Module 3 CT Option 1

Scott Miner

3/1/2020

# Boston housing data

The assignment asks the student to explore and summarize the BostonHousing.csv file. Additionally, the student will fit a multiple linear regression model to the median house price (MEDV) as a function of CRIM, CHAS, and RM.

## Descriptive statistics

### Read the file

# Read .csv into bostonHousing dataframe  
bostonHousing.df <- read.csv("BostonHousing.csv")

### Generate descriptive statistics

Generate means, sd, minimum, etc…

# Get the mean, sd, min, max, med, length, # of miss.val tol.perc of miss.val for all numeric variables  
  
(data.frame(mean = sapply(Filter(is.numeric, bostonHousing.df), mean, na.rm = TRUE),  
 sd = sapply(Filter(is.numeric, bostonHousing.df), sd, na.rm = TRUE),  
 min = sapply(Filter(is.numeric, bostonHousing.df), min, na.rm = TRUE),  
 max = sapply(Filter(is.numeric, bostonHousing.df), max, na.rm = TRUE),  
 median = sapply (Filter(is.numeric, bostonHousing.df), median, na.rm = TRUE ),  
 length = sapply(Filter(is.numeric, bostonHousing.df), length),  
 mis.val = sapply(Filter(is.numeric, bostonHousing.df), function(x) sum(is.na(x))),  
 cum.ratio = percent(cumsum(sapply(Filter(is.numeric,bostonHousing.df), function(x) mean(is.na(x)))))))

## mean sd min max median length  
## CRIM 3.61352356 8.6015451 0.00632 88.9762 0.25651 506  
## ZN 11.36363636 23.3224530 0.00000 100.0000 0.00000 506  
## INDUS 11.13677866 6.8603529 0.46000 27.7400 9.69000 506  
## CHAS 0.06916996 0.2539940 0.00000 1.0000 0.00000 506  
## NOX 0.55469506 0.1158777 0.38500 0.8710 0.53800 506  
## RM 6.28463439 0.7026171 3.56100 8.7800 6.20850 506  
## AGE 68.57490119 28.1488614 2.90000 100.0000 77.50000 506  
## DIS 3.79504269 2.1057101 1.12960 12.1265 3.20745 506  
## RAD 9.54940711 8.7072594 1.00000 24.0000 5.00000 506  
## TAX 408.23715415 168.5371161 187.00000 711.0000 330.00000 506  
## PTRATIO 18.45553360 2.1649455 12.60000 22.0000 19.05000 506  
## LSTAT 12.65306324 7.1410615 1.73000 37.9700 11.36000 506  
## MEDV 22.53280632 9.1971041 5.00000 50.0000 21.20000 506  
## CAT..MEDV 0.16600791 0.3724560 0.00000 1.0000 0.00000 506  
## mis.val cum.ratio  
## CRIM 0 0%  
## ZN 0 0%  
## INDUS 0 0%  
## CHAS 0 0%  
## NOX 0 0%  
## RM 0 0%  
## AGE 0 0%  
## DIS 0 0%  
## RAD 0 0%  
## TAX 0 0%  
## PTRATIO 0 0%  
## LSTAT 0 0%  
## MEDV 0 0%  
## CAT..MEDV 0 0%

The data set contains 506 records and no missing values. The averages, standard deviations, minimums, maximums, etc. are present for all numeric values. The average home price per town, for instance, is $22,000, with a standard deviation of $9,000. The minimum home value per town is $5,000, and the maximum home value is $50,000.

On average, the houses contain 6.28 rooms, with a standard deviation of 0.70. The minimum number of rooms per town is 3.56, while the maximum number of rooms per town is 8.78.

### Object Size

# check object size in memory  
object.size(bostonHousing.df)

## 50904 bytes

object.size(diamonds)

## 3456848 bytes

object.size(mpg)

## 23776 bytes

In memory, the data set contains just over 50,904 bytes. Comparatively, the diamonds dataset contains 3,456,848 bytes, while the mpg dataset contains 23,776 bytes.

### Dimensions

# get rows and dimensions  
dim(bostonHousing.df)

## [1] 506 14

The dataset contains 506 rows and 14 columns.

