CS 6301.004 R for Data Scientists

California Housing Prices

Linear Regression

Team Members:

Omkar Nandkumar Dixit (ond170030)

Rajitha Koppisetty (rxk164330)

Vagdevi Kasina (vxk180030)

Introduction

This is the dataset used in the second chapter of Aurélien Géron's recent book 'Hands-On Machine learning with Scikit-Learn and TensorFlow'. It serves as an excellent introduction to implementing machine learning algorithms because it requires rudimentary data cleaning, has an easily understandable list of variables and sits at an optimal size between being to toyish and too cumbersome.

The data contains information from the 1990 California census.

We were able to perform some linear regression techniques on this data to make our own judgments on the same.

Content

The data pertains to the houses found in a given California district and some summary stats about them based on the 1990 census data.

Source

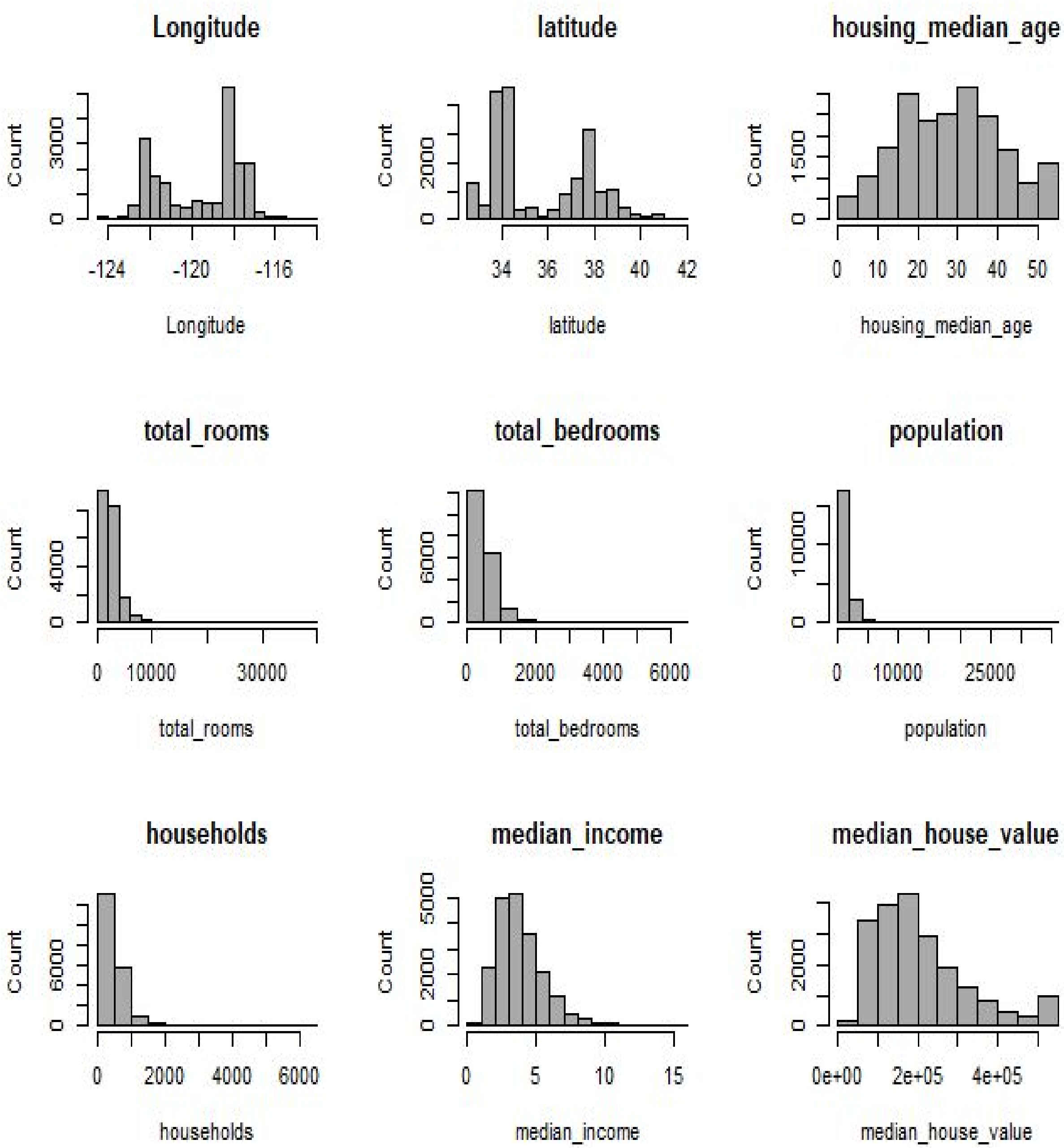
Kaggle

Summary of the Data

```
> summary(housing)
   longitude        latitude     housing_median_age  total_rooms     total_bedroms      population      households      median_income
 Min.   :-124.3   Min.   :32.54   Min.   : 1.00     Min.   :    2   Min.   :   1.0   Min.   :    3   Min.   :   1.0   Min.   : 0.4999
 1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00     1st Qu.: 1448   1st Qu.: 296.0   1st Qu.:  787   1st Qu.: 280.0   1st Qu.: 2.5634
 Median :-118.5   Median :34.26   Median :29.00     Median : 2127   Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
 Mean   :-119.6   Mean   :35.63   Mean   :28.64     Mean   : 2636   Mean   : 537.9   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
 3rd Qu.:-118.0   3rd Qu.:37.71   3rd Qu.:37.00     3rd Qu.: 3148   3rd Qu.: 647.0   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
 Max.   :-114.3   Max.   :41.95   Max.   :52.00     Max.   :39320   Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
                                                                    NA's   :207
 median_house_value   ocean_proximity
 Min.   : 14999     <1H OCEAN :9136
 1st Qu.:119600     INLAND    :6551
