STAT 3675Q Homework 4

Due date: **Thursday, September 25, at noon**

Zeshi Feng

Note:

- Ensure that your code is fully visible in the PDF and not cropped. If needed, break the code into multiple lines to fit.
- It is recommended to write descriptive answers outside of R code chunks (i.e., as text in the main body), while comments within the code chunks can be reserved for brief code annotations.
- In all homework questions, include a written explanation of any output to earn full credit.

Question 1 [30 points]

Use the Pima data at http://heather.cs.ucdavis.edu/FasteR/data/Pima.csv

a. Take a look at the first few rows of the dataset and its dimension.

Answer:

```
pima <- read.csv("http://heather.cs.ucdavis.edu/FasteR/data/Pima.csv", header = TRUE)
head(pima)
```

```
##   pregnant glucose diastolic triceps insulin  bmi diabetes age test
## 1        6     148        72      35       0 33.6    0.627  50    1
## 2        1      85        66      29       0 26.6    0.351  31    0
## 3        8     183        64       0       0 23.3    0.672  32    1
## 4        1      89        66      23      94 28.1    0.167  21    0
## 5        0     137        40      35     168 43.1    2.288  33    1
## 6        5     116        74       0       0 25.6    0.201  30    0
```

```
dim(pima)
```

```
## [1] 768   9
```

b. Find the frequencies of different glucose values. Hint: use `table()`. How many women had glucose = 68? How many women had glucose = 0?

Answer:

```
glucose_freq <- table(pima$glucose)
glucose_freq["68"]
```

```
## 68
##  3
```

```
glucose_freq["0"]
```

```
## 0
## 5
```

c. Presumably a zero glucose level is not physiologically possible. Define a version of the glucose data that excludes the 0s and save it as a vector named `pg1`. Do not change the original data frame.

Answer:

```
pg1 <- pima$glucose[pima$glucose != 0]
head(pg1)
```

```
## [1] 148  85 183  89 137 116
```

d. Modify the glucose variable in the dataframe by recoding 0s as NAs.

Answer:

```
pima$glucose[pima$glucose == 0] <- NA
```

e. Verify that we now have 5 NAs in the glucose variable. Hint: Use `sum()` and `is.na()`.

Answer:

```
sum(is.na(pima$glucose))
```

```
## [1] 5
```

f. Check the mean of the above variable.

Answer:

```
mean_glucose <- mean(pima$glucose, na.rm = TRUE)
mean_glucose
```

```
## [1] 121.6868
```

Question 2 [40 points]

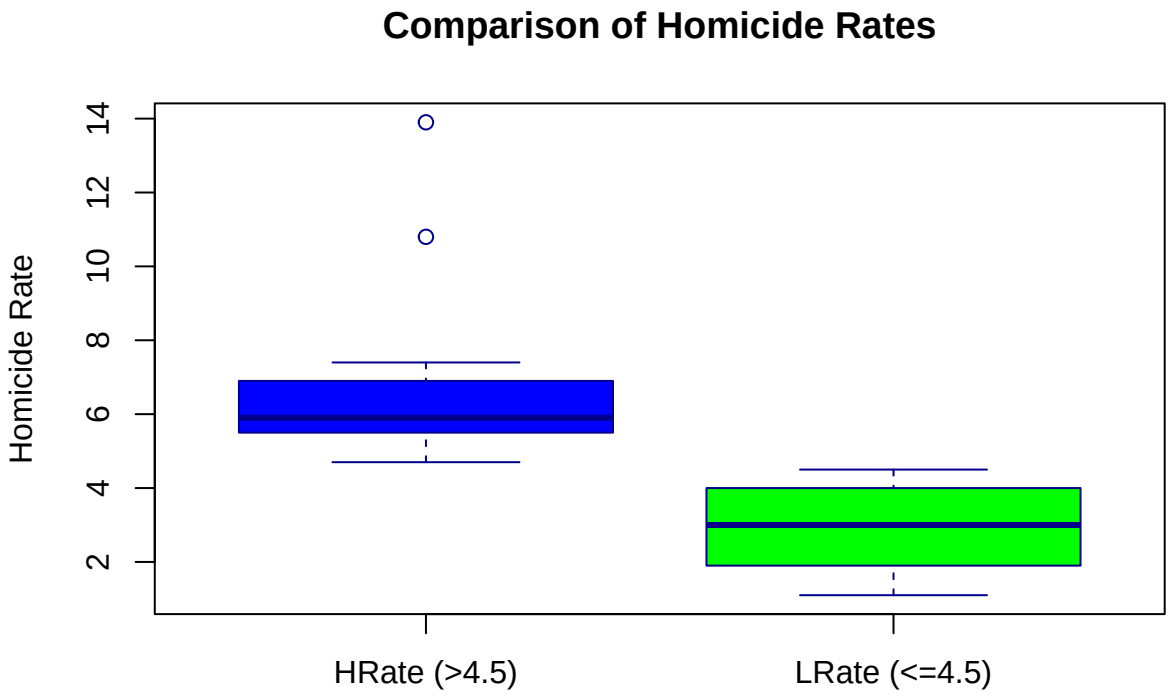
Reconsider the **gunData** data frame created in Quiz 2.

