AAD Assignment 1 - Group 26

Omar, Eloise, Alina, Sue

2025-04-11

Contents

---

1 Descriptive analysis of the data set

1.1 Data loading and cleaning

There are 18640 data points in the original data set. There are 10 features being observed and/or recorded, including the response variable, Median House Value ("medianHouseValue").

```
data <- read.csv("Assignt1_data.csv")
dim(data)
```

```
## [1] 18640    10
```

```
stargazer(data[-1], summary = TRUE, type = "text")
```

```
##
## =================================================================
## Statistic          N       Mean      St. Dev.     Min      Max
## -----------------------------------------------------------------
## longitude       18,640  -119.569      2.004    -124.350 -114.310
## latitude        18,640   35.630       2.136     32.550   41.950
## housingMedianAge 18,640  28.613      12.606       1        52
## aveRooms        18,640    5.437       2.535     0.846   141.909
## aveBedrooms     18,450    1.097       0.490     0.375    34.067
## population      18,640  1,426.684  1,135.967     3      35,682
## medianIncome    18,640    3.880       1.907     0.500    15.000
## medianHouseValue 18,640 207,241.900 115,651.600 14,999  500,001
## -----------------------------------------------------------------
```

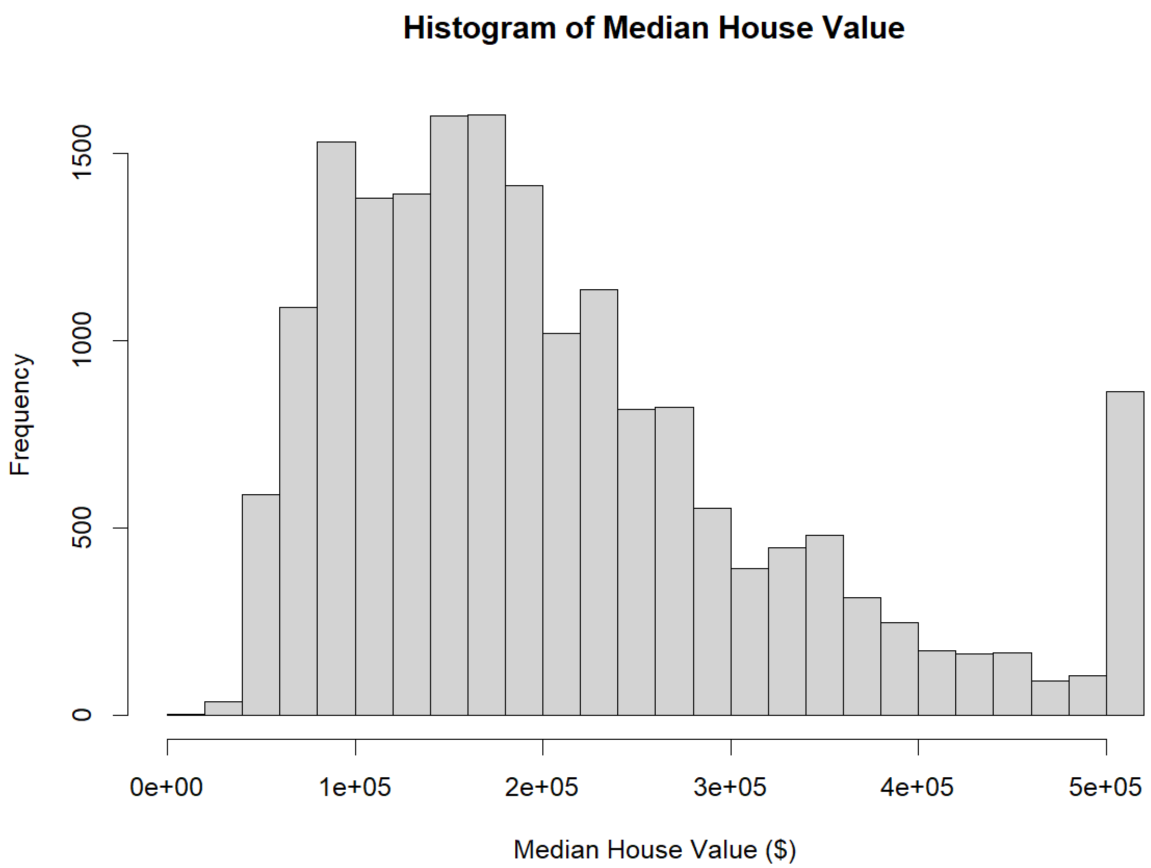
When inspecting the orginal data, we observe that there are 190 rows that have missing values (NA's) in the varaible "aveBedrooms". Since using data points with missing values can lead to biased results, and since 190 out of 18,640 data points is an immaterial number, we decided to remove these rows that contains NA's from the original data frame.

```r
data <- na.omit(data)
rownames(data) <- 1:nrow(data)
```

1.2 Preliminary analysis on the data

1.2.1 Histogram on the response variable and data censoring

```r
hist(data$medianHouseValue,
     xlab = "Median House Value ($)",
     ylab = "Frequency",
     main = "Histogram of Median House Value",
     breaks = 25)
```



From the histogram, it is evident that the medianHouseValue exhibits right-skewness. Interestingly, there is large a concentration of values at the right-end, where the frequency of values $500,000 is disproportionately high. This is likely due to data censoring, where all the medianHouseValue over a certain threshold is recorded at a capped value around $500,000.

Through further inspection we find out this cap for censoring is at $500,001.

```r
value <- max(data$medianHouseValue)
percentage <- mean(data$medianHouseValue == value) * 100
cat("Percentage of data with medianHouseValue =", value, "is", percentage, "%\n")
```

```
## Percentage of data with medianHouseValue = 500001 is 4.688347 %
```

```r
summary(data$medianHouseValue)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14999  119800  180000  207277  265600  500001
```

Ultimately, given the max. value is 500,001 and the fact that the frequency of this exact value is disproportionately high (4.7%), we are led to conclude that 500,001 is the censoring cap applied. This is likely due to the data collection methods used, where houses valued above $500,000 were simply reported as exactly $500,001. The impact of this is that the data may under-represent the true values and variation in higher-value homes. Moving forward, we have decided to keep these censored observations in but will maintain caution - especially in our interpretation of correlations, and coming to terms with the fact that our model will not be able to provide reliable predictions for values above 500,001.

Other censoring

```r
summary(data$medianIncome)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4999  2.5669  3.5446  3.8803  4.7608 15.0001
```

```r
value <- max(data$medianIncome)
percentage <- mean(data$medianIncome == value) * 100
cat("Percentage of data with medianIncome =", value, "is", percentage, "%\n")
```

```
## Percentage of data with medianIncome = 15.0001 is 0.2384824 %
```

It's clear that medianIncome is also censored. Both medianIncome and medianHouseValue are left censored and right censored. Since it is more significant, for this report we will focus on right censoring on medianHouseValue.

1.2.2 Correlations among variables

```r
numeric.data <-  data[sapply(data, is.numeric)]
head(numeric.data)
```

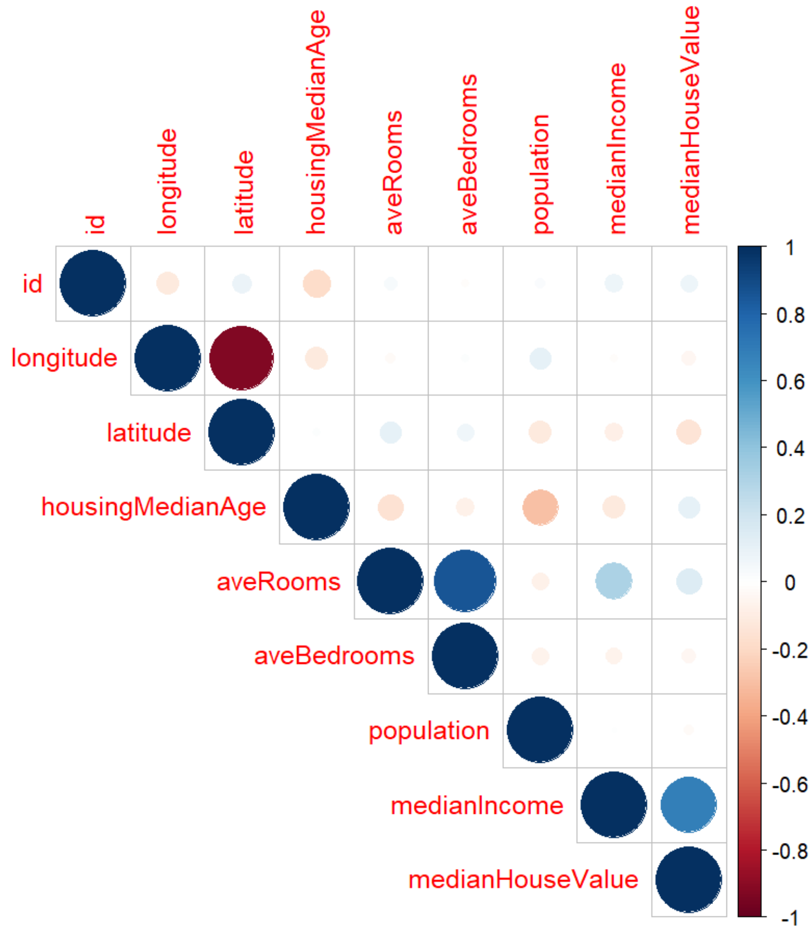
```
##   id longitude latitude housingMedianAge aveRooms aveBedrooms population
## 1  1   -122.23    37.88               41 6.984127   1.0238095        322
## 2  2   -122.22    37.86               21 6.238137   0.9718805       2401
## 3  3   -122.24    37.85               52 8.288136   1.0734463        496
## 4  4   -122.25    37.85               52 5.817352   1.0730594        558
## 5  5   -122.25    37.85               52 6.281853   1.0810811        565
## 6  6   -122.25    37.85               52 4.761658   1.1036269        413
##   medianIncome medianHouseValue
```

```
## 1       8.3252           452600
## 2       8.3014           358500
## 3       7.2574           352100
## 4       5.6431           341300
## 5       3.8462           342200
## 6       4.0368           269700
```

```
cor_medianHouseValue <- cor(data$medianHouseValue,
                            data[, c("longitude",
                                            "latitude",
                                            "housingMedianAge",
                                            "aveRooms",
                                            "aveBedrooms",
                                            "population",
                                            "medianIncome")])
print(cor_medianHouseValue)
```

```
##        longitude   latitude housingMedianAge aveRooms aveBedrooms  population
## [1,] -0.04646254 -0.1439446        0.1052129 0.148625 -0.04545692 -0.02394206
##      medianIncome
## [1,]     0.689536
```

```
corrplot(cor(data[, sapply(data, is.numeric)]), method = "circle",type =
          "upper")
```



From the correlation matrix plot we can see that among all the predictors, medianHouseValue has highest correlation with Median Income. This is pretty intuitive, as we expect high-income households purchasing more expensive houses.

Longitude and Latitude are highly negatively correlated, while Average Rooms and Average Bedrooms are highly positively correlated. This multicollinearity can lead to issues such as unstable coefficient estimates and inflated standard errors, as the model struggles to distinguish the individual effects of highly correlated predictors. Consequently, the interpretability of the regression results is compromised. We will address this issue later in section 2.4 Model improvements.

Thee rest of the correlations between the remaining variables with medianHouseValue have a magnitude below 0.3. However, they may still be very useful in predicting the response variable. But we do not give detailed discussion here.

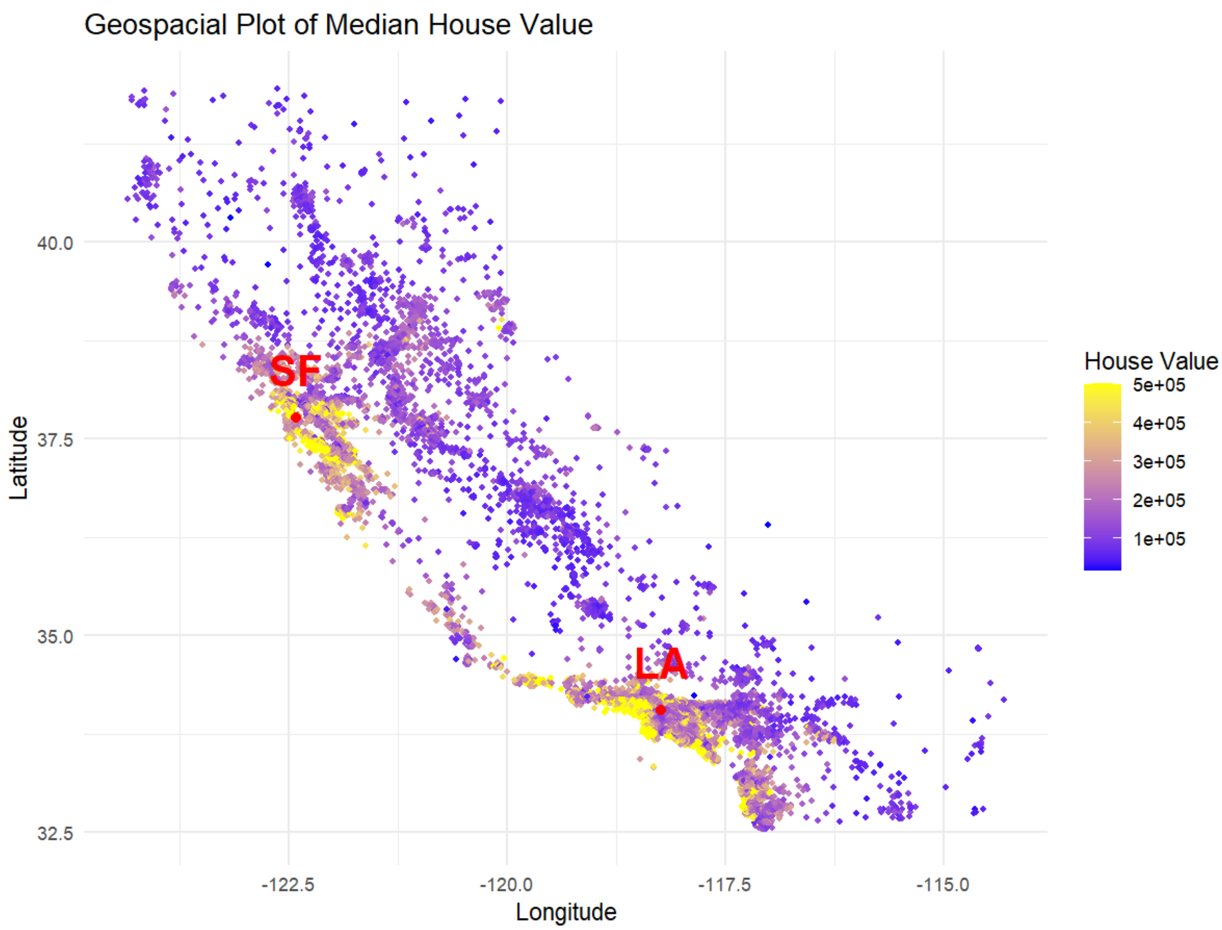
Note that ID is just for labeling purposes and it is not a relevant predictor, so we don't care about the correlation between ID and other variables.