 Median :179700     ISLAND    :   5
 Mean   :206856     NEAR BAY  :2290
 3rd Qu.:264725     NEAR OCEAN:2658
 Max.   :500001
```

Here we can see all the features in housing data. This just gives the summary of housing data. We can visualize the data using histograms and know how different attributes have data in the dataset,

- To learn more about the data lets use a histogram.
- The Histograms for all the features is shown below:



- We can see that there are some houses with old age homes in them.
- Total_bedrooms has about 207 N/A, we will fill them with median because using mean would be a bit volatile because of the effect of the outliers

- Also rather than having total_bedrooms and total_rooms, we can convert that into average_bedrooms and average_rooms which would more informative than total
- Drop total_bedrooms and total_rooms

```
housing$total_bedrooms[is.na(housing$total_bedrooms)]                    <-
median(housing$total_bedrooms, na.rm=TRUE)

housing$avg_bedrooms <- housing$total_bedrooms/housing$households
housing$avg_rooms <- housing$total_rooms/housing$households

housing    <-    housing[,    !(names(housing)    %in%    c('total_bedrooms',
'total_rooms'))]
```

- If we look at the structure of the housing data, we can see that ocean_proximity is a Factor and from the summary above we can see that there are about 5 types, NEAR BAY, <1H OCEAN, INLAND, NEAR OCEAN ISLAND

```
> str(housing)
'data.frame':   20640 obs. of  10 variables:
 $ longitude         : num  -122 -122 -122 -122 -122 ...
 $ latitude          : num  37.9 37.9 37.9 37.9 37.9 ...
 $ housing_median_age: num  41 21 52 52 52 52 52 52 42 52 ...
 $ population        : num  322 2401 496 558 565 ...
 $ households        : num  126 1138 177 219 259 ...
 $ median_income     : num  8.33 8.3 7.26 5.64 3.85 ...
 $ median_house_value: num  452600 358500 352100 341300 342200 ...
 $ ocean_proximity   : Factor w/ 5 levels "<1H OCEAN","INLAND",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ avg_bedrooms      : num  1.024 0.972 1.073 1.073 1.081 ...
 $ avg_rooms         : num  6.98 6.24 8.29 5.82 6.28 ...
```

