SURV727 Assignment # 1

Juan Gelvez-Ferreira. PhD Student

2022-09-23

First assignment

1. Which command do you use to determine the type of an object?

Typeof() to determines the type or storage mode of any object.

```
?typeof
```

```
## starting httpd help server ... done
```

2. What is the type of vector A? Using the command “typeof()”, the vector A is a character.

```
A <- c("2", "3", "4", "5", "6", "7", "8")
typeof(A)
```

```
## [1] "character"
```

3 Convert A into an integer vector To convert A into an integer, I use the “as.integer()” function to create A1.

```
?as.integer
A1 <- as.integer(A)
A1
```

```
## [1] 2 3 4 5 6 7 8
```

```
typeof(A1)
```

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

*4*Create an integer vector B containing the numbers one through ten

```
B <- (1:10)
B
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
typeof(B)
```

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

5. Create a new vector C from B which has the type “double”

```
C<- as.double(B)
C
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
typeof(C)
```

```
## [1] "double"
```

6 Change the third value of B to “3.5”

```
B
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
B1<-replace(B,B==3,3.5)
B1
```

```
##  [1]  1.0  2.0  3.5  4.0  5.0  6.0  7.0  8.0  9.0 10.0
```

7 Did this affect the type of B? How? Yes, it moved from integer to double. Since “B” was containing the numbers one through ten, and the new B “B1” has a decimal, the type changed from a integer to a double.

```
typeof(B1)
```

```
## [1] "double"
```

#Reading in data

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

```
angell_stata <- read_dta("data/angell.dta")
head(angell_stata)
```

```
## # A tibble: 6 x 5
##   city       morint ethhet geomob region
##   <chr>       <dbl>  <dbl>  <dbl> <chr>
## 1 Rochester    19     20.6   15   E
## 2 Syracuse     17     15.6   20.2 E
## 3 Worcester    16.4   22.1   13.6 E
## 4 Erie         16.2   14     14.8 E
## 5 Milwaukee    15.8   17.4   17.6 MW
## 6 Bridgeport   15.3   27.9   17.5 E
```

# 9. Read in the .txt version and store it in an object called angell_txt

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

```
##           V1   V2   V3   V4 V5
## 1  Rochester 19.0 20.6 15.0  E
## 2   Syracuse 17.0 15.6 20.2  E
## 3  Worcester 16.4 22.1 13.6  E
## 4       Erie 16.2 14.0 14.8  E
## 5  Milwaukee 15.8 17.4 17.6 MW
## 6 Bridgeport 15.3 27.9 17.5  E
```

10. Drop the first five observations in the angell_txt object

```
angell_txt2<-tail(angell_txt, -5)
head(angell_txt2)
```

```
##            V1   V2   V3   V4 V5
## 6  Bridgeport 15.3 27.9 17.5  E
## 7     Buffalo 15.2 22.3 14.7  E
## 8      Dayton 14.3 23.7 23.8 MW
## 9     Reading 14.2 10.6 19.4  E
## 10 Des_Moines 14.1 12.7 31.9 MW
## 11  Cleveland 14.0 39.7 18.6 MW
```

11. Select columns 2 and 3 of the agell_stata object and store them in a new object called angell_small

```
angell_small<- angell_stata %>%
  select(2,3)

head(angell_small)
```

```
## # A tibble: 6 x 2
##   morint ethhet
##    <dbl>  <dbl>
## 1   19     20.6
## 2   17     15.6
## 3   16.4   22.1
## 4   16.2   14
## 5   15.8   17.4
## 6   15.3   27.9
```

12. 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
```

```r
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

13. What is the type of the Boston object? Using the "typeof" command, it's a list.

```r
typeof(Boston)
```

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

14. What is the class of the Boston object? Boston is a dataframe.

```r
class(Boston)
```

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

15. How many of the suburbs in the Boston data set bound the Charles river? There are 35 suburbs in the Boston data set bound the Charles river

```r
nrow(subset(Boston, chas ==1))
```

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

16. 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.

