Project

Abu Usama

2023-08-16

Step 1

Data Collection

We're using the California Housing Prices dataset (housing.csv) from the following Kaggle site: Click Here. This data pertains to the houses found in a given California district and some summary stats about them based on the 1990 census data.
The dataset contains 20640 observations and 10 attributes. Below is a list of the variables with descriptions taken from the original Kaggle site given above.

> longitude: A measure of how far west a house is; a higher value is farther west
> latitude: A measure of how far north a house is; a higher value is farther north
> housing_median_age: Median age of a house within a block; a lower number is a newer building
> total_rooms: Total number of rooms within a block
> total_bedrooms: Total number of bedrooms within a block
> population: Total number of people residing within a block
> households: Total number of households, a group of people residing within a home unit, for a block
> median_income: Median income for households within a block of houses (measured in tens of thousands of US Dollars)
> ocean_proximity: Location of the house w.r.t ocean/sea
> median_house_value: Median house value for households within a block (measured in US Dollars)

```r
library(readr)
housing <- read_csv("housing.csv")
```

```
## Rows: 20640 Columns: 10
## ── Column specification ────────────────────────────────────────────────
──────
## Delimiter: ","
## chr (1): ocean_proximity
## dbl (9): longitude, latitude, housing_median_age, total_rooms, total_bedro
om...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this m
essage.
```

```
View(housing)
data <- housing
```

Step 2

Data Preprocessing:

```
print(colnames(data))
```

```
##  [1] "longitude"          "latitude"           "housing_median_age"
##  [4] "total_rooms"        "total_bedrooms"     "population"
##  [7] "households"         "median_income"      "median_house_value"
## [10] "ocean_proximity"
```

2.1.1 Overview of Missing Values

```
null_counts <- colSums(is.na(data))
print(null_counts)
```

```
##          longitude           latitude housing_median_age        total_room
s
##                  0                  0                  0
0
##     total_bedrooms         population         households      median_incom
e
##                207                  0                  0
0
## median_house_value    ocean_proximity
##                  0                  0
```

2.1.2 Dealing with Missing Values Through Median Imputation

```
data$total_bedrooms[is.na(data$total_bedrooms)] <- median(data$total_bedrooms
, na.rm = TRUE)
null_counts <- colSums(is.na(data))
print(null_counts)
```

```
##          longitude           latitude housing_median_age        total_room
s
##                  0                  0                  0
0
##     total_bedrooms         population         households      median_incom
e
##                  0                  0                  0
0
## median_house_value    ocean_proximity
##                  0                  0
```

2.2.1 Encode Categorical Variables

```
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
```

```
unique_values <- unique(data$ocean_proximity)
print(unique_values)
```

```
## [1] "NEAR BAY"   "<1H OCEAN"  "INLAND"     "NEAR OCEAN" "ISLAND"
```

```
label_dict <- setNames(1:length(unique_values), unique_values)
data_encoded <- data %>%
  mutate(ocean_proximity_encoded = label_dict[ocean_proximity])
data <- data_encoded
```

```
unique_values <- unique(data$ocean_proximity_encoded)
print(unique_values)
```

```
## [1] 1 2 3 4 5
```

2.3.1 Mathmatical Transformation / Add New variables

```
data$room_par_houshold <- data$total_rooms / data$households
data$bedroomd_per_room <- data$total_bedrooms / data$total_rooms
data$population_per_houshold <- data$population / data$households
print(colnames(data))
```

```
##  [1] "longitude"               "latitude"
##  [3] "housing_median_age"      "total_rooms"
##  [5] "total_bedrooms"          "population"
##  [7] "households"              "median_income"
##  [9] "median_house_value"      "ocean_proximity"
## [11] "ocean_proximity_encoded" "room_par_houshold"
## [13] "bedroomd_per_room"       "population_per_houshold"
```

Step 3

Exploratory Data Analysis (EDA)

```
summary(data)
```

```
##    longitude         latitude     housing_median_age  total_rooms
##  Min.   :-124.3   Min.   :32.54   Min.   : 1.00      Min.   :    2
##  1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00      1st Qu.: 1448
##  Median :-118.5   Median :34.26   Median :29.00      Median : 2127
```

```
##  Mean   :-119.6   Mean   :35.63   Mean   :28.64      Mean   : 2636
##  3rd Qu.:-118.0   3rd Qu.:37.71   3rd Qu.:37.00      3rd Qu.: 3148
##  Max.   :-114.3   Max.   :41.95   Max.   :52.00      Max.   :39320
##  total_bedrooms     population      households     median_income
##  Min.   :   1.0   Min.   :    3   Min.   :   1.0   Min.   : 0.4999
##  1st Qu.: 297.0   1st Qu.:  787   1st Qu.: 280.0   1st Qu.: 2.5634
##  Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
##  Mean   : 536.8   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
##  3rd Qu.: 643.2   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
##  Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
##  median_house_value ocean_proximity    ocean_proximity_encoded
##  Min.   : 14999     Length:20640       Min.   :1.000
##  1st Qu.:119600     Class :character   1st Qu.:2.000
##  Median :179700     Mode  :character   Median :2.000
##  Mean   :206856                        Mean   :2.465
##  3rd Qu.:264725                        3rd Qu.:3.000
##  Max.   :500001                        Max.   :5.000
##  room_par_houshold  bedroomd_per_room population_per_houshold
##  Min.   :  0.8461   Min.   :0.03715   Min.   :   0.6923
##  1st Qu.:  4.4407   1st Qu.:0.17522   1st Qu.:   2.4297
##  Median :  5.2291   Median :0.20316   Median :   2.8181
##  Mean   :  5.4290   Mean   :0.21379   Mean   :   3.0707
##  3rd Qu.:  6.0524   3rd Qu.:0.24013   3rd Qu.:   3.2823
##  Max.   :141.9091   Max.   :2.82468   Max.   :1243.3333
```

```
glimpse(data)
```

```
## Rows: 20,640
## Columns: 14
## $ longitude               <dbl> -122.23, -122.22, -122.24, -122.25, -122.2
5, -…
## $ latitude                <dbl> 37.88, 37.86, 37.85, 37.85, 37.85, 37.85,
37.8…
## $ housing_median_age      <dbl> 41, 21, 52, 52, 52, 52, 52, 52, 42, 52, 52
, 52…
## $ total_rooms             <dbl> 880, 7099, 1467, 1274, 1627, 919, 2535, 31
04, …
## $ total_bedrooms          <dbl> 129, 1106, 190, 235, 280, 213, 489, 687, 6
65, …
## $ population              <dbl> 322, 2401, 496, 558, 565, 413, 1094, 1157,
120…
## $ households              <dbl> 126, 1138, 177, 219, 259, 193, 514, 647, 5
95, …
## $ median_income           <dbl> 8.3252, 8.3014, 7.2574, 5.6431, 3.8462, 4.
0368…
## $ median_house_value      <dbl> 452600, 358500, 352100, 341300, 342200, 26
9700…
## $ ocean_proximity         <chr> "NEAR BAY", "NEAR BAY", "NEAR BAY", "NEAR
BAY"…
## $ ocean_proximity_encoded <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
1, 1…
## $ room_par_houshold       <dbl> 6.984127, 6.238137, 8.288136, 5.817352, 6.
2818…
## $ bedroomd_per_room       <dbl> 0.1465909, 0.1557966, 0.1295160, 0.1844584
, 0.…
## $ population_per_houshold <dbl> 2.555556, 2.109842, 2.802260, 2.547945, 2.
1814…
```

```
# summarize with dplyr packages
data %>%
  group_by(ocean_proximity) %>%
  summarise(mean(median_house_value), sd(median_house_value), min(median_hous
e_value), max(median_house_value))
```

```
## # A tibble: 5 × 5
##   ocean_proximity `mean(median_house_value)` `sd(median_house_value)`
##   <chr>                                <dbl>                    <dbl>
## 1 <1H OCEAN                          240084.                  106124.
## 2 INLAND                             124805.                   70008.
## 3 ISLAND                             380440                    80560.
## 4 NEAR BAY                           259212.                  122819.
## 5 NEAR OCEAN                         249434.                  122477.
## # ℹ 2 more variables: `min(median_house_value)` <dbl>,
## #   `max(median_house_value)` <dbl>
```

```
class(data)
```

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

```
head(data)
```

```
## # A tibble: 6 × 14
##   longitude latitude housing_median_age total_rooms total_bedrooms populat
ion
##       <dbl>    <dbl>              <dbl>       <dbl>          <dbl>      <d
bl>
## 1     -122.     37.9                 41         880            129
322
## 2     -122.     37.9                 21        7099           1106       2
401
## 3     -122.     37.8                 52        1467            190
496
## 4     -122.     37.8                 52        1274            235
558
## 5     -122.     37.8                 52        1627            280
565
## 6     -122.     37.8                 52         919            213
413
## # ℹ 8 more variables: households <dbl>, median_income <dbl>,
## #   median_house_value <dbl>, ocean_proximity <chr>,
```

```
## #   ocean_proximity_encoded <int>, room_par_houshold <dbl>,
## #   bedroomd_per_room <dbl>, population_per_houshold <dbl>
```

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

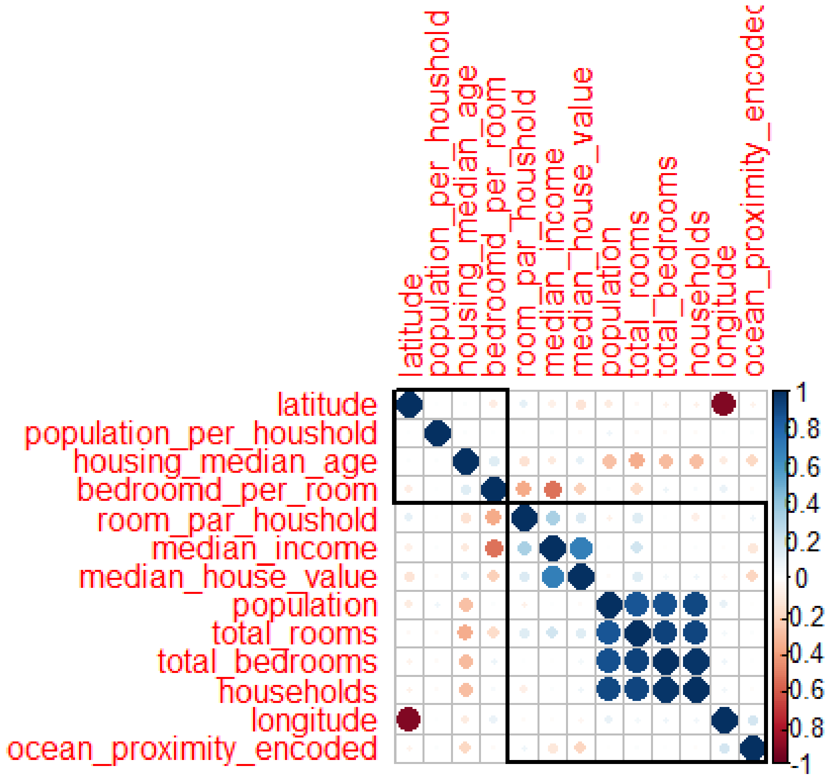
Convert the correlation matrix to a data frame

```r
numeric_df <- data %>%
  select_if(is.numeric)
m<-cor(numeric_df,method="kendall")
m <- cor(numeric_df)
m
```