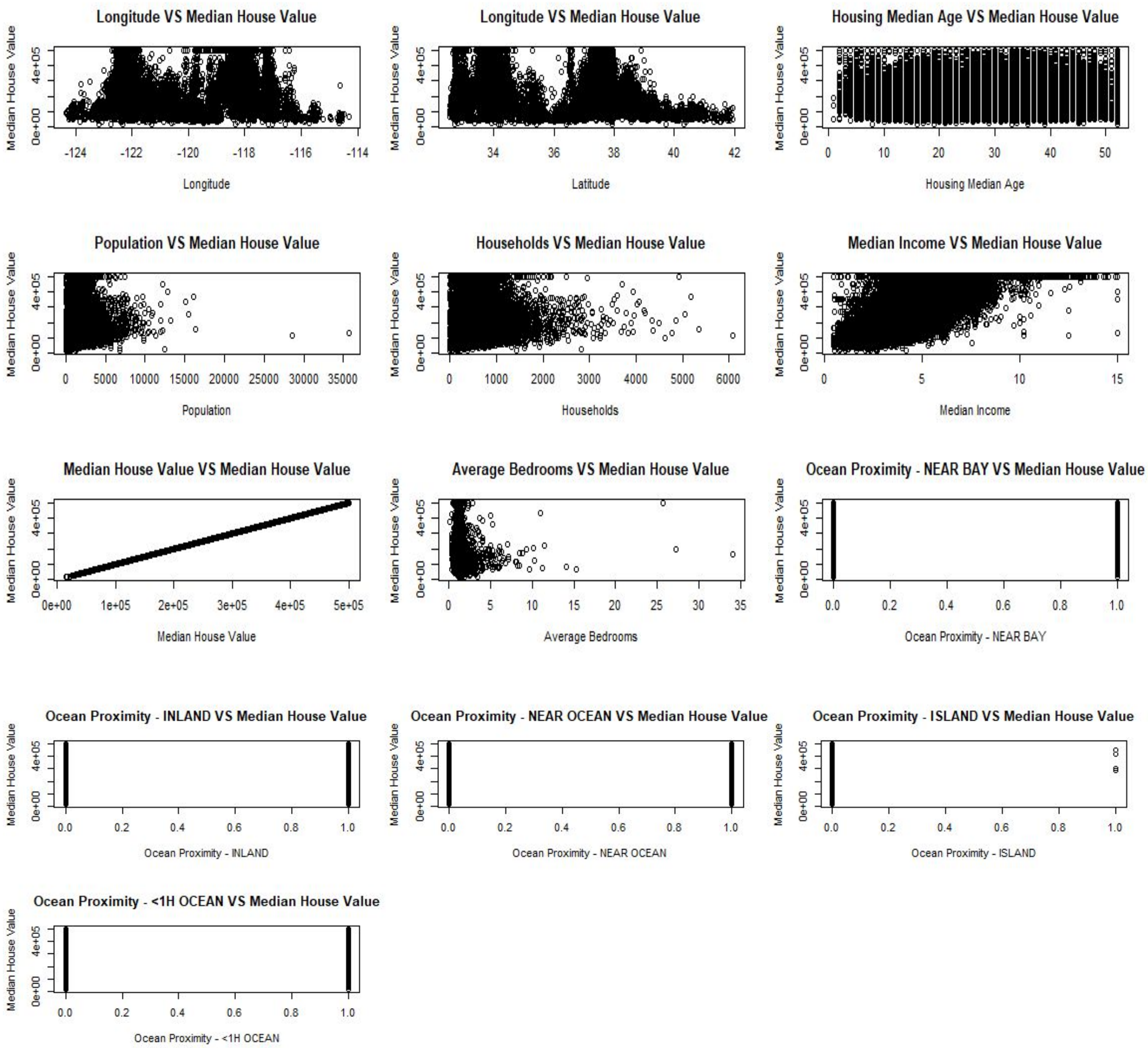
- Let's create a separate column for each of these types and then we can use it as a boolean column, and then drop ocean_proximity

```r
for(c in categories){
  housing[,c] <- rep(0, times=nrow(housing))
}
for (i in 1:nrow(housing)){
  c <- as.character(housing$ocean_proximity[i])
  housing[, c][i] <- 1
}
```

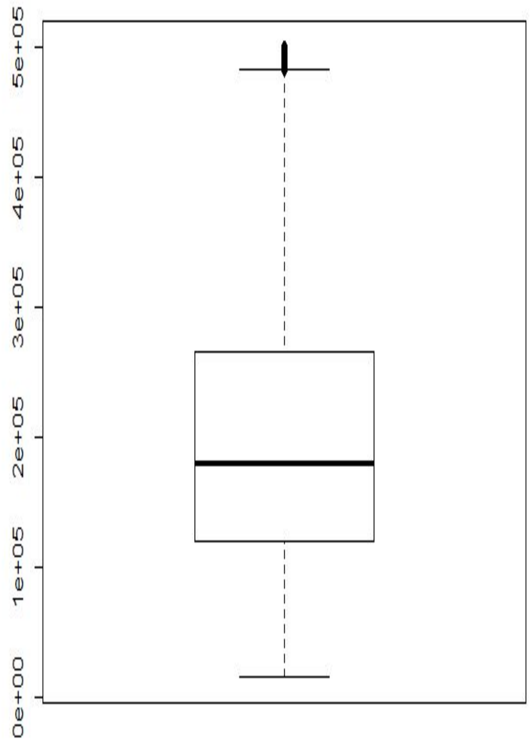
- Now if we take a look at our columns it will be something like

```
> colnames(housing)
 [1] "longitude"          "latitude"           "housing_median_age" "population"         "households"         "median_income"
 [7] "median_house_value" "avg_bedrooms"       "avg_rooms"          "NEAR BAY"           "<1H OCEAN"          "INLAND"
[13] "NEAR OCEAN"         "ISLAND"
```

