Homework 1

4375 Machine Learning with Dr. Mazidi

Anthony Martinez

9/5/21

This homework has two parts:

- Part 1 uses R for data exploration
- Part 2 uses C++ for data exploration

This homework is worth 100 points, 50 points each for Part 1 and Part 2.

Part 1: RStudio Data Exploration

Instructions: Follow the instructions for the 10 parts below. If the step asks you to make an observation or comment, write your answer in the white space above the gray code box for that step.

Step 1: Load and explore the data

- load library MASS (install at console, not in code)
- load the Boston dataframe using data(Boston)
- use str() on the data
- type ?Boston at the console
- Write 2-3 sentences about the data set below

Your commentary here: The Boston data frame from MASS provides the Housing values in the suburbs of Boston. The data frame has 506 rows and 14 columns. Some of the columns include, ‘crim’ which provides the crime rate by town per capita, ‘rm’ which provides the average number of rooms per dwelling and ‘ptratio’ which provides the pupuil-teacher ratio by town.

```
# step 1 code
if (!require("MASS")){
  install.packages("MASS")
}
```

```
## Loading required package: MASS
```

```
library(MASS)
data(Boston)
str(Boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

Step 2: More data exploration

Use R commands to:

- display the first few rows
- display the last two rows
- display row 5
- display the first few rows of column 1 by combining head() and using indexing
- display the column names

```
# step 2 code

# display the first few rows
head(Boston)
```

	crim <dbl>	zn <dbl>	indus <dbl>	chas <int>	nox <dbl>	rm <dbl>	age <dbl>	dis <dbl>	rad <int>
1	0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3
6	0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3

6 rows | 1-10 of 15 columns

```
# display the last two rows
tail(Boston, 2)
```

	crim <dbl>	zn <dbl>	indus <dbl>	chas <int>	nox <dbl>	rm <dbl>	age <dbl>	dis <dbl>	rad <int>
505	0.10959	0	11.93	0	0.573	6.794	89.3	2.3889	1
506	0.04741	0	11.93	0	0.573	6.030	80.8	2.5050	1

2 rows | 1-10 of 15 columns

```
#display row 5
Boston[5, ]
```

	crim <dbl>	zn <dbl>	indus <dbl>	chas <int>	nox <dbl>	rm <dbl>	age <dbl>	dis <dbl>	rad <int>
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3

1 row | 1-10 of 15 columns

```
# display the first few rows of column 1 by combining head() and using indexing
head(Boston[,1])
```

```
## [1] 0.00632 0.02731 0.02729 0.03237 0.06905 0.02985
```

```
#$ display the column names
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

Step 3: More data exploration

For the crime column, show:

- the mean
- the median
- the range

```
# step 3 code

# mean of crime column
mean(Boston$crim)
```

```
## [1] 3.613524
```

```
# median of crime column
median(Boston$crim)
```

```
## [1] 0.25651
```

```
# range of crime column
range(Boston$crim)
```