### Column Names

# get names of columns  
names(bostonHousing.df)

## [1] "CRIM" "ZN" "INDUS" "CHAS" "NOX"   
## [6] "RM" "AGE" "DIS" "RAD" "TAX"   
## [11] "PTRATIO" "LSTAT" "MEDV" "CAT..MEDV"

### Examine Records

The first 6 records, followed by the last 6 records…

# examine the first 6 records  
head(bostonHousing.df)

## CRIM ZN INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO LSTAT MEDV  
## 1 0.00632 18 2.31 0 0.538 6.575 65.2 4.0900 1 296 15.3 4.98 24.0  
## 2 0.02731 0 7.07 0 0.469 6.421 78.9 4.9671 2 242 17.8 9.14 21.6  
## 3 0.02729 0 7.07 0 0.469 7.185 61.1 4.9671 2 242 17.8 4.03 34.7  
## 4 0.03237 0 2.18 0 0.458 6.998 45.8 6.0622 3 222 18.7 2.94 33.4  
## 5 0.06905 0 2.18 0 0.458 7.147 54.2 6.0622 3 222 18.7 5.33 36.2  
## 6 0.02985 0 2.18 0 0.458 6.430 58.7 6.0622 3 222 18.7 5.21 28.7  
## CAT..MEDV  
## 1 0  
## 2 0  
## 3 1  
## 4 1  
## 5 1  
## 6 0

# examine the last 6 records  
tail(bostonHousing.df)

## CRIM ZN INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO LSTAT  
## 501 0.22438 0 9.69 0 0.585 6.027 79.7 2.4982 6 391 19.2 14.33  
## 502 0.06263 0 11.93 0 0.573 6.593 69.1 2.4786 1 273 21.0 9.67  
## 503 0.04527 0 11.93 0 0.573 6.120 76.7 2.2875 1 273 21.0 9.08  
## 504 0.06076 0 11.93 0 0.573 6.976 91.0 2.1675 1 273 21.0 5.64  
## 505 0.10959 0 11.93 0 0.573 6.794 89.3 2.3889 1 273 21.0 6.48  
## 506 0.04741 0 11.93 0 0.573 6.030 80.8 2.5050 1 273 21.0 7.88  
## MEDV CAT..MEDV  
## 501 16.8 0  
## 502 22.4 0  
## 503 20.6 0  
## 504 23.9 0  
## 505 22.0 0  
## 506 11.9 0

### Classes and Counts

# get the class of each variable  
sapply(bostonHousing.df, class)

## CRIM ZN INDUS CHAS NOX RM AGE   
## "numeric" "numeric" "numeric" "integer" "numeric" "numeric" "numeric"   
## DIS RAD TAX PTRATIO LSTAT MEDV CAT..MEDV   
## "numeric" "integer" "integer" "numeric" "numeric" "numeric" "integer"

# get the count of unique values per variable  
sapply(sapply(bostonHousing.df, unique), length)

## CRIM ZN INDUS CHAS NOX RM AGE   
## 504 26 76 2 81 446 356   
## DIS RAD TAX PTRATIO LSTAT MEDV CAT..MEDV   
## 412 9 66 46 455 229 2

Each variable is either numeric or an integer. There are just two binomial variables that assume the values ‘1’ or ‘0’: CAT..MEDV and CHAS. RAD contains the next least most unique values, at 9. What are they?

# use table function  
table(bostonHousing.df$RAD)

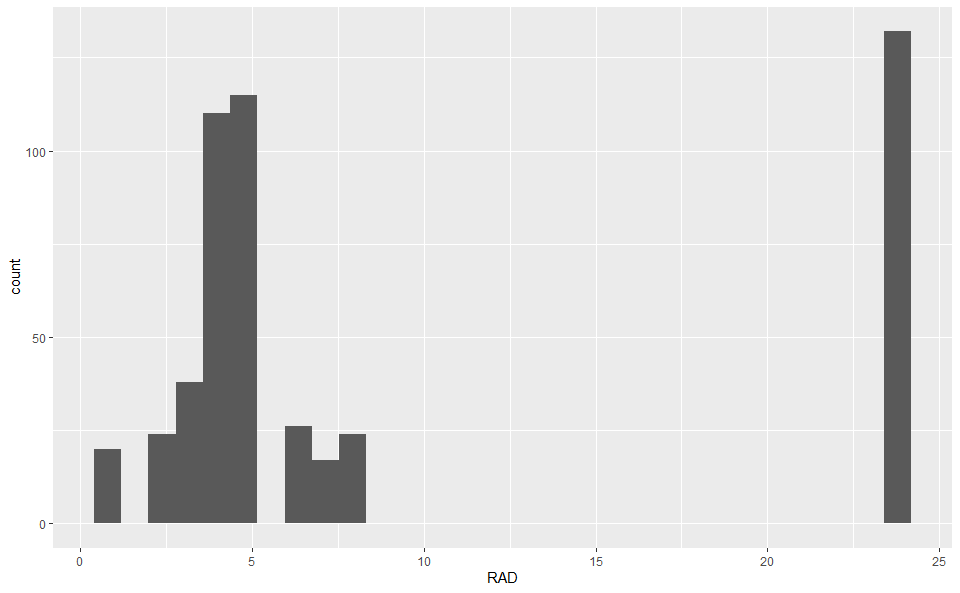
##   
## 1 2 3 4 5 6 7 8 24   
## 20 24 38 110 115 26 17 24 132

# use group\_by  
bostonHousing.df %>%  
 group\_by(RAD) %>%  
 count()

## # A tibble: 9 x 2  
## # Groups: RAD [9]  
## RAD n  
## <int> <int>  
## 1 1 20  
## 2 2 24  
## 3 3 38  
## 4 4 110  
## 5 5 115  
## 6 6 26  
## 7 7 17  
## 8 8 24  
## 9 24 132

# plot the distribution  
ggplot(bostonHousing.df, aes(RAD)) +  
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



The above displays the index, along with the counts.