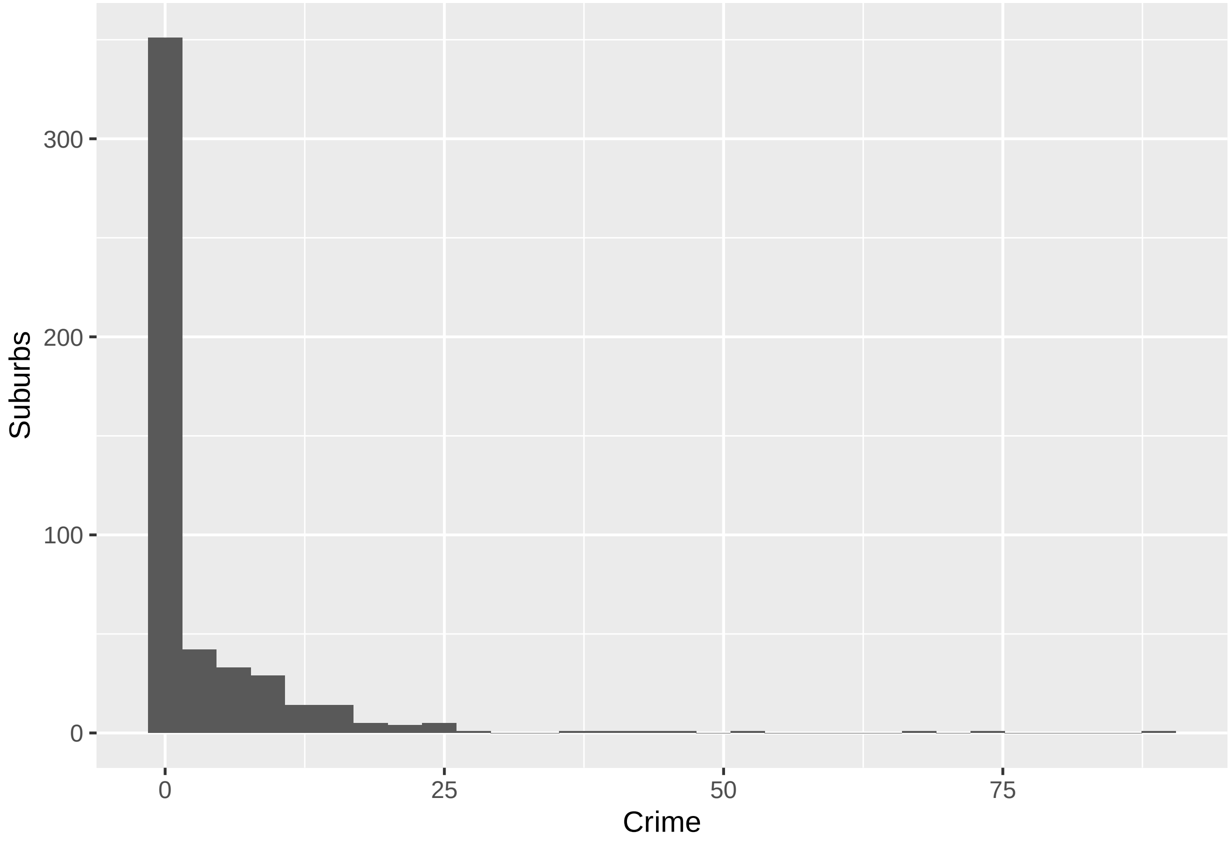
According to the summary and the histogram, it seems that the three variables have outliers. For instance, the crime variable shows a maximum of 88.97 crimes, which is far from the mean which is 3.6, and the minimum is 0.0063.

```r
summary(Boston$crim)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##  0.00632  0.08204  0.25651  3.61352  3.67708 88.97620
```

```
qplot(Boston$crim, xlab = "Crime", ylab="Suburbs" )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



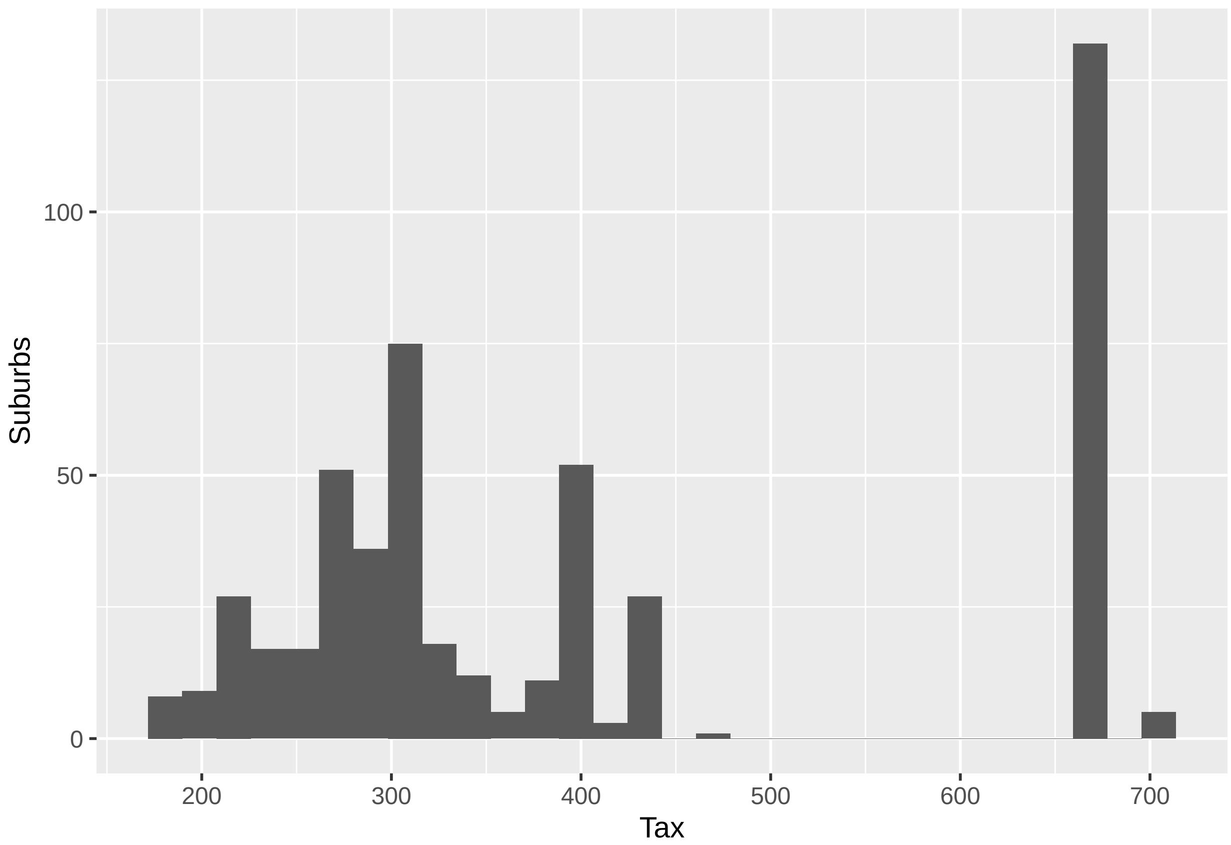
Likewise, the tax variable shows that there are a big range. The minimum is 187 and the maximum is 711, with a mean of 408.2.