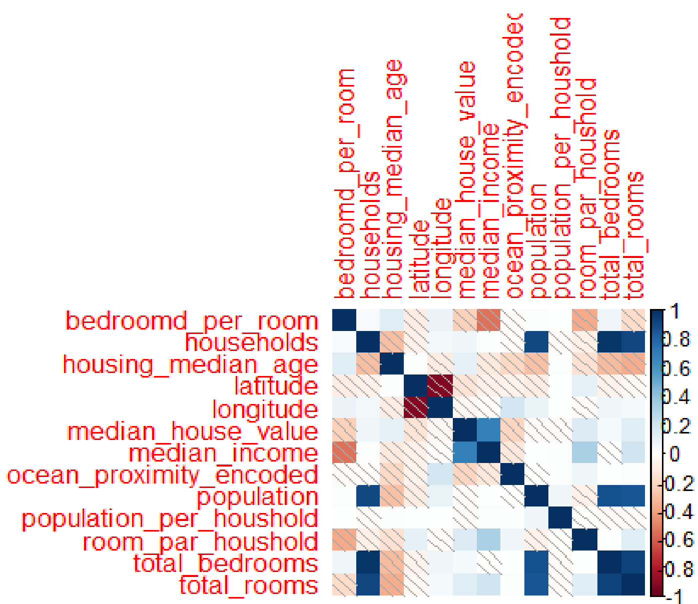
```
##                            longitude     latitude housing_median_age
## longitude                1.000000000 -0.924664434        -0.10819681
## latitude                -0.924664434  1.000000000         0.01117267
## housing_median_age      -0.108196813  0.011172674         1.00000000
## total_rooms              0.044567978 -0.036099596        -0.36126220
## total_bedrooms           0.069119698 -0.066483906        -0.31902633
## population               0.099773223 -0.108784747        -0.29624424
## households               0.055310093 -0.071035433        -0.30291601
## median_income           -0.015175865 -0.079809127        -0.11903399
## median_house_value      -0.045966615 -0.144160277         0.10562341
## ocean_proximity_encoded  0.180381158 -0.067585629        -0.20488238
## room_par_houshold       -0.027540054  0.106388965        -0.15327742
## bedroomd_per_room        0.081204771 -0.098618701         0.13562155
## population_per_houshold  0.002475816  0.002366182         0.01319136
##                         total_rooms total_bedrooms   population  household
s
## longitude                0.04456798    0.069119698  0.099773223  0.0553100
9
## latitude                -0.03609960   -0.066483906 -0.108784747 -0.0710354
3
## housing_median_age      -0.36126220   -0.319026332 -0.296244240 -0.3029160
1
## total_rooms              1.00000000    0.927058197  0.857125973  0.9184844
9
## total_bedrooms           0.92705820    1.000000000  0.873534861  0.9743662
9
## population               0.85712597    0.873534861  1.000000000  0.9072222
7
## households               0.91848449    0.974366294  0.907222266  1.0000000
0
## median_income            0.19804965   -0.007616874  0.004834346  0.0130330
5
## median_house_value       0.13415311    0.049456862 -0.024649679  0.0658426
5
## ocean_proximity_encoded  0.01481817    0.004075811 -0.008510881 -0.0169111
6
```

```
## room_par_houshold        0.13379843    0.001764969 -0.072212849 -0.0805977
1
## bedroomd_per_room       -0.18738114    0.071648570  0.010035266  0.0344980
6
## population_per_houshold -0.02458066   -0.028325168  0.069862730 -0.0273093
6
##                         median_income median_house_value
## longitude                -0.015175865        -0.04596662
## latitude                 -0.079809127        -0.14416028
## housing_median_age       -0.119033990         0.10562341
## total_rooms               0.198049645         0.13415311
## total_bedrooms           -0.007616874         0.04945686
## population                0.004834346        -0.02464968
## households                0.013033052         0.06584265
## median_income             1.000000000         0.68807521
## median_house_value        0.688075208         1.00000000
## ocean_proximity_encoded  -0.129135252        -0.21060028
## room_par_houshold         0.326895432         0.15194829
## bedroomd_per_room        -0.545297903        -0.23330293
## population_per_houshold   0.018766248        -0.02373741
##                         ocean_proximity_encoded room_par_houshold
## longitude                           0.180381158      -0.027540054
## latitude                           -0.067585629       0.106388965
## housing_median_age                 -0.204882385      -0.153277423
## total_rooms                         0.014818167       0.133798431
## total_bedrooms                      0.004075811       0.001764969
## population                         -0.008510881      -0.072212849
## households                         -0.016911158      -0.080597714
## median_income                      -0.129135252       0.326895432
## median_house_value                 -0.210600282       0.151948290
## ocean_proximity_encoded             1.000000000       0.066124225
## room_par_houshold                   0.066124225       1.000000000
## bedroomd_per_room                  -0.025433547      -0.370308327
## population_per_houshold             0.010449287      -0.004852295
##                         bedroomd_per_room population_per_houshold
## longitude                     0.081204771             0.002475816
## latitude                     -0.098618701             0.002366182
## housing_median_age            0.135621546             0.013191357
## total_rooms                  -0.187381141            -0.024580659
## total_bedrooms                0.071648570            -0.028325168
## population                    0.010035266             0.069862730
## households                    0.034498063            -0.027309356
## median_income                -0.545297903             0.018766248
## median_house_value           -0.233302927            -0.023737413
## ocean_proximity_encoded      -0.025433547             0.010449287
## room_par_houshold            -0.370308327            -0.004852295
## bedroomd_per_room             1.000000000             0.002600952
## population_per_houshold       0.002600952             1.000000000
```

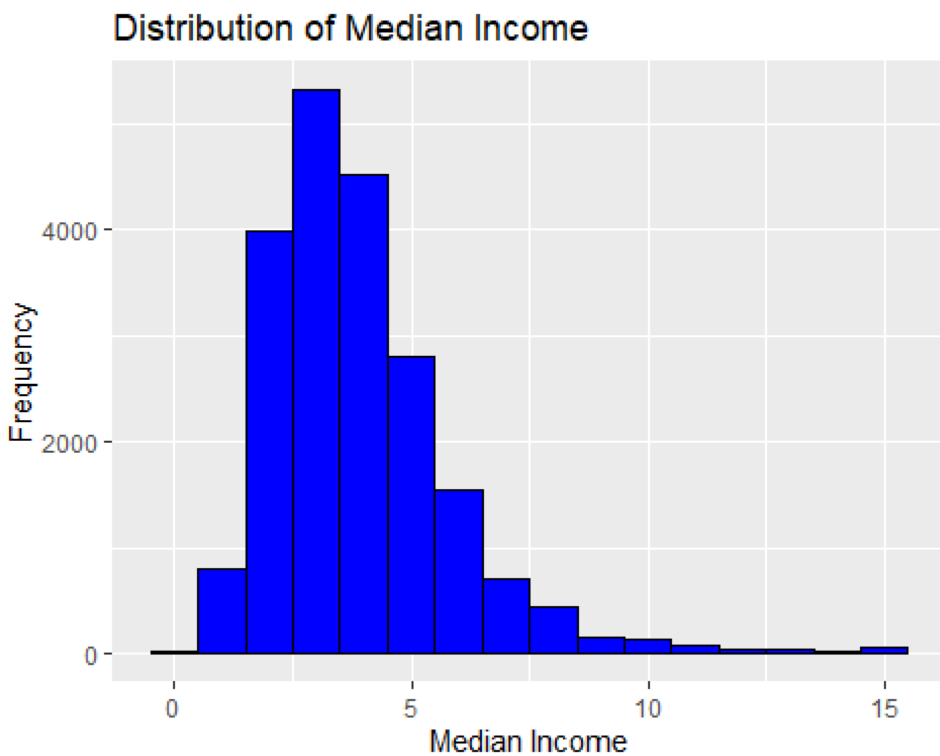
```r
corrplot(m,order="hclust",addrect=2)
```



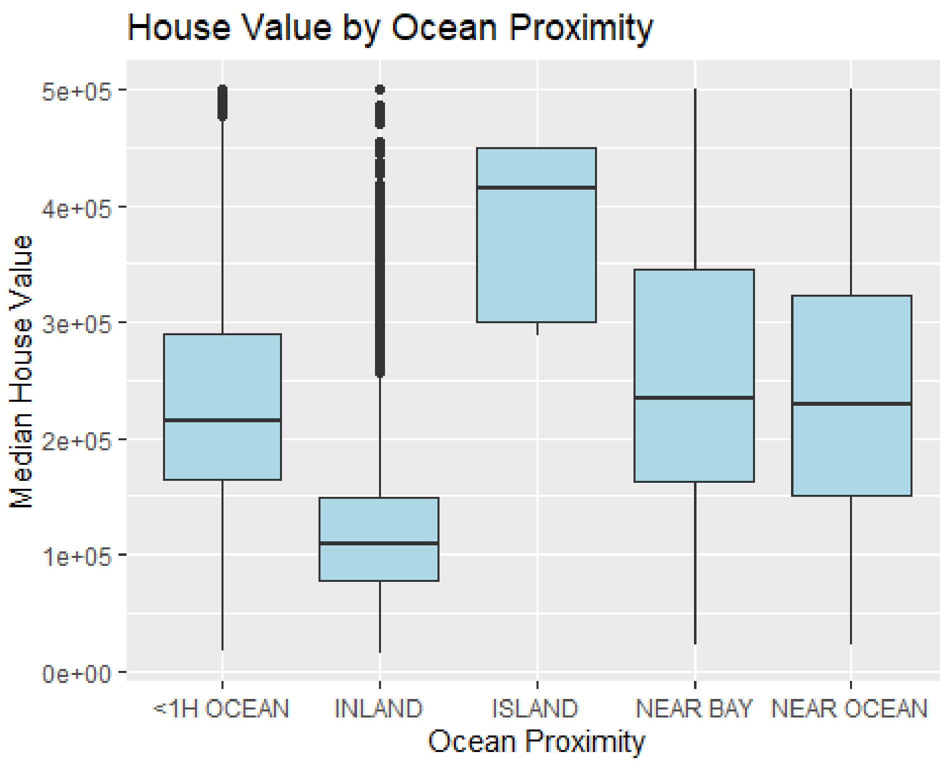

```r
# Specify a dark shade color
dark_shade_color <- "darkgray"

# Create the correlation plot with a dark shade
corrplot(m, method = "shade", order = "alphabet", shade.col = dark_shade_colo
r)
```