### Summaries

# summary statistics   
summary(bostonHousing.df)

## CRIM ZN INDUS CHAS   
## Min. : 0.00632 Min. : 0.00 Min. : 0.46 Min. :0.00000   
## 1st Qu.: 0.08204 1st Qu.: 0.00 1st Qu.: 5.19 1st Qu.:0.00000   
## Median : 0.25651 Median : 0.00 Median : 9.69 Median :0.00000   
## Mean : 3.61352 Mean : 11.36 Mean :11.14 Mean :0.06917   
## 3rd Qu.: 3.67708 3rd Qu.: 12.50 3rd Qu.:18.10 3rd Qu.:0.00000   
## Max. :88.97620 Max. :100.00 Max. :27.74 Max. :1.00000   
## NOX RM AGE DIS   
## Min. :0.3850 Min. :3.561 Min. : 2.90 Min. : 1.130   
## 1st Qu.:0.4490 1st Qu.:5.886 1st Qu.: 45.02 1st Qu.: 2.100   
## Median :0.5380 Median :6.208 Median : 77.50 Median : 3.207   
## Mean :0.5547 Mean :6.285 Mean : 68.57 Mean : 3.795   
## 3rd Qu.:0.6240 3rd Qu.:6.623 3rd Qu.: 94.08 3rd Qu.: 5.188   
## Max. :0.8710 Max. :8.780 Max. :100.00 Max. :12.127   
## RAD TAX PTRATIO LSTAT   
## Min. : 1.000 Min. :187.0 Min. :12.60 Min. : 1.73   
## 1st Qu.: 4.000 1st Qu.:279.0 1st Qu.:17.40 1st Qu.: 6.95   
## Median : 5.000 Median :330.0 Median :19.05 Median :11.36   
## Mean : 9.549 Mean :408.2 Mean :18.46 Mean :12.65   
## 3rd Qu.:24.000 3rd Qu.:666.0 3rd Qu.:20.20 3rd Qu.:16.95   
## Max. :24.000 Max. :711.0 Max. :22.00 Max. :37.97   
## MEDV CAT..MEDV   
## Min. : 5.00 Min. :0.000   
## 1st Qu.:17.02 1st Qu.:0.000   
## Median :21.20 Median :0.000   
## Mean :22.53 Mean :0.166   
## 3rd Qu.:25.00 3rd Qu.:0.000   
## Max. :50.00 Max. :1.000