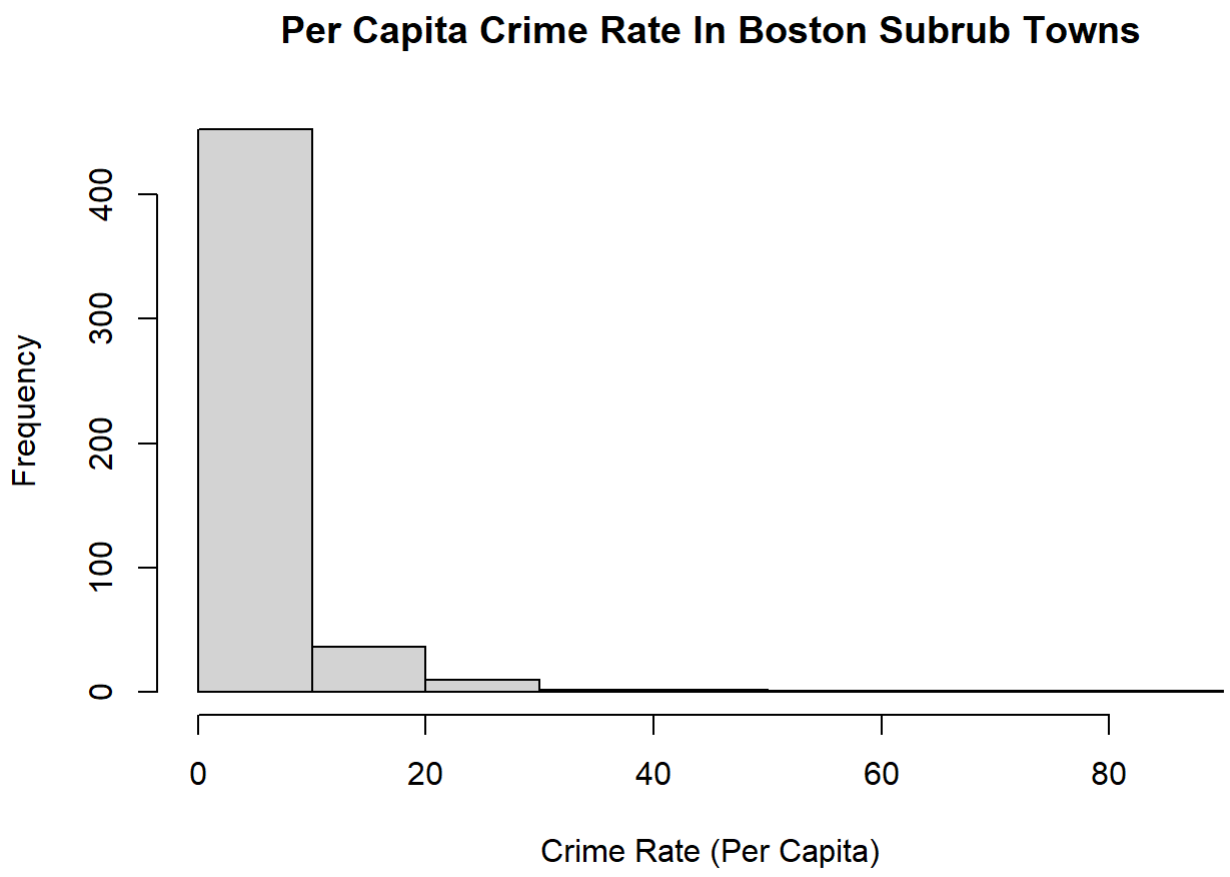
```
## [1]  0.00632 88.97620
```

Step 4: Data visualization

Create a histogram of the crime column, with an appropriate main heading. In the space below, state your conclusions about the crime variable:

Your commentary here: Based on the Boston data frame we can see from the histogram of the Crim coloumn that most towns in Boston have a per capita crime rate in 0-.1 range. A minority of towns in Boston have a per capita crime rate between .1 and .2. Very few towns have a per capita crime rate in the rage of .2 and .3. It is extremely rare for a town in Boston to have a per capita crime rate greater than .3.

```
# step 4 code
hist(Boston$crim, main = "Per Capita Crime Rate In Boston Subrub Towns", xlab="Crime Rate (Per C
apita)")
```



Step 5: Finding correlations

Use the cor() function to see if there is a correlation between crime and median home value. In the space below, write a sentence or two on what this value might mean. Also write about whether or not the crime column might be useful to predict median home value.

Your commentary here: Recall that a -1 correlation represents a strong negative correlation. This means that a -.38 cor() value for crim and medv represents a weak negative correlation between these columns. This means that the crime column might not be the best option for predicting the median home value.

```
# step 5 code

# correlation between crime and median home value
cor(Boston$crim,Boston$medv, use="complete")
```

```
## [1] -0.3883046
```