a. Create two numerical vectors in the global environment: **HRate** which contains all the `Homicide.rate` values that are greater than 4.5, and **LRate** which contains all

remaining `Homicide.rate` values. Produce a side-by-side boxplot comparing HRate and LRate. Use at least two more options in the `boxplot()` function to improve the plot. Hint: To draw a comparative boxplot for vectors `x` and `y`, use `boxplot(x, y)`.

Answer:

```
gun <- read.csv("gun.csv", header = TRUE)
HRate <- gun$Homicide.rate[gun$Homicide.rate > 4.5]
LRate <- gun$Homicide.rate[gun$Homicide.rate <= 4.5]

boxplot(HRate, LRate,
        names = c("HRate (>4.5)", "LRate (<=4.5)"),
        col = c("blue", "green"),
        main = "Comparison of Homicide Rates",
        ylab = "Homicide Rate",
        border = "darkblue",
        notch = FALSE)
```



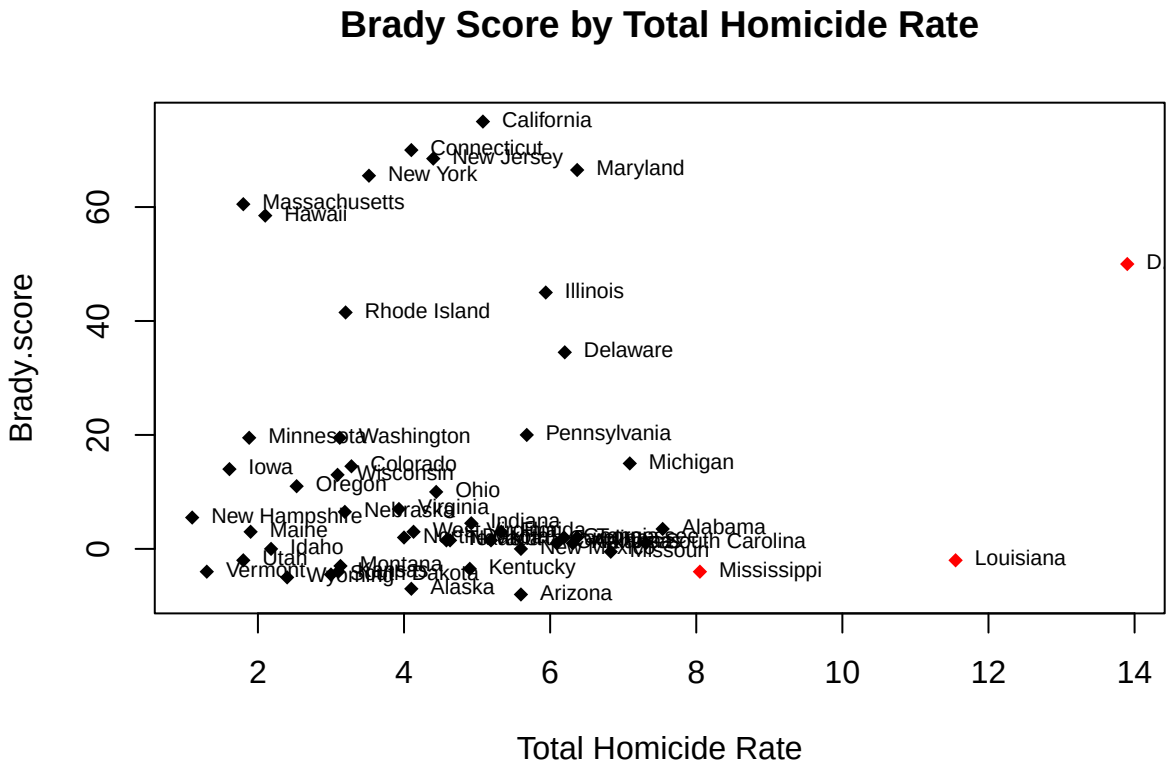
b. Produce a scatterplot of variables `Brady.score` (y-axis) versus `Sum` (x-axis):

