SURV727 HW #1

Yael Beshaw

2024-09-17

Git and Github

1) Provide the link to the GitHub repo that you used to practice git from Week 1 https://github.com/ybeshaw/test-f24-727, note that there were some merge conflicts from Week 1. After addressing these, I was able to update all changes and make notes in the README file.

Reading Data

Download both the Angell.dta (Stata data format) dataset and the Angell.txt dataset from this website: https://stats.idre.ucla.edu/stata/examples/ara/applied-regression-analysis-byfox-data-files/

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

```
#install.packages("foreign")
library(foreign)

## Warning: package 'foreign' was built under R version 4.3.3

angell_stata <- read.dta("/Users/yaelbeshaw/R Scripts and
Projects/angell.dta")
View(angell_stata)
```

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

```
angell_txt_link <-
  "https://stats.oarc.ucla.edu/wp-content/uploads/2016/02/angell.txt"
download.file(angell_txt_link, destfile=
                "/Users/yaelbeshaw/R Scripts and Projects/angell.txt")

angell_txt <- read.table("/Users/yaelbeshaw/R Scripts and
Projects/angell.txt")
View(angell_txt)
```

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

Between angell_stata and angell_txt we see there is a difference in the column names. In order to check for the classifications of each column we can utilize
class(df
$colname). Oncewedothisforeach, weseethattherearenodifferencesintheclassforeachpair(i.e., angel$
city=character= class(angell_txt) and so forth).

```
colnames(angell_stata)

## [1] "city"   "morint" "ethhet" "geomob" "region"

colnames(angell_txt)

## [1] "V1" "V2" "V3" "V4" "V5"

#both angell_stata and angell_txt are both dataframes
class(angell_stata)

## [1] "data.frame"

class(angell_txt)

## [1] "data.frame"

class(angell_stata$city) #character

## [1] "character"

class(angell_txt$V1) #character

## [1] "character"

class(angell_stata$morint) #numeric

## [1] "numeric"

class(angell_txt$V2) #numeric

## [1] "numeric"

class(angell_stata$ethhet) #numeric

## [1] "numeric"

class(angell_txt$V3) #numeric

## [1] "numeric"

class(angell_stata$geomob) #numeric

## [1] "numeric"

class(angell_txt$V4) #numeric

## [1] "numeric"

class(angell_stata$region) #character

## [1] "character"

class(angell_txt$V5) #character

## [1] "character"
```

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

We see that the main difference between these two dataframes is the column names. We can address this using colnames(angell_txt)<-c().

```r
colnames(angell_txt) <- c("city", "morint", "ethhet", "geomob", "region")
View(angell_txt)

#to verify this, take a look at the dimensions of the dataframe and that it
#matches!
dim(angell_txt)

## [1] 43  5

dim(angell_stata)

## [1] 43  5
```

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

The heterogeneity variable represents the percentages of nonwhite and foreign-born white residents within a city. The mean value of this variable is 31.37, indicating that on average across all cities represented, 31.37% of the residents are non-white and foreign-born white residents. The median value of this is 23.70%, in which half the cities in the dataset have a percentage of nonwhites and foreign born whites lower than this percentage while the other half have a percentage greater than 23.70%. The mean is greater than the median, indicating a skewdness to the right in this variable. The standard deviation is 20.41 while the variance is 416.63, this highlights a wide spread distribution of data and great variation from the mean.

In order to assess differences by region, we create a function in R, that will also us to assess the mean, median, sd, and variance according to each of the four regions that this dataset is split into. In the East (E) region we see that there are a total of 9 cities with a mean heterogeneity of 23.49% of their population and a median of 22.10%, we see that the difference between the mean and median still indicates a right skew but is of a lesser magnitude than that of the overall dataset. Additionally, it has a standard deviation of 10.77 and variance of 116.07, much smaller than that above. This indicates that there is less variance in this region subset. In the MidWest (MW) region, we observe similar outcomes for a total of 14 cities. With a mean of 21.67% and median of 19.25%, we again see that there is a right skew but less than that of the entire dataset. We also have a standard deviation of 9.08 and a variance of 82.54, indicating that there is still a spread of values and variability in the dataset but less than that of East cities and all cities as a whole. For the 14 cities in the South(S) region, we see that there is a much greater mean and median of 52.49% and 53.80%, respectively. This is the only region where the data is skewed to the left as the median is slightly greater than the mean. It also has the greatest standard deviation (21.44) and variance (459.71) compared to any other region and the entire dataset itself, this helps us understand where a large portion of the variability from the mean in our data comes from. Lastly, the 6 cities in the West (W) region, reflect the lowest heterogentiy on average at 16.55% with a median of 16.15%. Additionally, we see the lowest standard deviation (4.16) and variance (17.34) in this region. Overall, by grouping according to region, we are able to find patterns and identity characteristics of the dataset that help us understand the data as a whole.

```
#descriptive stats about Ethnic Heterogeneity
summary(angell_stata$ethhet)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   10.60   16.90   23.70   31.37   39.00   84.50

sd(angell_stata$ethhet)

## [1] 20.41149

var(angell_stata$ethhet)

## [1] 416.6287

library(dplyr)

## 
## Attaching package: 'dplyr'

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

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

ethhet_region <- angell_stata %>%
  group_by(region) %>%
  summarize(
    mean = mean(ethhet, na.rm = TRUE),
    med = median(ethhet, na.rm = TRUE),
    sd= sd(ethhet, na.rm = TRUE),
    var = var(ethhet, na.rm = TRUE),
    total = n()
  )

ethhet_region

## # A tibble: 4 × 6
##   region  mean   med    sd   var total
##   <chr>  <dbl> <dbl> <dbl> <dbl> <int>
## 1 E       23.5  22.1 10.8  116.      9
## 2 MW      21.7  19.2  9.08  82.5    14
## 3 S       52.5  53.8 21.4  460.     14
## 4 W       16.5  16.1  4.16  17.3     6
```