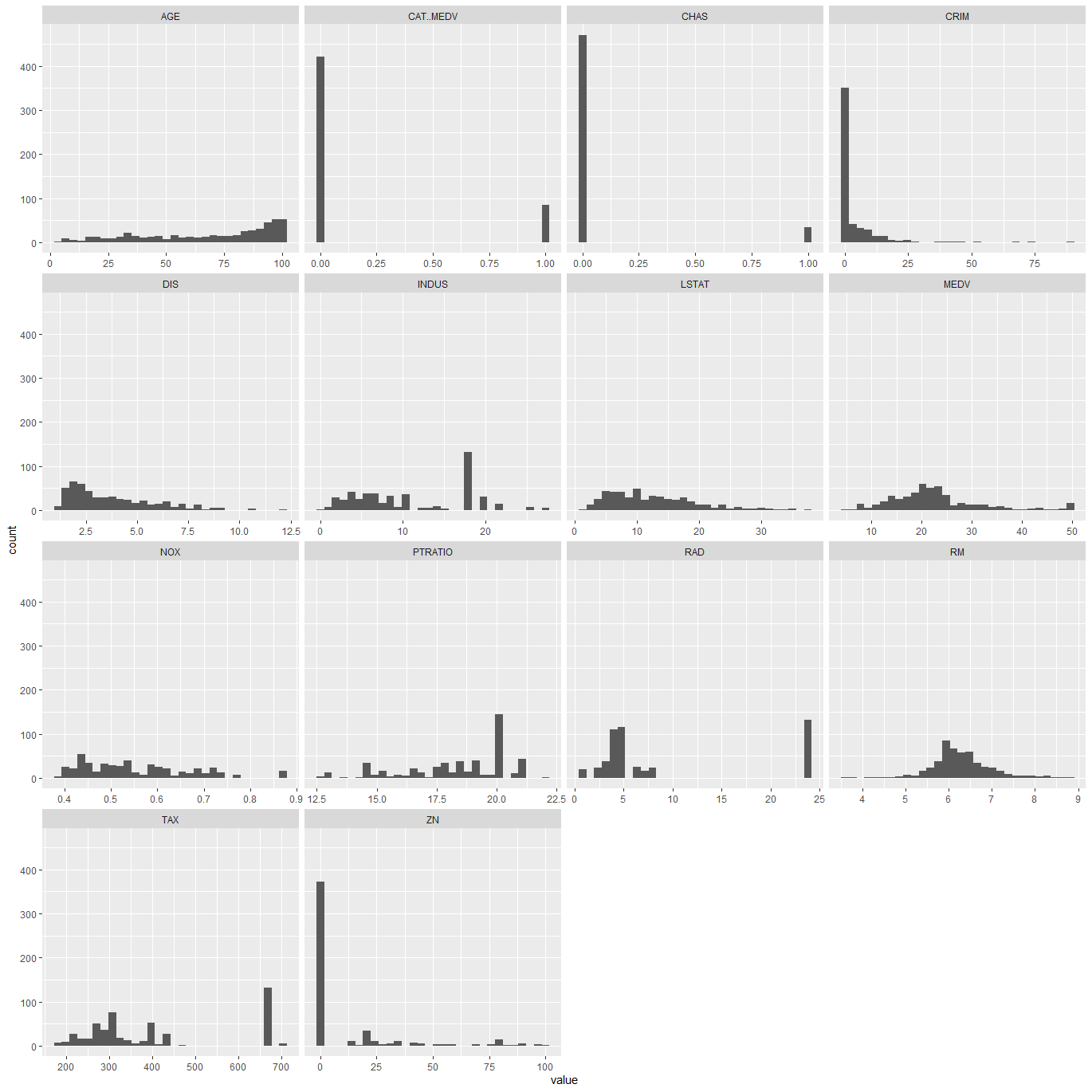
# glimpse  
glimpse(bostonHousing.df)

## Observations: 506  
## Variables: 14  
## $ CRIM <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.02985...  
## $ ZN <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5, 12....  
## $ INDUS <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87, 7.87...  
## $ CHAS <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...  
## $ NOX <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524, 0.5...  
## $ RM <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012, 6.1...  
## $ AGE <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1, 100....  
## $ DIS <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622, 5.56...  
## $ RAD <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4...  
## $ TAX <int> 296, 242, 242, 222, 222, 222, 311, 311, 311, 311, 31...  
## $ PTRATIO <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2, 15.2...  
## $ LSTAT <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.15, 29...  
## $ MEDV <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5...  
## $ CAT..MEDV <int> 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...

### Histograms and boxplots

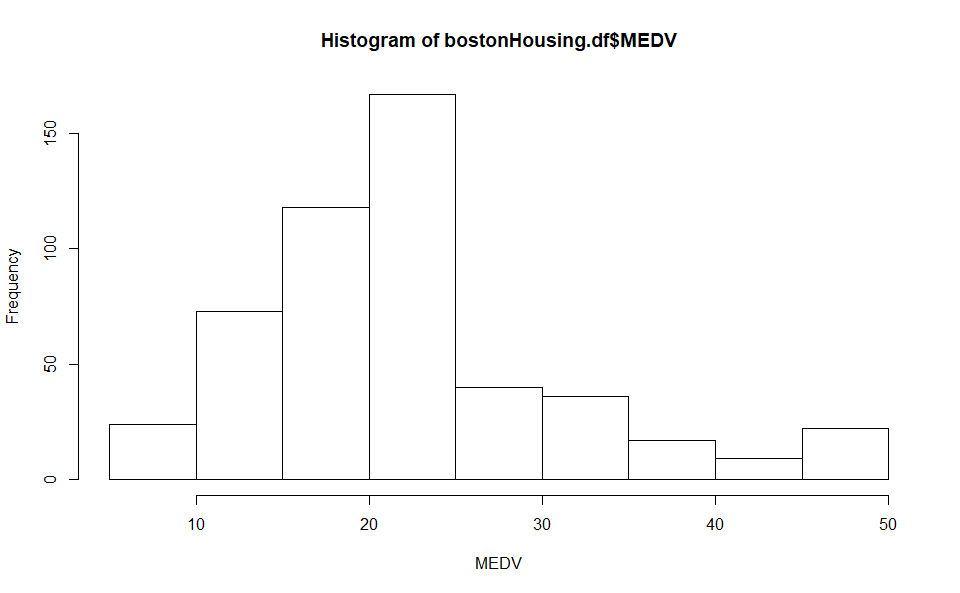
ggplot(gather(bostonHousing.df), aes(value)) +  
 geom\_histogram() +  
 facet\_wrap(~key, scales = 'free\_x')