```
summary(Boston$tax)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   187.0   279.0   330.0   408.2   666.0   711.0
```

```
qplot(Boston$tax, xlab = "Tax", ylab="Suburbs" )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



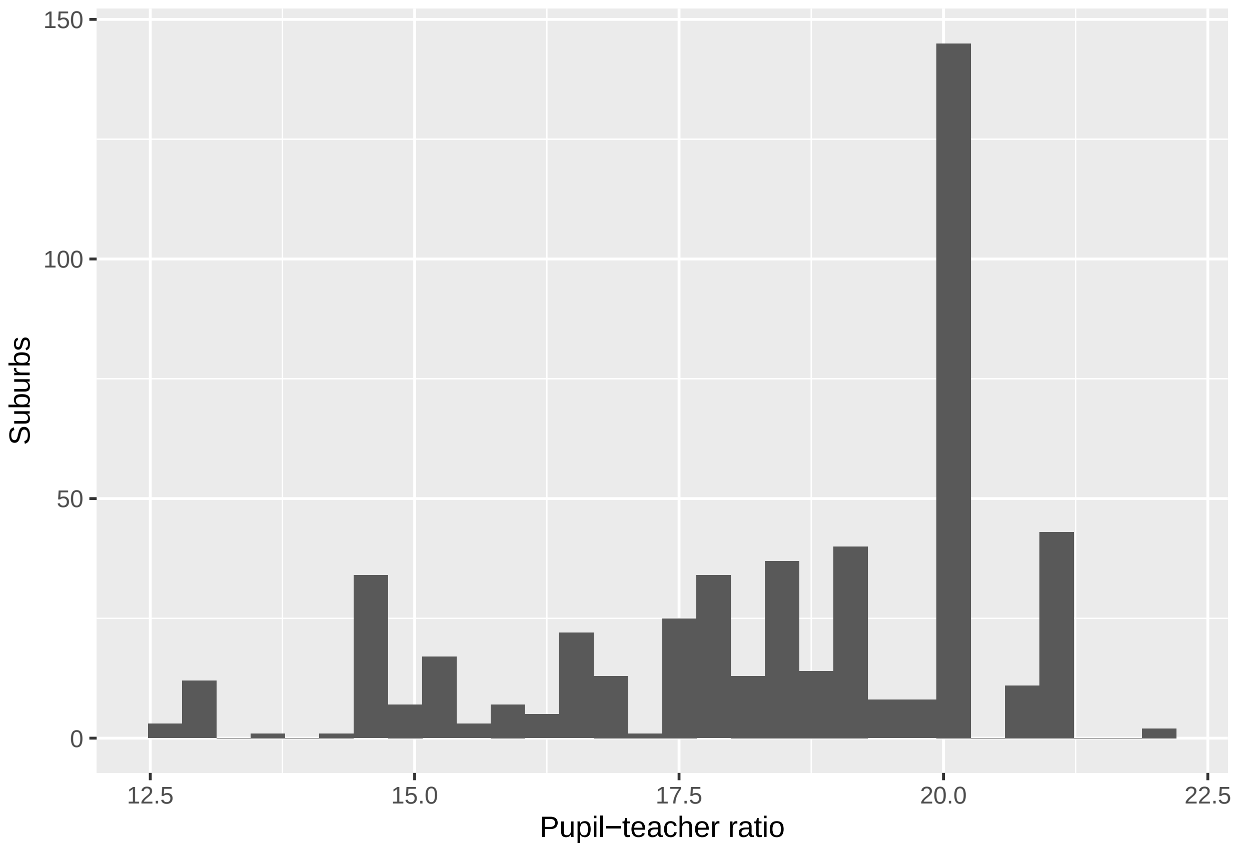
Finally, the pupil-teacher ratio minimum value is 12.6 which is far from the median (19.05) and the mean (18.46). Likewise, the maximum is 22.

```
summary(Boston$ptratio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   12.60   17.40   19.05   18.46   20.20   22.00
```

```
qplot(Boston$ptratio, xlab = "Pupil-teacher ratio", ylab="Suburbs" )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



17. What is the median pupil-teacher ratio among the towns in this data set that have a per capita crime rate larger than 1 ?

It’s 20.20

