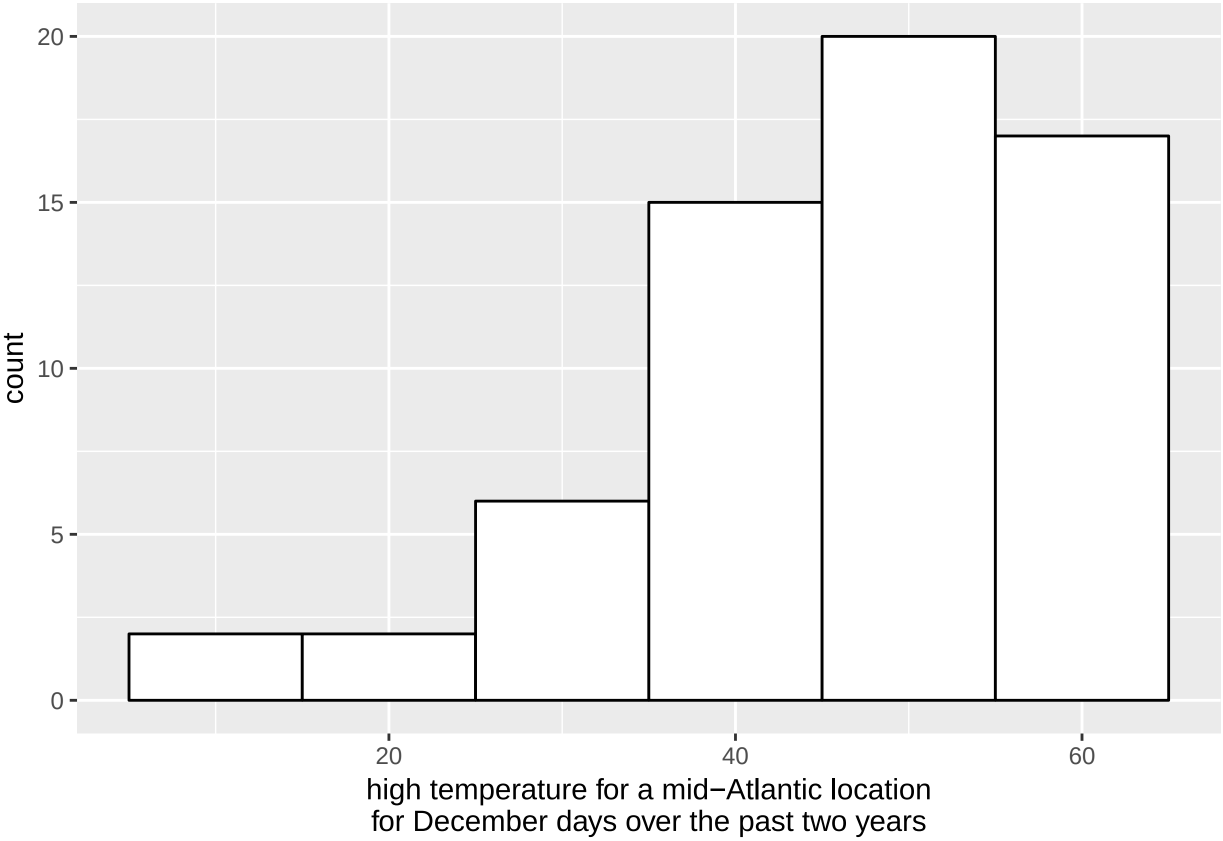
Homework 7

Max Ryoo

Problem 1

Part A

```
set.seed(12181998)
library(ggplot2)
setwd("/Users/maxryoo/Documents/Fall 2018/STAT3080/HW8")
weather = read.table("weather.txt", header = FALSE)
baseplot <- ggplot(weather, aes(x=V1))
plot <- baseplot + geom_histogram(binwidth=10,
                                  fill="white",
                                  color="black")
final <- plot +labs(
x="high temperature for a mid-Atlantic location
for December days over the past two years")
print(final)
```



I

read in the text file as a table then used the values in it to make a basic ggplot. I then made a histogram with this data with the binwidth being 10 (degrees). I also added a descriptive label. After graphing this plot we can see that the plot is skewed to the left.