Step 6: Finding potential correlations

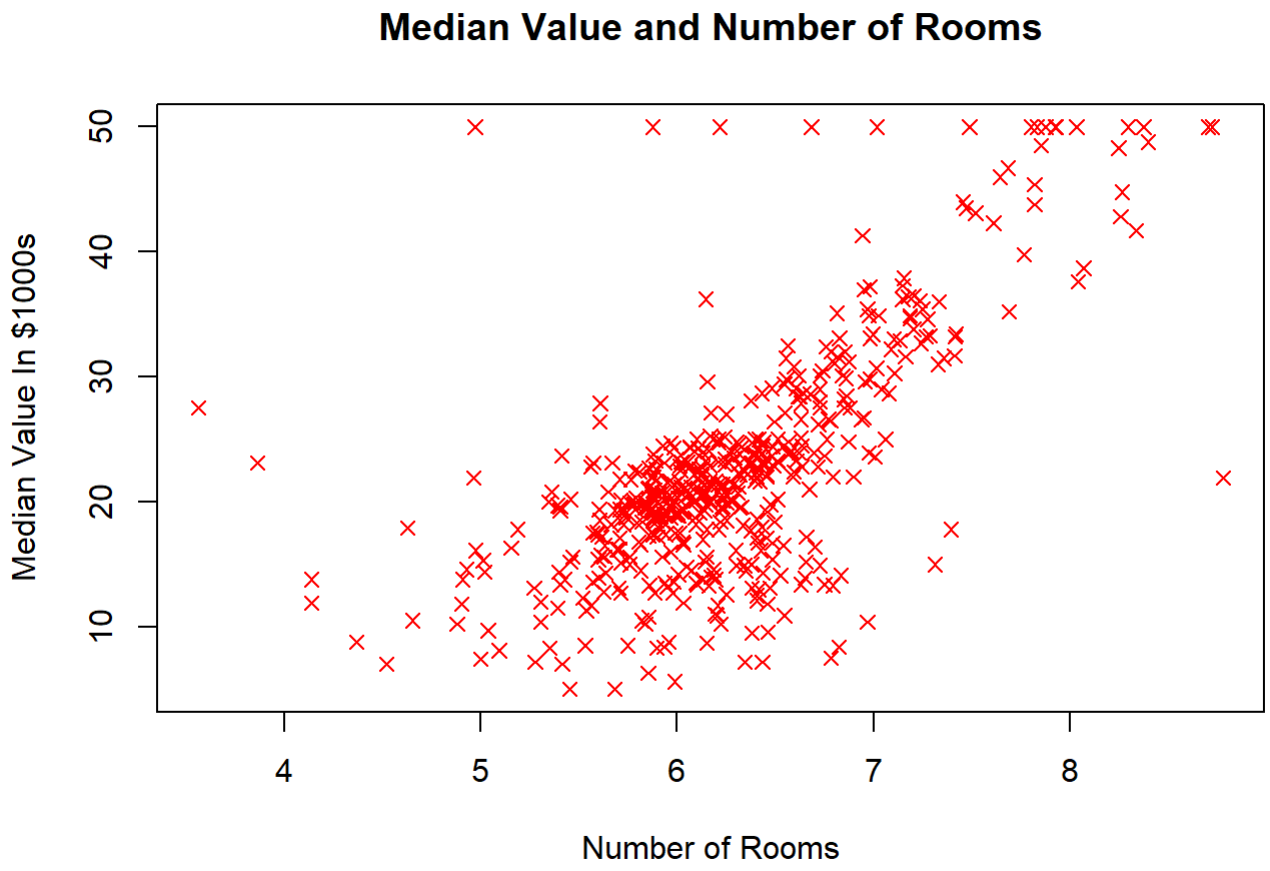
Create a plot showing the median value on the y axis and number of rooms on the x axis. Create appropriate main, x and y labels, change the point color and style. [Reference for plots(http://www.statmethods.net/advgraphs/parameters.html (http://www.statmethods.net/advgraphs/parameters.html))

Use the cor() function to quantify the correlation between these two variables. Write a sentence or two summarizing what the graph and correlation tell you about these 2 variables.

Your commentary here: Per the plot function, it can be seen that houses with more rooms tend to have a higher value. The cor() function supports this, and tells us that these two variables are closely correlated.

```
# step 6 code

# Create a plot showing median value on y and num of rooms on x, change point color and style
plot(Boston$rm, Boston$medv, main = "Median Value and Number of Rooms", xlab="Number of Rooms",y
lab="Median Value In $1000s", col="red", pch=4)
```




```
# use cor*() function to quantify the correlation between these two variables
cor(Boston$rm, Boston$medv, use="complete")
```

```
## [1] 0.6953599
```

