Assignment 1

Kangrui Liu

2023-09-13

Git and GitHub

1) Provide the link to the GitHub repo that you used to practice git from Week 1.

- https://github.com/krliu67/Liu-a1

Reading Data

2) Read in the .dta version and store in an object called angell_stata.

```
library(haven)
angell_stata <- read_dta("angell.dta")
```

3) Read in the .txt version and store it in an object called angell_txt.

```
angell_txt <- read.table("angell.txt")
```

4) What are the differences between angell_stata and angell_txt? Are there differences in the classes of the individual columns?

```
print(sapply(angell_stata, class))
```

```
##        city      morint      ethhet      geomob      region
## "character"   "numeric"   "numeric"   "numeric" "character"
```

```
print(sapply(angell_txt, class))
```

```
##          V1          V2          V3          V4          V5
## "character"   "numeric"   "numeric"   "numeric" "character"
```

- The column name of two data sets are different. There are differences between the classes of columns, which are numeric and character.

5) Make any updates necessary so that angell_txt is the same as angell_stata.

```
library(plyr)
angell_txt <- plyr::rename(angell_txt, c("V1"="city","V2"="morint","V3"="ethhet","V4"="geomob","V5"="reg
print(class(angell_stata) == class(angell_txt))
```

```
## [1] FALSE FALSE  TRUE
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
angell_txt <- as_tibble(angell_txt)
print(class(angell_stata) == class(angell_txt))
```

```
## [1] TRUE TRUE TRUE
```

- The columun names of two data are different and the classes of two data are also different, so I alter the column name of angell_txt and convert its type to which angell_stata is.

6) Describe the Ethnic Heterogeneity variable. Use descriptive statistics such as mean, median, standard deviation, etc. How does it differ by region?

```
t1 <- aggregate(ethhet ~ region, angell_stata, FUN = mean)
t2 <- aggregate(ethhet ~ region, angell_stata, FUN = sd)
t3 <- aggregate(ethhet ~ region, angell_stata, FUN = median)
t4 <- merge(t1, t2, by = "region")
rm(t1,t2)
p6 <- merge(t3, t4, by = "region")
rm(t3,t4)
plyr::rename(p6, c("ethhet"="mean","ethhet.x"="sd","ethhet.y"="median"))
```

```
##   region  mean       sd    median
## 1      E 22.10 23.48889 10.773398
## 2     MW 19.25 21.67857  9.084914
## 3      S 53.80 52.48571 21.440897
## 4      W 16.15 16.55000  4.164012
```

```
rm(p6)
```

- As from the table, S region has highest mean, sd and median number, whereas W region has the least mean, sd and median number.

Describing Data

7) Install the "MASS" package, load the package. Then, load the Boston dataset.

```
# install.packages('MASS')
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
data(Boston)
```

8) What is the type of the Boston object?

```
typeof(Boston)
```

```
## [1] "list"
```

- The type of Boston object is "list".

9) What is the class of the Boston object?

```
class(Boston)
```

```
## [1] "data.frame"
```

- The class of Boston object is "data.frame".

10) How many of the suburbs in the Boston data set bound the Charles river?

```
#print(sapply(Boston, class))
sum(Boston$chas == 1, na.rm = TRUE) # filter
```

```
## [1] 35
```

- There are 35 suburbs in the Boston data set bound the Charles river.

11) Do any of the suburbs of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the range of each variable.

```
c(min(Boston$crim),max(Boston$crim))
```

```
## [1]  0.00632 88.97620
```

```
c(min(Boston$tax),max(Boston$tax))
```

```
## [1] 187 711
```

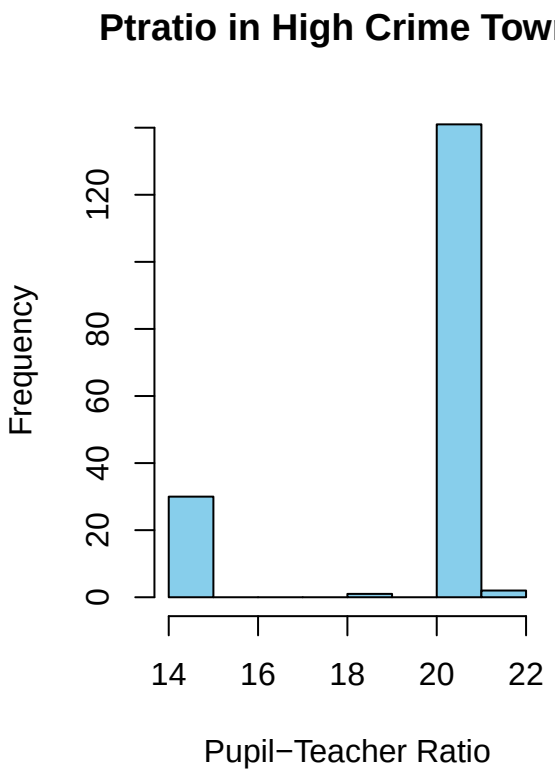
```
c(min(Boston$ptratio),max(Boston$ptratio))
```