1.2.3 Geospacial plot of medianHouseValue

Housing values can vary based on their proximity to major cities. By analysing the provided longitude and latitude, we can deduce that the dataset originates from California. To visualise the relative variation in median house values, we create a heatmap that highlights the influence of major cities, specifically Los Angeles (LA) and San Francisco (SF).

```r
cities <- data.frame(
  city = c("LA", "SF"),
  longitude = c(-118.24, -122.42),
  latitude = c(34.05, 37.77)
)

ggplot(data,
       aes(x = longitude,
           y = latitude,
           colour = medianHouseValue)) +
  geom_point(size = 1) +
  scale_colour_gradient(low = "blue", high = "yellow") +
  geom_point(data = cities,
             aes(x = longitude, y = latitude),
             colour = "red",
             size = 2) +
  geom_text(data = cities,
            aes(label = city),
            vjust = -1,
            colour = "red",
            size = 7,
            fontface = "bold") +
  labs(title = "Geospacial Plot of Median House Value",
       x = "Longitude",
       y = "Latitude",
       colour = "House Value") +
  theme_minimal()
```

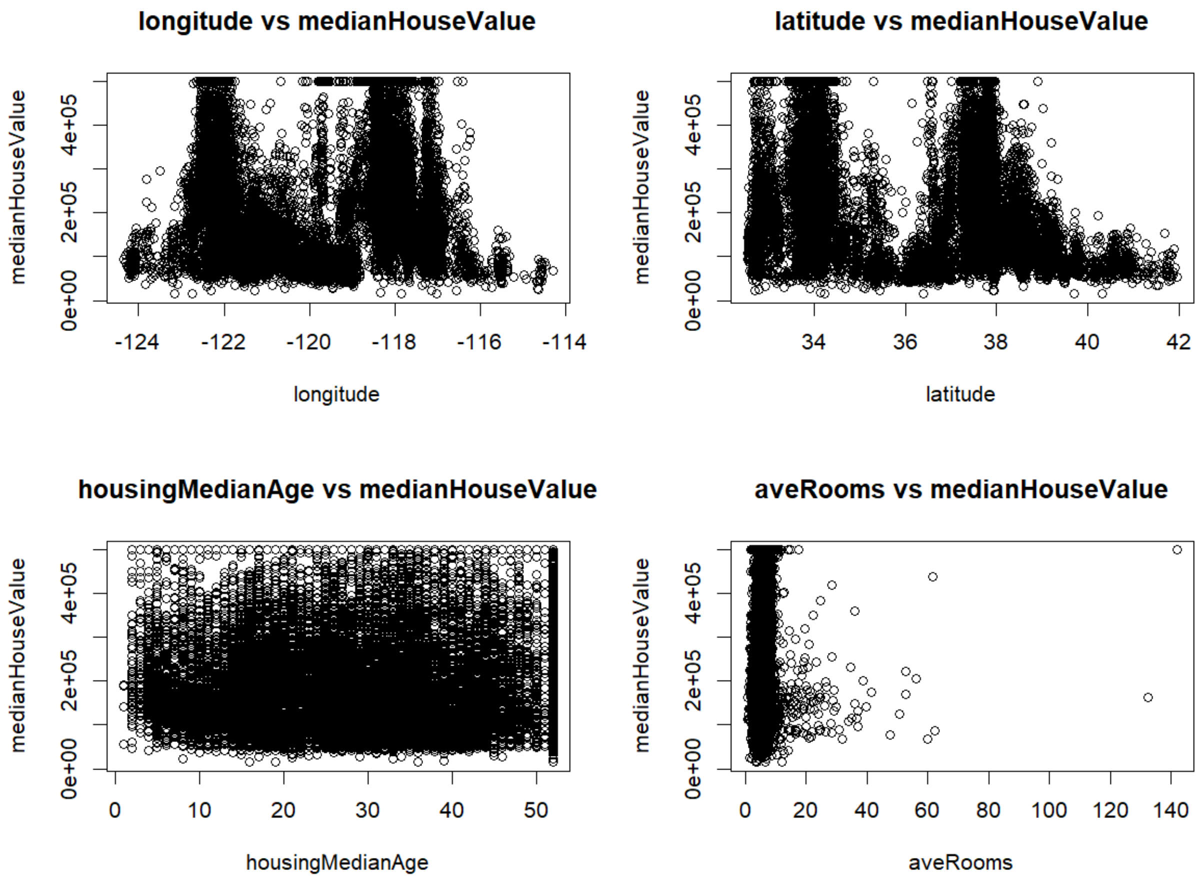


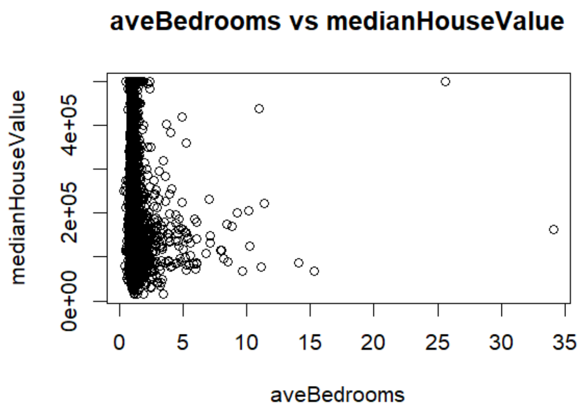
Clearly, median housing value is higher in the SF and LA regions. Therefore, it is reasonable to integrate a region's closeness to these major cities in our regression model later. They are also an appropriate proxy for censoring.

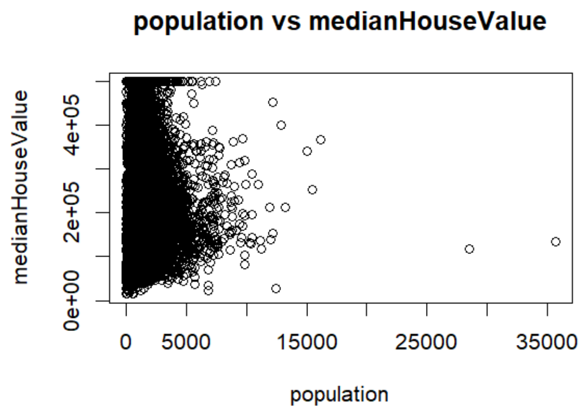
1.2.4 Plots of the response variable vs different features

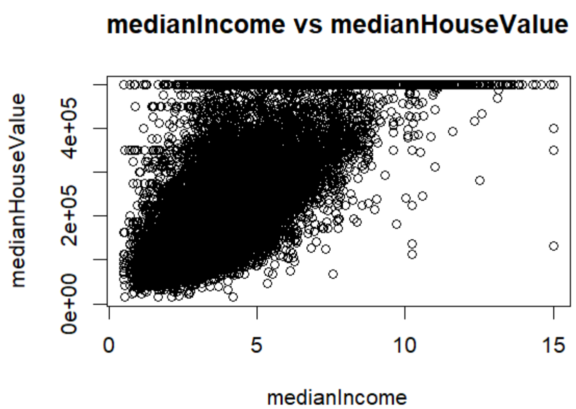
```
par(mfrow = c(2, 2))  # 2 rows, 2 columns of plots per "page"
target_col <- "medianHouseValue"
exclude <- c("id","oceanProximity",target_col)

for (col in names(data)) {
  if (!(col %in% exclude)) {
    plot(data[[col]], data[[target_col]],
         xlab = col,
         ylab = target_col,
         main = paste(col, "vs", target_col))
  }
}
```


longitude vs medianHouseValue
medianHouseValue
0e+00
2e+05
4e+05
-124
-122
-120
-118
-116
-114
longitude
latitude vs medianHouseValue
medianHouseValue
0e+00
2e+05
4e+05
34
36
38
40
42
latitude
housingMedianAge vs medianHouseValue
medianHouseValue
0e+00
2e+05
4e+05
0
10
20
30
40
50
housingMedianAge
aveRooms vs medianHouseValue
medianHouseValue
0e+00
2e+05
4e+05
0
20
40
60
80
100
120
140
aveRooms





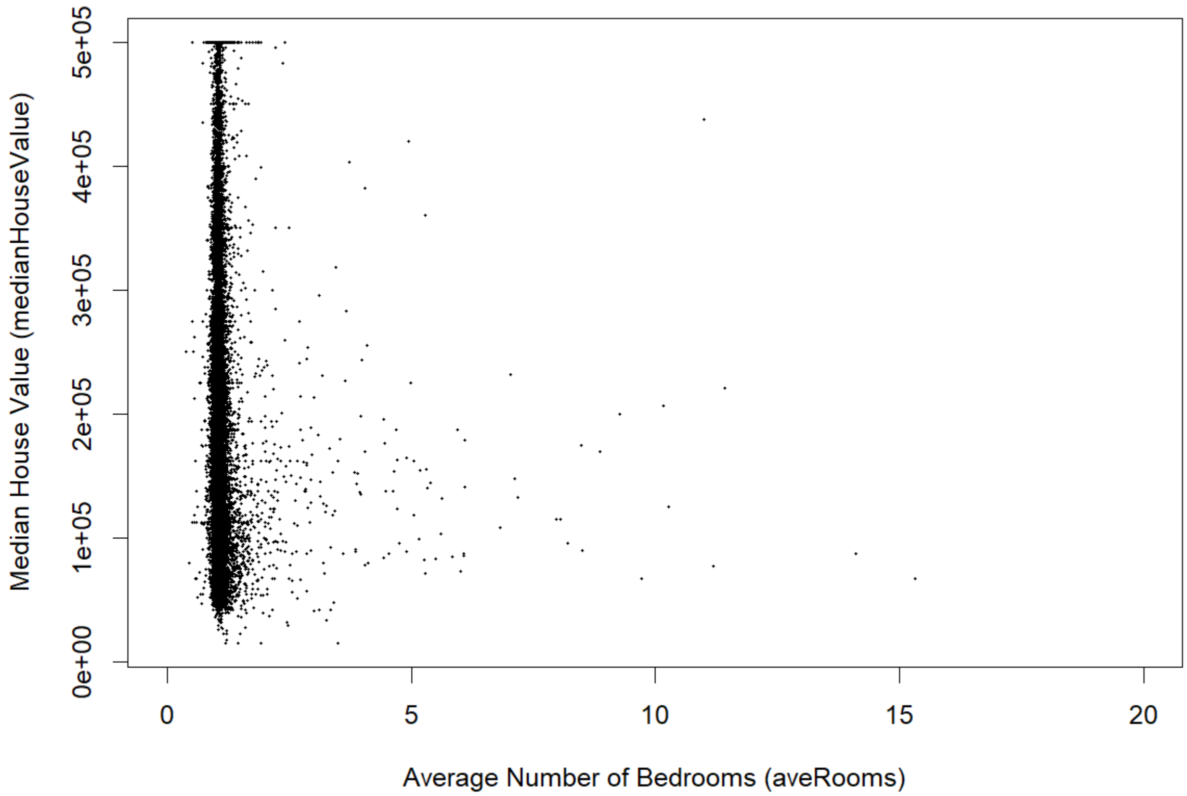


Median House Value vs Longitude/Latitude: there is no clear linear relationship or overall trend. But we can see that for some specific longtitude/latitude the housing value is higher. This is consistent with our findings in the geospacial plot - the peaks in the above plot has to do with the relative location to major cities, SF and LA. This motivates us to do some transformation on the raw longitude and latitude data while integrating relevant information in the MLR framwork later on.

Median House Value vs Housing Media Age: No clear pattern again just by looking at the plot. However the high density of data on the rightmost band (representing housingMedianAge = 52) shows another sign of data censoring.

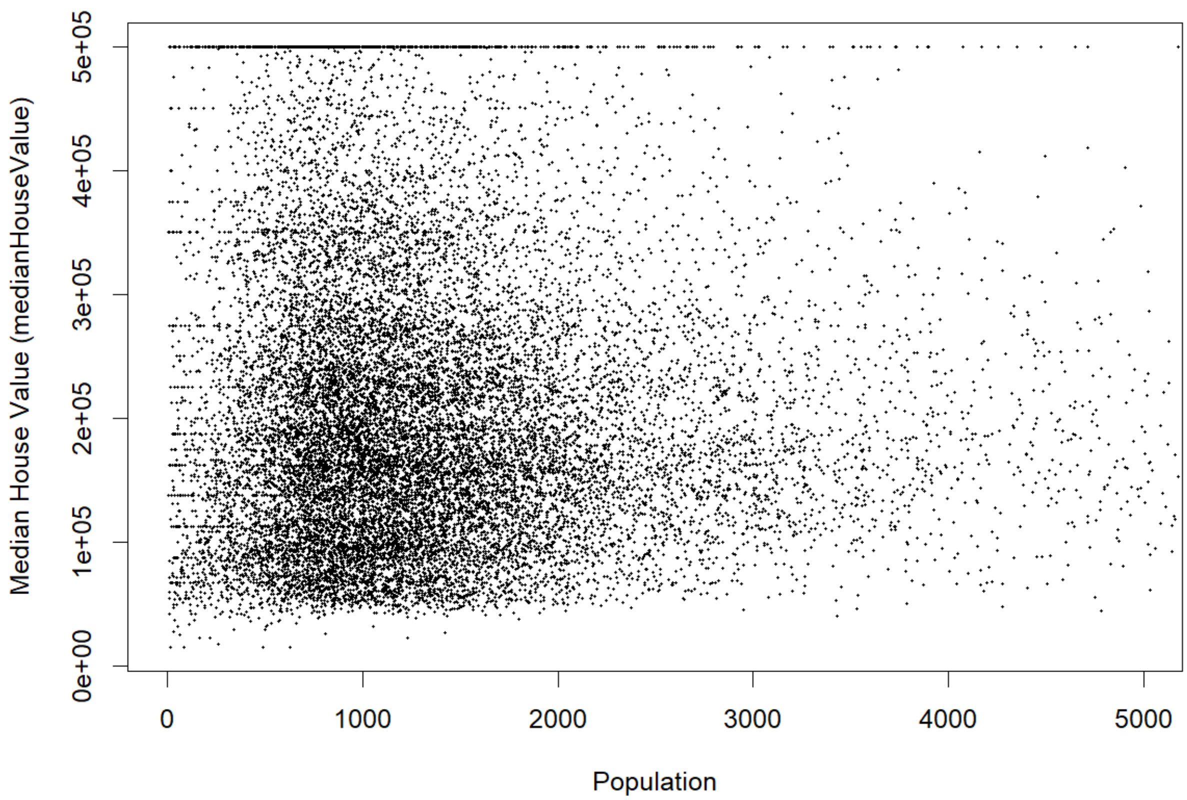
Median House Value vs Average Rooms / Average Bedrooms: The linear pattern is not clear when we plot against the entire data range for rooms. These two plots exposes some issue of high leverage points - some regions have average number of bedrooms over 30 and average number of rooms over 140. These abnormalities are worthy to be investigated and treated when we fit the MLR.

```
# Focus on the range of [0, 20] for aveRooms
plot(data$aveBedrooms, data$medianHouseValue,
     xlim = c(0, 20),
     pch = 16,
     xlab = "Average Number of Bedrooms (aveRooms)",
     ylab = "Median House Value (medianHouseValue)",
     cex = 0.3)
```