Step 7: Evaluating potential predictors

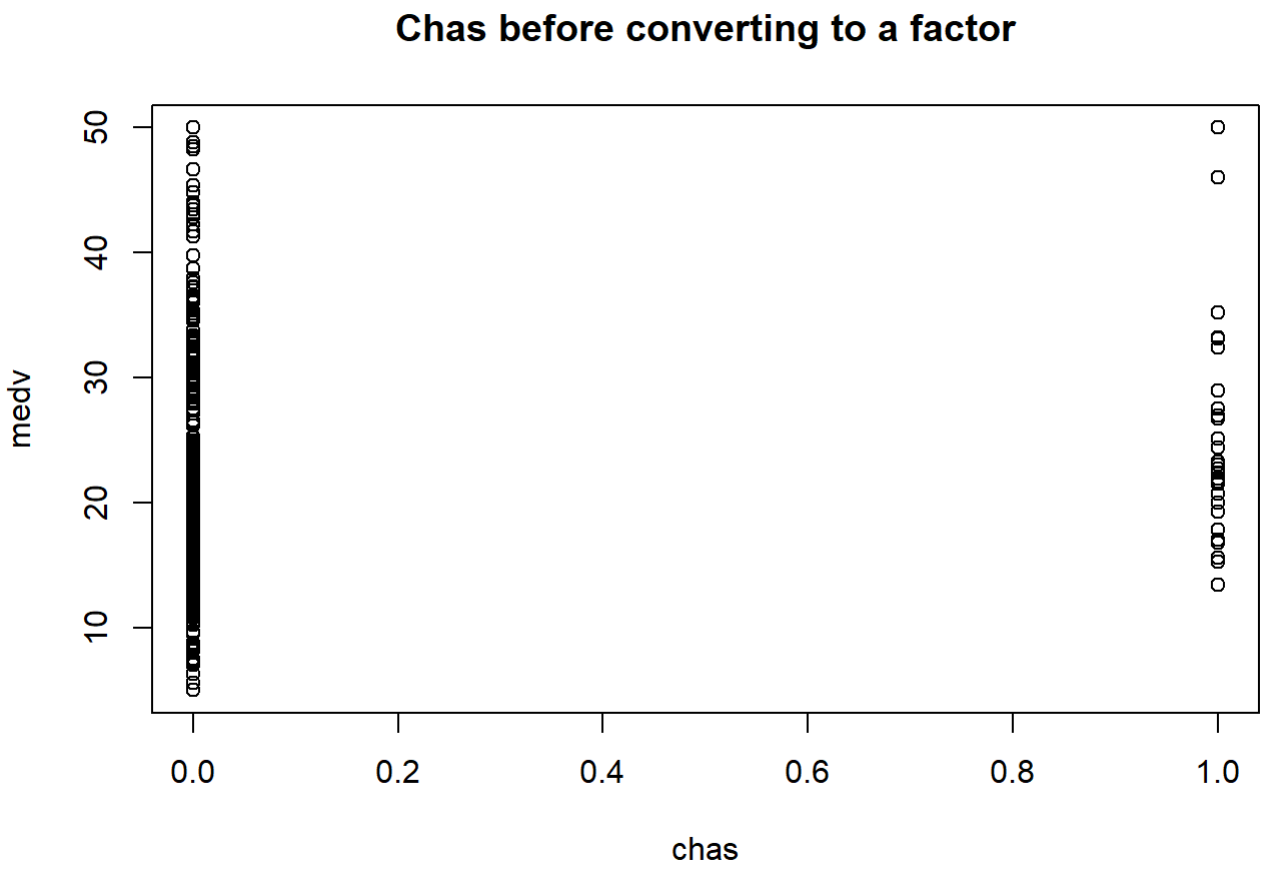
Use R functions to determine if variable chas is a factor. Plot median value on the y axis and chas on the x axis. Make chas a factor and plot again.

Comment on the difference in meaning of the two graphs. Look back the description of the Boston data set you got with the ?Boston command to interpret the meaning of 0 and 1.

Your commentary here: Before making chas a factor, R plotted it with a scatter plot. After turning chas into a factor it was plotted using a Box plot. The chas variable in Boston is made of up only 0s and 1s. 1, if tract bounds the Charles River and 0 otherwise. The box plot is a more effective graph to represent the chas column since the content can only be one of two options. Using the scatter plot does not make sense. As you can see, R thought there could be more than options; '.2', '.4', '6', '8'. We have to use as.factor() function to accurately plot columns that should be factors.