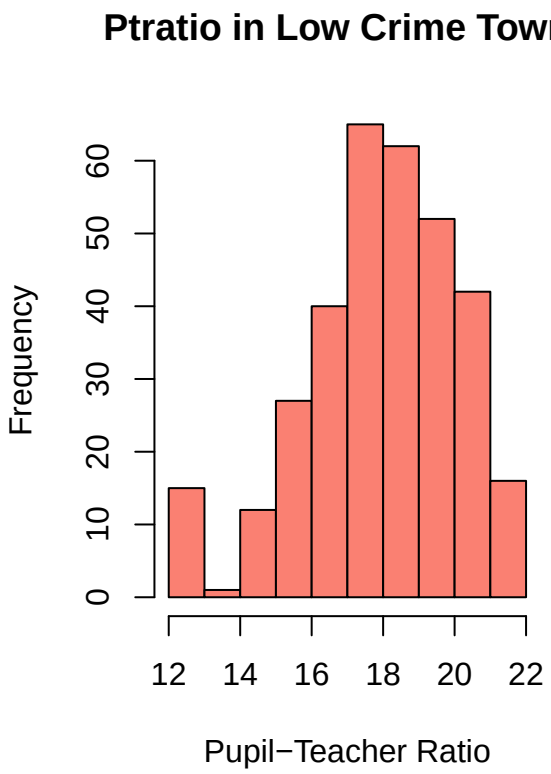
```
## [1] 12.6 22.0
```

- The range of crime rates in Boston is $[0.00632, 88.97620]$, Tax rates is $[187, 711]$, and Pupil-teacher ratios is $[12.6, 22.0]$.

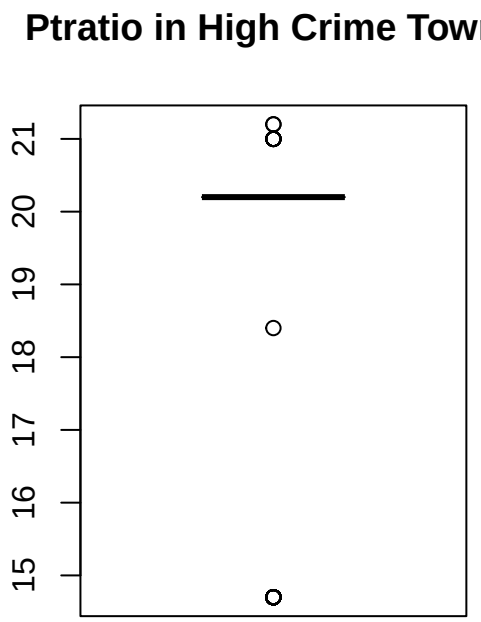
12) Describe the distribution of pupil-teacher ratio among the towns in this data set that have a per capita crime rate larger than 1. How does it differ from towns that have a per capita crime rate smaller than 1?

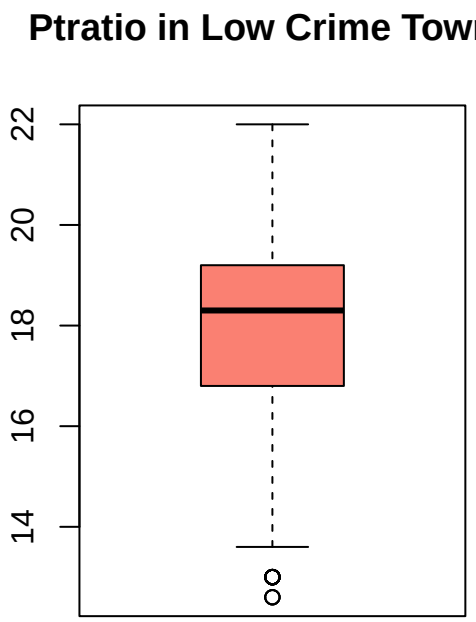
```
bos1 <- subset(Boston, Boston$crim > 1)
bos2 <- subset(Boston, Boston$crim <= 1)
layout(matrix(c(1, 2), nrow = 1, ncol = 2, byrow = TRUE))
hist(bos1$ptratio, main="Ptratio in High Crime Towns", xlab="Pupil-Teacher Ratio", col="skyblue")
hist(bos2$ptratio, main="Ptratio in Low Crime Towns", xlab="Pupil-Teacher Ratio", col="salmon")
```






```r
layout(matrix(c(1, 2), nrow = 1, ncol = 2, byrow = TRUE))
boxplot(bos1$ptratio, main="Ptratio in High Crime Towns", col="skyblue")
boxplot(bos2$ptratio, main="Ptratio in Low Crime Towns", col="salmon")
```





