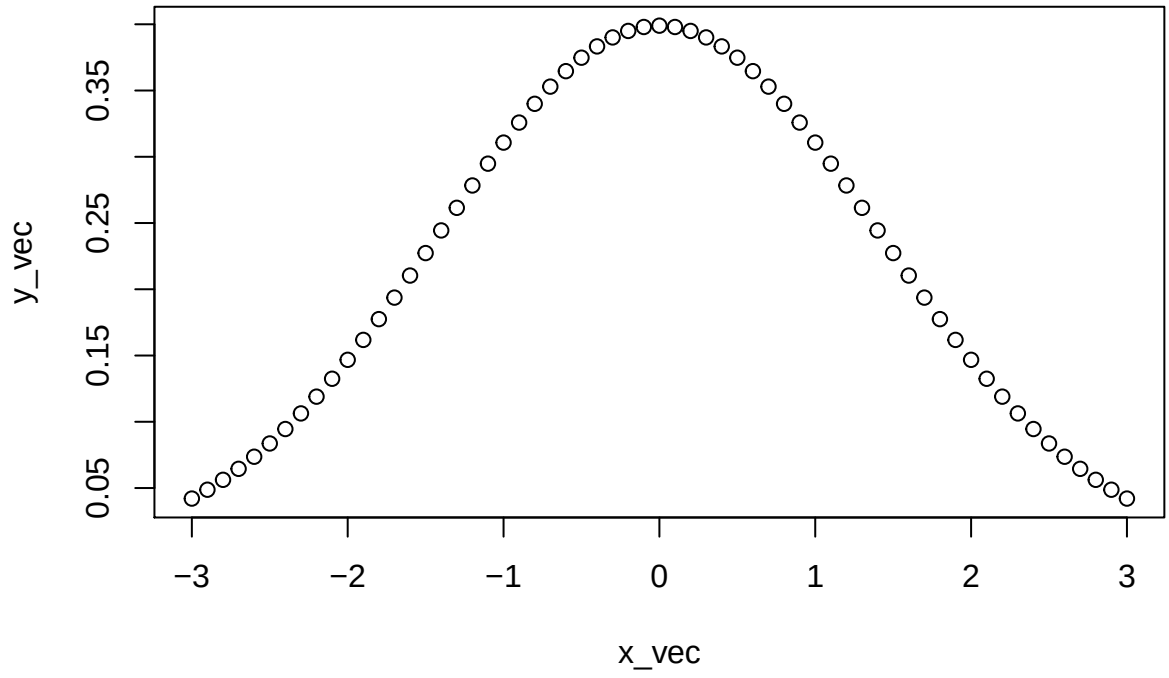
```r
mean(numbers_removed)
```

```
## [1] 841.4564
```

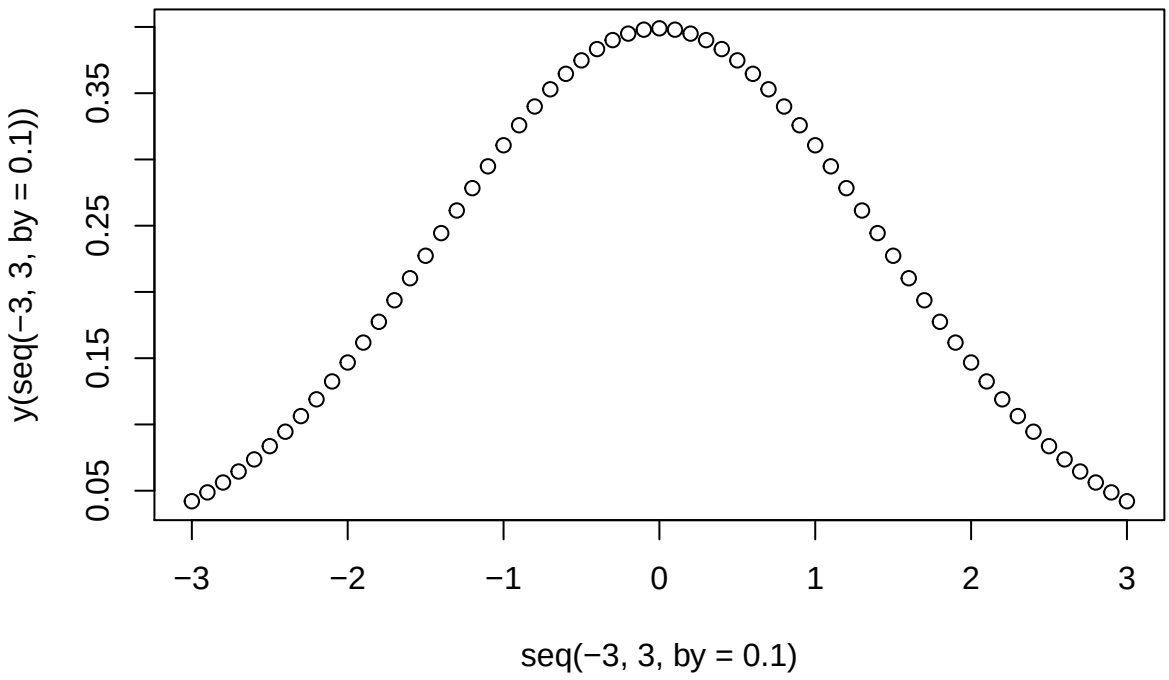
So, with the outliers, the mean is 1263612, and the mean without the outliers is 841.4564.