Part B

As we can see the figure is skewed to the left, which means the mean will be heavily affected by thsee values. Mean is a numerical summary that can be heavily influeneced by outlier points meanwhile median takes into consideration these points or is not affected by these outliers. We can thus say that the median is a better measure for this data.

Part C

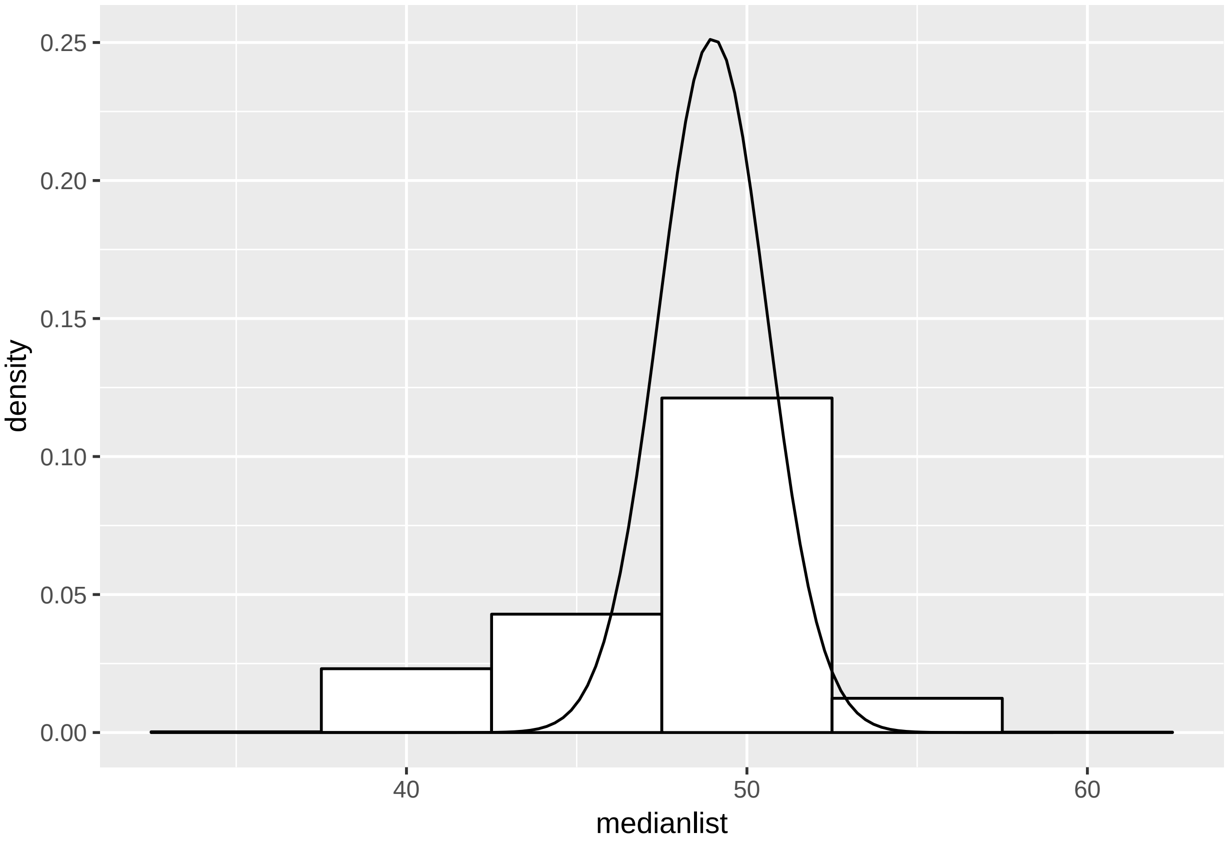
```
samps <- replicate(10000, sample(weather$V1, 15))
medianlist <- apply(samps,2,median)
meanlist <- mean(medianlist)
sd <- sd(medianlist)
ans <- c(meanlist, sd)
print(ans)
```

```
## [1] 48.27990  3.65542
```

I replicated taking the sample 15 times 10,000 times. I then took the median then took the mean of the list. i also got the sd of the list and concated them into another vector to display the ans.