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

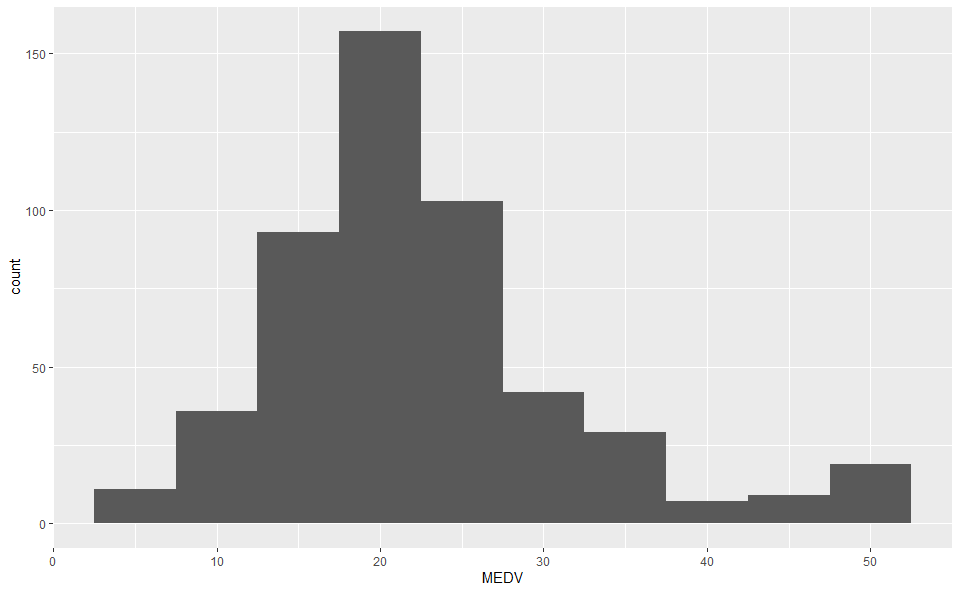


#### Histogram of Median Value

hist(bostonHousing.df$MEDV, xlab="MEDV")