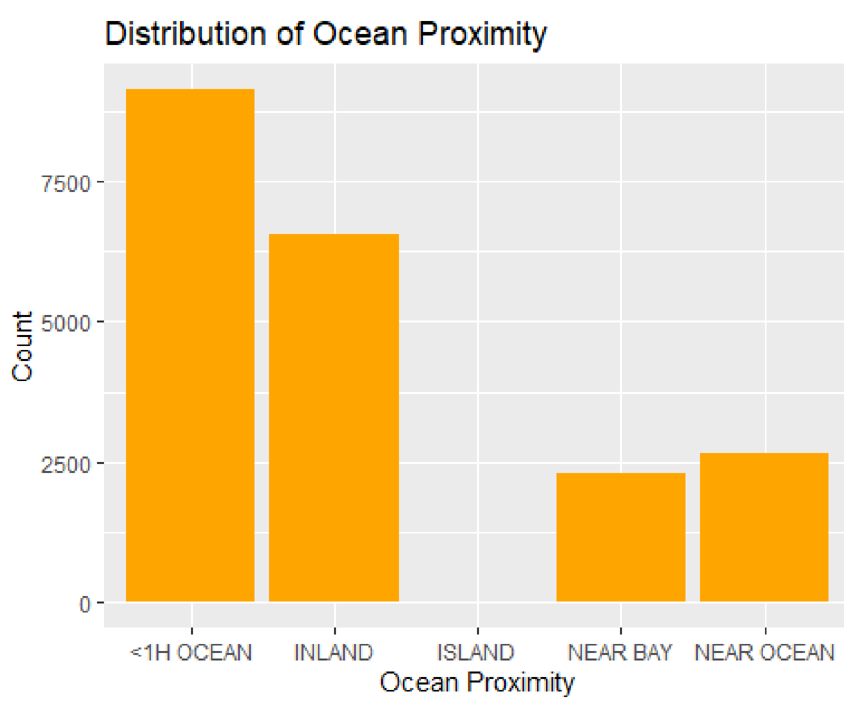
```r
# Load necessary library
library(ggplot2)

# Plot histograms for numeric variables
ggplot(data, aes(x = median_income)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  labs(title = "Distribution of Median Income", x = "Median Income", y = "Fre
quency")
```



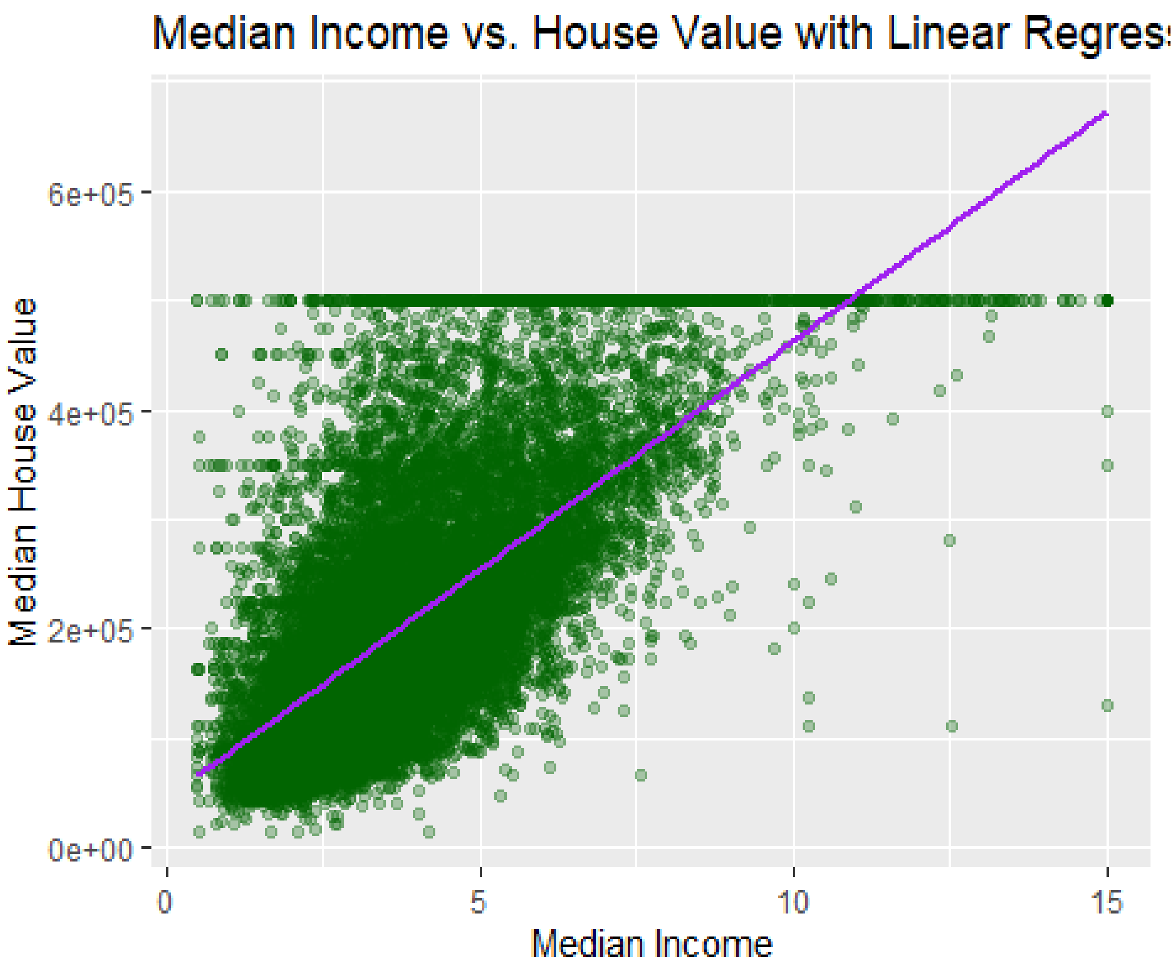

```r
# Plot box plots for 'ocean_proximity_encoded' vs. 'median_house_value'
ggplot(data, aes(x = data$ocean_proximity, y = median_house_value)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "House Value by Ocean Proximity", x = "Ocean Proximity", y = "
Median House Value")
```



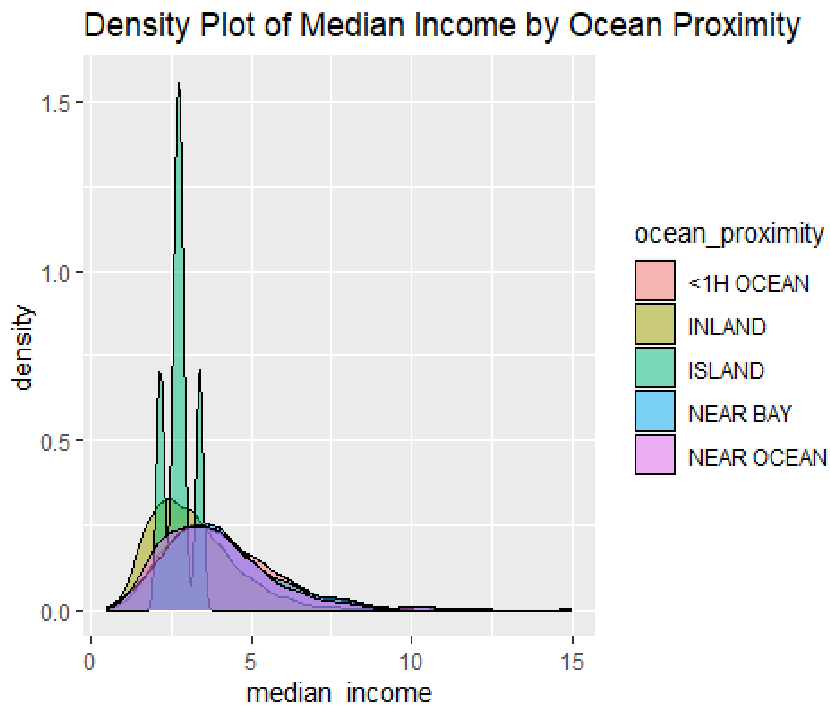

```
# Bar plot for 'ocean_proximity_encoded'
ggplot(data, aes(x = (ocean_proximity))) +
  geom_bar(fill = "orange") +
  labs(title = "Distribution of Ocean Proximity", x = "Ocean Proximity", y =
"Count")
```



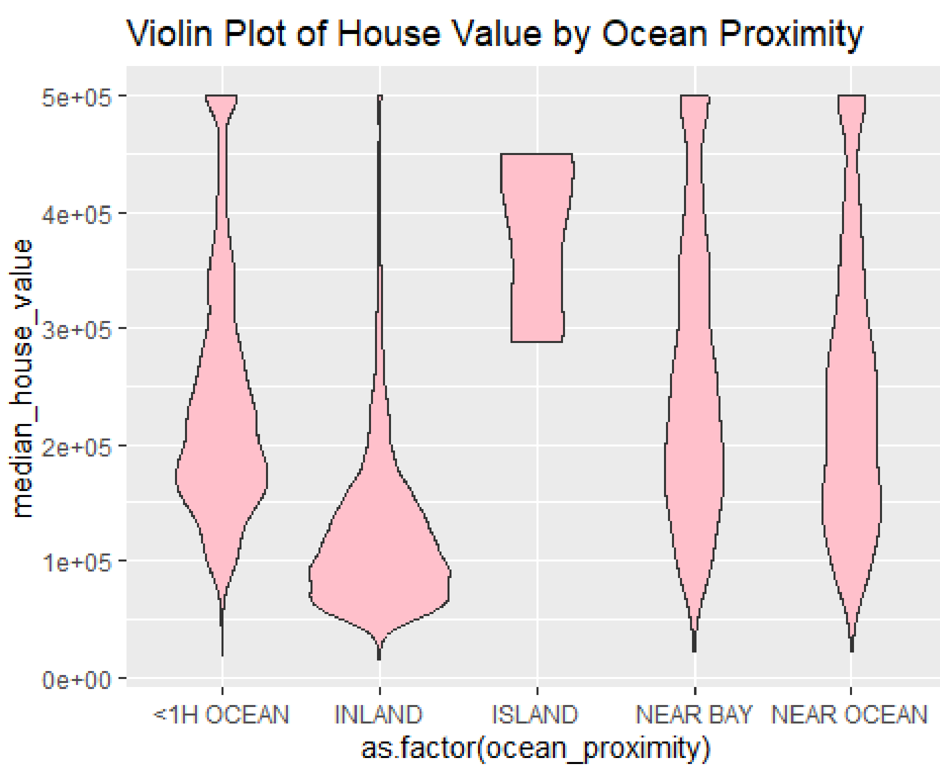
```r
# Scatter plot with linear regression line
ggplot(data, aes(x = median_income, y = median_house_value)) +
  geom_point(alpha = 0.3, color = "darkgreen") +
  geom_smooth(method = "lm", se = FALSE, color = "purple") +
  labs(title = "Median Income vs. House Value with Linear Regression Line",
       x = "Median Income", y = "Median House Value")
```