Median House Value vs Population: The relationship appear non-linear even if we zoom in to the population data range 0 to 5000:

```
plot(data$population, data$medianHouseValue,
    xlim = c(0, 5000),
     pch = 16,
     xlab = "Population",
     ylab = "Median House Value (medianHouseValue)",
     cex = 0.3)
```



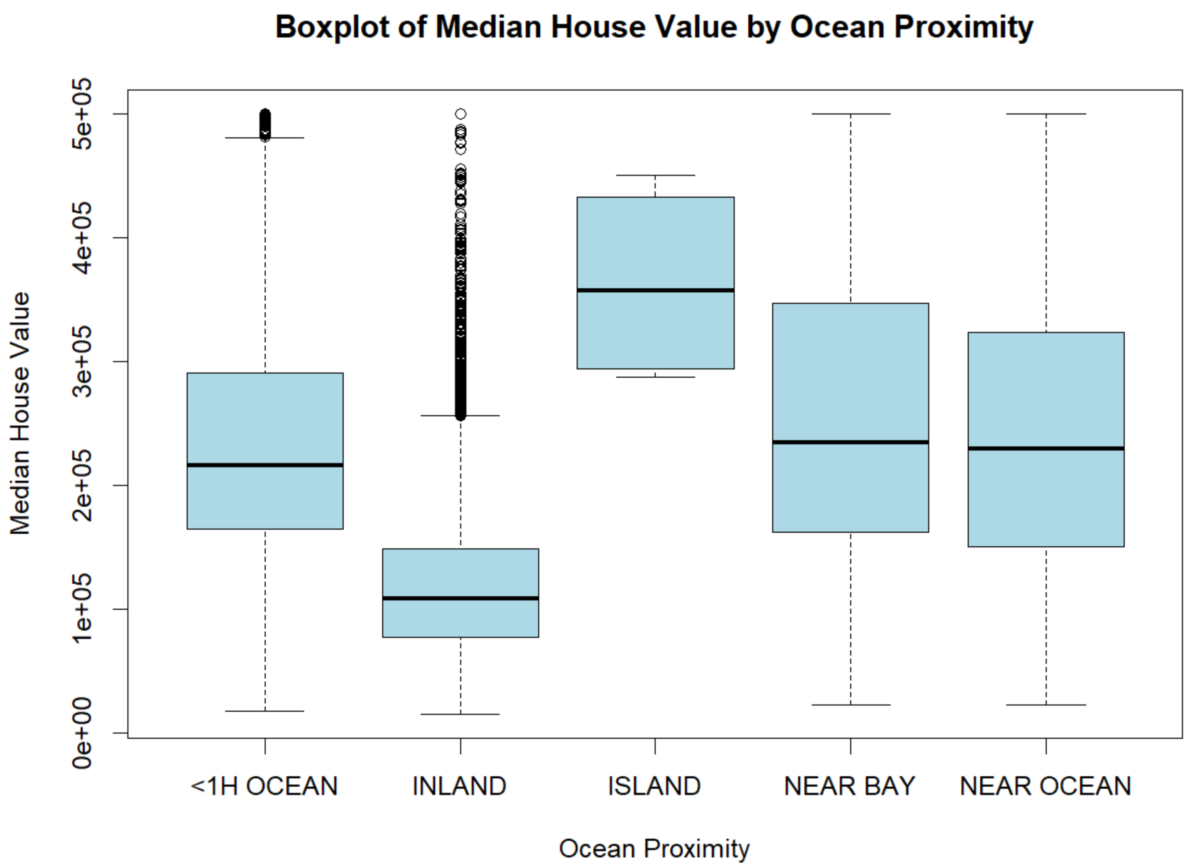
Median House Value vs Median Income: We can see some clear positive trend. This resonates with the correlation plot result, where these two features are positively correlated.

For those pairs where linear relationship is unclear, we will explore log transformations and polynomials in section 1.2.6 single linear regression.

1.2.5 Boxplot of Median House Value by Ocean Proximity

For the qualitative variable, Ocean Proximity, we can examine the differences across different classes via a boxplot:

```
boxplot(medianHouseValue ~ oceanProximity, data = data,
        main = "Boxplot of Median House Value by Ocean Proximity",
        xlab = "Ocean Proximity",
        ylab = "Median House Value",
        col = "lightblue")
```

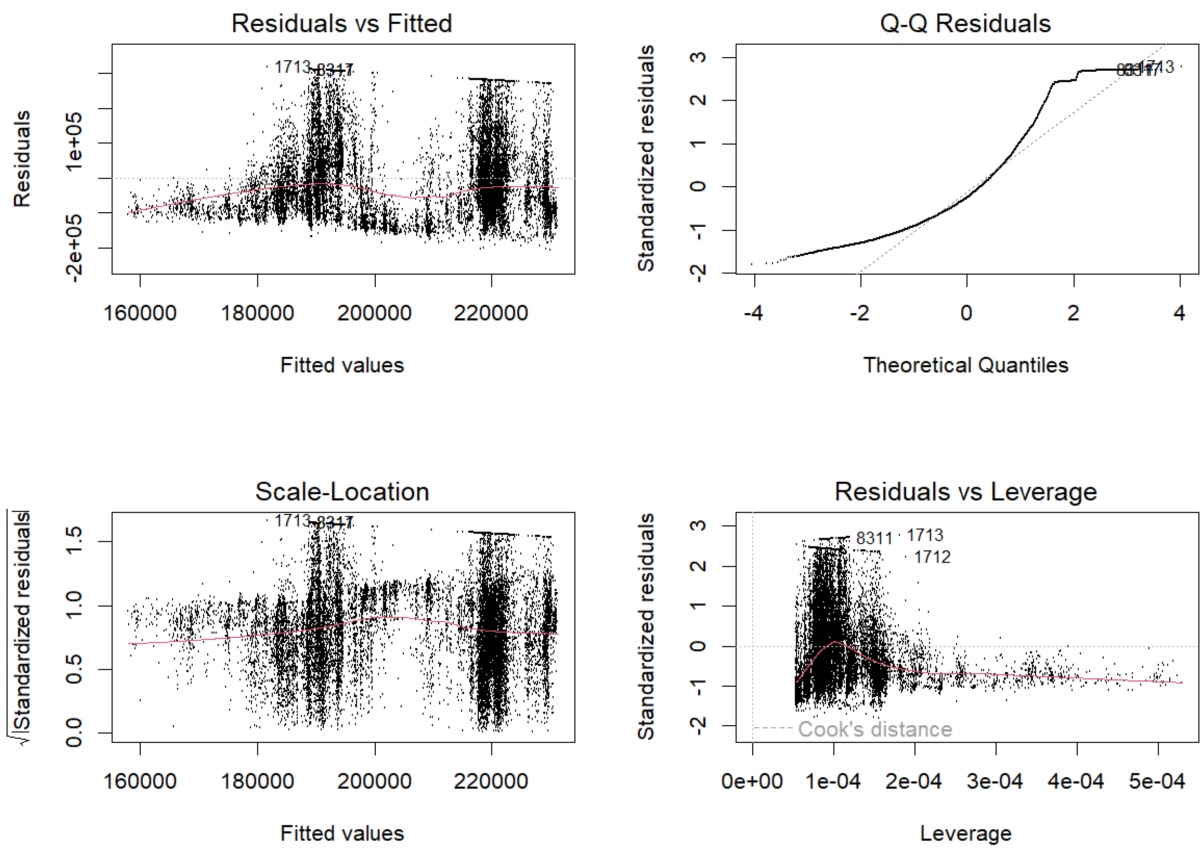


From the plot, island housing appears to be most valuable on average, whereas inland housing appears to be least valuable. The median value for other 3 classes (<1h ocean, near bay and near ocean) seems to be close to each other.

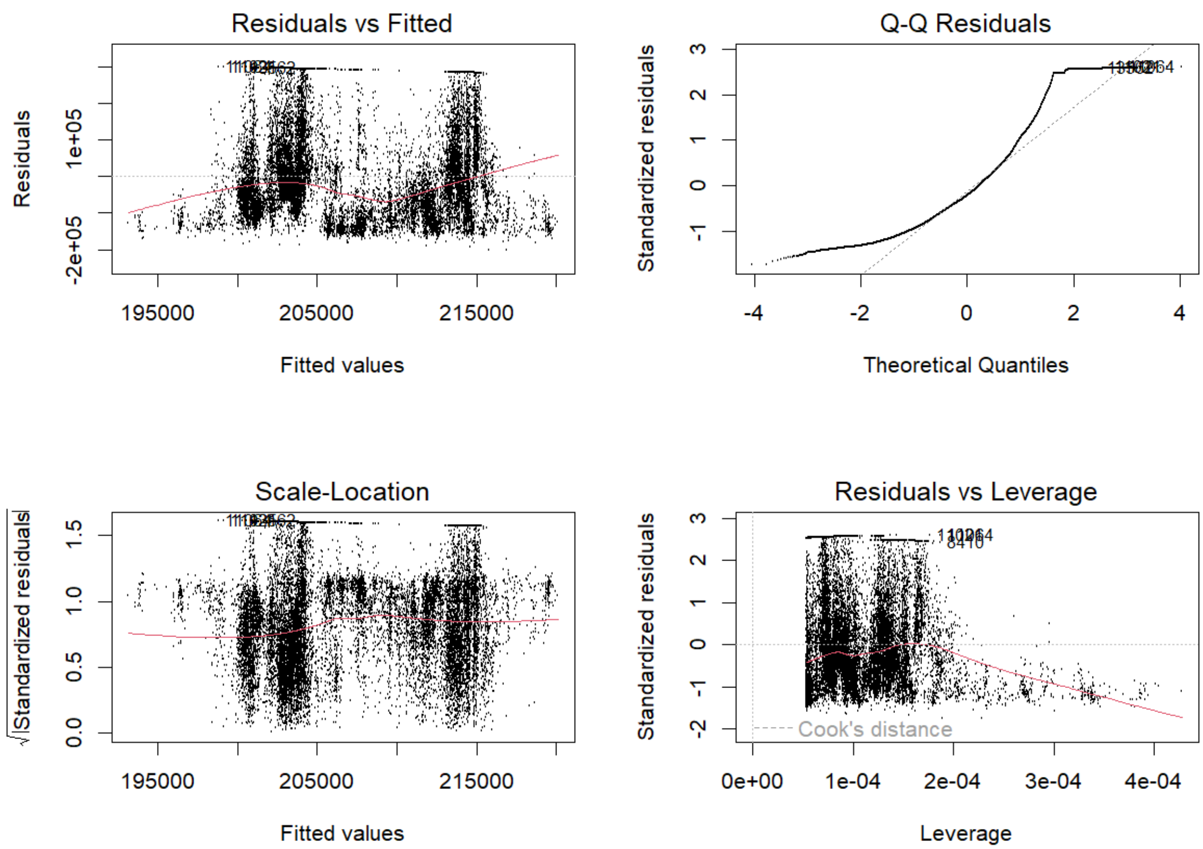
1.2.6 Non-linear considerations

By performing single linear regressions, we find that individual pair of relationships are all statistically significant. However, some assumptions necessary for OLS are clearly violated (e.g. heteroscedasiticty, normality), and some relationship are clearly non-linear. We therefore consider polynomial and log transformations for the following pairwise relationships:

```
lm.latitude = lm(medianHouseValue ~ latitude, data = data)
lm.longitude = lm(medianHouseValue ~ longitude, data = data)
par(mfrow = c(2,2))
plot(lm.latitude, cex = 0.1)
```




```r
plot(lm.longitude, cex = 0.1)
```



Single regression of medianHouseValue on longitude and latitude shows clear non-linearity. This is evident from the residual plot, where the red reference line shows curved patterns.

```r
# logitude and latitude up to degree 3:
lm.latitude.poly2 = lm(medianHouseValue ~ poly(latitude, 2, raw = T), data = data)
lm.latitude.poly3 = lm(medianHouseValue ~ poly(latitude, 3, raw = T), data = data)
lm.longitude.poly2 = lm(medianHouseValue ~ poly(longitude, 2, raw = T), data = data)
lm.longitude.poly3 = lm(medianHouseValue ~ poly(longitude, 3, raw = T), data = data)
# par(mfrow = c(2,2))
# plot(lm.latitude, cex = 0.1)
# plot(lm.latitude.poly2, cex = 0.1)
# plot(lm.latitude.poly3, cex = 0.1)

anova(lm.latitude, lm.latitude.poly2, lm.latitude.poly3)
```

```
## Analysis of Variance Table
##
## Model 1: medianHouseValue ~ latitude
## Model 2: medianHouseValue ~ poly(latitude, 2, raw = T)
## Model 3: medianHouseValue ~ poly(latitude, 3, raw = T)
##   Res.Df        RSS Df  Sum of Sq       F    Pr(>F)
## 1  18448 2.4188e+14
## 2  18447 2.3566e+14  1 6.2128e+12 486.911 < 2.2e-16 ***
## 3  18446 2.3537e+14  1 2.9719e+11  23.291 1.403e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

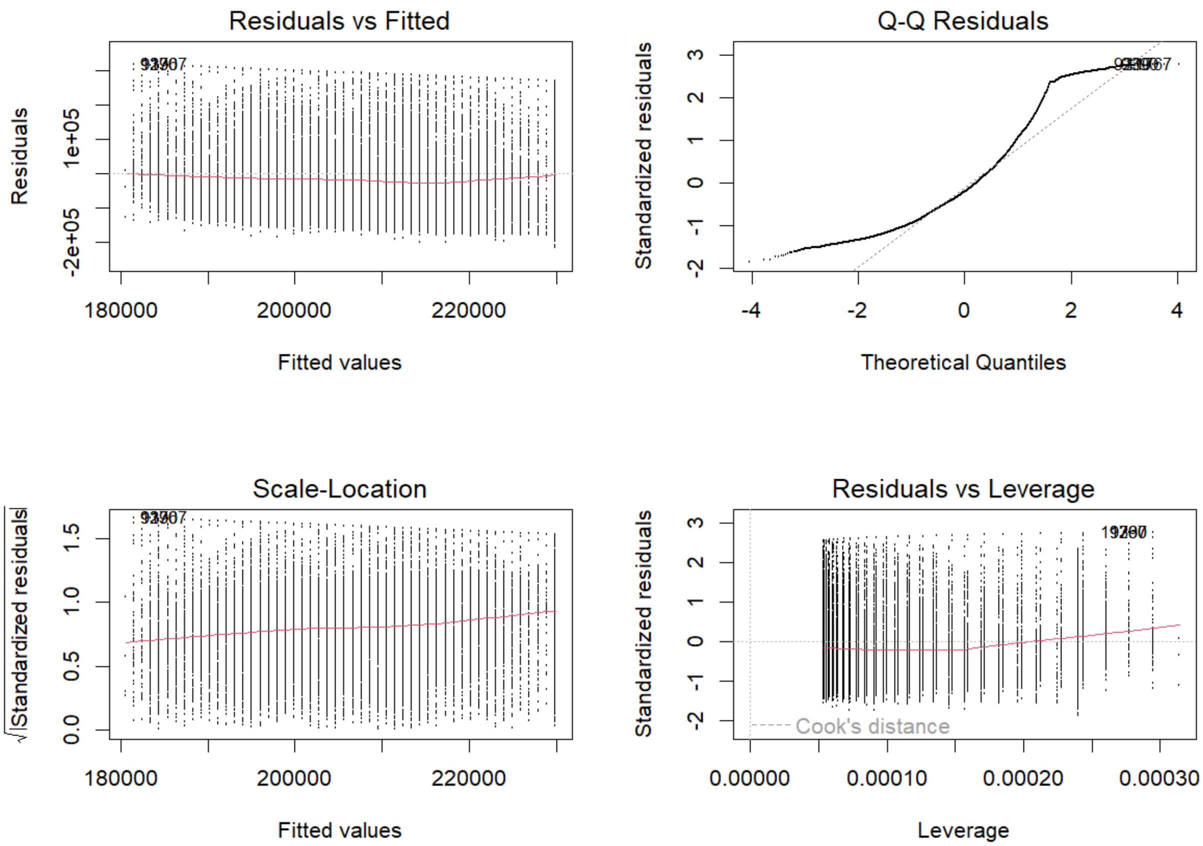
```r
anova(lm.longitude, lm.longitude.poly2, lm.longitude.poly3)
```