Part D

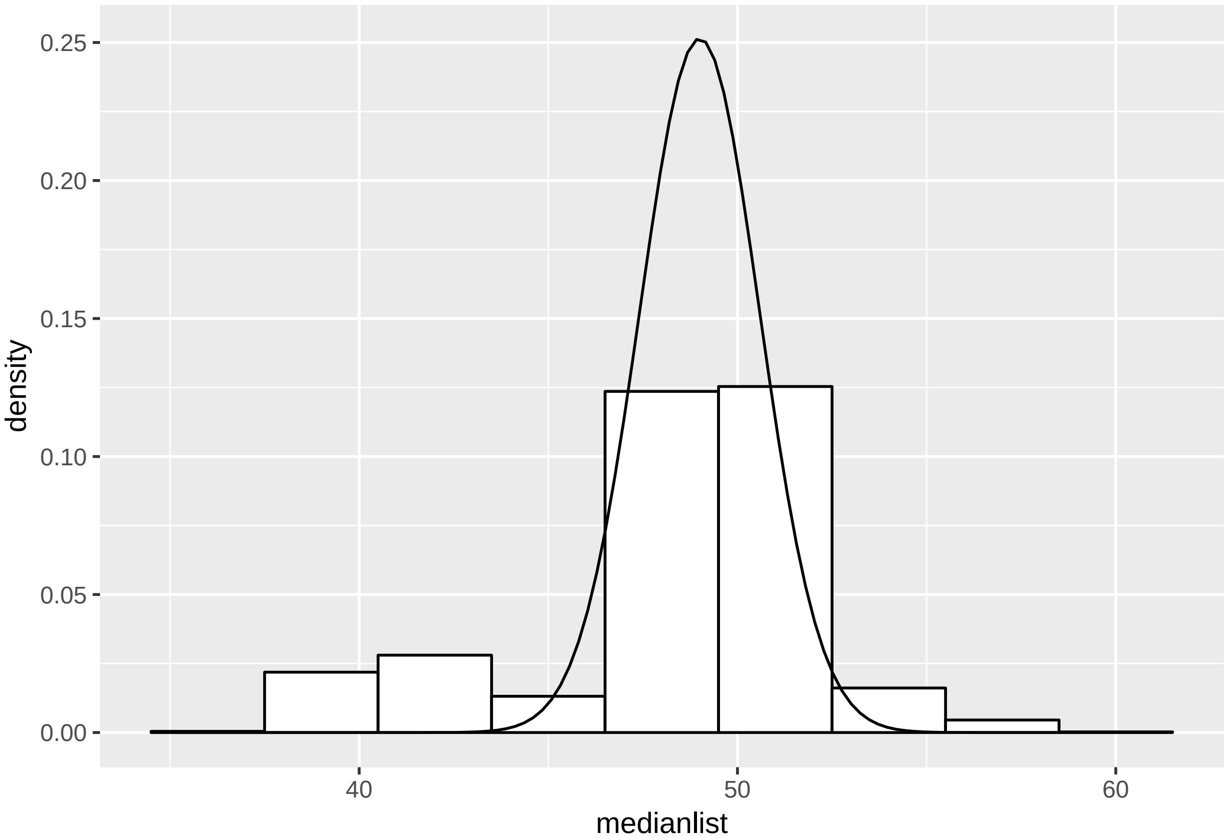
```
graph <- ggplot(as.data.frame(medianlist), aes(x=medianlist)) +
  geom_histogram(binwidth = 5, fill="white",colour="black",aes(y=..density..)) +
  stat_function(fun = dnorm, args = list(mean = median(weather$V1),
                                         sd=sd(weather$V1)/
                                           sqrt(length(weather$V1))))
print(graph)
```



I first made a base plot with the x being the midianlist. I then made a histogram with binwidth of 5 and white fill and black color. On this histogram I fitted a normal density curve using the stat_function().