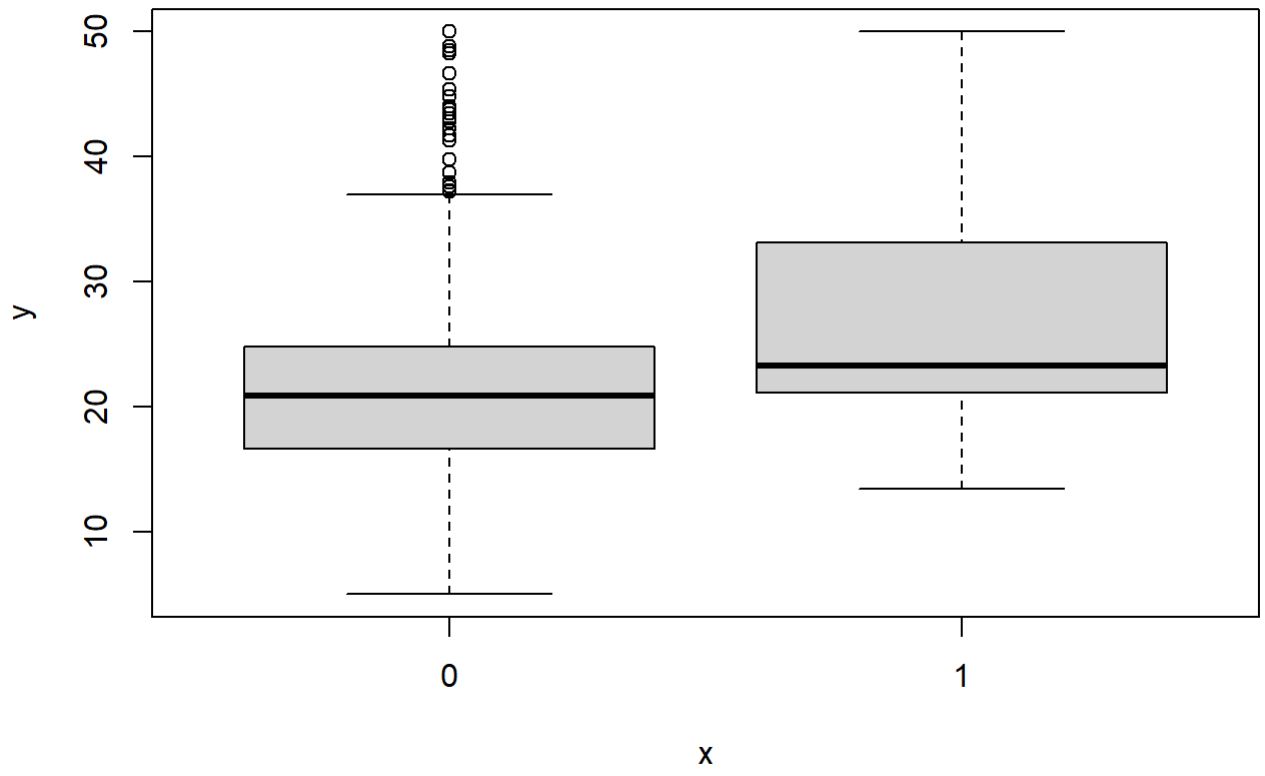
```
# step 7 code

# plot chas before factor
plot(Boston$chas, Boston$medv, main="Chas before converting to a factor", xlab="chas",ylab="med
v")
```




```
# make factor
Boston$chas <- as.factor(Boston$chas)

# plot again
plot(Boston$chas, Boston$medv)
```



Step 8: Evaluating potential predictors

Explore the rad variable. What kind of variable is rad? What information do you get about this variable with the summary() function? Does the unique() function give you additional information? Use the sum() function to determine how many neighborhoods have rad equal to 24. Use R code to determine what percentage this is of the neighborhoods.