```
## Analysis of Variance Table
##
## Model 1: medianHouseValue ~ longitude
## Model 2: medianHouseValue ~ poly(longitude, 2, raw = T)
## Model 3: medianHouseValue ~ poly(longitude, 3, raw = T)
##   Res.Df        RSS Df  Sum of Sq       F    Pr(>F)
## 1  18448 2.4646e+14
## 2  18447 2.4623e+14  1 2.3032e+11  17.547 2.815e-05 ***
## 3  18446 2.4212e+14  1 4.1097e+12 313.102 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

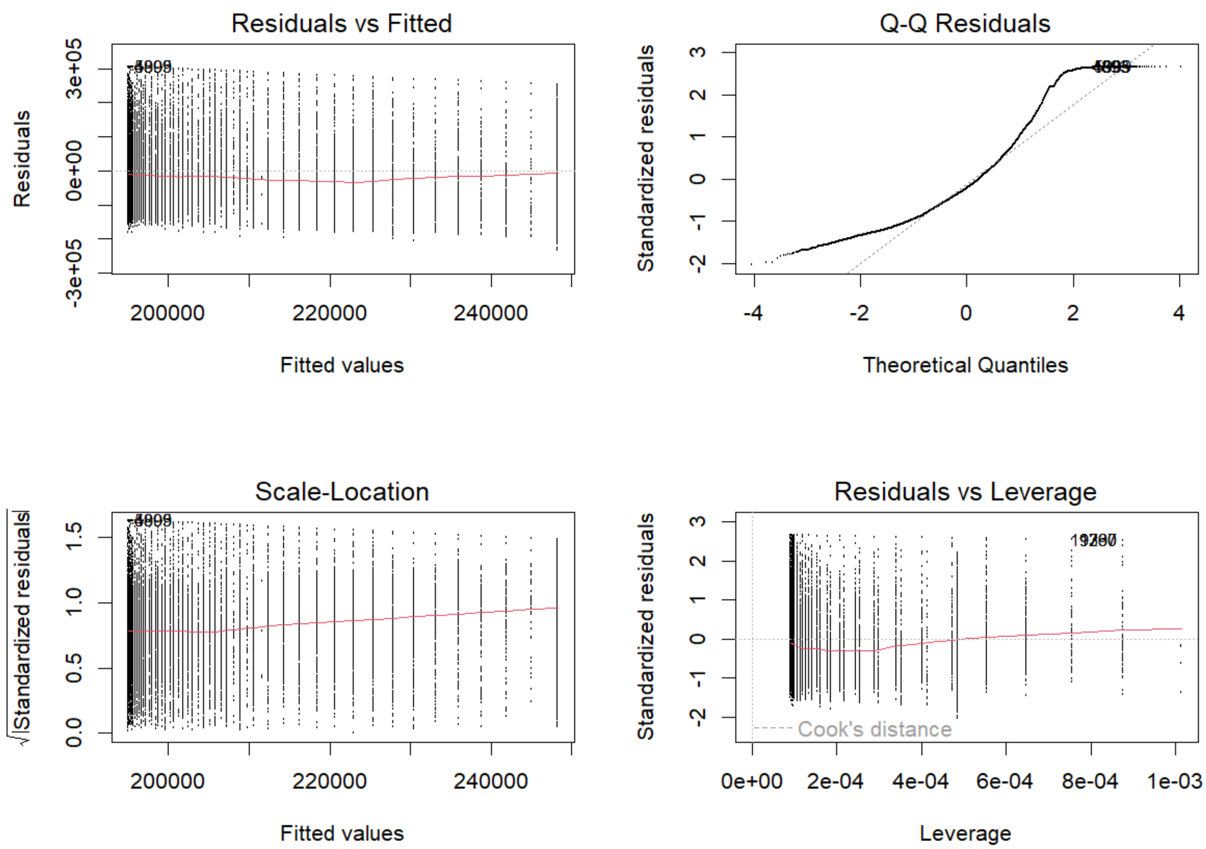
By fitting a polynomial regression up to degree 3, we observe an improvement in Adjusted R^2, and the residual plot pattern also improves. The ANOVA test further supports a significant model improvement. To avoid overfitting, we choose not to explore higher-degree polynomials.

```r
lm.housingMedianAge <- lm(medianHouseValue ~ housingMedianAge, data = data)
lm.housingMedianAge.poly2 <- lm(medianHouseValue
                                ~ poly(housingMedianAge, 2, raw = T),
                                data = data)

par(mfrow = c(2,2)); plot(lm.housingMedianAge, cex = 0.1)
```




```r
par(mfrow = c(2,2)); plot(lm.housingMedianAge.poly2, cex = 0.1)
```




```r
summary(lm.housingMedianAge)
```

```
##
## Call:
## lm(formula = medianHouseValue ~ housingMedianAge, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -214856  -85248  -25850   58318  318407
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      179663.30    2100.05   85.55   <2e-16 ***
## housingMedianAge    965.22      67.17   14.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115100 on 18448 degrees of freedom
## Multiple R-squared:  0.01107,    Adjusted R-squared:  0.01102
## F-statistic: 206.5 on 1 and 18448 DF,  p-value: < 2.2e-16
```

```r
summary(lm.housingMedianAge.poly2)
```

```
##
## Call:
## lm(formula = medianHouseValue ~ poly(housingMedianAge, 2, raw = T),
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -233248  -85747  -25328   59177  304946
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         213548.81    3919.03   54.49  < 2e-16 ***
## poly(housingMedianAge, 2, raw = T)1  -1917.96     289.70   -6.62 3.68e-11 ***
## poly(housingMedianAge, 2, raw = T)2     49.72       4.86   10.23  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 114700 on 18447 degrees of freedom
## Multiple R-squared:  0.01665,    Adjusted R-squared:  0.01654
## F-statistic: 156.1 on 2 and 18447 DF,  p-value: < 2.2e-16
```

```r
anova(lm.housingMedianAge,
      lm.housingMedianAge.poly2)
```

```
## Analysis of Variance Table
##
## Model 1: medianHouseValue ~ housingMedianAge
## Model 2: medianHouseValue ~ poly(housingMedianAge, 2, raw = T)
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  18448 2.4426e+14
## 2  18447 2.4288e+14  1 1.3777e+12 104.64 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The normality assumption appears violated for housingMedianAge, as seen in the QQ plot. Although the residual pattern remains largely unchanged, the second-degree polynomial model improves Adjusted R^2 (from 0.011 to 0.017) and yields a statistically significant improvement in fit (ANOVA $p < 2.2e-16$). Therefore, we include the degree-2 polynomial in the model selection stage.

```r
lm.medianIncome <- lm(medianHouseValue ~ medianIncome, data = data)
# summary(lm.medianIncome)
# par(mfrow = c(2,2)); plot(lm.medianIncome, cex = 0.2)

lm.medianIncome.poly2 <- lm(medianHouseValue ~ poly(medianIncome, 2, raw = T),
                            data = data)
# summary(lm.medianIncome.poly2)
# par(mfrow = c(2,2)); plot(lm.medianIncome.poly2, cex = 0.2)

lm.medianIncome.poly3 <- lm(medianHouseValue ~ poly(medianIncome, 3, raw = T),
                            data = data)
```

```r
# summary(lm.medianIncome.poly3)
# par(mfrow = c(2,2)); plot(lm.medianIncome.poly3, cex = 0.2)

lm.medianIncome.poly4 <- lm(medianHouseValue ~ poly(medianIncome, 4, raw = T),
                            data = data)
anova(lm.medianIncome,
      lm.medianIncome.poly2,
      lm.medianIncome.poly3,
      lm.medianIncome.poly4)
```

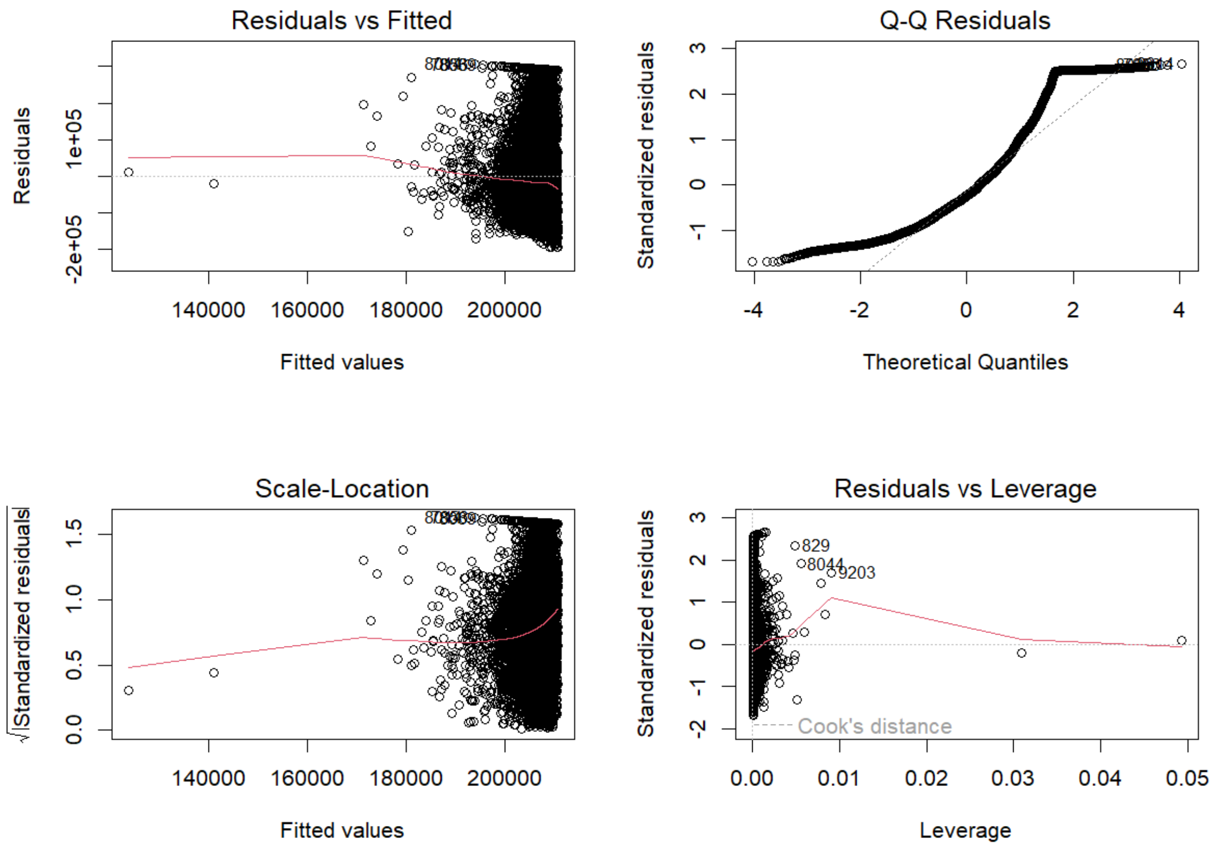
```
## Analysis of Variance Table
##
## Model 1: medianHouseValue ~ medianIncome
## Model 2: medianHouseValue ~ poly(medianIncome, 2, raw = T)
## Model 3: medianHouseValue ~ poly(medianIncome, 3, raw = T)
## Model 4: medianHouseValue ~ poly(medianIncome, 4, raw = T)
##   Res.Df        RSS Df  Sum of Sq        F  Pr(>F)
## 1  18448 1.2956e+14
## 2  18447 1.2844e+14  1 1.1210e+12 163.6184 < 2e-16 ***
## 3  18446 1.2639e+14  1 2.0433e+12 298.2294 < 2e-16 ***
## 4  18445 1.2637e+14  1 2.1319e+10   3.1117 0.07775 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Polynomial regression of medianHouseValue on medianIncome up to degree 3 is also shown to have a better fit than lower degree polynomial as well as linear model. In particular, with the evidence from the anova F-test, we conclude that degree-3 is significantly better than degree-2, whereas moving to higher degree (i.e. quartic) does not bring additional significant benefit.

```r
lm.pop <- lm(medianHouseValue ~ population, data = data)
summary(lm.pop)
```

```
##
## Call:
## lm(formula = medianHouseValue ~ population, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -195709  -87189  -26995   58504  307357
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.108e+05  1.366e+03 154.254  < 2e-16 ***
## population  -2.437e+00  7.492e-01  -3.253  0.00114 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115700 on 18448 degrees of freedom
## Multiple R-squared:  0.0005732,  Adjusted R-squared:  0.000519
## F-statistic: 10.58 on 1 and 18448 DF,  p-value: 0.001145
```