Part E

```
graph <- ggplot(as.data.frame(medianlist),
                aes(x=medianlist)) +
  geom_histogram(binwidth = 3,
                 fill="white",colour="black",
                 aes(y=..density..)) +
  stat_function(fun = dnorm,
                args = list(mean = median(weather$V1),
                            sd=sd(weather$V1)/
                              sqrt(length(weather$V1))))
print(graph)
```



I

repeated the procedure on part D with binwidth of 3 instead of 5.

Part F

The hisogram is clearly not symmetric. Also, the sample size is too small to hold the CLT for this example.

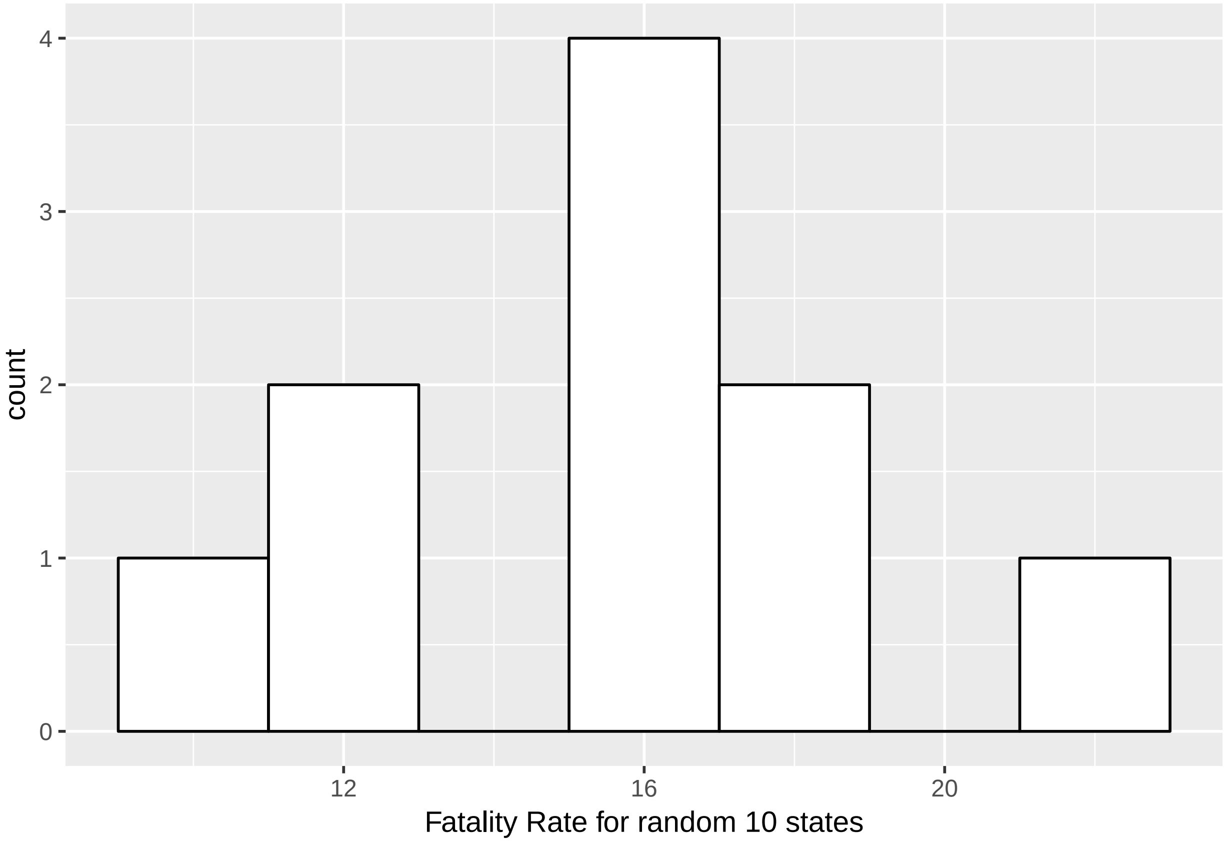
Problem 2

Part A

```
fatal = read.csv("fatality rates.csv", header = TRUE)
fatal
```

```
##         State Fatality.rate
## 1    Florida         15.40
## 2    Georgia         15.07
## 3   Oklahoma         17.41
## 4   Kentucky         18.80
## 5       Utah          9.21
## 6  Louisiana         16.17
## 7    Alabama         21.34
## 8  Tennessee         15.65
## 9   Delaware         12.50
## 10    Alaska         11.32
```

```
baseplot2 <- ggplot(fatal, aes(x=fatal$Fatality.rate))
plot <- baseplot2 + geom_histogram(binwidth=2, fill="white", color="black")
final <- plot + labs(x="Fatality Rate for random 10 states")
print(final)
```



I first read in the csv as fatal then made a base plot of this data witht he x axis being fatality rate. I then made a histogram with the binwidth being 2, which means 2 percent. I then added a descriptive title for the x axis. Although hard to make an argument since there is a small sample size, we can still say that the general shape is normally distributed.

Part B

I would still say that it is normally distributed.

Part C

```
samp_mean <- mean(fatal$Fatality.rate)
boot_samp <- replicate(10000,
                       sample(fatal$Fatality.rate, replace=TRUE))
boots_means <- apply(boot_samp,2,mean)