Your commentary here: The rad variable is an integer. With the summary() function we get the min integer, the median, mean, 1st quartile, 3rd quartile and the max integer. The unique function gives us every integer that appears in the column. In other words, it gives us every unique integer in the column.

```
# step 8 code

# What kind of variable is rad?
typeof(Boston$rad)
```

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

```
# What information do you get about this variable with the summary() function?
summary(Boston$rad)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   5.000   9.549  24.000  24.000
```

```
# Does the unique() function give you additional information?
unique(Boston$rad)
```

```
## [1]  1  2  3  5  4  8  6  7 24
```

```
# Use the sum() function to determine how many neighborhoods have rad equal to 24.
equals24 = sum(Boston$rad==24)
equals24
```

```
## [1] 132
```

```
# Use R code to determine what percentage this is of the neighborhoods.
total <- length(Boston$rad)
percentage <- equals24/total
paste(round(percentage*100),"%")
```

```
## [1] "26 %"
```

Step 9: Adding a new potential predictor

Create a new variable called “far” using the ifelse() function that is TRUE if rad is 24 and FALSE otherwise. Make the variable a factor. Plot far and medv. What does the graph tell you?

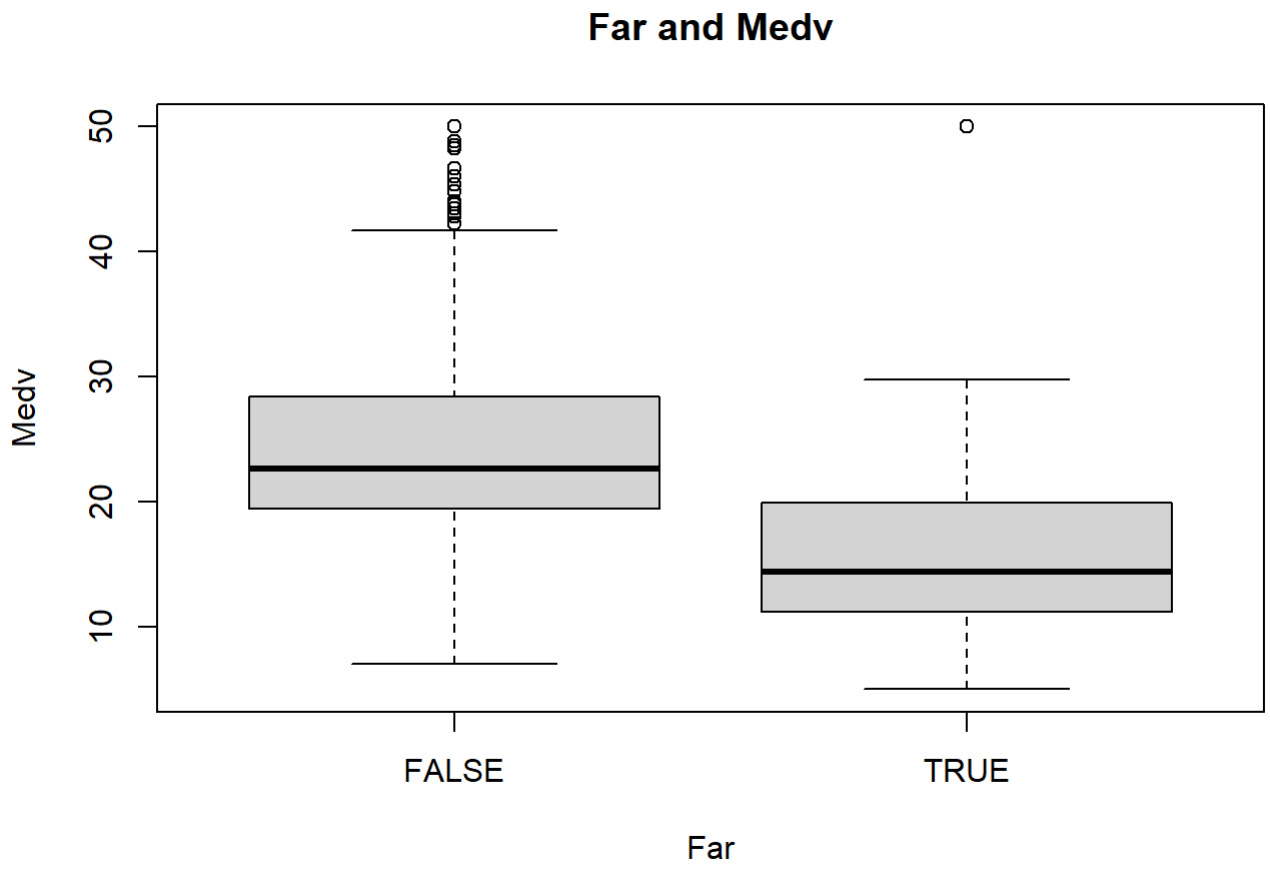
Your commentary here: The graph below tells us that the houses with a rad index of 24 have a lower median medv value than the houses that do not have an index value of 24.

```
# step 9 code

# far is true if rad is 24 and false otherwise
far <- ifelse(Boston$rad == 24, T , F)
far
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
## [361]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [373]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [385]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [397]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [409]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [421]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [433]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [445]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [457]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [469]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [481]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
## [493] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [505] FALSE FALSE
```

```
# make far a factor and plot
far <- as.factor(far)
plot(far, Boston$medv, main = "Far and Medv", xlab="Far", ylab="Medv")
```