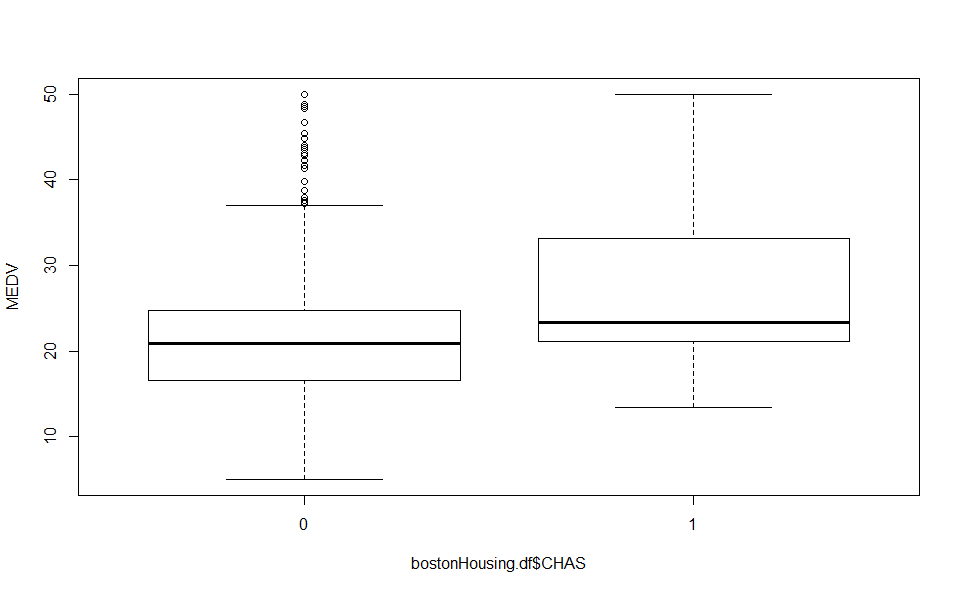


# alternative plot with ggplot  
ggplot(bostonHousing.df) + geom\_histogram(aes(MEDV), binwidth = 5)

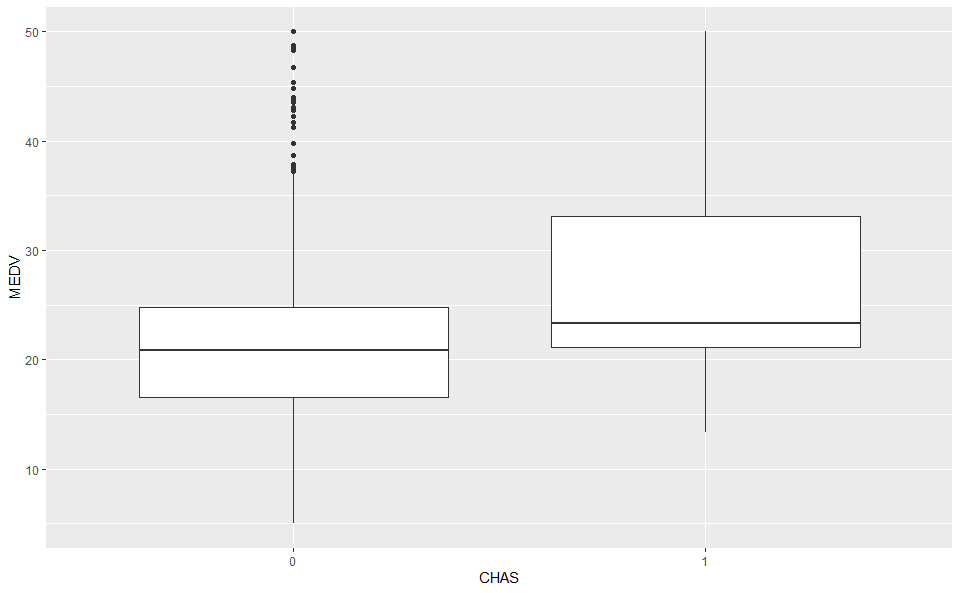


#### Boxplots of MEDV x CHAS

## boxplot of MEDV for different values of CHAS  
boxplot(bostonHousing.df$MEDV ~ bostonHousing.df$CHAS, ylab = "MEDV")

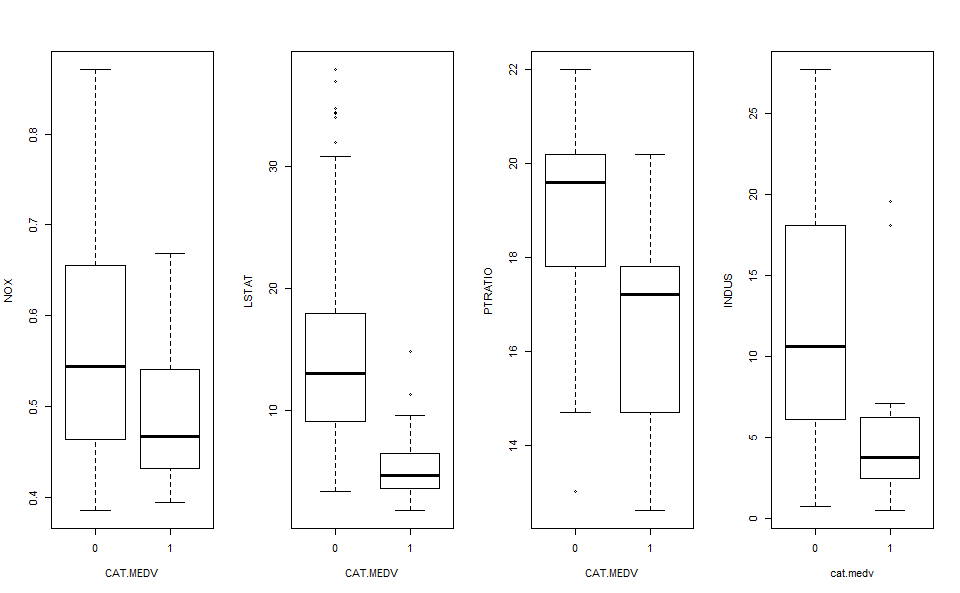


# alternative plot with ggplot  
ggplot(bostonHousing.df) + geom\_boxplot(aes(as.factor(CHAS), y = MEDV)) + xlab("CHAS")



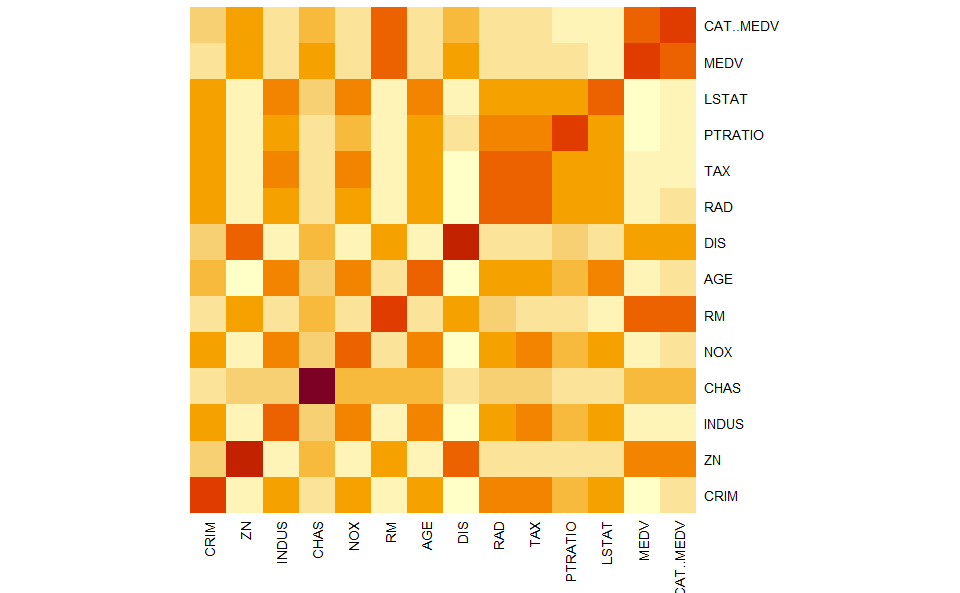
The Median value is higher when the tract bounds the Charles River.