boots_err <- boots_means - samp_mean
boots_err_sort <- sort(boots_err)
p1 <- 10000*0.05
p90 <- 10000*0.95
boot_ci <- samp_mean - boots_err_sort[c(p90,p1)]
boot_ci
```

```
## [1] 13.554 17.059
```

I created samp_mean which holds the mean of the fatality rate column. I then used teh sample function with a replication of 10,000 times. I then applied the mean function to get a boots_means function. I then got the error between boots_means - samp_mean, which I also sorted. I then found the confidence interval.

Part D

The true average does not fall within this confidence interval, which means that the sample is not a good representation of the population.

Problem 3

Part A

```
flights = read.table("flights.txt", header = FALSE)
p95 <- quantile(flights$V1, 0.95)
flights_95 <- flights[flights < p95]
fbootsamps <- replicate(10000,
                        sample(flights_95, replace=TRUE))
fmeanboot <- apply(fbootsamps,2,mean)
fbooteer <- fmeanboot - mean(flights_95)
sortedfbooterr <- sort(fbooteer)
fp1 <- 10000*0.005
fp99 <- 10000*0.995
flights_bootCI <- mean(flights_95) - sortedfbooterr[c(fp99,fp1)]
print(flights_bootCI)
```

```
## [1] 41.41379 49.60345
```

I conducected a bootstrap test with the textfile fo flights. I took 10000 samples and got the mean. Used the similar procedure to the previous question but instead used 10000*0.005 and 10000*0.995 as the boundaries since the interval value changed

Part B

The endpoint is the the higher number of the interval given above in part A. The number of passengers in a plane will be less than this upper bound provided. It is more likely that the flights will not be overbooked due to this probability.

Part C

The endpoint is less than 55, which means that the algorithm did not work.

References

1. https://www.itl.nist.gov/div898/handbook/eda/section3/eda3672.htm
2. https://www.statisticshowto.datasciencecentral.com/standardized-test-statistic/