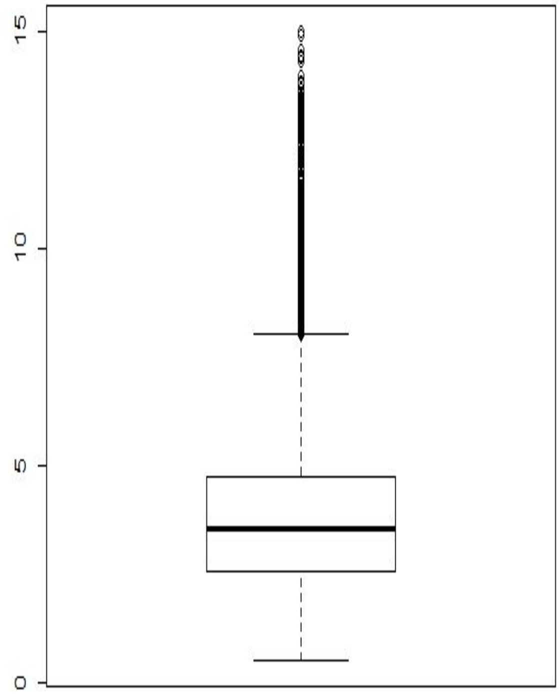
- We used scatterplots to check what predicting variables have a linear relationship with median_house_value



- Looks like median_income is the only feature with a linear relationship with median_house_value
- To identify outliers, let's use boxplots to find out outliers if any



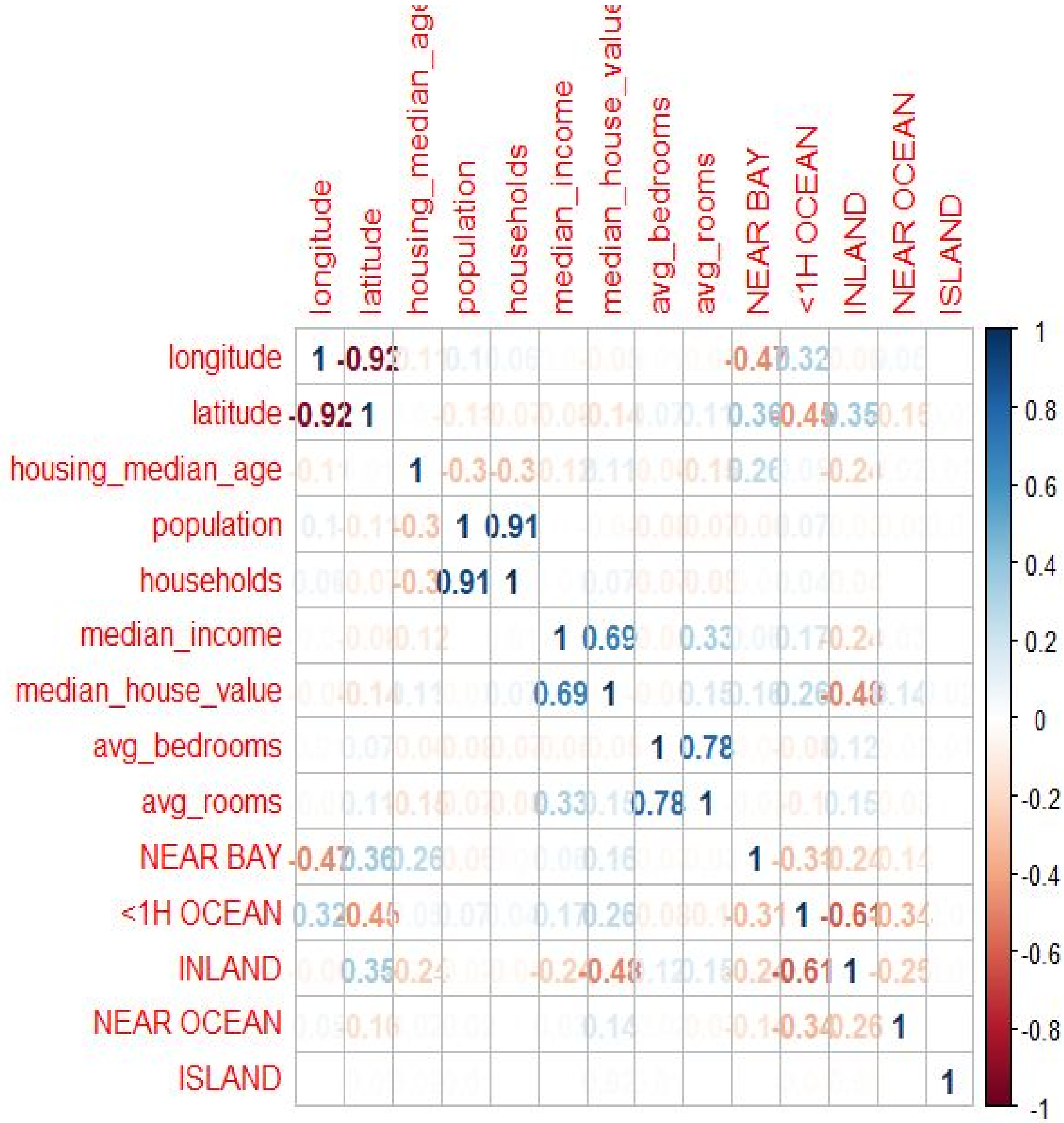
Boxplot (Median House Value)