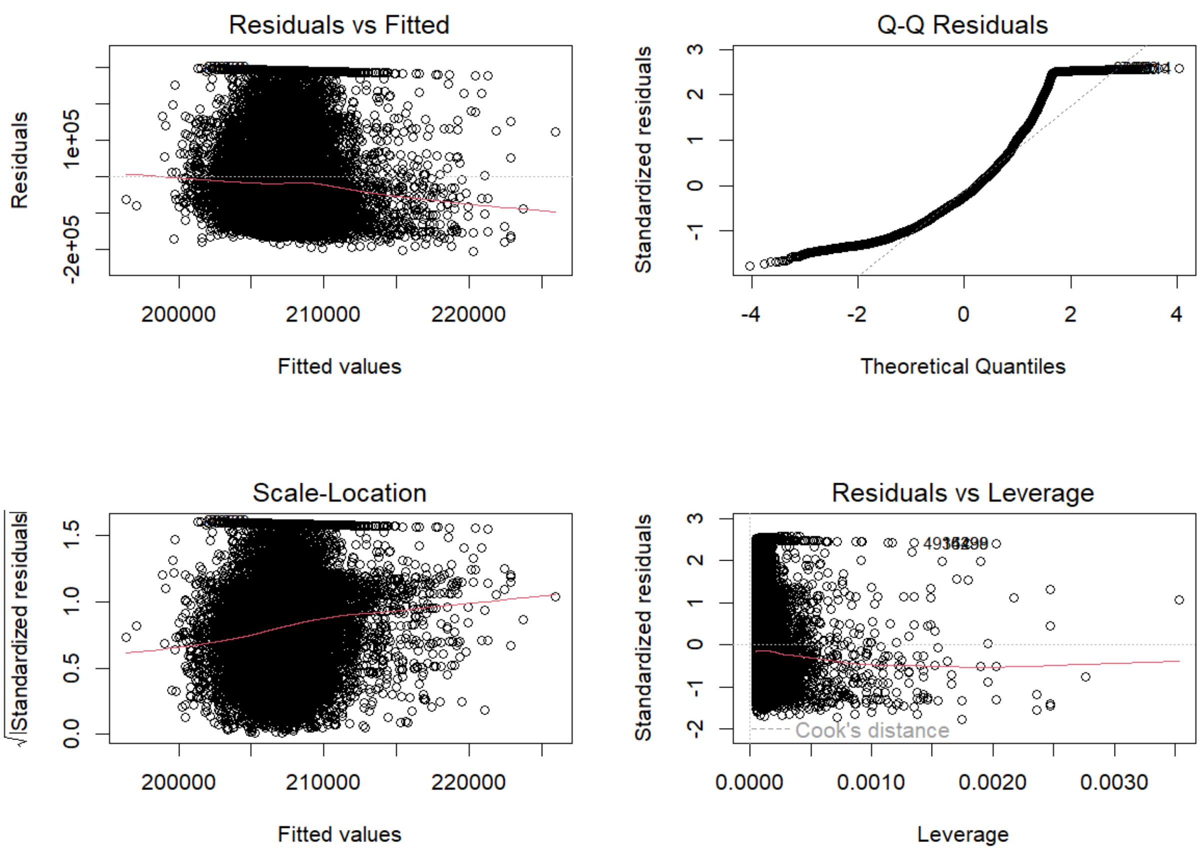
```r
par(mfrow = c(2,2)); plot(lm.pop)
```




```r
lm.log.pop <- lm(medianHouseValue ~ log(population), data = data)
summary(lm.log.pop)
```

```
##
## Call:
## lm(formula = medianHouseValue ~ log(population), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -205287  -87246  -27031   58539  298671
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         229383       8132  28.209  < 2e-16 ***
## log(population)      -3147       1151  -2.734  0.00627 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115700 on 18448 degrees of freedom
## Multiple R-squared:  0.0004049,  Adjusted R-squared:  0.0003507
## F-statistic: 7.472 on 1 and 18448 DF,  p-value: 0.006272
```

```r
par(mfrow = c(2,2)); plot(lm.log.pop)
```



The predictor "population" suffers from high leverage point and heteroscedasticity. To address this issue, a log transform is applied to the original data. From the pre-log-transform residual plot, the residuals are clustered at one end, whereas the post-log plot show better random scattering. This indicates that a log transform improves linearity between the response and the predictor, which motivates us to implement this improvement in our final model.

---

2 Multiple linear regression

2.1 Initial MLR model using all appropriate predictors

We will be using the all the given predictors except for id, since it is irrelevant.

```r
# Initial_data is a copy of data
initial_data <- read.csv("Assignt1_data.csv")
initial_data <- na.omit(initial_data)

initial_fit <- lm(medianHouseValue ~ . - id, data = initial_data)
summary(initial_fit)
```

```
##
## Call:
## lm(formula = medianHouseValue ~ . - id, data = initial_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -632258  -45300  -11782   30245  443819
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -2.272e+06  9.759e+04 -23.278  < 2e-16 ***
## longitude                 -2.655e+04  1.131e+03 -23.473  < 2e-16 ***
## latitude                  -2.473e+04  1.119e+03 -22.103  < 2e-16 ***
## housingMedianAge           8.205e+02  4.761e+01  17.235  < 2e-16 ***
## aveRooms                  -8.784e+03  6.219e+02 -14.126  < 2e-16 ***
## aveBedrooms                5.338e+04  2.973e+03  17.956  < 2e-16 ***
## population                -6.741e-01  4.942e-01  -1.364 0.172593
## medianIncome               4.209e+04  4.471e+02  94.142  < 2e-16 ***
## oceanProximityINLAND      -3.845e+04  1.931e+03 -19.906  < 2e-16 ***
## oceanProximityISLAND       1.290e+05  3.586e+04   3.597 0.000322 ***
## oceanProximityNEAR BAY     4.059e+03  2.090e+03   1.942 0.052115 .
## oceanProximityNEAR OCEAN  8.919e+03  1.714e+03   5.204 1.97e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71650 on 18438 degrees of freedom
## Multiple R-squared:  0.6168, Adjusted R-squared:  0.6166
## F-statistic:  2698 on 11 and 18438 DF,  p-value: < 2.2e-16
```

2.2 Discussion of the initial model

The p-value of the F-test is practically zero, therefore we have sufficient evidence to reject the null hypothesis that all regression coefficients are zero. This means that the model has overall significance, and that at least one of the predictors are useful in explaining the variation in medianHouseValue.

Having concluded that this model (initial_fit) has overall significance, we can gauge the significance of individual predictors through the t-test p-values: At 5% level of significance, we cannot say that population is useful, as we do not have sufficient evidence to reject the null hypothesis that its corresponding coefficient is non-zero. The dummy variable, "NEAR BAY", associated to the categorical predictor, "oceanProximity", is also shown to be insignificant. This might be an indication that comparing to the base case (<1H OCEAN), NEAR BAY does not lead to significant change in Median House Price. A potential real-world explanation of why nearbay might not be influential is that "bay" is a very broad concept; some could be more favourable than others (eg. close to amenities OR prone to environmental issues) so its hard to deduce a clear relationship between proximity to bay and house value.

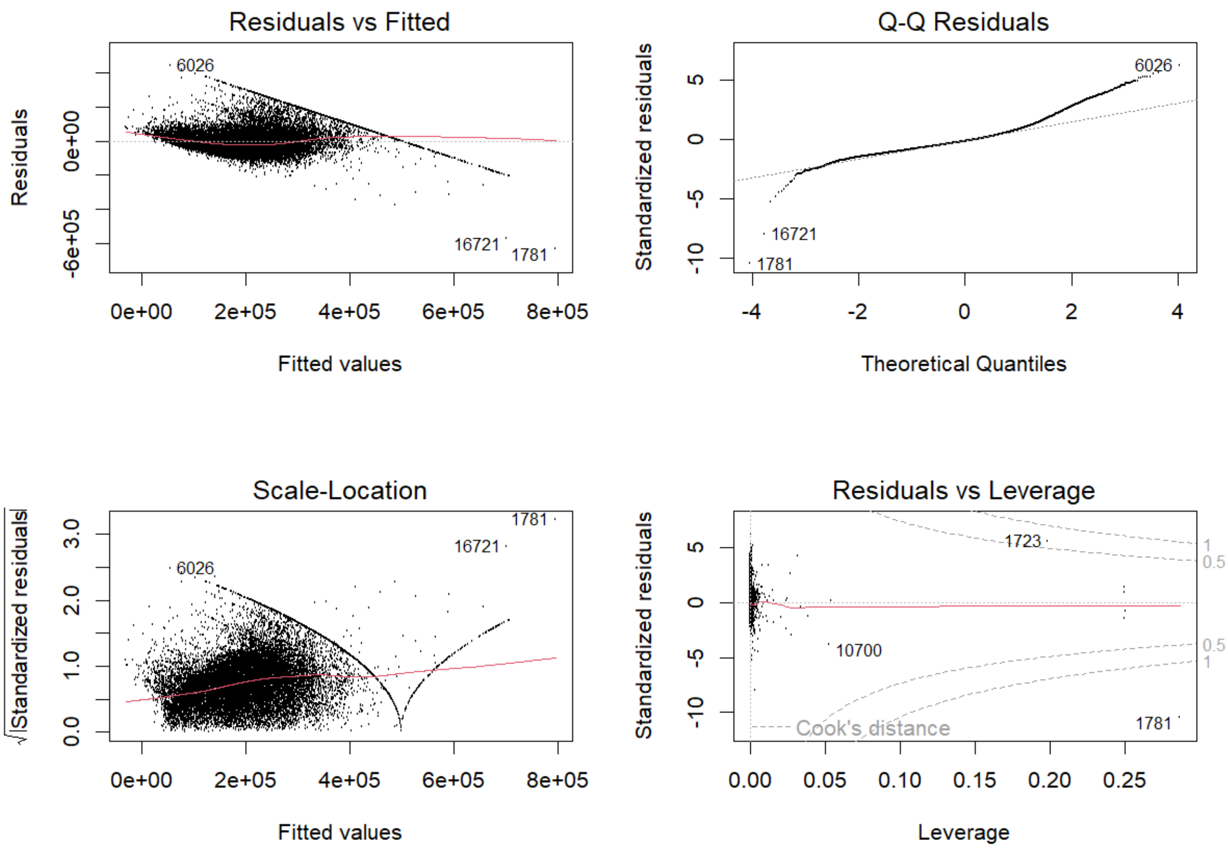
Moreover, the R-squared and adjusted R-squared both equal to 0.617 when corrected to 3 decimal places. This means that 61.7% of the variation in the predictors are useful in explaining the variation in the response variable (medianHouseValue).

Additionally, this initial model also reveals some clear relationships among the varaibles: Median income has a large, positive coefficient so its a strong positive predictor as we anticipated from the preliminary analysis.

longitude and latitude both have large negative coefficients which reiterates our findings before regarding geographic (major cities) relevance.

2.3 Checking issues in the initial model

```r
par(mfrow = c(2,2))
plot(initial_fit, cex = 0.1)
```



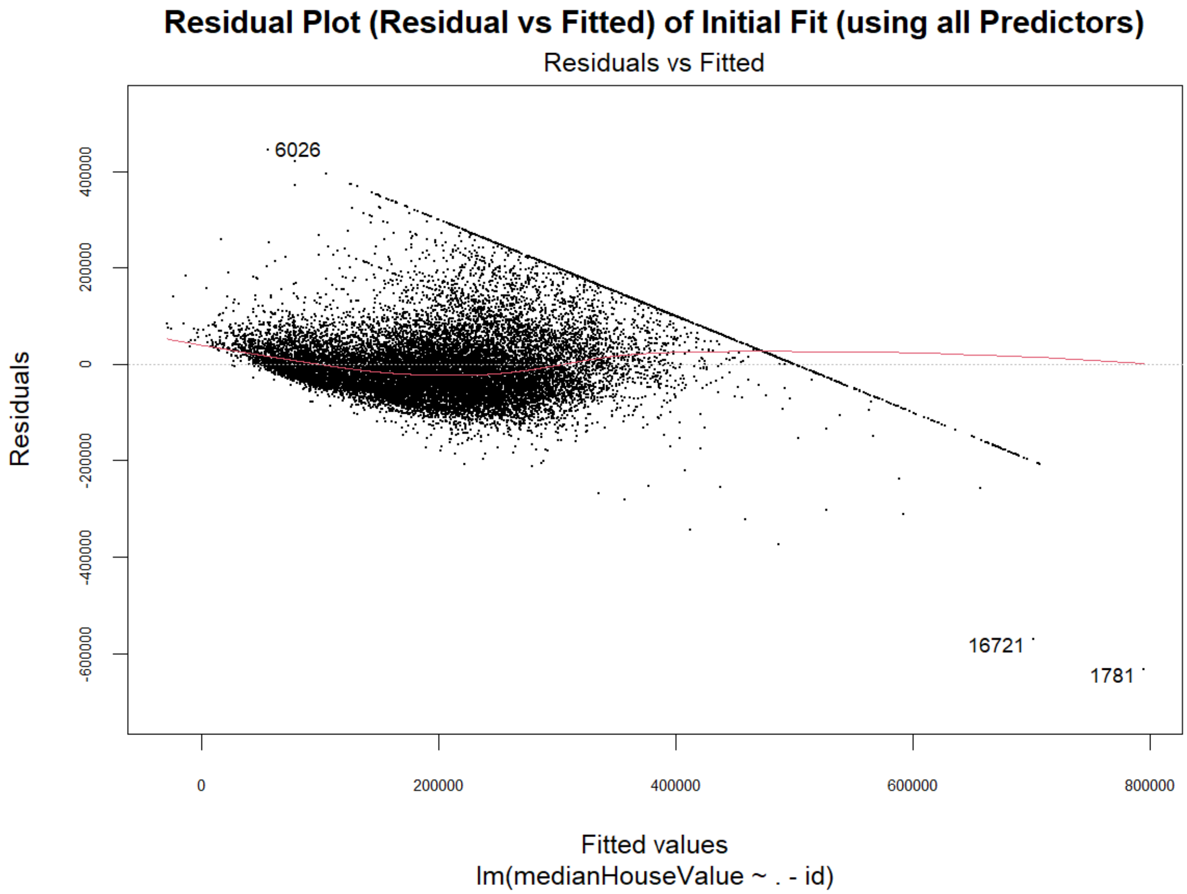
2.3.1 Residual Plot:

The residual plot shows that the points are not quite randomly scattered around zero. This means that the homoscedasiticity and linear assumption might be questionable. To address non-linearity, we can consider polynomial terms or interactions.

The funnel shape (the spread of residuals seems to increase as the fitted values increase) within the fitted value range (0, 400000) strengthens my doubt of heteroscedasticity. We may consider transforming some of our predictors later.

There are some outliers in the residuals that are far away from zero. These influential points may be high-leverage or outliers or both - should be investigated later.

```r
par(mfrow = c(1, 1))
options(scipen = 999)
plot(initial_fit, which = 1, cex = 0.2, pch = 16, cex.axis = 0.6,
     main = "Residual Plot (Residual vs Fitted) of Initial Fit (using all Predictors)")
```



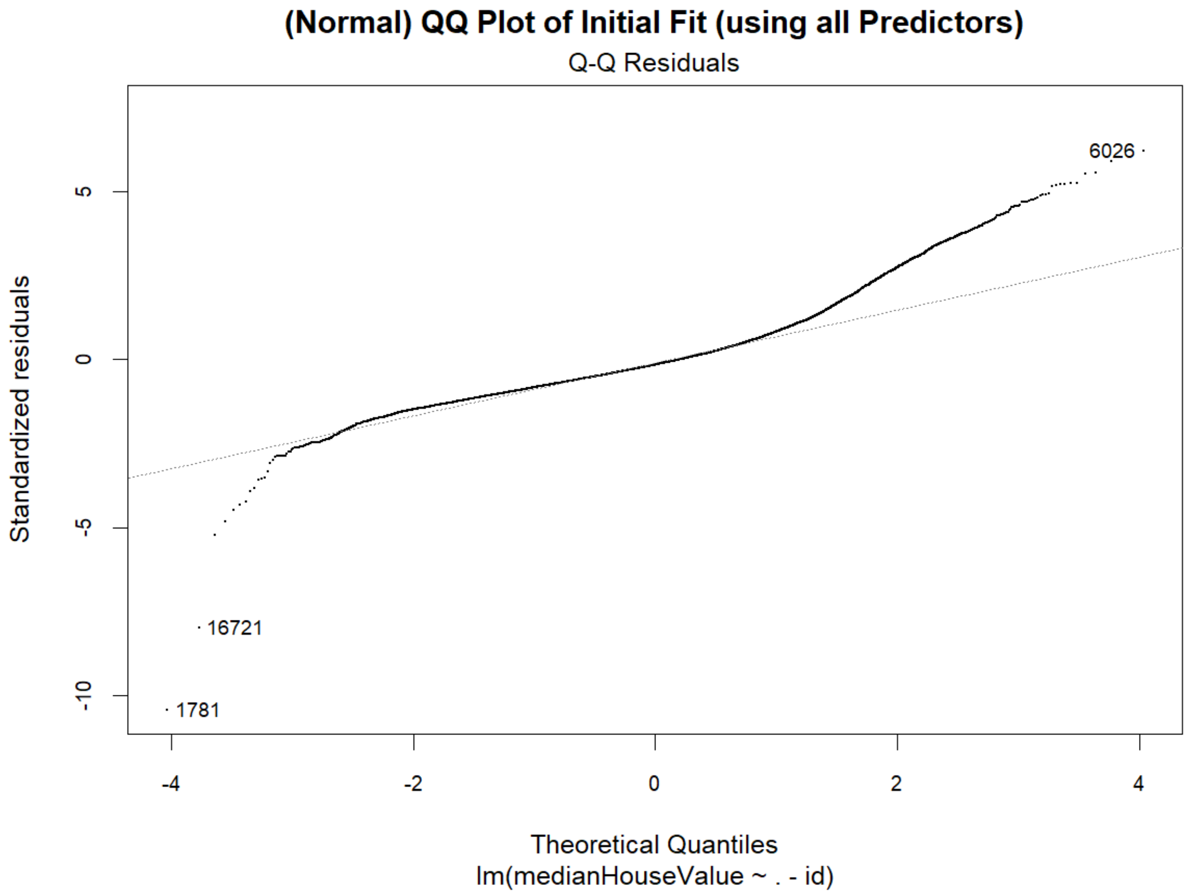
2.3.2 QQ Plot for Residuals:

A key assumption behind generalised linear model is that the error term is normally distributed. This is why t-statistics and F-statistics can be used in our previous testing.