par(mfcol = c(1, 4))  
boxplot(bostonHousing.df$NOX ~ bostonHousing.df$CAT..MEDV, xlab = "CAT.MEDV", ylab = "NOX")  
boxplot(bostonHousing.df$LSTAT ~ bostonHousing.df$CAT..MEDV, xlab = "CAT.MEDV", ylab = "LSTAT")  
boxplot(bostonHousing.df$PTRATIO ~ bostonHousing.df$CAT..MEDV, xlab = "CAT.MEDV", ylab = "PTRATIO")  
boxplot(bostonHousing.df$INDUS ~ bostonHousing.df$CAT..MEDV, xlab = "cat.medv", ylab = "INDUS")

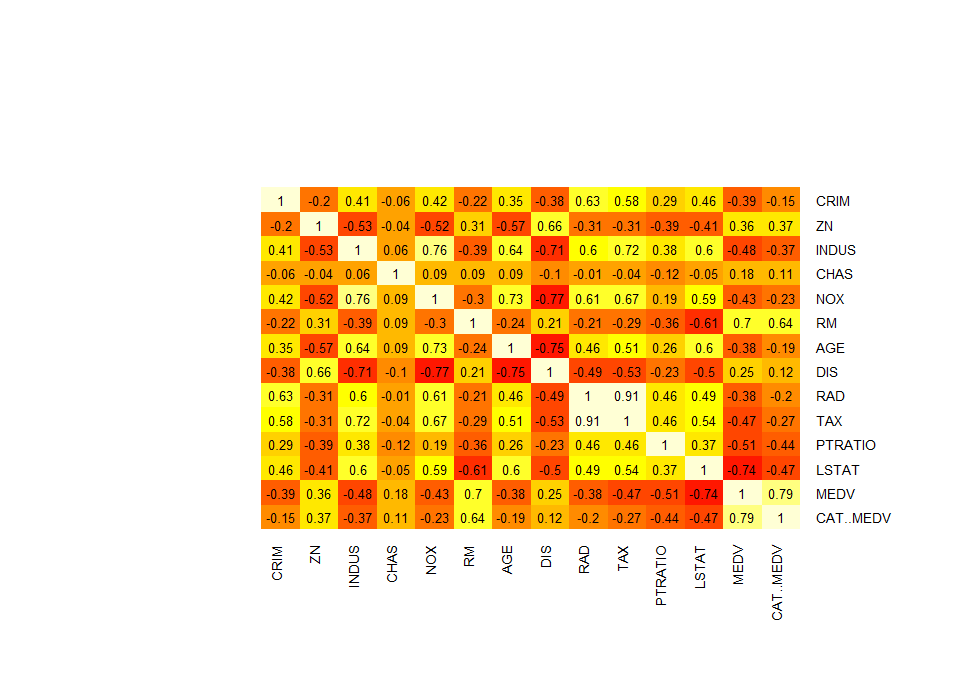


### Heatmaps: Visualizing Correlations and Missing Values

## simple heatmap of correlations (without values)  
heatmap(cor(bostonHousing.df), Rowv = NA, Colv = NA)



## heatmap with values  
heatmap.2(cor(bostonHousing.df), Rowv = FALSE, Colv = FALSE, dendrogram = "none",   
 cellnote = round(cor(bostonHousing.df),2),  
 notecol = "black", key = FALSE, trace = 'none', margins = c(10,10))



The average number of rooms is strongly correlated with the median house value per town. These variables are positvely correlated (Correlation coefficient = .7). Percentage of lower status of the population is negatively correlated with the median house value as well(Correlation coefficient = -0.74).