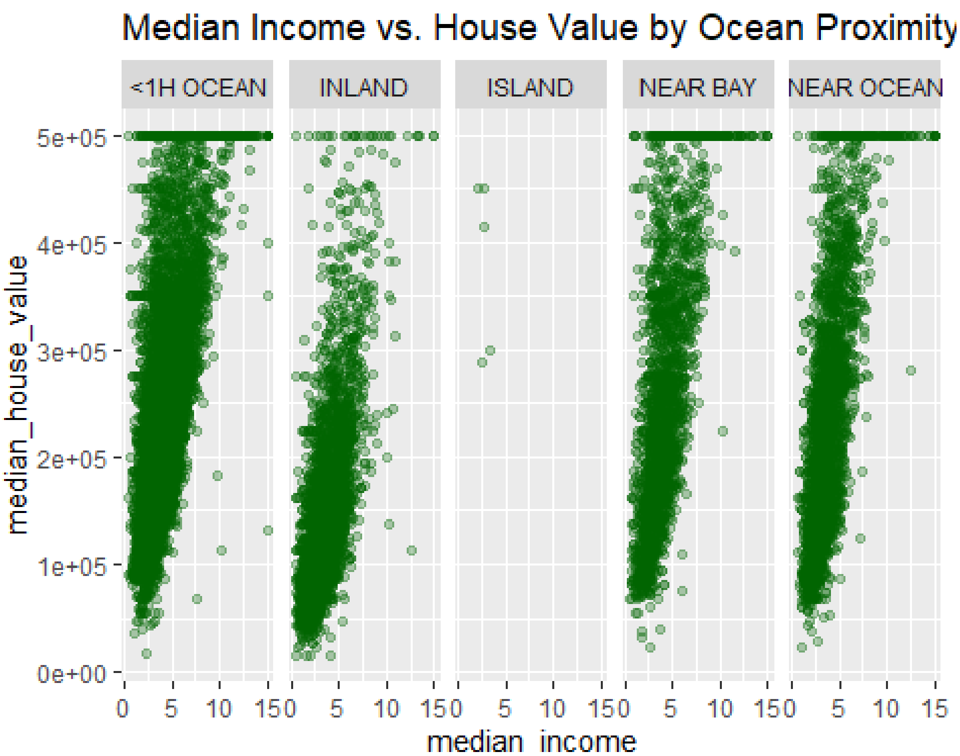

```r
# Density plot for 'median_income'
ggplot(data, aes(x = median_income, fill = ocean_proximity)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot of Median Income by Ocean Proximity")
```




```r
# Violin plot for 'ocean_proximity' vs. 'median_house_value'
ggplot(data, aes(x = as.factor(ocean_proximity), y = median_house_value)) +
  geom_violin(fill = "pink") +
  labs(title = "Violin Plot of House Value by Ocean Proximity")
```



```
# Create a facet grid of scatter plots
ggplot(data, aes(x = median_income, y = median_house_value)) +
  geom_point(alpha = 0.3, color = "darkgreen") +
  facet_grid(. ~ ocean_proximity) +
  labs(title = "Median Income vs. House Value by Ocean Proximity")
```



Step 4

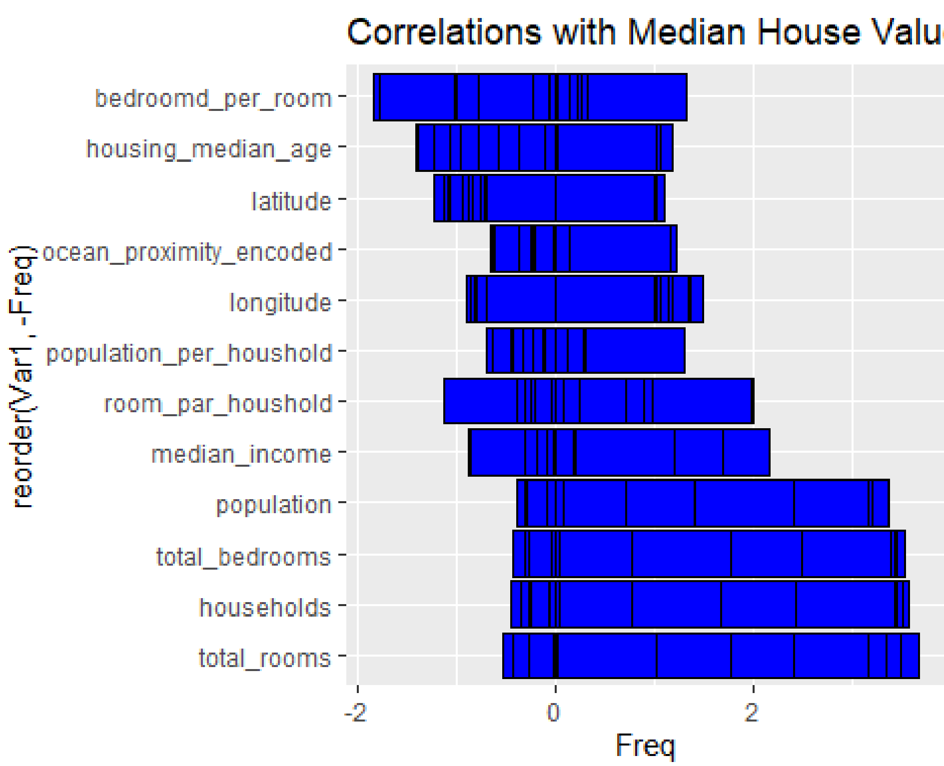
Variable or Feature Selection

```
# Calculate correlations between variables and median_house_value
#library(corrplot)
numeric_df <- data %>%
  select_if(is.numeric)
correlations<-cor(numeric_df,method="kendall")
```

```
# Sort correlations in descending order
sorted_correlations <- sort(correlations[,"median_house_value"], decreasing =
TRUE)
```

```
# Select features with high correlations
threshold <- 0.2  # Adjust as needed
selected_features <- names(sorted_correlations[sorted_correlations > threshol
d])
```

```r
# Convert the correlation matrix to a data frame
cor_data <- as.data.frame(as.table(correlations))

# Filter out correlations with the dependent variable
cor_data_filtered <- cor_data %>%
  filter(Var1 != "median_house_value")

# Visualize correlations using a bar plot
library(ggplot2)
ggplot(data = cor_data_filtered, aes(x = reorder(Var1, -Freq), y = Freq)) +
  geom_bar(stat = "identity", fill = "blue", color = "black") +
  coord_flip() +
  labs(title = "Correlations with Median House Value")
```



Step 5

Regression Modeling:

```r
model <- lm(median_house_value ~ median_income +
            data$housing_median_age,data = data)
```

Step 6

Model Evaluation:

```
summary(model)
```

```
##
## Call:
## lm(formula = median_house_value ~ median_income + data$housing_median_age,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -596748  -53834  -15000   36719  446725
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -10189.03    1915.41   -5.32 1.05e-07 ***
## median_income            43169.19     298.36  144.69  < 2e-16 ***
## data$housing_median_age   1744.13      45.04   38.73  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80850 on 20637 degrees of freedom
## Multiple R-squared:  0.5091, Adjusted R-squared:  0.5091
## F-statistic: 1.07e+04 on 2 and 20637 DF,  p-value: < 2.2e-16
```

Interpretation

The **“Residual standard error”** represents the estimated standard deviation of the residuals. It indicates how much the predicted values vary around the actual values. In this case, the estimated residual standard error is 80850.
The **“Coefficients”** section provides the estimated coefficients of the regression model:
(Intercept): The estimated intercept term, representing the predicted median house value when both “median_income” and “housing_median_age” are zero. In this case, it is -10189.03. The associated t-value indicates the significance of the intercept, and the p-value (< 0.001) suggests that the intercept is significantly different from zero.
median_income: The estimated coefficient for the “median_income” predictor variable is 43169.19. This indicates that for every unit increase in median income, the median house value is expected to increase by $43169.19. The very low p-value (< 0.001) indicates strong statistical significance.
housing_median_age: The estimated coefficient for the “housing_median_age” predictor variable is 1744.13. This means that for every unit increase in housing median age, the

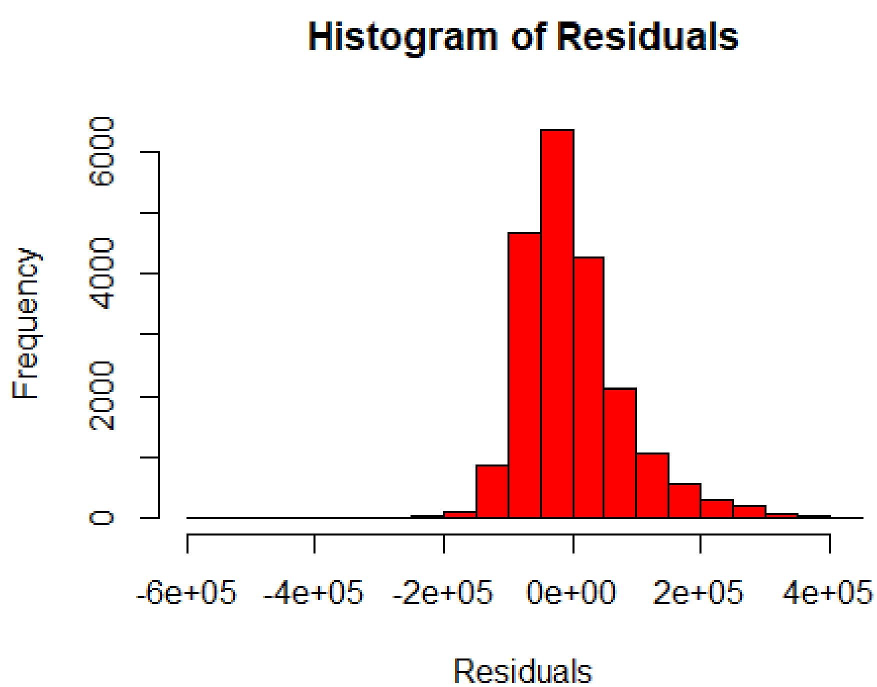
median house value is expected to increase by $1744.13. The very low p-value (< 0.001) indicates strong statistical significance.
The multiple **R-squared** value (0.5091) represents the proportion of the variability in the dependent variable (median_house_value) that is explained by the predictor variables (median_income and housing_median_age). **Adjusted R-squared** adjusts for the number of predictors in the model.
The **F-statistic** assesses the overall significance of the model. A very low p-value (< 0.001) suggests that the model is statistically significant overall and can explain a significant amount of the variability in the dependent variable.