Describing Data

R comes also with many built-in datasets. The “MASS” package, for example, comes with the “Boston” dataset.

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

View(Boston)
```

8) What is the type of the Boston object?

It is a list

```
typeof(Boston) #list of vectors

## [1] "list"
```

9) What is the class of the Boston object?

It is a data frame

```
class(Boston) #data.frame, 2D set of vectors

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

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

35 of the suburbs in the Boston dataset bound the Charles River.

```
#Firstly, understand the coding for each variable by looking at documentation
??MASS::Boston

#this dataset looks at the housing values in suburbs of Boston, all suburbs
#chas variable is a dummy variable for Charles River where 1= bound
#and 0= otherwise

bound2_cr<- sum(Boston$chas == 1) #will output the number of suburbs where
#chas== 1 and therefore the suburb bounds the Charles River
bound2_cr #35

## [1] 35
```

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.

Based on the range of these variables, yes, we do see particularly high rates in some cases. In the case of crime rate, we see a huge range between 0.006 and 88.976. The maximum here is particularly high and when compared against the mean value of 3.61, it is safe to say that there are suburbs of Boston that have a high crime rate. Additionally, in relation to the tax rate, we see that there is a range between 187 and 711, in order to assess if there are particularly high tax rates, I compare the range against the mean of 408.24, indicating that while there are suburbs with higher tax rates, it is not particularly high compared to the average. Additionally, in terms of the pupil-teacher ratio, we see that the range between 12.6 and 22.0 does not have as much distance compared to the other two variables, however when compared against the mean 18.45, we are actually able to state that there are particularly lower ratios.

```r
#crime rates, crim = per capita crime rate by town
range(Boston$crim) #(0.006, 88.976)

## [1]  0.00632 88.97620

mean(Boston$crim) # 3.61

## [1] 3.613524

#tax rates, tax= full-value property-tax rate per $10,000
range(Boston$tax) #(187, 711)

## [1] 187 711

mean(Boston$tax) # 408.24

## [1] 408.2372

#pupil-teacher ratio, ptration= pupil-teacher ratio by town
range(Boston$ptratio) #(12.6, 22.0)

## [1] 12.6 22.0

mean(Boston$ptratio) # 18.45

## [1] 18.45553
```

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?

In order to do this we have to filter out the suburbs that have a per capita crime rate larger than one, describe the distribution of pt ratio within it and repeat the same steps for per capita crime rate smaller than one. To compare the two dataframes created, I bound the rows for both the high_crim and low_crim to observe the side by side comparisons. What we are able to first observe is the different in the range between the high_crim and low_crim. We see that for suburbs with crime rates above 1, that the range of pupil to teacher ratio is (14.7, 21.2) whereas the range for suburbs with crime rates below 1 is (12.6, 22.0). Although there is a greater range for suburbs with less crime, the mean and median of pupil to teacher ratio in the lower crime towns is also lower (by 1.28 and 1.9, respectively). Additionally something to note is that the standard deviation and variance for each are similar although there is more variance in towns with higher crime rates. Lastly, there are almost double the number of towns with low crime rate relative to those with high crime rate, indicating that these low crime rate suburbs are more common in Boston with less variance in the pupil to teacher ratios, an important factor to consider in education policy and improvement.

```
library(dplyr)
crim_high <- Boston %>%
  filter(crim > 1) %>%
  summarize(
    mean= mean(ptratio, na.rm = TRUE),
    med= median(ptratio, na.rm = TRUE),
    min= min(ptratio, na.rm = TRUE),
    max= max(ptratio, na.rm = TRUE),
    sd= sd(ptratio, na.rm = TRUE),
    var= var(ptratio, na.rm = TRUE),
    total= n()
)
crim_high

##       mean  med  min  max       sd      var total
## 1 19.29425 20.2 14.7 21.2 2.117576 4.484129   174

crim_low <- Boston %>%
  filter(crim < 1) %>%
  summarize(
    mean= mean(ptratio, na.rm = TRUE),
    med= median(ptratio, na.rm = TRUE),
    min= min(ptratio, na.rm = TRUE),
    max= max(ptratio, na.rm = TRUE),
    sd= sd(ptratio, na.rm = TRUE),
    var=var(ptratio, na.rm = TRUE),
    total= n()
)
crim_low

##       mean  med  min max       sd      var total
## 1 18.01596 18.3 12.6  22 2.059995 4.243581   332

crim_compare <- bind_rows(crim_high, crim_low)
crim_compare

##       mean  med  min  max       sd      var total
## 1 19.29425 20.2 14.7 21.2 2.117576 4.484129   174
## 2 18.01596 18.3 12.6 22.0 2.059995 4.243581   332
```

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."