- Add the plot title **Brady Score by Total Homicide Rate**, x-axis label **Total Homicide Rate**, and y-axis label **Brady.score**, and change the point symbol to solid rhombus. Hint: Click the link to find more plot options.
- Note that the variable **Jurisdiction** contains the name of each state. Use the function `text()` to label each point with corresponding names. Hint: link
- In addition, color the 3 states with the highest total homicide rates red. The other states remain black. Hint: Use the argument `col=ifelse(Sum>=cutoff,'red','black')` to

set the color, where cutoff is the third-highest total homicide rate.

Answer:

```
gun$Sum <- gun$Homicide.rate + gun$Gun.accident.rate
cutoff <- sort(gun$Sum, decreasing = TRUE)[3]
plot(gun$Sum, gun$Brady.score,
     main = "Brady Score by Total Homicide Rate",
     xlab = "Total Homicide Rate",
     ylab = "Brady.score",
     pch = 18,
     col = ifelse(gun$Sum >= cutoff, "red", "black"))

text(gun$Sum, gun$Brady.score, labels = gun$Jurisdiction,
     pos = 4, cex = 0.7)
```



Question 3 [30 points]

a. From the `county_2019` dataset in the `usdata` R package, create a data frame `subdata` by extracting the columns pop, median_household_income, and mean_household_income for counties with a population (pop) greater than 3,000,000. Then, sort `subdata` by pop in ascending order.

Answer:

```r
library(usdata)
data("county_2019")
subdata <- subset(county_2019,
                  pop > 3000000,
                  select = c(pop, median_household_income, mean_household_income))

subdata <- subdata[order(subdata$pop), ]
subdata
```

```
##           pop median_household_income mean_household_income
## 216   3168044                  90234                122488
## 223   3316073                  78980                106600
## 104   4328810                  64468                 89019
## 2624  4646630                  61705                 91486
## 611   5198275                  64660                 95677
## 205  10081570                  68044                 99133
```