```
boston2 <- Boston[ which(Boston$crim > 1), ]
summary(boston2$ptratio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.70   20.20   20.20   19.29   20.20   21.20
```

18. Write a function that calculates the squareroot of an integer

```
function.square <- function(a) {
   for(i in a:a) {
      b <- sqrt(i)
      print(b)
   }
}

function.square(6) #for example number 6
```

```
## [1] 2.44949
```

19.. 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 output of the function should read “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)

```
my_CI <- function(x, y) {
  lower <- x - 1.96 * y
  upper <- x + 1.96 * y
  paste("The 95% CI upper bound of point estimate", x, "with standard error", y,
        "is", upper,".", "The lower bound is", lower,".")
}

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 i
```

20. Write a function that converts all negative numbers in the following dataset into NA Use as little code as possible and try to avoid code repetition

```
set.seed(1002)
df <- data.frame(replicate(10, sample(c(1:10, c(-99,-98,-5)), 6, rep = TRUE)))
names(df) <- letters[1:6]
df
```

```
##     a  b   c d  e f  NA  NA  NA  NA
## 1 -98  6   1 6  7 1 -98   6   5 -98
## 2   9 -5  10 4 -5 7 -99   3   2   2
## 3  -5  3   5 3  2 2   5  10   7  -5
## 4   7  8 -98 9  9 2  10   4   3 -99
## 5   4  1   5 3  6 6  10   7 -99   6
## 6  -5 -5   3 9  3 7  10 -98   7   6
```

```
class(df)
```

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

```
#function
into_NA <- function (x) {
  x <- replace(x, x < 0, NA)
  print(x)
}
into_NA(df)
```

```
##    a  b  c d  e f NA NA NA NA
## 1 NA  6  1 6  7 1 NA  6  5 NA
## 2  9 NA 10 4 NA 7 NA  3  2  2
## 3 NA  3  5 3  2 2  5 10  7 NA
## 4  7  8 NA 9  9 2 10  4  3 NA
## 5  4  1  5 3  6 6 10  7 NA  6
## 6 NA NA  3 9  3 7 10 NA  7  6
```

21. Use your function to convert all negative numbers in the dataset into NA without changing the class of the object.

```
#With the function above, the class of the object did not change
class(df)
```

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

22. Change the function you wrote above such that it turns any negative number into NA!

```
set.seed(1002)
df <- data.frame(replicate(10, sample(c(1:10, c(-99,-98,-5)), 6, rep = TRUE)))
names(df) <- letters[1:6]
df
```

```
##     a  b   c d  e f  NA  NA  NA  NA
## 1 -98  6   1 6  7 1 -98   6   5 -98
## 2   9 -5  10 4 -5 7 -99   3   2   2
## 3  -5  3   5 3  2 2   5  10   7  -5
## 4   7  8 -98 9  9 2  10   4   3 -99
## 5   4  1   5 3  6 6  10   7 -99   6
## 6  -5 -5   3 9  3 7  10 -98   7   6
```

```
class(df)
```

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

```
#function
into_NA_ <- function (x) {
  x <- replace(x, x < 0, "NA!")
  print(x)
}
into_NA_(df)
```

```
##     a   b   c d   e f  NA  NA  NA  NA
## 1 NA!   6   1 6   7 1 NA!   6   5 NA!
## 2   9 NA!  10 4 NA! 7 NA!   3   2   2
## 3 NA!   3   5 3   2 2   5  10   7 NA!
## 4   7   8 NA! 9   9 2  10   4   3 NA!
## 5   4   1   5 3   6 6  10   7 NA!   6
## 6 NA! NA!   3 9   3 7  10 NA!   7   6
```