T-statistics are robust under some mild deviation from normality, but under extreme non-normality, these statistics become less reliable.

In our plot, the points deviates from the reference line (dashed line) for larger and smaller quantiles (roughly outside of this range: (-2, 1.5)), indicating non-normality (especially high skewness) and influence of outliers especially at the tails.

```r
plot(initial_fit, which = 2, cex = 0.2, pch = 16, cex.axis = 0.8,
     main = "(Normal) QQ Plot of Initial Fit (using all Predictors)")
```



2.3.3 Outliers

Outlier identification: as a rule of thumb, we consider those whose studentised residual has a magnitude greater than 3 as outliers here.

However we cannot just simply remove the outliers in this case. This is because these outliers could be attributable to model specification or other problems. This is partially addressed in 2.4.1 by bedroomsPerRoom. Note: outliers are not a big consideration in this data set because the response variable is both left and right censored, this is also reflected in the final model.

```
residuals_initial_fit <- residuals(initial_fit)
stdresiduals_initial_fit <- rstandard(initial_fit)
outlier_row_number <- which(abs(stdresiduals_initial_fit) > 3)

length(outlier_row_number) # gives how many outliers are there
```

```
## [1] 314
```

2.3.4 High Leverage Points

We will compute the leverage statistic hi and to see whether it is » (p + 1)/n. Where p is the number of predictors in the model and n is the sample size.

Having 3,317 high leverage points out of 18,450 data points means that about 18% of the data has high leverage. This indicates that our regression line can change dramatically with small changes in the predictors. One possible reason is that we are overfitting the data - and a reason to this is having too many predictor variables.

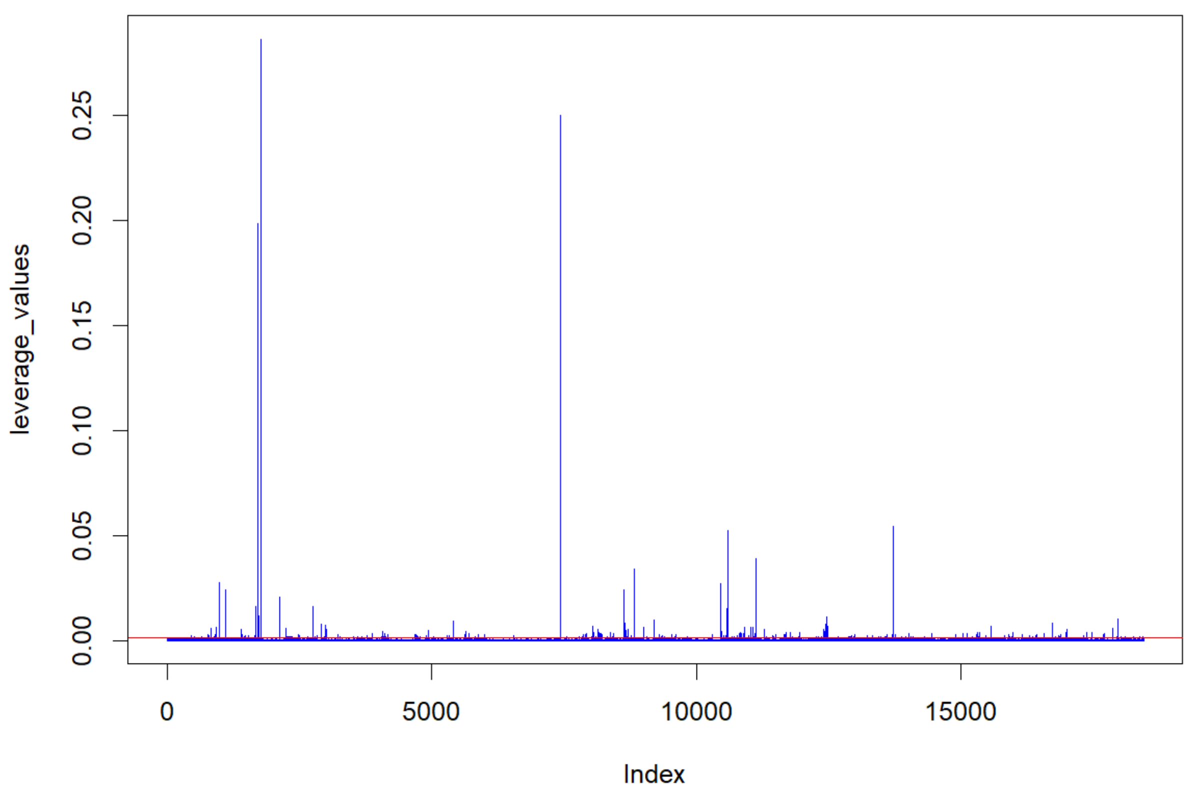
```r
leverage_values <- hatvalues(initial_fit)

p_lvg <- length(coef(initial_fit))
n_lvg <- nrow(data)
threshold_lvg <- (p_lvg + 1) / n_lvg

highlvg_row_number <- which(leverage_values > threshold_lvg)
length(highlvg_row_number)
```

```
## [1] 3317
```

```r
plot(leverage_values, type = "h", col = "blue")
abline(h = 2 * mean(leverage_values), col = "red")
```



2.3.5 Collinearity

The arbitrary threshold of severe collinearity is VIF greater or equal to 5. Here, all the predictors are shown to have a non-severe VIF. However, Longitude and Latitude shows relatively high VIF comparing to other predictors - a cause of this is the high correlation between the two.

```
vif(initial_fit)
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## longitude          18.464342  1        4.297015
## latitude           20.529183  1        4.530914
## housingMedianAge    1.295871  1        1.138363
## aveRooms            8.994891  1        2.999148
## aveBedrooms         7.637752  1        2.763648
## population          1.134318  1        1.065044
## medianIncome        2.610282  1        1.615637
## oceanProximity      4.089926  4        1.192517
```

2.4 Model Improvements

2.4.1 New predictors added

```
# New fit
data$bedroomsPerRoom <- data$aveBedrooms / data$aveRooms
data$incomePerRoom = data$medianIncome / data$aveRooms

# Dist from LA and SF
library(geosphere)
# Need to reference this
la_coords <- c(-118.24, 34.05)   # (longitude, latitude)
sf_coords <- c(-122.42, 37.77)

# Add distance to LA
data$distToLA <- apply(data[, c("longitude", "latitude")], 1, function(coord) {
  distGeo(coord, la_coords) / 1000  # convert meters to km
})

#Add distance to SF
data$distToSF <- apply(data[, c("longitude", "latitude")], 1, function(coord) {
  distGeo(coord, sf_coords) / 1000
})


# Compute direction angles
data$dirToLA <- atan2(data$latitude - la_coords[2], data$longitude - la_coords[1])
data$dirToSF <- atan2(data$latitude - sf_coords[2], data$longitude - sf_coords[1])

# Encode directions using dsin and cos
data$cosDirToLA <- cos(data$dirToLA)
data$sinDirToLA <- sin(data$dirToLA)

data$cosDirToSF <- cos(data$dirToSF)
data$sinDirToSF <- sin(data$dirToSF)

# Remove intermediate angle variables
```

```
data$dirToLA <- NULL
data$dirToSF <- NULL

data$cityProximityScore <- 1 / (1 + data$distToLA) + 1 / (1 + data$distToSF)

#distance to centre
center_lat <- mean(data$latitude)
center_lon <- mean(data$longitude)
data$distToCenter <- sqrt((data$latitude - center_lat)^2 + (data$longitude - center_lon)^2)
```

A number of new predictors have been added that are transformations of previous predictors, going through sequentially these are the justifications for each addition.

- bedroomsPerRoom: this additional predictor allows for the information in aveRooms and aveBedrooms to be included without running into the issue of collinearity in the model since as seen above aveRooms and aveBedrooms are correlated. Additionally it mitigates the issue of high leverage points which were very apparent in both aveRooms and aveBedrooms by standardising. Given the response variable is censored there is no value add for these high leverage points. Finally it is an affluence metric for the house in question. Low values for bedrooms per room meaning that there are far more rooms than bedrooms indicating that the house is less cramped and there's more space, it would make sense that these houses are more valuable.
- incomePerRoom: this feature was included because it captures how much income supports each unit of housing space, how much money is "backing" each room, a relationship that neither raw income nor room count alone can fully express, and one that likely correlates more strongly with housing prices.
- Distances and directions from Los Angeles and San Fransisco. As seen in the Geospatial plot, California has two major cities and those cities have the highest housing prices. In particular they are good proxies for whether medianHouseValue has been censored, because the vast majority of expensive homes, those over $500,001 are in those cities. These engineered features introduce geospatial context by quantifying both the Euclidean distance and relative bearing from each observation to Los Angeles and San Francisco, two primary centers of economic activity in California. Distance to these cities is a strong predictor of housing prices: since proximity to urban centers often means better access to jobs, amenities, and higher demand. Directional features add geographic nuance by distinguishing not just how far a home is, but where it lies in relation to the city, which can reflect differences in development, terrain, or desirability. Encoding direction using sine and cosine avoids issues with angle discontinuity and makes it easier the model to learn spatial patterns.
- cityProximityScore: this feature combines inverse-distance relationships to Los Angeles and San Francisco, assigning higher scores to homes that are closer to either city. It creates a smooth, nonlinear decay of influence with distance, ensuring that proximity to urban centers has a diminishing but continuous effect. This score effectively quantifies urban accessibility, helping the model prioritise locations that are geographically well-positioned relative to high-value economic hubs.
- distToCenter: this feature introduces a global spatial reference by computing the Euclidean distance from each observation to the dataset's centroid, defined by the mean latitude and longitude. This addresses potential spatial drift or boundary bias, where model performance may degrade at the geographic edges of the dataset due to uneven sampling or regional sparsity. By capturing centrality relative to the full spatial domain, not just to high-demand cities, it helps the model learn broad spatial gradients and corrects for systematic variation in housing value that might arise from being on the dataset's periphery rather than near economic centers.

2.4.2 Choosing interaction terms and non-linear transformations

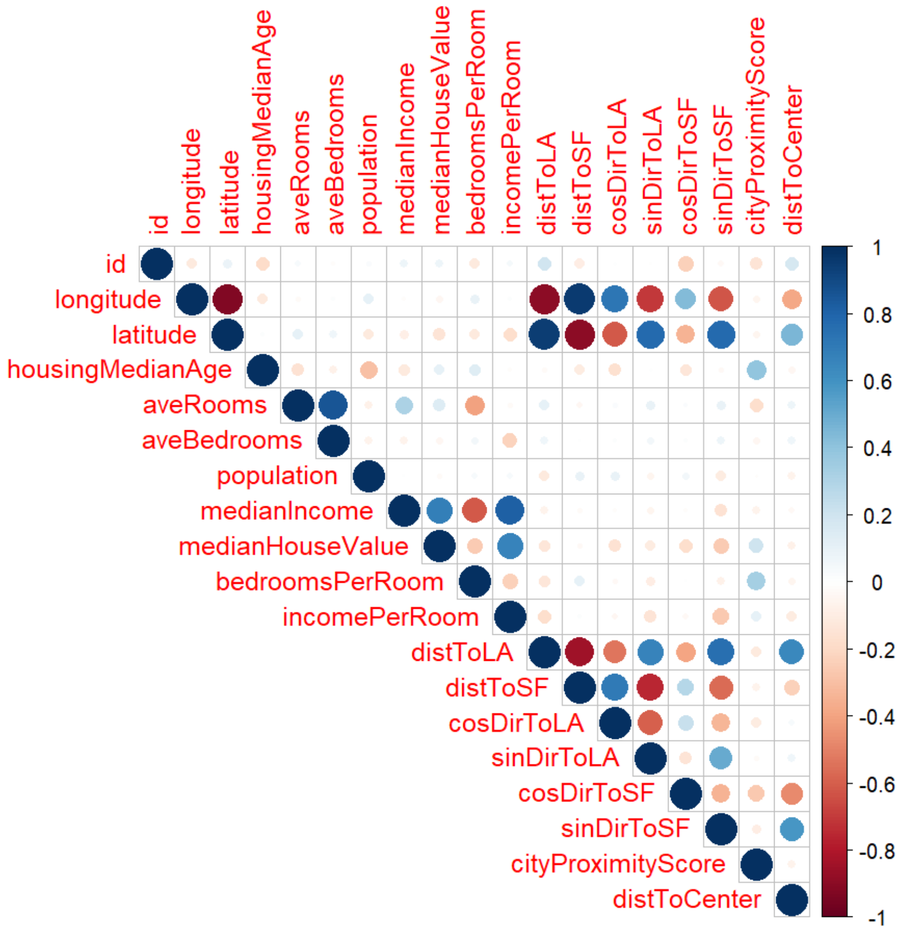
```r
# Select all numeric columns except 'ID'
numeric.data <- data[sapply(data, is.numeric)]
numeric.data$ID <- NULL

# Correlation between medianHouseValue and all other numeric variables
cor_medianHouseValue <- cor(data$medianHouseValue, numeric.data)
print(cor_medianHouseValue)
```

2.4.2.1 Choosing interaction terms from covariance matrix

```
##               id   longitude   latitude housingMedianAge aveRooms aveBedrooms
## [1,] 0.07198962 -0.04646254 -0.1439446        0.1052129 0.148625 -0.04545692
##       population medianIncome medianHouseValue bedroomsPerRoom incomePerRoom
## [1,] -0.02394206     0.689536                1      -0.2554662      0.665614
##        distToLA    distToSF cosDirToLA sinDirToLA cosDirToSF sinDirToSF
## [1,] -0.1309826 -0.03126119 -0.1547648 -0.1069453 -0.1746119  -0.251576
##      cityProximityScore distToCenter
## [1,]          0.2062844  -0.07148864
```

```r
# Correlation plot for all numeric variables
library(corrplot)
corrplot(cor(numeric.data), method = "circle",type =
         "upper")
```



Interaction choices for the model

Note: Some relate to covariance matrix others do not

- longitude:latitude
  - The variables longitude and latitude are not strongly correlated with medianHouseValue on their own but show strong negative correlation with each other, but together they define a unique spatial location. Their interaction enables the model to capture regional effects that are not purely east-west or north-south, but a combination of both.
  - The other reason why to include an interaction term is to reduce multi-collinearity between these two variables. This interaction represents the geographic positioning of each home, allowing the model to pick up spatial clusters in property value.
- bedroomsPerRoom:medianIncome
  - bedroomsPerRoom is negatively correlated with medianIncome. Pairing them allows the model to test whether the quality or crowding of space varies in importance across income levels. This interaction captures how the value of spaciousness or room density changes depending on neighbourhood affluence.
- medianIncome:housingMedianAge
  - medianIncome is strongly positively correlated with medianHouseValue, while housingMedianAge shows a weaker positive or near-zero correlation. The two variables are not strongly correlated

with each other, making them suitable for interaction. This interaction captures whether wealth is concentrated in newer or older neighbourhoods, revealing how the relationship between income and housing value shifts with the age of the housing stock.