Step 7

Testing of Assumptions of Regression Analysis

Chacking Normality

```
# Graphical Test: Histogram of residuals
hist(model$residuals, main = "Histogram of Residuals", xlab = "Residuals",col
="red")
```




```
# Normality test (Kamagorv test)
ks.test(model$residuals,"pnorm")
```

```
## Warning in ks.test.default(model$residuals, "pnorm"): ties should not be
## present for the Kolmogorov-Smirnov test

##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  model$residuals
## D = 0.58411, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
# Outliear
# Assuming 'model' is your regression model
#residuals <- model$residuals
```

```
# Create a boxplot of residuals
#library(ggplot2)
#ggplot(data.frame(Residuals = residuals), aes(y = Residuals,fill=data$ocean_
proximity)) +
  #geom_boxplot() +
  #labs(title = "Boxplot of Residuals", y = "Residuals") +
  #theme_minimal()
```

Using Log Transformation to Normalize Data

```
# Load necessary libraries
library(dplyr)
library(caret)
```

```
## Loading required package: lattice
```

```
# Logarithm transformation of the variables
data <- data %>%
  mutate(
    Log_MedianIncome = log(median_income + 1),  # Adding 1 to avoid taking lo
g of zero
    Log_HousingMedianAge = log(housing_median_age + 1)
    # Add more log-transformed variables if needed
  )
```

```
#Model
model <- lm(median_house_value ~ Log_MedianIncome+Log_HousingMedianAge ,data
=data)
```

```
summary(model)
```

```
##
## Call:
## lm(formula = median_house_value ~ Log_MedianIncome + Log_HousingMedianAge,
##     data = data)
##
```

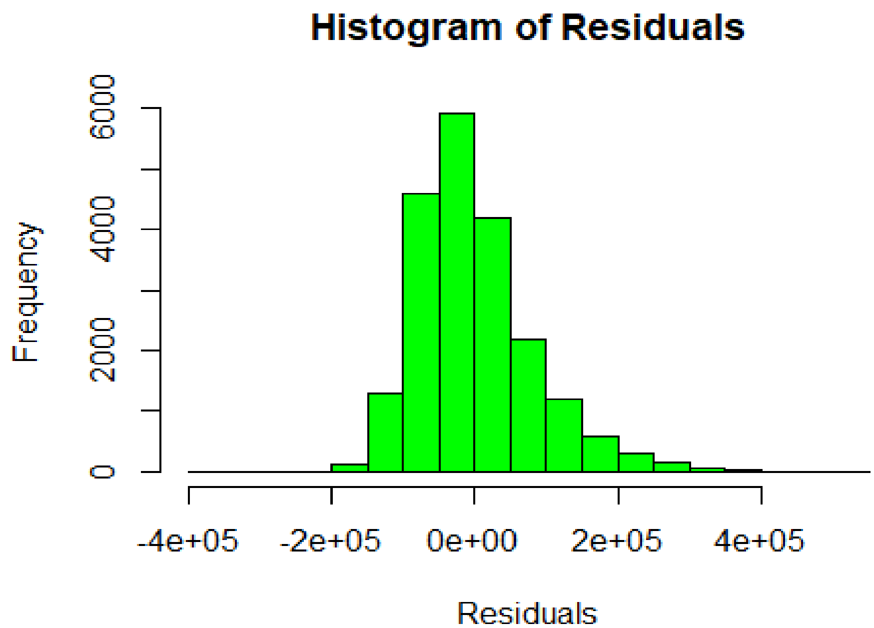
```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -385572  -57028  -13773   40194  538933 
## 
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           -265327       4705  -56.40   <2e-16 ***
## Log_MedianIncome       224554       1630  137.73   <2e-16 ***
## Log_HousingMedianAge    40199       1101   36.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 83040 on 20637 degrees of freedom
## Multiple R-squared:  0.4822, Adjusted R-squared:  0.4822 
## F-statistic:  9610 on 2 and 20637 DF,  p-value: < 2.2e-16
```

```r
# Get the residuals
residuals <- residuals(model)
```

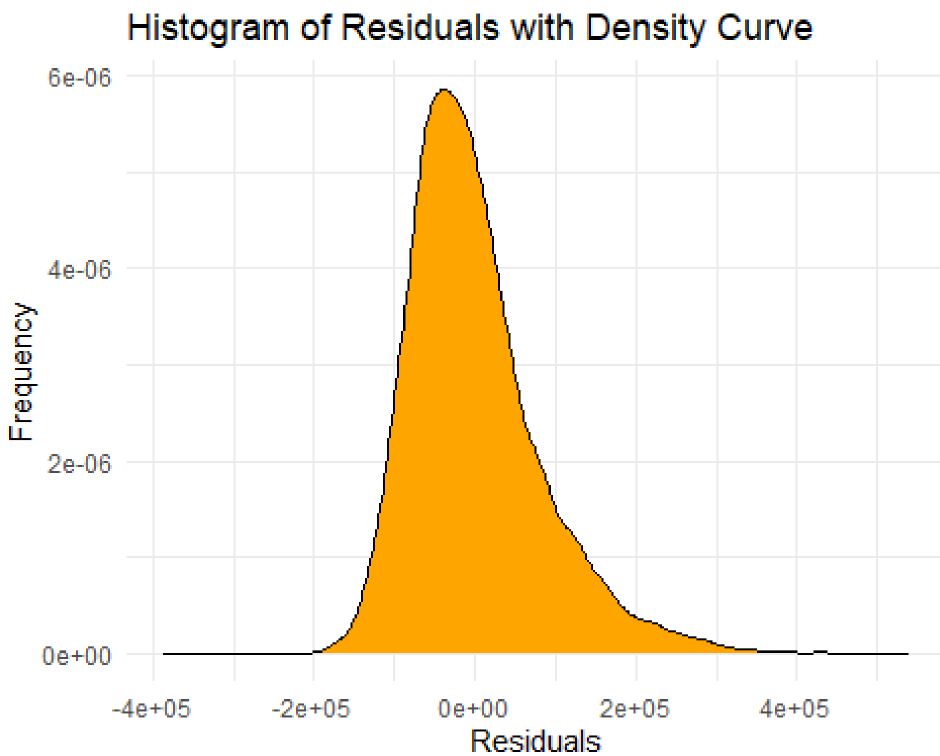
```r
# Normality Test
#library(nortest)
#ad_test <- ad.test(residuals)
# Normality test (Kamagorv test)
ks.test(model$residuals,"pnorm")
```

```
## Warning in ks.test.default(model$residuals, "pnorm"): ties should not be
## present for the Kolmogorov-Smirnov test

## 
##  Asymptotic one-sample Kolmogorov-Smirnov test
## 
## data:  model$residuals
## D = 0.57742, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```r
# Create a histogram of residuals
hist(model$residuals, main = "Histogram of Residuals", xlab = "Residuals",col
="green")
```

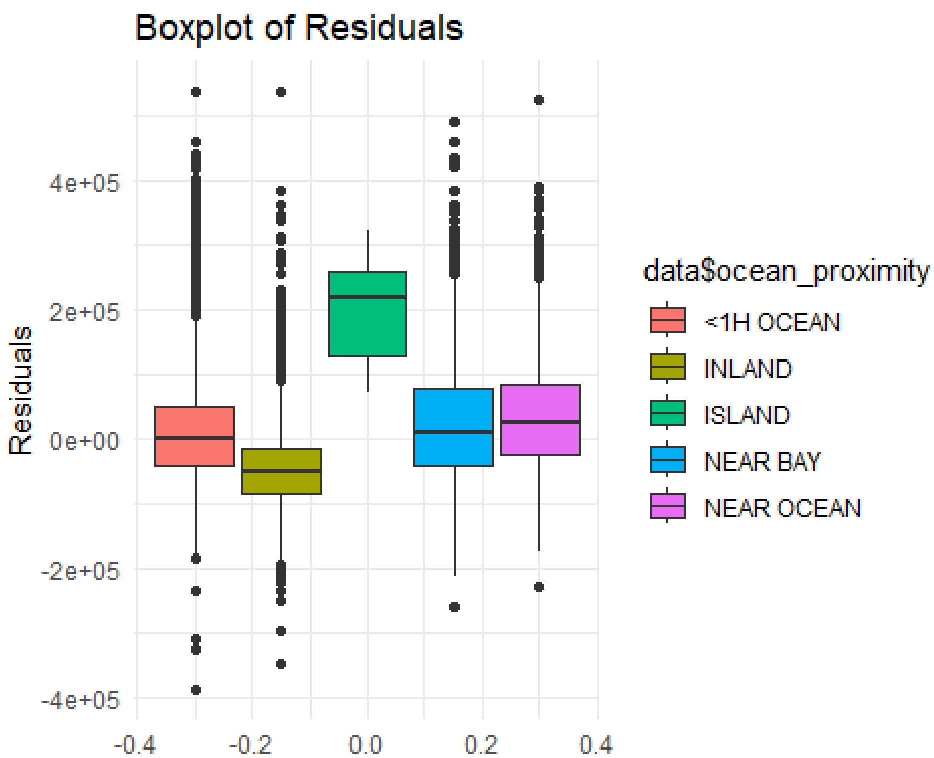



```
library(ggplot2)

# Create a histogram of residuals with fitted density curve
ggplot(data.frame(residuals = model$residuals), aes(x = residuals)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
  geom_density(fill = "orange") +
  labs(title = "Histogram of Residuals with Density Curve", x = "Residuals",
y = "Frequency") +
  theme_minimal()
```