2)

```r
# One way
x_vec <- seq(-3,3,0.1)
y_vec <- (1/sqrt(2*pi))*exp(-(x_vec/2)^2)

plot(x_vec, y_vec)
```



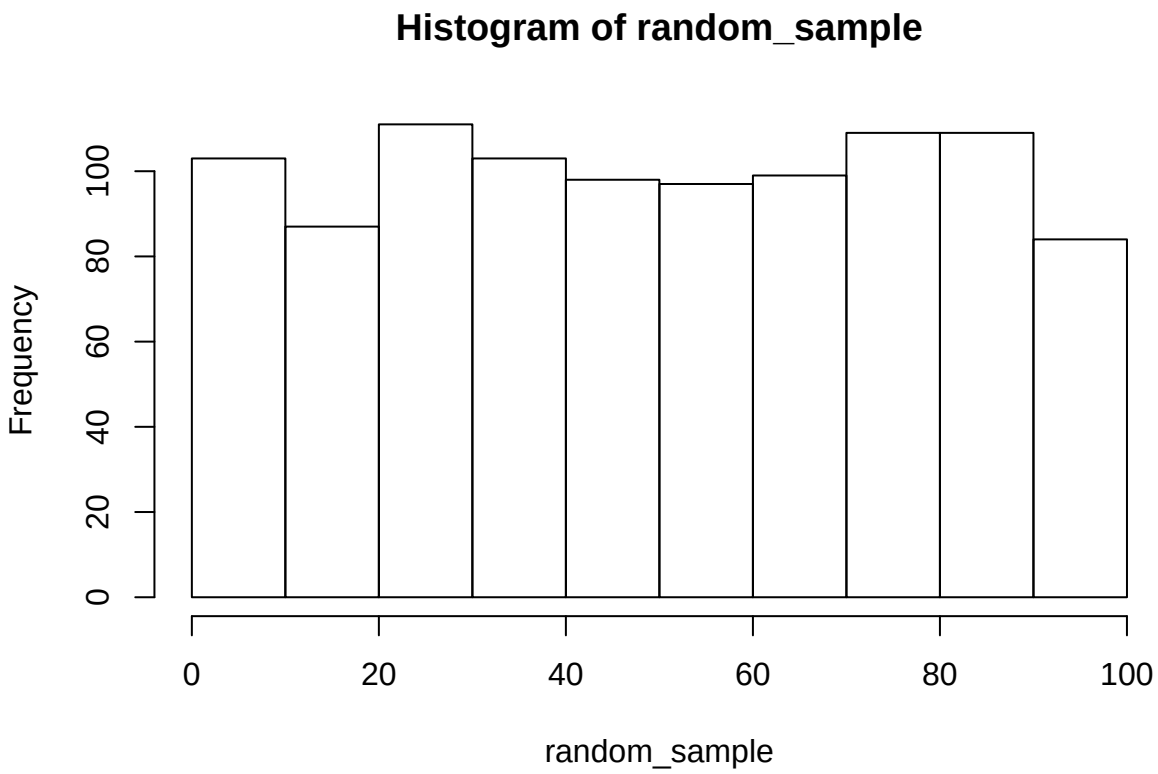

```r
# Another way using function
y <- function(x){
  (1/sqrt(2*pi))*exp(-(x/2)^2)
}

plot(seq(-3,3, by = .1), y(seq(-3,3,by=.1)))
```



3)