- distToCenter:medianIncome
  - This interaction was chosen because it captures how the influence of income on housing value may vary with geographic centrality. High income in a central, urban location may signal access to premium markets, whereas high income in a peripheral location might reflect different lifestyle choices (e.g. rural affluence or luxury sprawl). This term allows the model to distinguish between urban wealth and suburban or rural wealth, identifying how location context changes the effect of income on house prices.
- latitude:incomePerRoom
  - latitude and incomePerRoom are not highly correlated with each other. Including the interaction helps model spatial economic patterns that are not captured by either variable alone. This interaction reflects how the value of economic density varies along the north–south axis of California.
- bedroomsPerRoom:distToLA
  - bedroomsPerRoom and distToLA are weakly correlated with each other and with medianHouseValue. Their interaction enables the model to explore whether the meaning of room layout or density shifts based on urban proximity. This expresses how space efficiency is valued differently depending on distance from Los Angeles.
- cityProximityScore:housingMedianAge
  - cityProximityScore is positively correlated with medianHouseValue, and weakly so with housingMedianAge. Their interaction tests whether proximity to major cities boosts or dampens the value of older housing stock. This models whether older homes near urban centres are more desirable or more heavily discounted.
- cityProximityScore:bedroomsPerRoom
  - These variables are weakly correlated with each other and allow the model to investigate if spatial efficiency (bedroom density) is more or less valuable near high-demand areas. This reflects how crowding or layout impacts price differently in urban versus more distant areas.
- housingMedianAge:incomePerRoom
  - These two have low correlation but both relate to housing quality and socioeconomic context. Their interaction allows for the effect of economic density to vary depending on housing stock age. This models how the combination of compact affluence and housing maturity affects property value.
- The four interaction terms between medianIncome and directional components (cosDirToLA, sinDirToLA, cosDirToSF, sinDirToSF) were included to model how the effect of income on housing value changes depending on where a home is situated relative to Los Angeles and San Francisco. Income is the strongest individual predictor of housing value, but its impact is not spatially uniform. These directional interactions allow the model to account for how the purchasing power and market influence of income varies depending on urban proximity, regional economies, and land use patterns. In essence, they help capture whether income drives prices more strongly in certain directions, reflecting localised economic geographies and how spatial context modifies the effect of wealth.

2.4.2.2 Choosing non-linear terms

- log(medianIncome / distToCenter)
  - Dividing income by distance to center captures the economic value of location, higher income closer to the center is associated with more valuable housing. This term measures the "effective income" adjusted for geographic desirability. Log scaling reduces the impact of extreme values and helps linearise the relationship in the context of multi-linear regression. It ensures the model captures diminishing returns, the impact of a change in income or distance is smaller at higher levels.
- I(medianIncome / housingMedianAge)
  - Combining income with housing age highlights areas where economic resources are mismatched with infrastructure age. High income in areas with older housing can indicate redevelopment potential, while low income and old housing may correlate with lower prices.
- log(incomePerRoom) where incomePerRoom = medianIncome / aveRooms
  - Dividing income by average rooms captures income per unit of housing capacity. This reflects how wealth is distributed relative to housing size and can act as a proxy for crowding or luxury. Logging controls for skewed distributions and allows the model to interpret multiplicative effects additively. It helps the model distinguish between different housing market conditions more effectively, especially at the extremes of income or room size.
- All three terms are related to medianIncome because: medianIncome is the strongest individual predictor of house prices in the California Housing dataset.It has the highest correlation with medianHousevalue. Housing prices are highly influenced by local purchasing power, and income reflects the ability of residents to pay for housing. Modifying and combining medianIncome with other variables can reveal more complex and informative relationships that are not captured by the raw variable alone.
- See **1.2.6** for the justification of other non-linear terms used in the final model (i.e. log and polynomial transformations).

2.4.3 Best subset selection with new predictors and interaction terms

New function - Must run to run next chunk. Creates method for regsubsets in predict().

```
predict.regsubsets=function(object,newdata,id,...){
  #... allows for other arguments to be passed into the function
  form=as.formula(object$call[[2]])
  mat=model.matrix(form,newdata)
  coefi=coef(object,id=id)
  xvars=names(coefi)
  mat[,xvars]%*%coefi
}
```

Checking whether all of the chosen predictors are worth adding. Do they result in the **best** model?

```
library(car)
# k-fold CV

formula_string <- "  medianHouseValue ~
```

```r
    . - id - oceanProximity - aveBedrooms - aveRooms - population +


    longitude:latitude +
    bedroomsPerRoom:medianIncome +
    medianIncome:housingMedianAge +
    distToCenter:medianIncome +
    latitude:incomePerRoom +
    bedroomsPerRoom:distToLA +
    medianIncome:log(population) +
    medianIncome:cosDirToLA +
    medianIncome:sinDirToLA +
    medianIncome:cosDirToSF +
    medianIncome:sinDirToSF +
    cityProximityScore:housingMedianAge +
    cityProximityScore:bedroomsPerRoom +
    housingMedianAge:incomePerRoom +



    I(medianIncome^2) +
    I(medianIncome^3) +
    I(housingMedianAge^2) +
    I(medianIncome / housingMedianAge) +
    I(latitude^2) +
    I(longitude^2) +
    I(latitude^3) +
    I(longitude^3) +


    log(population) +
    log(incomePerRoom) +
    log(medianIncome / distToCenter)"


lm.fit <- lm(formula = as.formula(formula_string),data = data)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = as.formula(formula_string), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -408775  -38162   -8192   26297  444997
##
## Coefficients:
##                                        Estimate     Std. Error t value
## (Intercept)                        -365958026.525  190786730.783  -1.918
## longitude                            -9304695.009    4688340.863  -1.985
## latitude                              1171910.406     595662.514   1.967
## housingMedianAge                        -2328.347        246.687  -9.438
## medianIncome                           -14082.986       6798.503  -2.071
```

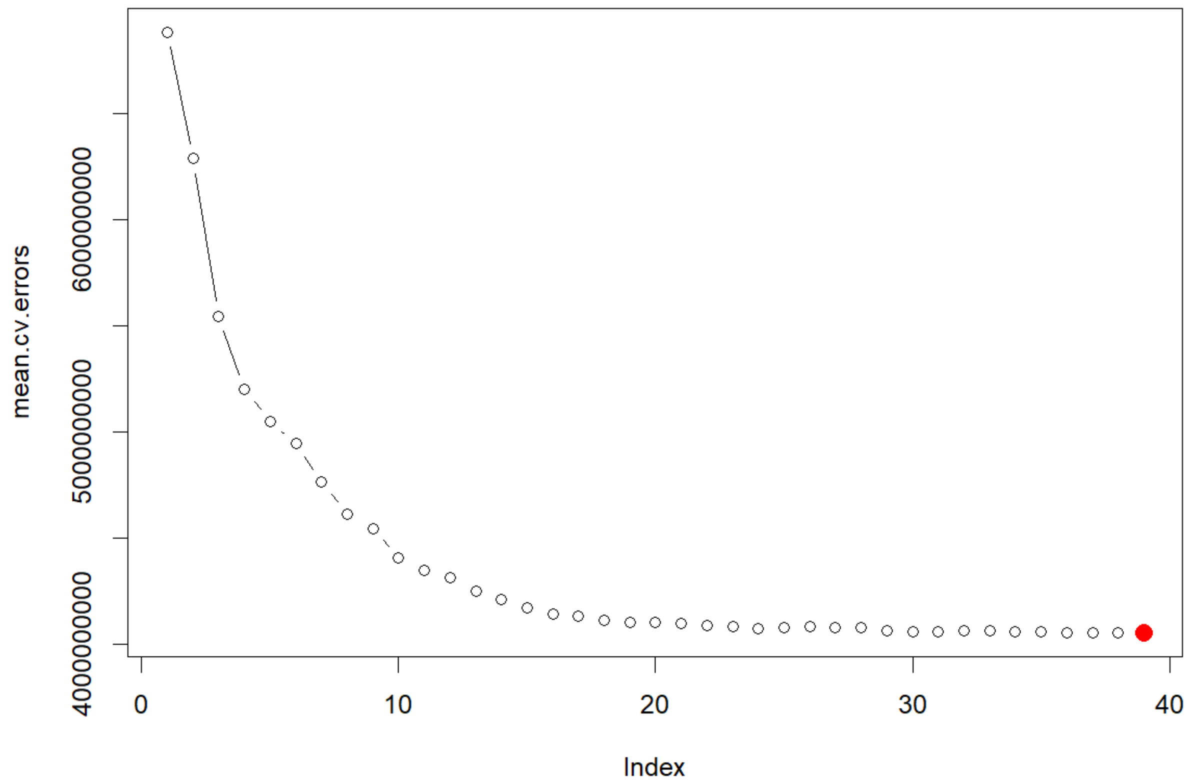
```
## bedroomsPerRoom                         137551.352       32167.239   4.276
## incomePerRoom                          -343090.014       62679.259  -5.474
## distToLA                                  -310.932          33.858  -9.184
## distToSF                                  -467.020          35.233 -13.255
## cosDirToLA                              -19265.312        2402.240  -8.020
## sinDirToLA                               53824.936        2908.701  18.505
## cosDirToSF                              -64289.055        4695.444 -13.692
## sinDirToSF                              -11800.223        4073.349  -2.897
## cityProximityScore                      -70396.767       55366.333  -1.271
## distToCenter                             12262.237        8609.351   1.424
## I(medianIncome^2)                         7139.209         634.332  11.255
## I(medianIncome^3)                         -416.072          24.983 -16.654
## I(housingMedianAge^2)                       14.487           3.646   3.973
## I(medianIncome/housingMedianAge)         -5212.574        3161.921  -1.649
## I(latitude^2)                             9340.986       17325.572   0.539
## I(longitude^2)                          -79974.600       39010.420  -2.050
## I(latitude^3)                               70.823         148.808   0.476
## I(longitude^3)                            -244.235         108.532  -2.250
## log(population)                         -13148.920        1372.858  -9.578
## log(incomePerRoom)                       20511.023        5166.048   3.970
## log(medianIncome/distToCenter)          -39790.608        7790.252  -5.108
## longitude:latitude                       18099.176        1352.990  13.377
## medianIncome:bedroomsPerRoom            115893.385        9909.695  11.695
## housingMedianAge:medianIncome              164.677          37.389   4.404
## medianIncome:distToCenter                 1194.777         485.961   2.459
## latitude:incomePerRoom                    4897.396        1765.195   2.774
## bedroomsPerRoom:distToLA                  -149.341          47.965  -3.114
## medianIncome:log(population)              2400.154         302.941   7.923
## medianIncome:cosDirToLA                  -2340.453         490.125  -4.775
## medianIncome:sinDirToLA                  -8633.563         657.784 -13.125
## medianIncome:cosDirToSF                   4353.736         932.040   4.671
## medianIncome:sinDirToSF                  -1604.394         775.812  -2.068
## housingMedianAge:cityProximityScore       3729.316         835.767   4.462
## bedroomsPerRoom:cityProximityScore     -169499.900      101452.926  -1.671
## housingMedianAge:incomePerRoom            1497.168         258.598   5.790
##                                                   Pr(>|t|)
## (Intercept)                                        0.05511 .
## longitude                                          0.04720 *
## latitude                                           0.04915 *
## housingMedianAge                     < 0.0000000000000002 ***
## medianIncome                                       0.03833 *
## bedroomsPerRoom                        0.00001911266371441 ***
## incomePerRoom                          0.00000004463901963 ***
## distToLA                             < 0.0000000000000002 ***
## distToSF                             < 0.0000000000000002 ***
## cosDirToLA                             0.00000000000000112 ***
## sinDirToLA                           < 0.0000000000000002 ***
## cosDirToSF                           < 0.0000000000000002 ***
## sinDirToSF                                         0.00377 **
## cityProximityScore                                 0.20358
## distToCenter                                       0.15438
## I(medianIncome^2)                    < 0.0000000000000002 ***
## I(medianIncome^3)                    < 0.0000000000000002 ***
## I(housingMedianAge^2)                  0.00007124761918128 ***
```

```
## I(medianIncome/housingMedianAge)                  0.09926 .
## I(latitude^2)                                     0.58979
## I(longitude^2)                                    0.04037 *
## I(latitude^3)                                     0.63413
## I(longitude^3)                                    0.02444 *
## log(population)                      < 0.0000000000000002 ***
## log(incomePerRoom)                     0.00007203953579710 ***
## log(medianIncome/distToCenter)         0.00000032928291737 ***
## longitude:latitude                   < 0.0000000000000002 ***
## medianIncome:bedroomsPerRoom         < 0.0000000000000002 ***
## housingMedianAge:medianIncome          0.00001066545658764 ***
## medianIncome:distToCenter                         0.01396 *
## latitude:incomePerRoom                            0.00554 **
## bedroomsPerRoom:distToLA                          0.00185 **
## medianIncome:log(population)           0.00000000000000245 ***
## medianIncome:cosDirToLA                0.00000180896994856 ***
## medianIncome:sinDirToLA              < 0.0000000000000002 ***
## medianIncome:cosDirToSF                0.00000301578362543 ***
## medianIncome:sinDirToSF                           0.03865 *
## housingMedianAge:cityProximityScore    0.00000816237931750 ***
## bedroomsPerRoom:cityProximityScore                0.09479 .
## housingMedianAge:incomePerRoom         0.00000000717233811 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63430 on 18410 degrees of freedom
## Multiple R-squared:  0.7001, Adjusted R-squared:  0.6995
## F-statistic:  1102 on 39 and 18410 DF,  p-value: < 0.00000000000000022
```

```r
nvmax <- length(coef(lm.fit)) - 1


k=10
set.seed(3)
folds=sample(1:k,nrow(data),replace=TRUE)
cv.errors=matrix(NA,k,nvmax, dimnames=list(NULL, paste(1:nvmax))) # NA means no data, NULL means no row


for(j in 1:k){
  best.fit=regsubsets(x = as.formula(formula_string),data = data[folds!=j,],nvmax=nvmax)
  for(i in 1:nvmax){
    pred=predict(best.fit,data[folds==j,],id=i)
    cv.errors[j,i]=mean( (data$medianHouseValue[folds==j]-pred)^2)
  }
}
mean.cv.errors=apply(cv.errors,2,mean)
par(mfrow=c(1,1))
best.model.size <- as.numeric(names(which.min(mean.cv.errors)))
plot(mean.cv.errors,type='b')
points(best.model.size, mean.cv.errors[best.model.size], col = "red", pch = 19, cex = 1.5)
```




```
# Obtain the best subset model using the full data and CV selected id
reg.best=regsubsets(x = as.formula(formula_string),data = data, nvmax=nvmax)

best.predictors = names(coef(reg.best, best.model.size))[-1]  # remove intercept
# Create formula dynamically
formula.best = as.formula(paste("medianHouseValue ~", paste(best.predictors, collapse = " + ")))

# Fit the model using lm
model.best = lm(formula.best, data = data)
summary(model.best)
```

```
##
## Call:
## lm(formula = formula.best, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -408775  -38162   -8192   26297  444997
##
## Coefficients:
##                                         Estimate     Std. Error t value
## (Intercept)                       -365958026.525  190786730.783  -1.918
## longitude                           -9304695.009    4688340.863  -1.985
```

```
## latitude                              1171910.406    595662.514   1.967
## housingMedianAge                        -2328.347       246.687  -9.438
## medianIncome                           -14082.986      6798.503  -2.071
## bedroomsPerRoom                        137551.352     32167.239   4.276
## incomePerRoom                         -343090.014     62679.259  -5.474
## distToLA                                 -310.932        33.858  -9.184
## distToSF                                 -467.020        35.233 -13.255
## cosDirToLA                             -19265.312      2402.240  -8.020
## sinDirToLA                              53824.936      2908.701  18.505
## cosDirToSF                             -64289.055      4695.444 -13.692
## sinDirToSF                             -11800.223      4073.349  -2.897
## cityProximityScore                     -70396.767     55366.333  -1.271
## distToCenter                            12262.237      8609.351   1.424
## I(medianIncome^2)                        7139.209       634.332  11.255
## I(medianIncome^3)                        -416.072        24.983 -16.654
## I(housingMedianAge^2)                      14.487         3.646   3.973
## I(medianIncome/housingMedianAge)        -5212.574      3161.921  -1.649
## I(latitude^2)                            9340.986     17325.572   0.539
## I(longitude^2)                         -79974.600     39010.420  -2.050
## I(latitude^3)                              70.823       148.808   0.476
## I(longitude^3)                           -244.235       108.532  -2.250
## log(population)                        -13148.920      1372.858  -9.578
## log(incomePerRoom)                      20511.023      5166.048   3.970
## log(medianIncome/distToCenter)         -39790.608      7790.252  -5.108
## longitude:latitude                      18099.176      1352.990  13.377
## medianIncome:bedroomsPerRoom           115893.385      9909.695  11.695
## housingMedianAge:medianIncome             164.677        37.389   4.404
## medianIncome:distToCenter                1194.777       485.961   2.459
## latitude:incomePerRoom                   4897.396      1765.195   2.774
## bedroomsPerRoom:distToLA                 -149.341        47.965  -3.114
## medianIncome:log(population)             2400.154       302.941   7.923
## medianIncome:cosDirToLA                 -2340.453       490.125  -4.775
## medianIncome:sinDirToLA                 -8633.563       657.784 -13.125
## medianIncome:cosDirToSF                  4353.736       932.040   4.671
## medianIncome:sinDirToSF                 -1604.394       775.812  -2.068
## housingMedianAge:cityProximityScore      3729.316       835.767   4.462
## bedroomsPerRoom:cityProximityScore    -169499.900    101452.926  -1.671
## housingMedianAge:incomePerRoom           1497.168       258.598   5.790
##                                                   Pr(>|t|)
## (Intercept)                                        0.05511 .
## longitude                                          0.04720 *
## latitude                                           0.04915 *
## housingMedianAge                    < 0.0000000000000002 ***
## medianIncome                                       0.03833 *
## bedroomsPerRoom                       0.00001911266371441 ***
## incomePerRoom                         0.00000004463901963 ***
## distToLA                            < 0.0000000000000002 ***
## distToSF                            < 0.0000000000000002 ***
## cosDirToLA                            0.00000000000000112 ***
## sinDirToLA                          < 0.0000000000000002 ***
## cosDirToSF                          < 0.0000000000000002 ***
## sinDirToSF                                         0.00377 **
## cityProximityScore                                 0.20358
## distToCenter                                       0.15438
```

```
## I(medianIncome^2)                   < 0.0000000000000002 ***
## I(medianIncome^3)                   < 0.0000000000000002 ***
## I(housingMedianAge^2)                 0.00007124761918128 ***
## I(medianIncome/housingMedianAge)                0.09926 .
## I(latitude^2)                                   0.58979
## I(longitude^2)                                  0.04037 *
## I(latitude^3)                                   0.63413
## I(longitude^3)                                  0.02444 *
## log(population)                     < 0.0000000000000002 ***
## log(incomePerRoom)                    0.00007203953579710 ***
## log(medianIncome/distToCenter)        0.00000032928291737 ***
## longitude:latitude                  < 0.0000000000000002 ***
## medianIncome:bedroomsPerRoom        < 0.0000000000000002 ***
## housingMedianAge:medianIncome         0.00001066545658764 ***
## medianIncome:distToCenter                       0.01396 *
## latitude:incomePerRoom                          0.00554 **
## bedroomsPerRoom:distToLA                        0.00185 **
## medianIncome:log(population)          0.00000000000000245 ***
## medianIncome:cosDirToLA               0.00000180896994856 ***
## medianIncome:sinDirToLA             < 0.0000000000000002 ***
## medianIncome:cosDirToSF               0.00000301578362543 ***
## medianIncome:sinDirToSF                         0.03865 *
## housingMedianAge:cityProximityScore  0.00000816237931750 ***
## bedroomsPerRoom:cityProximityScore              0.09479 .
## housingMedianAge:incomePerRoom        0.00000000717233811 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63430 on 18410 degrees of freedom
## Multiple R-squared:  0.7001, Adjusted R-squared:  0.6995
## F-statistic:  1102 on 39 and 18410 DF,  p-value: < 0.00000000000000022
```