Outliers Check

```
# Assuming 'model' is your regression model
residuals <- model$residuals

# Create a boxplot of residuals
library(ggplot2)
ggplot(data.frame(Residuals = residuals), aes(y = Residuals,fill=data$ocean_p
roximity)) +
  geom_boxplot() +
  labs(title = "Boxplot of Residuals", y = "Residuals") +
  theme_minimal()
```

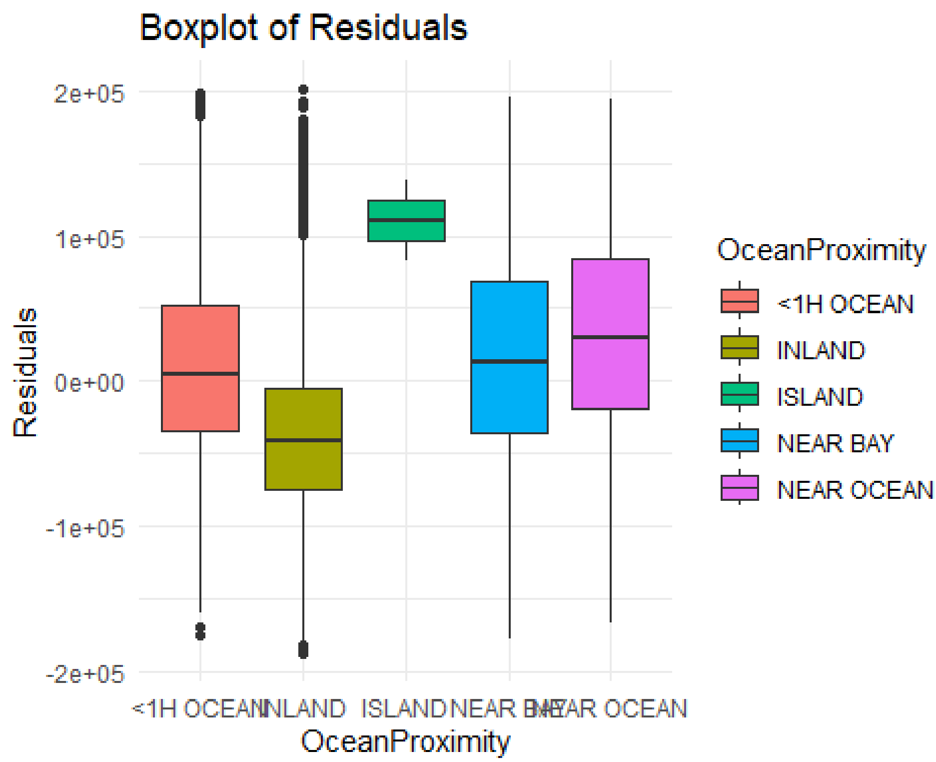


Outliear Removel

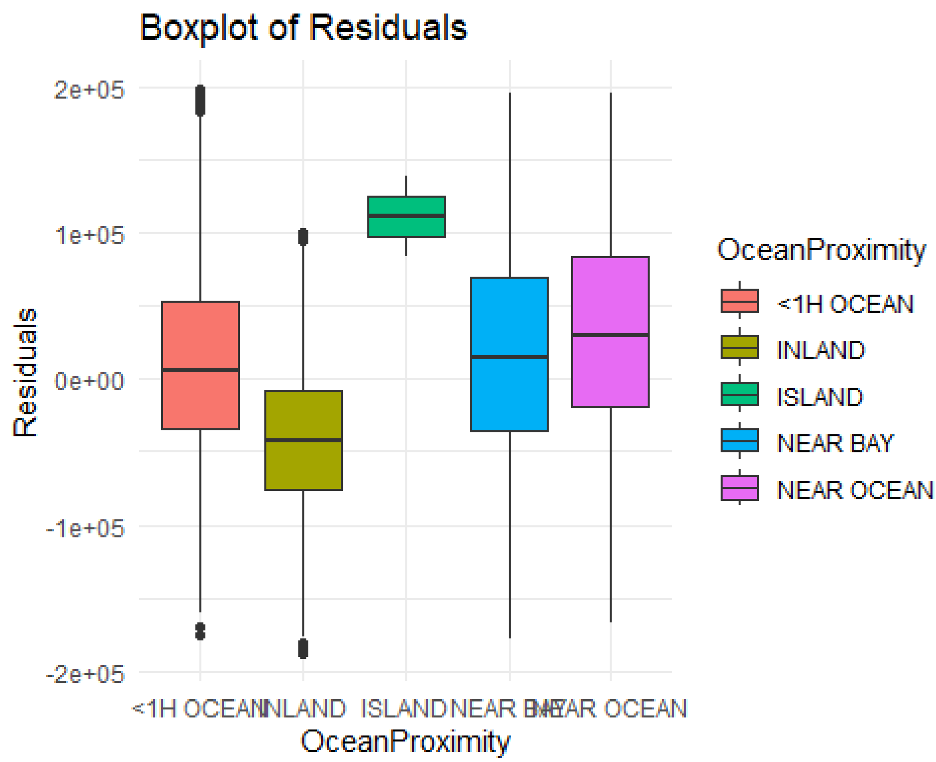
```
q1 <- quantile(model$residuals, 0.25)
q3 <- quantile(model$residuals, 0.75)
iqr <- q3 - q1
data<-data[!(model$residuals < (q1 - 1.5 * iqr) | model$residuals > (q3 + 1.5
* iqr)), ]
```

```
#create model
model <- lm(median_house_value ~ data$Log_MedianIncome +
               data$Log_HousingMedianAge, data = data)
# Assuming 'model5' is your regression model after removing outliers and usin
g dummy variables
residuals <- residuals(model)
```

```
# Create a boxplot of residuals colored by Ocean Proximity
ggplot(data.frame(Residuals = residuals, OceanProximity = data$ocean_proximit
y), aes(x = OceanProximity, y = Residuals, fill = OceanProximity)) +
  geom_boxplot() +
  labs(title = "Boxplot of Residuals", y = "Residuals") +
  theme_minimal()
```

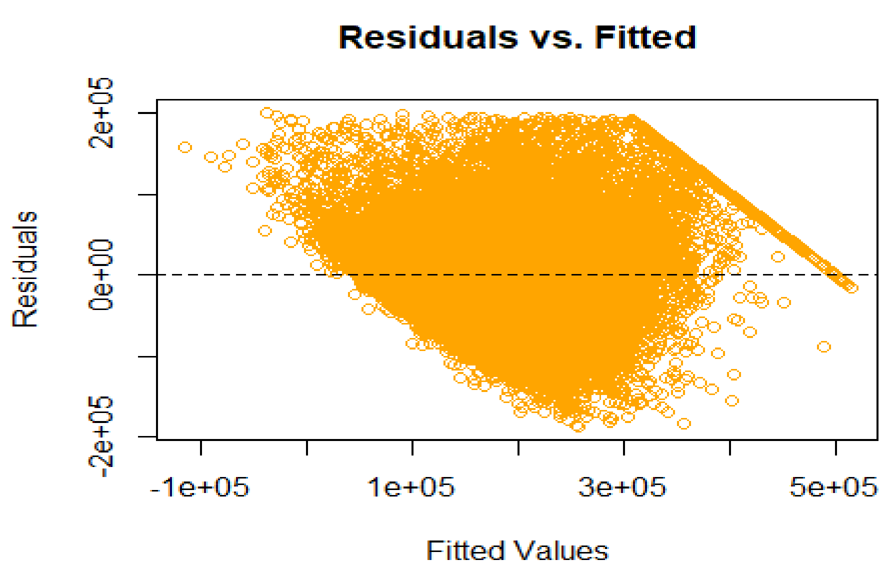



```
#khud sa values remove karna boxplot ko dakhtay hua
#Calculate residuals for the entire dataset
residuals <- data$median_house_value - predict(model, newdata = data)

# Add residuals to the data frame
data$residuals <- residuals

library(dplyr)
data <- data %>%
  filter(!(ocean_proximity=="INLAND" & residuals > 1e5)) %>%
  filter(!(ocean_proximity=="<1H OCEAN" & residuals > 2e5))

residuals <- data$median_house_value - predict(model, newdata = data)

combined_data <- data.frame(
  Residuals = residuals,
  OceanProximity = data$ocean_proximity
)

# Create the boxplot of residuals colored by Ocean Proximity
ggplot(combined_data, aes(x = OceanProximity, y = Residuals, fill = OceanProx
imity)) +
  geom_boxplot() +
  labs(title = "Boxplot of Residuals", y = "Residuals") +
  theme_minimal()
```



Heteroscedasticity

```
# Graphical Test: Residuals vs. Fitted plot
plot(model$fitted.values, model$residuals, main = "Residuals vs. Fitted", xla
b = "Fitted Values", ylab = "Residuals",col="orange")
abline(h = 0, col = "black", lty = 2)
```



```r
library(lmtest)
```

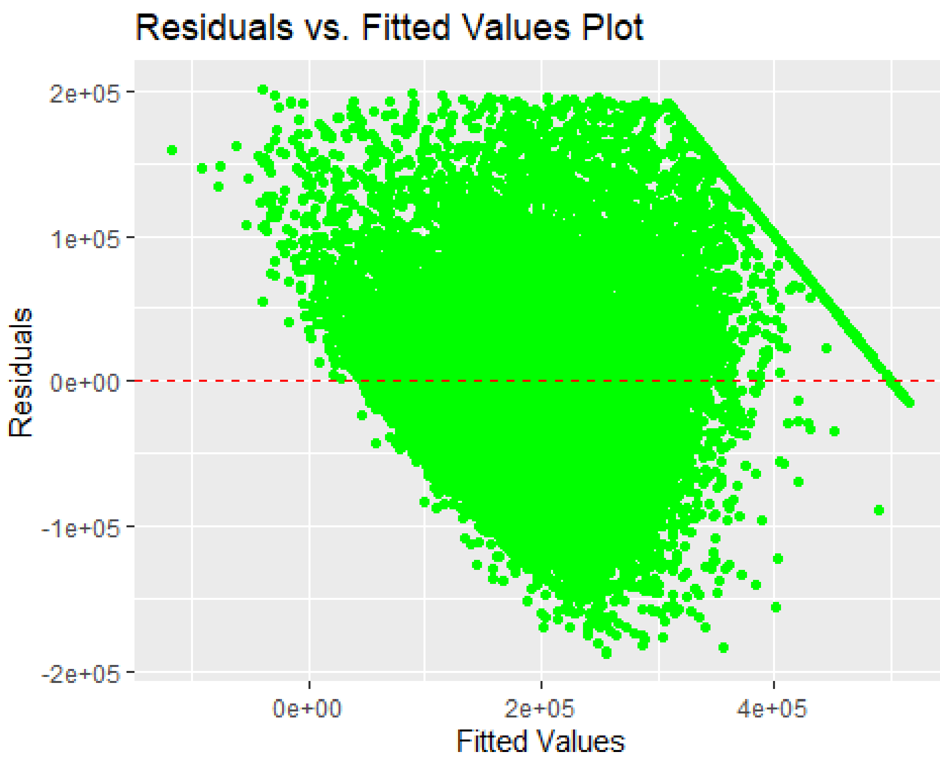
```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 632.26, df = 2, p-value < 2.2e-16
```