* Observing from the above graphs, the distribution of pupil-teacher ratio which has a per capita crime rate larger than 1 is dense, gathering in 20, and 2 of 4 extreme points are less than 19 and the rest outliers are larger than 20. However, that have less than 1 per capita crime rate is more chaos, separating from almost 14 to 22, and only 2 outliers are less than 14.

```
summary(bos1)
```

```
##       crim                zn             indus            chas              nox        
##  Min.   : 1.002   Min.   :0   Min.   : 8.14   Min.   :0.00000   Min.   :0.532  
##  1st Qu.: 3.489   1st Qu.:0   1st Qu.:18.10   1st Qu.:0.00000   1st Qu.:0.605  
##  Median : 6.759   Median :0   Median :18.10   Median :0.00000   Median :0.679  
##  Mean   :10.139   Mean   :0   Mean   :17.84   Mean   :0.08621   Mean   :0.677  
##  3rd Qu.:12.024   3rd Qu.:0   3rd Qu.:18.10   3rd Qu.:0.00000   3rd Qu.:0.713  
##  Max.   :88.976   Max.   :0   Max.   :21.89   Max.   :1.00000   Max.   :0.871  
##        rm              age              dis              rad        
##  Min.   :3.561   Min.   : 29.30   Min.   :1.130   Min.   : 4.00  
##  1st Qu.:5.657   1st Qu.: 87.90   1st Qu.:1.590   1st Qu.:24.00  
##  Median :6.113   Median : 94.85   Median :1.951   Median :24.00  
##  Mean   :6.022   Mean   : 90.34   Mean   :2.125   Mean   :19.34  
##  3rd Qu.:6.405   3rd Qu.: 98.78   3rd Qu.:2.429   3rd Qu.:24.00  
##  Max.   :8.780   Max.   :100.00   Max.   :4.499   Max.   :24.00  
##       tax           ptratio          black            lstat      
##  Min.   :304.0   Min.   :14.70   Min.   :  0.32   Min.   : 1.73  
##  1st Qu.:666.0   1st Qu.:20.20   1st Qu.:250.04   1st Qu.:13.14  
##  Median :666.0   Median :20.20   Median :367.44   Median :17.18  
##  Mean   :597.4   Mean   :19.29   Mean   :297.77   Mean   :17.82
```

```
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:393.65   3rd Qu.:22.70
##  Max.   :666.0   Max.   :21.20   Max.   :396.90   Max.   :37.97
##       medv
##  Min.   : 5.00
##  1st Qu.:12.53
##  Median :15.20
##  Mean   :17.61
##  3rd Qu.:20.60
##  Max.   :50.00
```

```
summary(bos2)
```

```
##       crim                zn              indus             chas
##  Min.   :0.00632   Min.   :  0.00   Min.   : 0.460   Min.   :0.00000
##  1st Qu.:0.05493   1st Qu.:  0.00   1st Qu.: 3.970   1st Qu.:0.00000
##  Median :0.11101   Median :  0.00   Median : 6.200   Median :0.00000
##  Mean   :0.19356   Mean   : 17.32   Mean   : 7.626   Mean   :0.06024
##  3rd Qu.:0.24637   3rd Qu.: 25.00   3rd Qu.: 9.742   3rd Qu.:0.00000
##  Max.   :0.98843   Max.   :100.00   Max.   :27.740   Max.   :1.00000
##       nox               rm              age              dis
##  Min.   :0.3850   Min.   :5.093   Min.   :  2.90   Min.   : 1.612
##  1st Qu.:0.4370   1st Qu.:5.963   1st Qu.: 33.45   1st Qu.: 2.894
##  Median :0.4890   Median :6.296   Median : 58.05   Median : 4.361
##  Mean   :0.4906   Mean   :6.422   Mean   : 57.17   Mean   : 4.670
##  3rd Qu.:0.5380   3rd Qu.:6.759   3rd Qu.: 82.65   3rd Qu.: 6.083
##  Max.   :0.6470   Max.   :8.725   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   :1.000   Min.   :187.0   Min.   :12.60   Min.   : 70.8
##  1st Qu.:4.000   1st Qu.:264.0   1st Qu.:16.80   1st Qu.:387.1
##  Median :4.000   Median :300.0   Median :18.30   Median :393.4
##  Mean   :4.416   Mean   :309.1   Mean   :18.02   Mean   :387.5
##  3rd Qu.:5.000   3rd Qu.:335.0   3rd Qu.:19.20   3rd Qu.:396.9
##  Max.   :8.000   Max.   :711.0   Max.   :22.00   Max.   :396.9
##      lstat             medv
##  Min.   : 1.980   Min.   : 7.00
##  1st Qu.: 5.987   1st Qu.:19.88
##  Median : 9.060   Median :22.95
##  Mean   : 9.948   Mean   :25.11
##  3rd Qu.:12.922   3rd Qu.:28.77
##  Max.   :30.810   Max.   :50.00
```