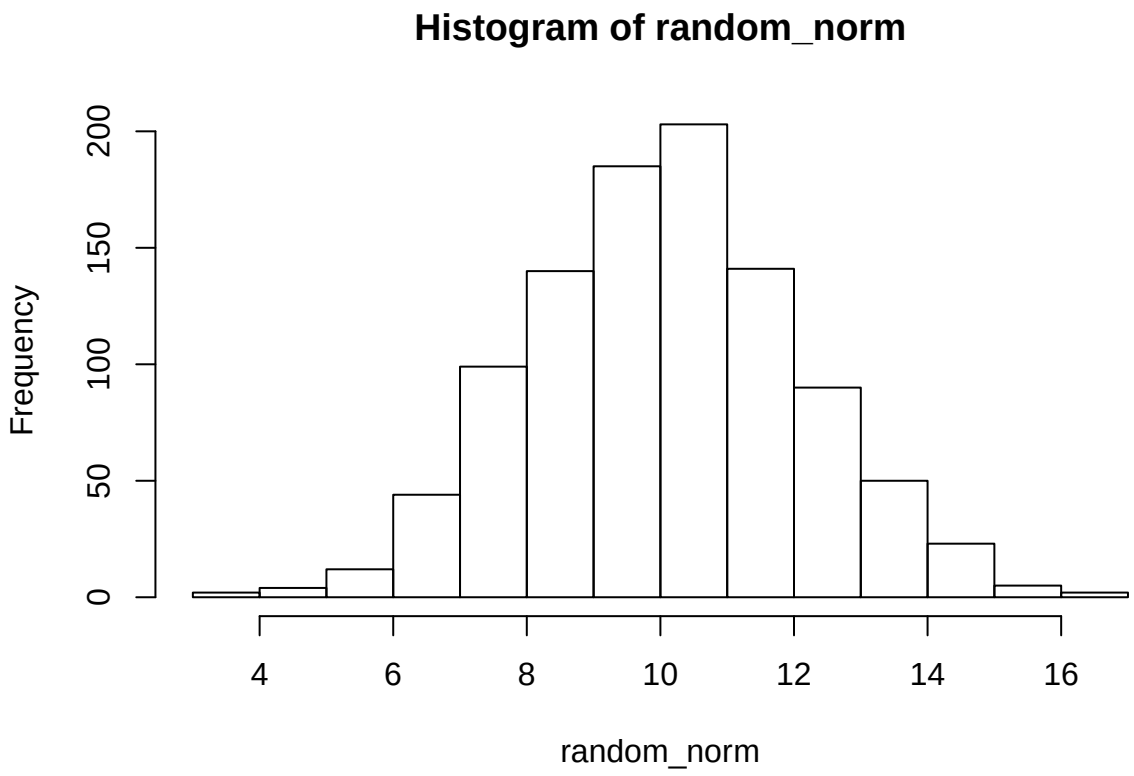
a)

```
random_sample <- sample(seq(1,100), 1000, replace = T)

hist(random_sample)
```



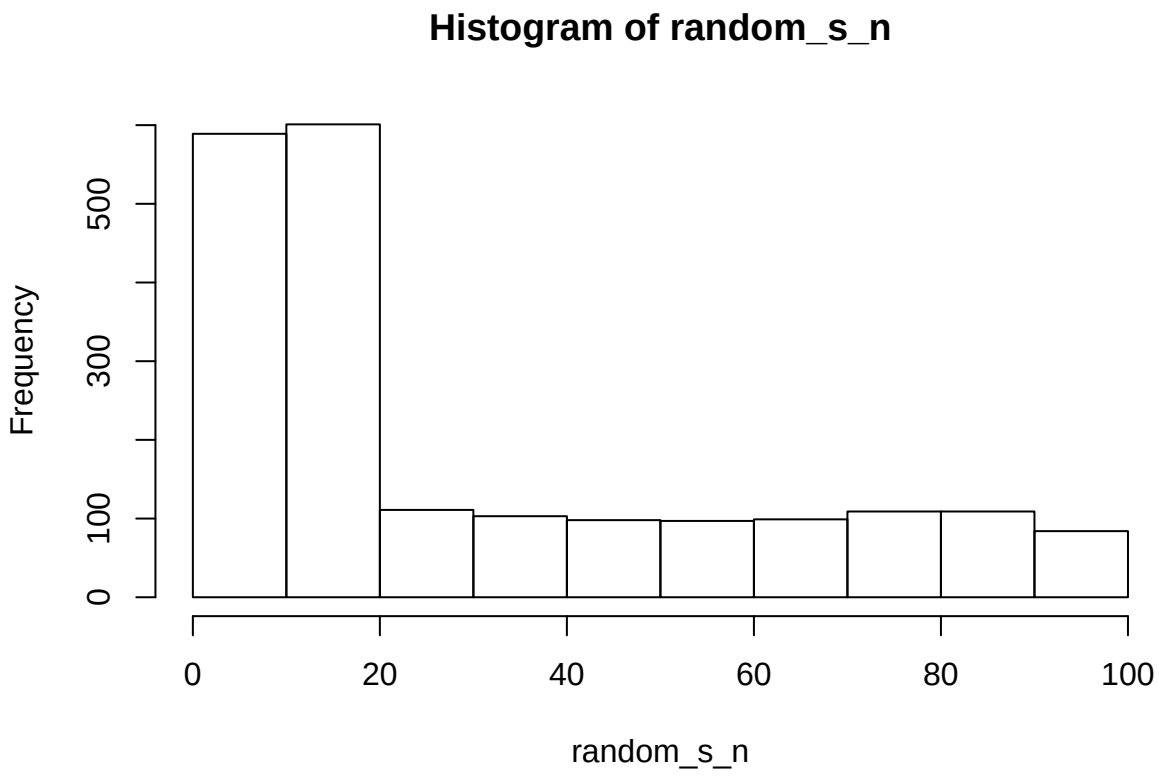
b)

```
random_norm <- rnorm(1000, mean = 10, sd = 2)

hist(random_norm)
```