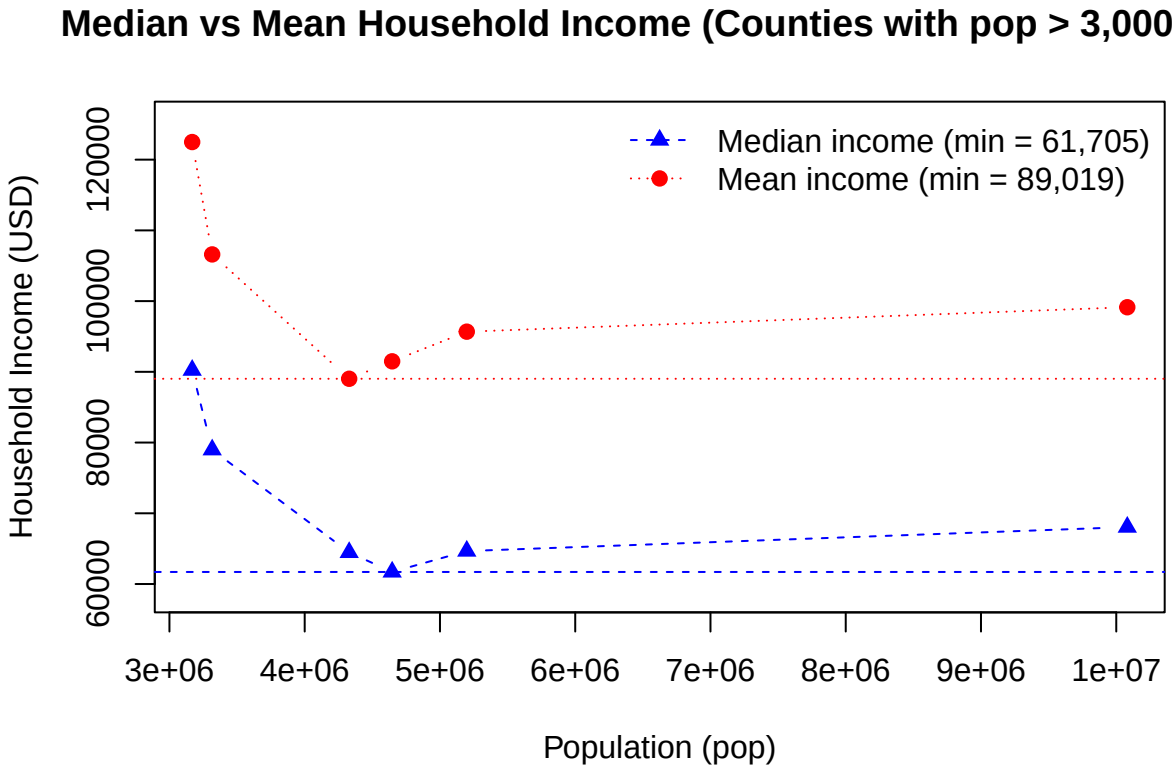
b. Attach the `subdata`. Create a plot styled similarly to the one on page 36 of Lecture 4 slides, with `pop` corresponding to the x-axis, and two lines representing `median_household_income` and `mean_household_income`.

- Add two horizontal reference lines at the minimum values of `median_household_income` and `mean_household_income`. Set `cex=1`. Add the legend at the top right of the plot.
- Adjust the ylim and the text in labels, title, and legend accordingly.
- Use at least one different value for `pch` and `lty` than those on the lecture slide. Make sure the settings are consistent for the data points and legend.

Answer:

```r
attach(subdata)
min_median <- min(median_household_income, na.rm = TRUE)
min_mean   <- min(mean_household_income,   na.rm = TRUE)

y_all <- c(median_household_income, mean_household_income, min_median, min_mean)
pad <- diff(range(y_all)) * 0.05
ylim_use <- c(min(y_all) - pad, max(y_all) + pad)

pch_median <- 17
pch_mean   <- 19
lty_median <- 2
lty_mean   <- 3

plot(pop, median_household_income, type = "b",
     pch = pch_median, lty = lty_median, cex = 1,
     col = "blue",
```

```
     xlab = "Population (pop)",
     ylab = "Household Income (USD)",
     main = "Median vs Mean Household Income (Counties with pop > 3,000,000)",
     ylim = ylim_use)

lines(pop, mean_household_income, type = "b",
      pch = pch_mean, lty = lty_mean, cex = 1,
      col = "red")

abline(h = min_median, lty = lty_median, col = "blue")
abline(h = min_mean,   lty = lty_mean,   col = "red")

legend("topright",
       legend = c(paste0("Median income (min = ", format(min_median, big.mark=","), ")"),
                  paste0("Mean income (min = ",   format(min_mean,   big.mark=","), ")")),
       col = c("blue", "red"),
       pch = c(pch_median, pch_mean),
       lty = c(lty_median, lty_mean),
       cex = 1, bty = "n")
```




```
detach(subdata)
```

"The plot shows that mean household income consistently exceeds the median, indicating right-skewed income distributions across large counties. The gap between mean and median

widens for the most populous counties, highlighting greater income inequality."