Writing Functions

13) Write a function that calculates 95% confidence intervals for a point estimate. The function should be called my_CI. When called with my_CI(2, 0.2), the function should print out "The 95% CI upper bound of point estimate 2 with standard error 0.2 is 2.392. The lower bound is 1.608."

```
my_CI <- function(point_estimate, se){
  lower_bound <- point_estimate - 1.96 * se
```

```
  upper_bound <- point_estimate + 1.96 * se
  paste0("The 95 percent CI upper bound of point estimate 2 with standard error 0.2 is ",upper_bound,".
}
my_CI(2, 0.2)
```

```
## [1] "The 95 percent CI upper bound of point estimate 2 with standard error 0.2 is 2.392. The lower b
```

14) Create a new function called my_CI2 that does that same thing as the my_CI function but outputs a vector of length 2 with the lower and upper bound of the confidence interval instead of printing out the text. Use this to find the 95% confidence interval for a point estimate of 0 and standard error 0.4.

```
my_CI2 <- function(point_estimate, se){
  lower_bound <- point_estimate - 1.96 * se
  upper_bound <- point_estimate + 1.96 * se
  c(lower_bound,upper_bound)
}
my_CI2(0,0.4)
```

```
## [1] -0.784  0.784
```

- The 95% confidence interval for a point estimate of 0 and standard error 0.4 is $[-0.784, 0.784]$.

15) Update the my_CI2 function to take any confidence level instead of only 95%. Call the new function my_CI3. You should add an argument to your function for confidence level.

```
my_CI3 <- function(point_estimate, se, alpha){
  a <- 1 - alpha  # 1 - alpha
  z <- qnorm(1 - a/2)
  lower_bound <- point_estimate - z * se
  upper_bound <- point_estimate + z * se
  c(lower_bound,upper_bound)
}
my_CI3(0,0.4,0.9)# test with a 90 percent confidence interval
```

```
## [1] -0.6579415  0.6579415
```

16) Without hardcoding any numbers in the code, find a 99 percent confidence interval for Ethnic Heterogeneity in the Angell dataset. Find the standard error by dividing the standard deviation by the square root of the sample size.

```r
mean_eth <- mean(angell_stata$ethhet)
std_eth <- sd(angell_stata$ethhet)/sqrt(nrow(angell_stata))
my_CI_eth <- function(point_estimate, se, alpha){
  a <- 1 - alpha  # 1 - alpha
  z <- qnorm(1 - a/2)
  lower_bound <- point_estimate - z * se
  upper_bound <- point_estimate + z * se
  c(lower_bound,upper_bound)
}
my_CI_eth(mean_eth,std_eth,0.99)
```

```
## [1] 23.35425 39.38993
```

- The 99% confidence interval for Ethnic Heterogeneity is $[23.35425, 39.38993]$.

17) Write a function that you can apply to the Angell dataset to get 95% confidence intervals. The function should take one argument: a vector. Use if-else statements to output NA and avoid error messages if the column in the data frame is not numeric or logical.

```r
my_CI4 <- function(input_vector){
  if(!is.numeric(input_vector) && !is.logical(input_vector)) {
    return(NA)
  }
  mean_temp <- mean(input_vector)
  std_temp <- sd(input_vector)/sqrt(length(input_vector))
  a <- 1 - 0.95  # 1 - alpha
  z <- qnorm(1 - a/2)
  lower_bound <- mean_temp - z * std_temp
  upper_bound <- mean_temp + z * std_temp
  c(lower_bound,upper_bound)
}
lapply(angell_stata, my_CI4)
```

```
## $city
## [1] NA
##
## $morint
## [1] 10.13242 12.26758
##
## $ethhet
## [1] 25.27127 37.47292
##
## $geomob
## [1] 24.67187 30.52347
##
## $region
## [1] NA
```