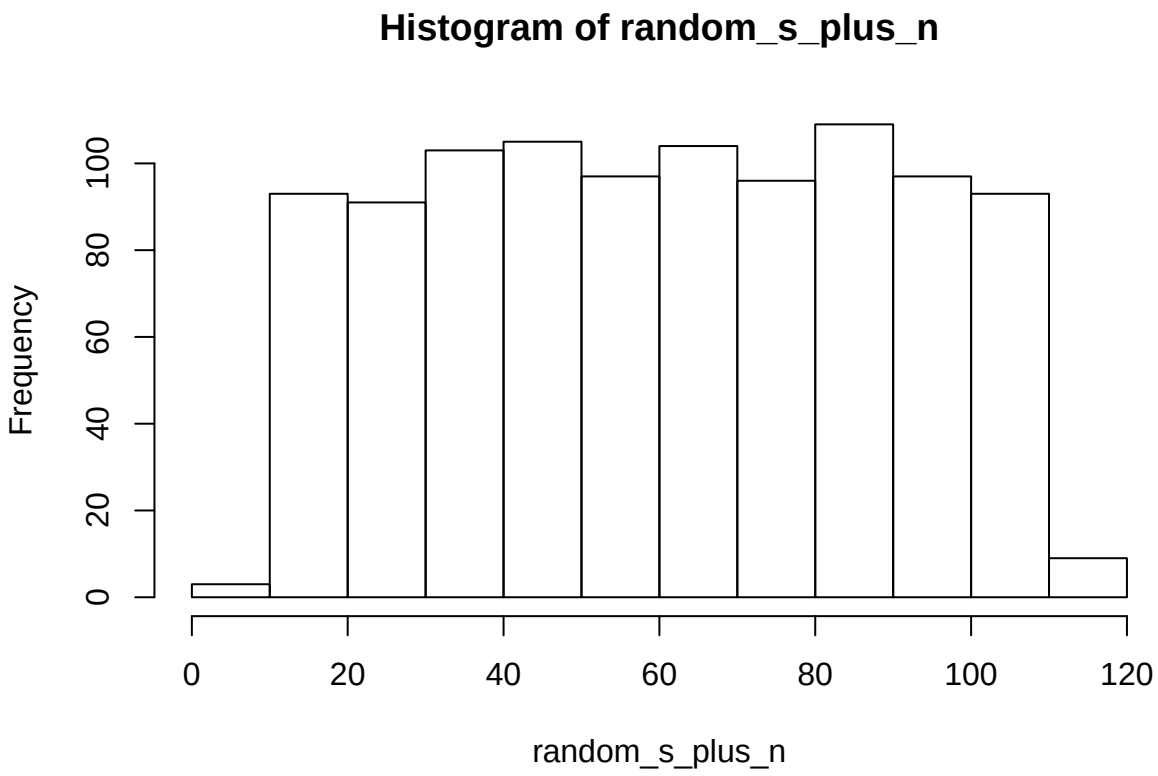
C)

```
random_s_n <- c(random_norm, random_sample)

hist(random_s_n)
```



d)

```
random_s_plus_n <- random_norm + random_sample

hist(random_s_plus_n)
```



e)

These histograms look very different. In part c, since we are just making a new vector with both vectors concatinated, it has a skew right, because there are a lot of data from the normal vector around the mean, 10. In part d, the vectors are being added together, so the maximum changes and we see a relitively uniform histogram, not including the end classes.