Boxplot (Median Income)

- The data has a good amount of outliers
- Now that we know median house value is the only feature with a linear relation with median income which was very evident from scatterplots, we can use a correlation plot to show us a correlation with the output variable
- First, we create a corrplot that shows the correlation of each variable with all others

```
corMat <- as.data.frame(corrplot(cor(housing),method = "number"))
```



- To train a model its better to have predictor which has more than 50% of correlation with the output variable

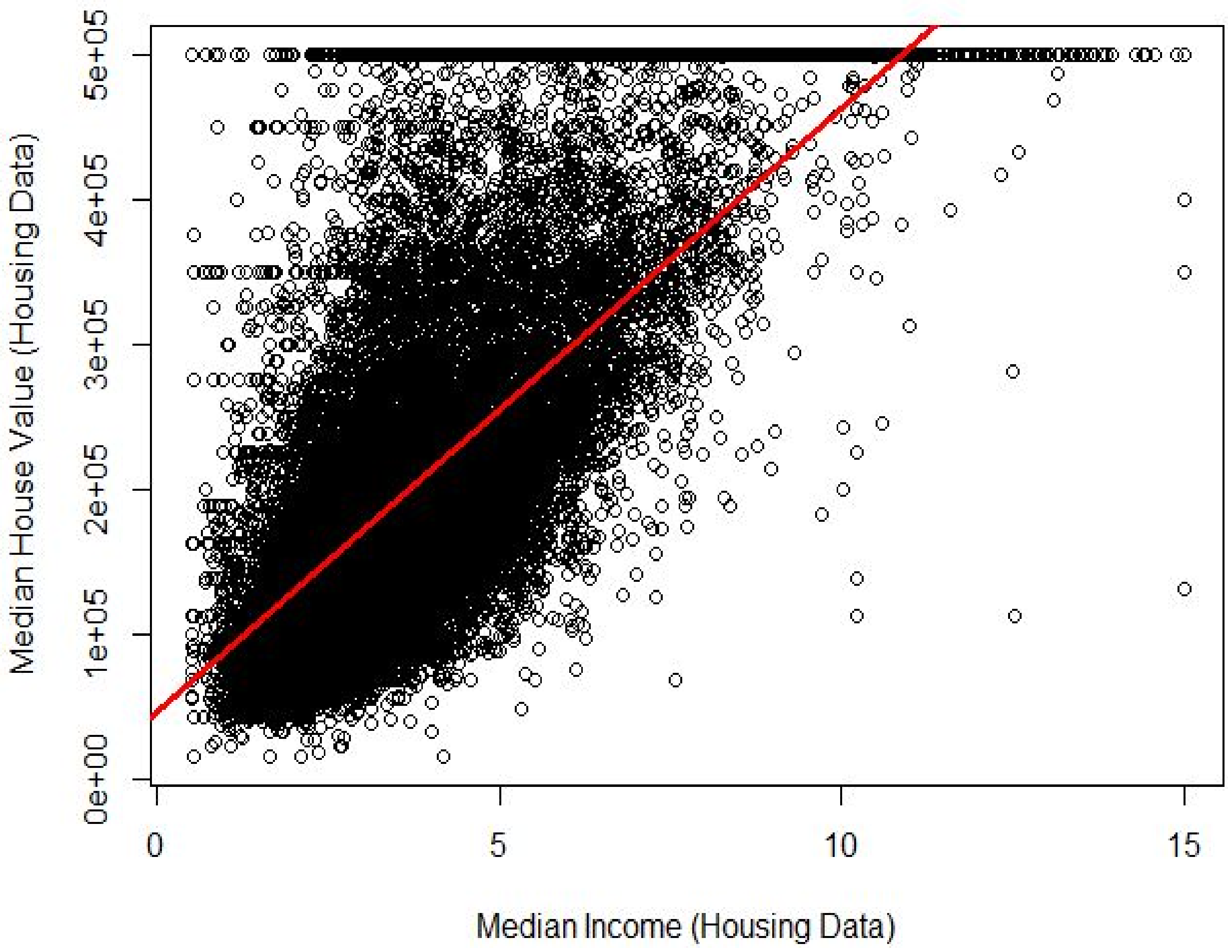
```
> row.names(corMat)[abs(corMat$median_house_value) > 0.50]
[1] "median_income"      "median_house_value"
```