### Linear regression models

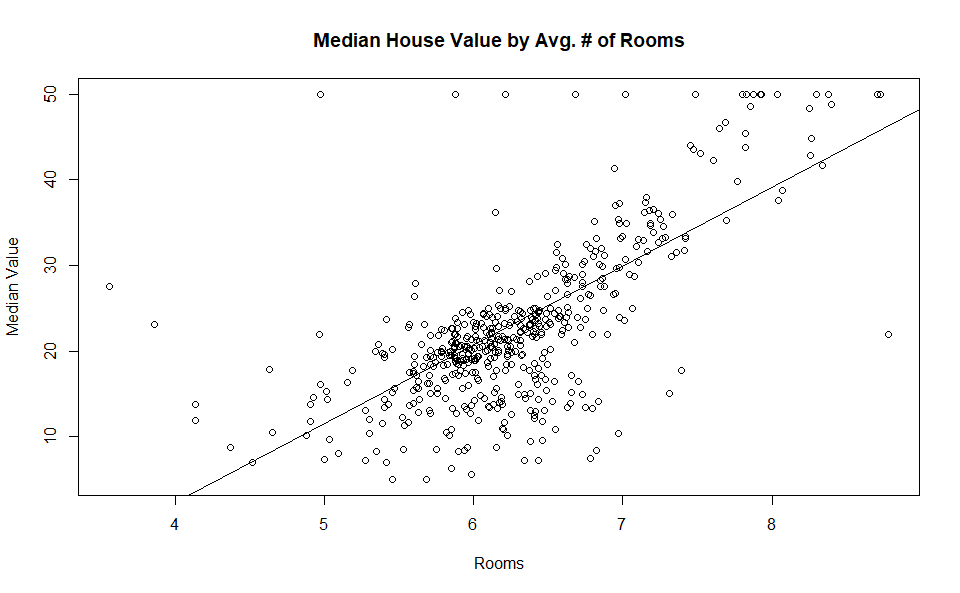
# select variables for regression  
selected.var <- c("CRIM", "CHAS", "RM", "MEDV")  
# partition data  
set.seed(111) # set seed for reproducing the partition  
train.index <- sample(row.names(bostonHousing.df), 0.6\*dim(bostonHousing.df)[1])  
valid.index <- setdiff(row.names(bostonHousing.df), train.index)  
train.df <- bostonHousing.df[train.index, selected.var]  
valid.df <- bostonHousing.df[valid.index, selected.var]  
  
# use lm() to run a linear regression of median house price (MEDV) on CRIM, CHAS, and RM.  
bh.lm <- lm(MEDV ~ ., data = train.df)  
  
### use options() to ensure numbers are not displayed in scientific notation.  
options(scipen = 999)  
summary(bh.lm)

##   
## Call:  
## lm(formula = MEDV ~ ., data = train.df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16.752 -2.786 -0.474 2.361 40.042   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -34.4126 3.4861 -9.871 < 0.0000000000000002 \*\*\*  
## CRIM -0.2669 0.0447 -5.971 0.00000000667 \*\*\*  
## CHAS 6.1577 1.4451 4.261 0.00002729752 \*\*\*  
## RM 9.1908 0.5462 16.827 < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.231 on 299 degrees of freedom  
## Multiple R-squared: 0.568, Adjusted R-squared: 0.5637   
## F-statistic: 131.1 on 3 and 299 DF, p-value: < 0.00000000000000022

### Scatterplot

Create a scatterplot with the plot() function with the following attributes:

plot(bostonHousing.df$MEDV ~ bostonHousing.df$RM, xlab = "Rooms", ylab = "Median Value",  
 main = "Median House Value by Avg. # of Rooms")  
abline(-34.4126, 9.1908)



The assignment asks the student to explore the ***BostonHousing.csv*** file and to create an R Markdown file that outputs into Microsoft Word. After using many exploratory functions to examine the data set, the student fits a multiple linear regression model to the median house price (MEDV) as a function of the CRIM (crime rate), CHAS (whether tract borders the Charles River), and RM (average number of rooms per dwelling) variables.

The dataset contains information collected by the U.S. Census Service concerning housing in Boston, Massachusetts (Boston Dataset, n.d.). The dataset contains 506 records and 14 variables. Four of the variables are of type integer, while the remaining ten are of type numeric. The average median price per house is $22,530. The minimum median price per house is $5,000, and the maximum price is $50,000. The houses, on average, contain 6.2 rooms, with a range between 3.5 - 8.78 rooms. Boxplots tell us that houses bordering the Charles River have a higher median value than houses that do not.

Additionally, we see houses with a median value greater than $30,000 have lower concentrations of nitric oxide, lower percentages of the lower status of the population, lower percentages of land occupied by nonretail business, and lower pupil-to-teacher ratios. The average number of rooms per dwelling (RM) positively correlates most strongly (+0.7) with the median price per household (MEDV). On the other hand, the percentage of the lower status of the population (LSTAT) negatively correlates most strongly (-0.74) with the median price per household (MEDV). We demonstrate this via heatmaps. The average number of rooms per dwelling was found to most strongly influence the median household value. Next was the crime rate, and lastly, whether the tract bounds the Charles River.

References

Boston Dataset. (n.d.). Retrieved March 1, 2020, from https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html