#Independancy

```r
# Load necessary libraries
library(tibble)
library(dplyr)
library(ggplot2)



# Calculate residuals
residuals <- resid(model)

# Create a dataframe with fitted values and residuals
fitted_resid_df <- tibble(
  Fitted_Values = predict(model),
  Residuals = residuals
)

# Plot residuals vs. fitted values
ggplot(fitted_resid_df, aes(x = Fitted_Values, y = Residuals)) +
  geom_point(col="green") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  ggtitle("Residuals vs. Fitted Values Plot") +
  xlab("Fitted Values") +
  ylab("Residuals")
```

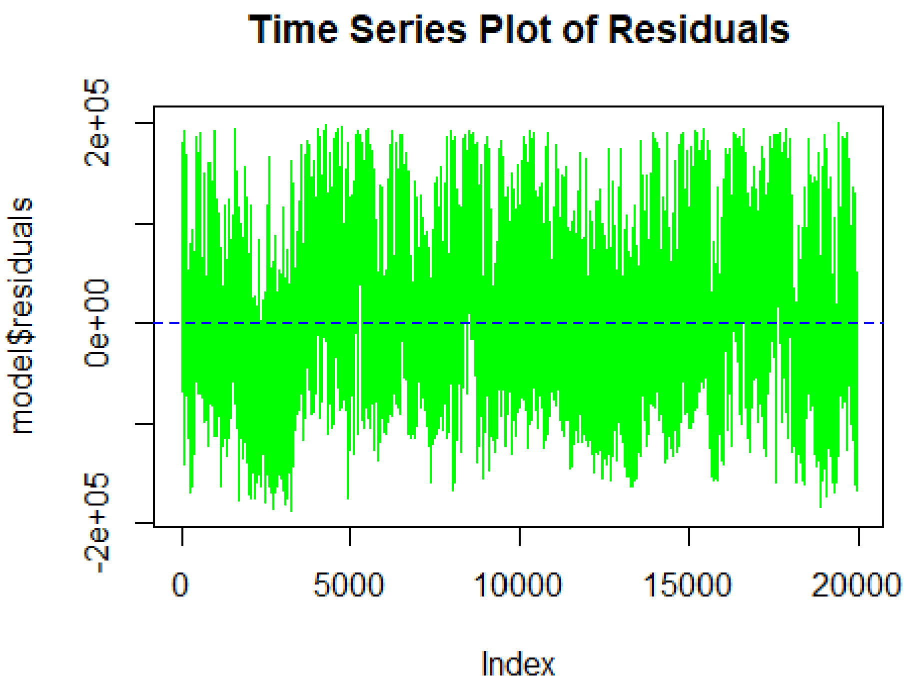



```
library(lmtest)
dw_test <- dwtest(model)
dw_test
```

```
##
##  Durbin-Watson test
##
## data:  model
## DW = 0.84541, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Auto correlation

```
# Graphical Test: Time series plot of residuals
plot(model$residuals, type = "l", main = "Time Series Plot of Residuals",
     col="green")
abline(h = 0, col = "blue", lty = 2)
```




```
# Statistical Test: Durbin-Watson test for autocorrelation
library(zoo)
dwtest(model)
```

```
##
##  Durbin-Watson test
##
## data:  model
## DW = 0.84541, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Removing AutoCorrelation

```
# Load necessary libraries
library(dplyr)
library(caret)
library(lmtest)
```

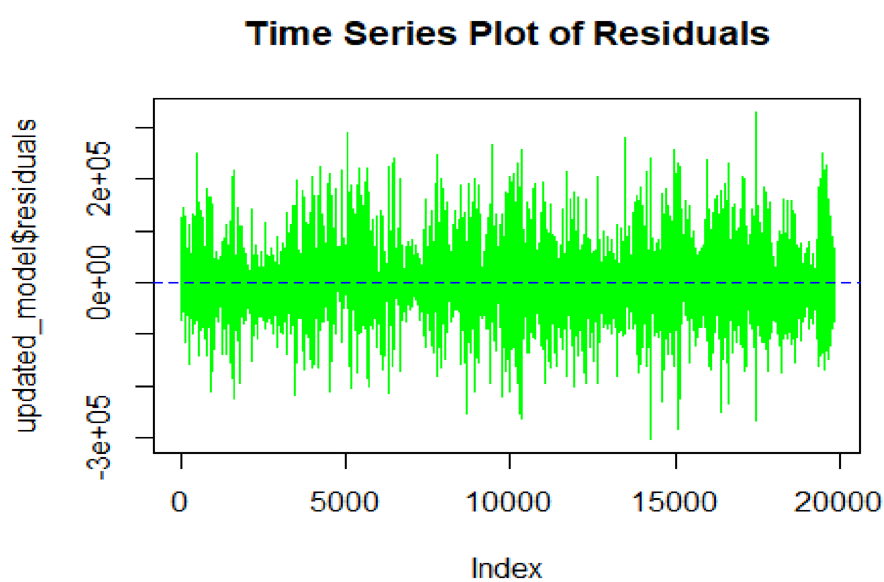
```
dw_test <- dwtest(model)
```

```
# If Durbin-Watson test value is less than 1.5, consider addressing autocorre
lation
if (dw_test$statistic < 1.5) {
  # Create lagged variables for the dependent variable
  data1 <- data %>%
    mutate(lagged_median_house_value = lag(median_house_value))
```

```r
  # Update the model to include lagged variables
  updated_model <- lm(median_house_value ~ Log_MedianIncome + Log_HousingMedi
anAge +
                      lagged_median_house_value, data = data1)

  # Print the updated model summary
  summary(updated_model)

  # Check for autocorrelation in the updated model
  dw_test_updated <- dwtest(updated_model)
  print(dw_test_updated)
} else {
  # No autocorrelation issue
  print("Durbin-Watson test indicates no autocorrelation issue.")
}
```

```
##
##  Durbin-Watson test
##
## data:  updated_model
## DW = 2.129, p-value = 1
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
# Graphical Test: Time series plot of residuals
plot(updated_model$residuals, type = "l", main = "Time Series Plot of Residua
ls",
     col="green")
abline(h = 0, col = "blue", lty = 2)
```

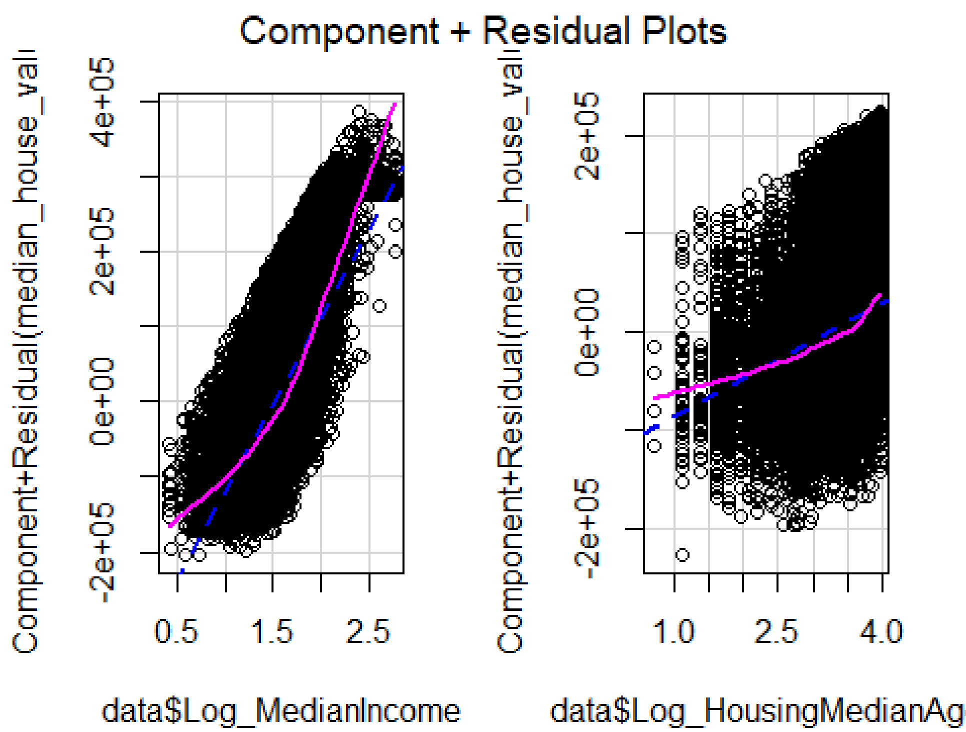


Multicollinearity

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
# Create component plus residual (CR) plots
crPlots(model)
```

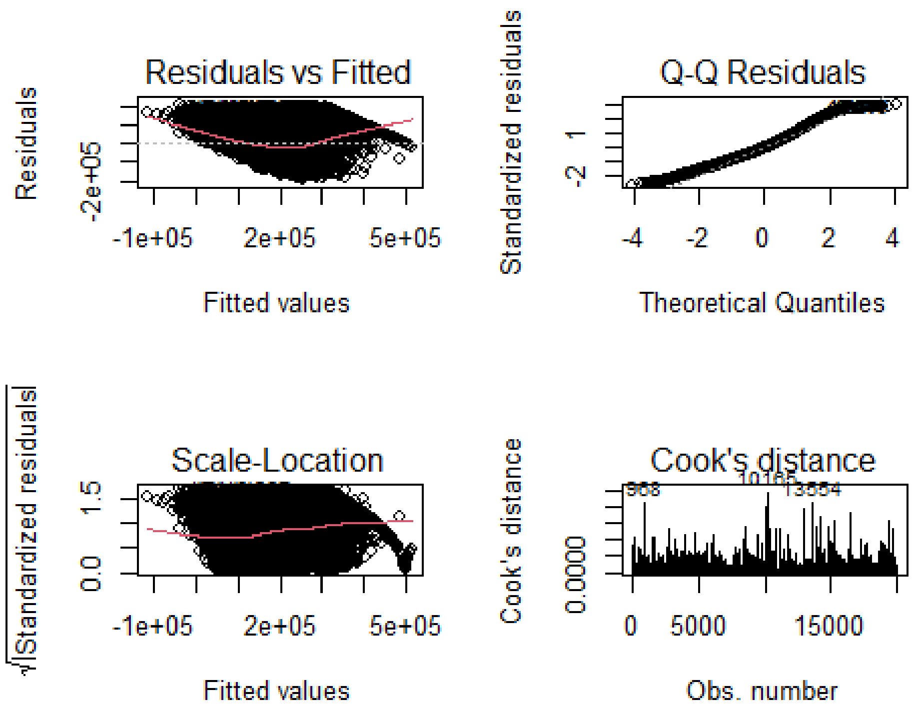