Note: The function should take a point estimate and its standard error as arguments. You may use the formula for 95% CI: point estimate +/- 1.96*standard error.

Hint: Pasting text in R can be done with: paste() and paste0()

```
my_CI <- function(pt_est, st_err) {
  lower_bound <- pt_est - 1.96*st_err
  upper_bound <- pt_est + 1.96*st_err

  paste("The 95% CI upper bound of point estimate",pt_est,"with standard
error", st_err,"is",upper_bound,".","The lower bound is",lower_bound)
}

#test
my_CI(2, 0.2)

## [1] "The 95% CI upper bound of point estimate 2 with standard error 0.2 is
2.392 . The lower bound is 1.608"
```

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.

The confidence for a point estimate of 0 and a standard error of 0.4 is -0.784, 0.784.

```
my_CI2 <- function(pt_est,st_err) {
    lower_bound <- pt_est - 1.96*st_err
    upper_bound <- pt_est + 1.96*st_err

    output <-return(c(lower_bound, upper_bound)) #c(,) == vector of n=2
}

#test
my_CI2(0, 0.4)

## [1] -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.

Hint: Use the qnorm function to find the appropriate z-value. For example, for a 95% confidence interval, using qnorm(0.975) gives approximately 1.96.

Note-> Utilizing the hint we can work backwards to create a function for the CI within qnorm. If qnorm(0.975) is for a 95% CI, we know that (1-0.95) will give us our alpha, and 1-alpha/2 will give us the appropriate z-value. Therefore, we can add an equation 1-(1-CI)/2 into qnorm.

```
?qnorm() #check properities of qnorm first


my_CI3 <- function(pt_est, st_err, CI_level) {
    z_score <- qnorm(1-(1-CI_level)/2)
    lower_bound <- pt_est - z_score*st_err
    upper_bound <- pt_est + z_score*st_err

    return(c(lower_bound, upper_bound)) #c() == vector
}

#check answers
my_CI3(0, 0.4, 0.95) #correct!

## [1] -0.7839856  0.7839856

my_CI3(2, 0.2, 0.95) #correct!

## [1] 1.608007 2.391993
```

16) Without hardcoding any numbers in the code, find a 99% 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.

The 99% confidence interval is (30.76074, 31.98345).

```
#we know that CI are computed utilizing the point estimate, z_score, and
#standard error.

#point-estimate in this case will be equivalent to the mean
#z_score will be equivalent to qnorm(1-(1-conf)/2), simillar to question 15,
#and as stated in the question, our standard error will be computed by
#dividing the standard deviation by the square root of the sample size



#point_est <- mean(angell_stata$ethhet)
#conf <- 0.99
#st_error <- sd(angell_stata$ethhet)/ (n(angell_stata$ethhet))^1/2



ethnic_hetero <- function(point_est, st_error, conf) {
    z_score <- qnorm(1-(1-conf)/2)
    lower_bound <- point_est - z_score*st_error
    upper_bound <- point_est + z_score*st_error
    return(c(lower_bound, upper_bound)) #c() == vector
}

ethnic_hetero(mean(angell_stata$ethhet),
              (sd(angell_stata$ethhet)/(length(angell_stata$ethhet))^1/2),
              (0.99))

## [1] 30.76074 31.98345
```

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.

```
#Using is.numeric() and is.logical(), we first verify that the columns in the
#df are numeric or logical, we see that 3/5 are numeric and none are logical.

#Write the function
angell_ci95 <- function(x){

    if (is.numeric(x) || is.logical(x)){ #if it is numeric or logical
      x <- na.omit(x)

      z_score <- qnorm(0.975)
      lower_bound <- mean(x) - z_score*(sd(x)/(length(x))^1/2)
      upper_bound <- mean(x) + z_score*(sd(x)/(length(x))^1/2)

    return(c(lower_bound, upper_bound)) #c() == vector
    } else{ #if it is not numeric or logical
      return(c(NA,NA))
    }
}

lapply(angell_stata, angell_ci95) #this will give us a list for each x

## $city
## [1] NA NA
##
## $morint
## [1] 11.1186 11.2814
##
## $ethhet
## [1] 30.90691 31.83728
##
## $geomob
## [1] 27.37458 27.82076
##
## $region
## [1] NA NA
```