- It's confirmed that median_income will be the best predictor for median_house_value, which has a linear relationship with median_house_value and also has more than 50% of correlation with median_house_value

Model 1

```
> lm1.fit <- lm(median_house_value~median_income, data=housing)
> lm1.fit

Call:
lm(formula = median_house_value ~ median_income, data = housing)

Coefficients:
  (Intercept)  median_income
        45086          41794

> summary(lm1.fit)

Call:
lm(formula = median_house_value ~ median_income, data = housing)

Residuals:
    Min      1Q  Median      3Q     Max
-540697  -55950  -16979   36978  434023

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    45085.6     1322.9   34.08   <2e-16 ***
median_income  41793.8      306.8  136.22   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 83740 on 20638 degrees of freedom
Multiple R-squared:  0.4734,  Adjusted R-squared:  0.4734
F-statistic: 1.856e+04 on 1 and 20638 DF,  p-value: < 2.2e-16

> names(lm1.fit)
  [1]  "coefficients"    "residuals"          "effects"              "rank"
"fitted.values" "assign"        "qr"             "df.residual"
```

```
 [9] "xlevels"       "call"          "terms"         "model"
> confint(lm1.fit)
                 2.5 %   97.5 %
(Intercept)   42492.64 47678.51
median_income 41192.49 42395.21
> plot(housing$median_income, housing$median_house_value)
> abline(lm1.fit, lwd=3, col="red")
```