```
library(car)
vif(model)
```

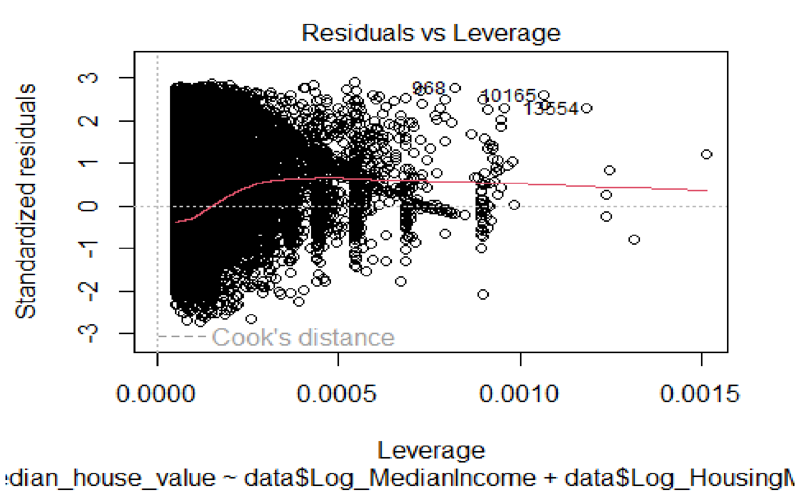
```
##     data$Log_MedianIncome data$Log_HousingMedianAge
##                  1.025512                  1.025512
```

Leverage

```r
# Generate the matrix of four plots
par(mfrow = c(2, 2))
plot(model, which = 1:4)
```




```r
# Graphical Test: Leverage plot (Cook's distance plot)
plot(model, which = 5)
```



```r
# Test for leverage value
# Assuming 'model' is your regression model
cooksd <- cooks.distance(model)
```

```r
# Identify influential observations using a threshold
threshold <- 4 / nrow(model$qr$qr)
influential <- cooksd > threshold
influential_indices <- which(influential)
```

```r
# Display influential observation indices
head(influential_indices)
```

```
## 49 51 69 71 73 75
## 49 51 69 71 73 75
```

Step 9

Dummy Variables

```r
# Load necessary libraries
library(dplyr)
library(caret)
```

```r
# Create dummy variables for ocean_proximity
dummy_vars <- dummyVars(~ ocean_proximity, data = data)
```

```r
# Transform the data with the dummy variables
df <- predict(dummy_vars, newdata = data)
# Convert the matrix/array 'df' into a data frame
df <- as.data.frame(df)
```

```r
# Combine the transformed data with the original dataset
df <- cbind(data, df)
```

```r
# Combine the transformed data with the original dataset

model <- lm(median_house_value ~ Log_HousingMedianAge + Log_MedianIncome +
            `ocean_proximity<1H OCEAN` + ocean_proximityINLAND +
            ocean_proximityISLAND + `ocean_proximityNEAR OCEAN`,
            data = df)
```

```r
summary(model)
```

```
##
## Call:
## lm(formula = median_house_value ~ Log_HousingMedianAge + Log_MedianIncome
+
##     `ocean_proximity<1H OCEAN` + ocean_proximityINLAND + ocean_proximityIS
```

```
LAND +
##     `ocean_proximityNEAR OCEAN`, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -198960  -43198   -6514   35964  195018
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -159588       4375 -36.477  < 2e-16 ***
## Log_HousingMedianAge            22085        886  24.927  < 2e-16 ***
## Log_MedianIncome               206144       1311 157.283  < 2e-16 ***
## `ocean_proximity<1H OCEAN`     -10854       1498  -7.247 4.42e-13 ***
## ocean_proximityINLAND          -74008       1626 -45.518  < 2e-16 ***
## ocean_proximityISLAND           87300      43501   2.007   0.0448 *
## `ocean_proximityNEAR OCEAN`      7093       1821   3.894 9.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61490 on 19803 degrees of freedom
## Multiple R-squared:  0.6689, Adjusted R-squared:  0.6688
## F-statistic:  6667 on 6 and 19803 DF,  p-value: < 2.2e-16
```

Interpretation

The **"Coefficients"** section provides the estimated coefficients of the regression model for each predictor variable:

- **`(Intercept)`**: The estimated intercept term, representing the predicted median house value when all other predictor variables are zero. In this case, it is -159588. The associated t-value is large, and the p-value (< 0.001) suggests that the intercept is significantly different from zero.
- **`df$Log_MedianIncome`**: The estimated coefficient for the log-transformed "median_income" predictor variable is 206144. This indicates that for a one-unit increase in log-transformed median income, the median house value is expected to increase by $206144. The very low p-value (< 0.001) indicates strong statistical significance.
- **`df$Log_HousingMedianAge`:** The estimated coefficient for the log-transformed "housing_median_age" predictor variable is 22085. This means that for a one-unit increase in log-transformed housing median age, the median house value is expected to increase by $22085. The very low p-value (< 0.001) indicates strong statistical significance.
- **df$ocean_proximity<1H OCEAN``:** The estimated coefficient for the "ocean_proximity<1H OCEAN" dummy variable is -10854. This suggests that houses

	located in the category "1H OCEAN" have a lower median house value compared to the reference category.
•	**df$ocean_proximityINLAND:** The estimated coefficient for the "ocean_proximityINLAND" dummy variable is -74008. This suggests that houses located inland have a significantly lower median house value compared to the reference category.
•	**df$ocean_proximityISLAND:** The estimated coefficient for the "ocean_proximityISLAND" dummy variable is 87300. This indicates that houses located on an island have a higher median house value compared to the reference category, but the p-value is relatively high (0.0448), suggesting a lower level of statistical significance.
•	**df$ocean_proximityNEAR OCEAN``:** The estimated coefficient for the "ocean_proximityNEAR OCEAN" dummy variable is 7093. This indicates that houses located near the ocean have a higher median house value compared to the reference category. The "Residual standard error" represents the estimated standard deviation of the residuals. It indicates how much the predicted values vary around the actual values.
•	The **Multiple R-squared** value (0.6689) represents the proportion of the variability in the dependent variable (median_house_value) that is explained by the independent variables in the model.
•	The **Adjusted R-squared** value (0.6688) adjusts the Multiple R-squared value for the number of predictor variables in the model.
•	The **F-statistic** assesses the overall significance of the model. A very low p-value (< 0.001) suggests that the model is statistically significant overall and can explain a significant amount of the variability in the dependent variable.

Step 11

Overall Interpretation

Certainly! I'll rephrase the explanations using the word "I" to reflect that the interpretation is being done by you.

Step 1: Data Collection

In this step, I introduce my project and mention that I'm using the California Housing Prices dataset from Kaggle. I provide a brief description of the dataset and its attributes.

Step 2: Data Preprocessing

2.1.1 Overview of Missing Values

I start by checking for missing values in the dataset and calculating the count of missing values for each column.

2.1.2 Dealing with Missing Values Through Median Imputation

I perform median imputation to fill in missing values for the "total_bedrooms" column with the median value of the column.

2.2.1 Encode Categorical Variables

I encode the categorical variable "ocean_proximity" using label encoding and create a new variable "ocean_proximity_encoded" to represent the encoded values.

2.3.1 Mathematical Transformation / Add New Variables

I create new variables by performing mathematical transformations on existing variables, such as calculating "room_par_houshold," "bedroomd_per_room," and "population_per_houshold."

Step 3: Exploratory Data Analysis (EDA)

I provide summary statistics and visualizations to explore the dataset:

- Summary statistics of the dataset.
- Glimpse of the dataset using the `glimpse` function.
- Summary statistics (mean, standard deviation, min, max) grouped by "ocean_proximity."
- Checking the class of the dataset.
- Displaying the first few rows of the dataset.
- Creating a correlation plot using the `corrplot` library.
- Creating a histogram of the "median_income" variable.
- Creating box plots for "ocean_proximity_encoded" vs. "median_house_value."
- Creating a bar plot for the distribution of "ocean_proximity."
- Creating scatter plots and density plots to explore relationships between variables.

Step 4: Variable or Feature Selection

I perform variable selection using correlation analysis:

- Calculating correlations between variables and "median_house_value."

- Sorting correlations in descending order.
- Selecting features with high correlations.
- Creating a bar plot to visualize correlations with “median_house_value.”

Step 5: Regression Modeling

I create a linear regression model to predict “median_house_value” using “median_income” and “housing_median_age” as predictor variables.

Step 6: Model Evaluation

I provide an overview of model evaluation by analyzing the model summary:

- Summary statistics of the model.
- Explanation of the coefficients of the model, including intercept, “median_income,” and “housing_median_age.”
- Interpretation of the R-squared values and F-statistic.

Step 7: Testing of Assumptions of Regression Analysis

I test several assumptions of regression analysis:

- Checking for normality using graphical and statistical tests.
- Checking for outliers using box plots.
- Transforming data using logarithmic transformation.
- Checking for autocorrelation using time series plots and statistical tests (Durbin-Watson test).
- Checking for heteroscedasticity using graphical and statistical tests.
- Checking for multicollinearity using component plus residual (CR) plots and variance inflation factor (VIF) analysis.
- Checking for leverage and influential observations using graphical tests and Cook’s distance.

Conclusion

My project covers a comprehensive range of steps, from data preprocessing and exploratory data analysis to regression modeling and assumption testing. Each step contributes to a thorough understanding of the dataset and the creation of a regression model for predicting house prices. This well-organized approach ensures that I am taking the necessary precautions and making informed decisions throughout the analysis process.