Step 10: Data exploration

- Create a summary of Boston just for columns 1, 6, 13 and 14 (crim, rm, lstat, medv)
- Use the which.max() function to find the neighborhood with the highest median value. See p. 176 in the pdf
- Display that row from the data set, but only columns 1, 6, 13 and 14
- Write a few sentences comparing this neighborhood and the city as a whole in terms of: crime, number of rooms, lower economic percent, median value.

Your commentary here: The Neighborhood with the highest median value is 162. The crime rate per capita is lower than the mean crime rate of the entire city of Boston. The average number of rooms in neighborhood 162 is higher than the rest of the city. The lower status of the population for 162 is at 1.73 while the mean of lstat for the rest of Boston is much higher at 12.65. The median value for 162 is at 50, while for the rest of the city it is about 22.53. In summary 162 has a lower crime rate, a higher number of rooms, a lower lsat score and a higher median value of homes, than the other neighborhoods in Boston.

```
# step 10 code

# Summary for columns 1,6,13,14
summary(Boston[, c(1,6,13,14)])
```

```
##       crim                rm            lstat            medv      
##  Min.   : 0.00632   Min.   :3.561   Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 0.08205   1st Qu.:5.886   1st Qu.: 6.95   1st Qu.:17.02  
##  Median : 0.25651   Median :6.208   Median :11.36   Median :21.20  
##  Mean   : 3.61352   Mean   :6.285   Mean   :12.65   Mean   :22.53  
##  3rd Qu.: 3.67708   3rd Qu.:6.623   3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :88.97620   Max.   :8.780   Max.   :37.97   Max.   :50.00
```

```
# use which.max() function to find the neighborhood with highest med
i <- which.max(Boston$medv)
Boston[i,]
```

	crim	zn	indus	chas	nox	rm	age	dis	rad
	<dbl>	<dbl>	<dbl>	<fctr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
162	1.46336	0	19.58	0	0.605	7.489	90.8	1.9709	5

1 row | 1-10 of 15 columns

```
# display that row but only columns 1,6,13,14
subset(Boston[162, ], select=c("crim", "rm", "lstat", "medv"))
```

	crim	rm	lstat	medv
	<dbl>	<dbl>	<dbl>	<dbl>
162	1.46336	7.489	1.73	50

1 row

Part 2: C++

In this course we will get some experience writing machine learning algorithms from scratch in C++, and comparing performance to R. Part 2 of Homework 1 is designed to lay the foundation for writing custom machine learning algorithms in C++.

To complete Part 2, first you will read in the Boston.csv file which just contains columns rm and medv.

In the C++ IDE of your choice:

1 Read the csv file (now reduced to 2 columns) into 2 vectors of the appropriate type.

2 Write the following functions:

- a function to find the sum of a numeric vector
- a function to find the mean of a numeric vector
- a function to find the median of a numeric vector
- a function to find the range of a numeric vector
- a function to compute covariance between rm and medv (see formula on p. 74 of pdf)

- a function to compute correlation between rm and medv (see formula on p. 74 of pdf); Hint: sigma of a vector can be calculated as the square root of variance(v, v)

3 Call the functions described in a-d for rm and for medv. Call the covariance and correlation functions. Print results for each function.