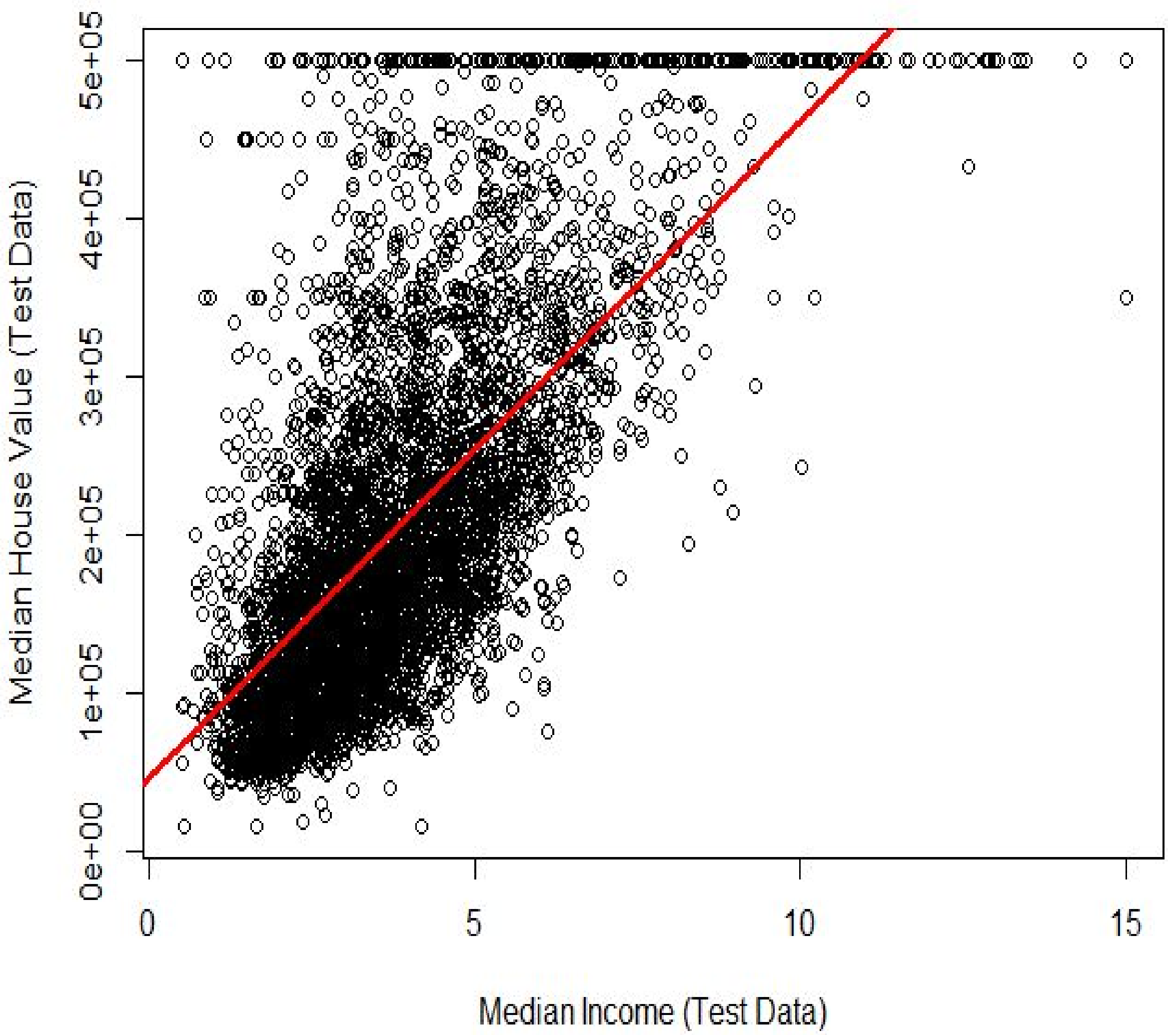


Model 2

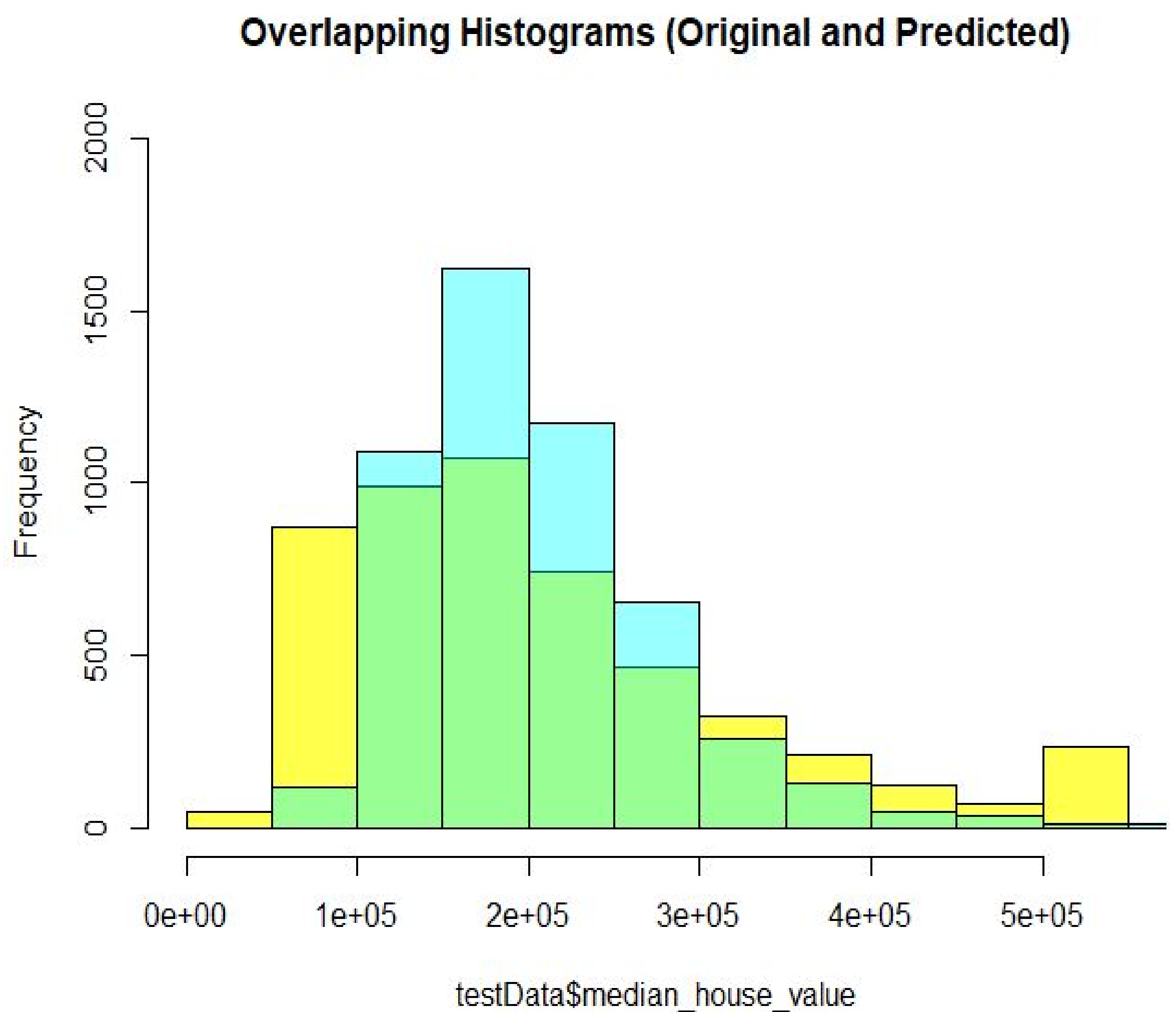
- In this model, we decide to divide the model into train and test data of 75% and 25% respectively

```
> # This will be used as to calculate sample size what is 75% for now to use it as training Data and rest as testing Data
> sample.size <- floor(0.75 * nrow(housing))
>
> # Setting seed will make sure you get some random numbers generated
> set.seed(123)
>
> # Stores Random rownumbers in trainIndices
> trainIndices <- sample(seq_len(nrow(housing)), size=sample.size)
>
> # Creates training dataset with row numbers stored in trainIndices
> trainData <- housing[trainIndices,]
>
> # All the ones excluding the ones in trainIndices are stored as testing Data
> testData <- housing[-trainIndices, ]
>
> lm2.fit <- lm(median_house_value ~ median_income, data=trainData)
> lm2.fit

Call:
lm(formula = median_house_value ~ median_income, data = trainData)

Coefficients:
  (Intercept)  median_income
        45499          41628

> summary(lm2.fit)

Call:
lm(formula = median_house_value ~ median_income, data = trainData)

Residuals:
    Min      1Q  Median      3Q     Max
-538629  -55936  -16821   37543  433692
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    45498.7     1515.9   30.01   <2e-16 ***
median_income  41628.4      350.4  118.80   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 83510 on 15478 degrees of freedom
Multiple R-squared:  0.4769,  Adjusted R-squared:  0.4769
F-statistic: 1.411e+04 on 1 and 15478 DF,  p-value: < 2.2e-16

> preds <- predict(lm2.fit, testData)
> mse <- mse(housing$median_house_value, preds)
> root_mse <- sqrt(mse)
> plot(testData$median_income, testData$median_house_value)
> abline(lm2.fit, lwd=3, col="red")
>   hist(testData$median_house_value,   ylim   =   c(0,   2000),   col   =
rgb(1,1,0,0.7), main = "Overlapping Histograms (Original and Predicted)")
> hist(preds,col=rgb(0,1,1,0.4), add=T)
>
```



Median House Value (Test Data)
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
0
5
10
15
Median Income (Test Data)



Conclusion

From both the models we can see that the p-value is less than 0.05 so we reject the null Hypothesis which said that there is no relationship between the two variables, and we accept the alternate hypothesis concluding that there is some relationship between the variables.

References

https://swcarpentry.github.io/r-novice-inflammation/11-supp-read-write-csv/

http://web.utk.edu/~wfeng1/html/pre.html