From best subset selection, all predictors provided above improve the model.

2.4.4 Adding in oceanProximity to model

Since oceanProximity is a factor variable, best selection cannot be performed on it. Either all factors relating to oceanProximity are included or none are. Given the p-value's for the factors relating to oceanProximity in the initial model (2.1) are significant, in particular "Inland", oceanProximity will be included in the model.

In additional to oceanProximity these two interaction terms have been added:

- oceanProximity:distToCenter
  - This term captures how the effect of distance to the geographic center varies by coastal category. For example, INLAND areas may become less valuable with distance from the center, while NEAR BAY areas may be valuable regardless of centrality.
  - This term allows the slope of distToCenter to change across oceanProximity groups. Instead of assuming a single, fixed effect of distance for all areas, the model can learn group-specific sensitivities to distance.
- oceanProximity:medianIncome

- Income may affect house prices differently across coastal regions. High income near the ocean could indicate luxury real estate, while the same income level inland may not produce the same price signal. This term allows the effect of income on price to vary by oceanProximity.
- The model estimates different coefficients for income in each region, capturing regional differences in how income translates into housing value.

```
# With oceanProximity
new.formula = update(formula.best, . ~ . + oceanProximity +
                        oceanProximity:distToCenter +oceanProximity:medianIncome)
model.best = lm(new.formula, data = data)

summary(model.best)
```

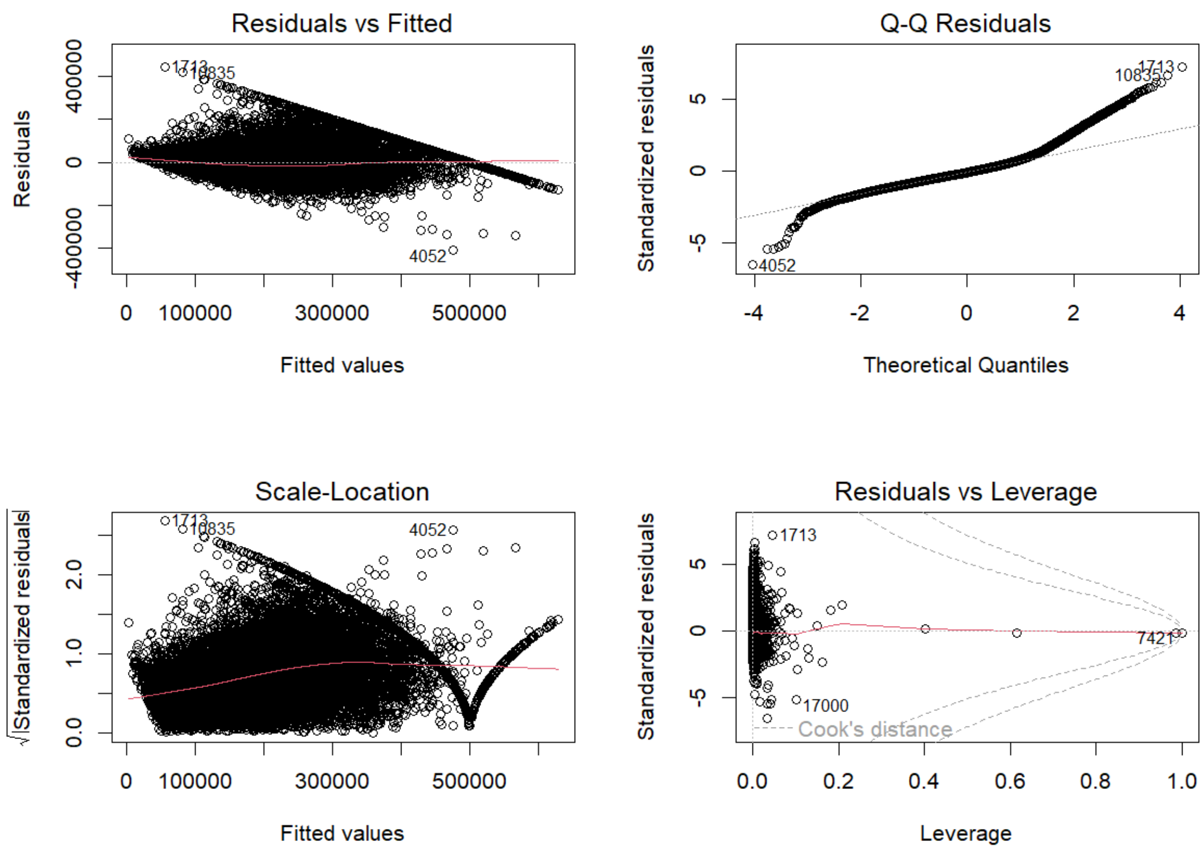
```
##
## Call:
## lm(formula = new.formula, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -408514  -37711   -8488   26111  444487
##
## Coefficients:
##                                          Estimate     Std. Error t value
## (Intercept)                        1221195367.532  308730841.015   3.956
## longitude                            31625764.372    7801341.024   4.054
## latitude                              2575546.922     674208.586   3.820
## housingMedianAge                        -2312.125        246.099  -9.395
## medianIncome                            -7452.574       7058.028  -1.056
## bedroomsPerRoom                        144592.331      32309.499   4.475
## incomePerRoom                         -343692.666      63133.380  -5.444
## distToLA                                 -361.874         36.713  -9.857
## distToSF                                 -324.034         45.098  -7.185
## cosDirToLA                             -23920.844       2528.925  -9.459
## sinDirToLA                              50782.338       3048.435  16.658
## cosDirToSF                             -68159.116       4853.541 -14.043
## sinDirToSF                             -11680.409       4617.229  -2.530
## cityProximityScore                     -44241.047      55916.370  -0.791
## distToCenter                            42674.364       9635.788   4.429
## I(medianIncome^2)                        6598.428        647.080  10.197
## I(medianIncome^3)                        -394.358         25.434 -15.505
## I(housingMedianAge^2)                      14.117          3.641   3.877
## I(medianIncome/housingMedianAge)        -5921.504       3153.363  -1.878
## I(latitude^2)                          -38946.219      19661.659  -1.981
## I(longitude^2)                         267182.061      65655.990   4.069
## I(latitude^3)                             444.931        168.191   2.645
## I(longitude^3)                            741.101        184.529   4.016
## log(population)                        -13019.223       1369.370  -9.507
## log(incomePerRoom)                      19042.460       5171.140   3.682
## log(medianIncome/distToCenter)         -36045.347       7886.918  -4.570
## oceanProximityINLAND                    41610.139       8483.764   4.905
## oceanProximityISLAND                 -2416681.302    1320271.246  -1.830
## oceanProximityNEAR BAY                  -6038.556      35011.840  -0.172
## oceanProximityNEAR OCEAN                18980.973       6702.481   2.832
```

```
## longitude:latitude                       12863.459      1616.064   7.960
## medianIncome:bedroomsPerRoom            111176.839      9887.108  11.245
## housingMedianAge:medianIncome              122.941        37.936   3.241
## medianIncome:distToCenter                  771.541       526.126   1.466
## latitude:incomePerRoom                    4836.782      1775.712   2.724
## bedroomsPerRoom:distToLA                  -120.748        48.247  -2.503
## medianIncome:log(population)              2290.172       302.512   7.571
## medianIncome:cosDirToLA                  -1553.312       536.264  -2.897
## medianIncome:sinDirToLA                  -7518.266       705.836 -10.652
## medianIncome:cosDirToSF                   5639.558       963.433   5.854
## medianIncome:sinDirToSF                  -1121.863       891.543  -1.258
## housingMedianAge:cityProximityScore       3904.937       834.823   4.678
## bedroomsPerRoom:cityProximityScore     -198869.663    102624.752  -1.938
## housingMedianAge:incomePerRoom            1614.170       258.626   6.241
## distToCenter:oceanProximityINLAND       -12188.084      2369.785  -5.143
## distToCenter:oceanProximityISLAND      1327063.411    589026.767   2.253
## distToCenter:oceanProximityNEAR BAY       1615.530      9939.055   0.163
## distToCenter:oceanProximityNEAR OCEAN       60.914      1736.678   0.035
## medianIncome:oceanProximityINLAND        -4167.721      1007.566  -4.136
## medianIncome:oceanProximityISLAND      -300389.435    132274.194  -2.271
## medianIncome:oceanProximityNEAR BAY       1815.726      1078.701   1.683
## medianIncome:oceanProximityNEAR OCEAN     -143.401       845.553  -0.170
##                                                   Pr(>|t|)
## (Intercept)                            0.00007665005233137 ***
## longitude                              0.00005058050980433 ***
## latitude                                          0.000134 ***
## housingMedianAge                     < 0.0000000000000002 ***
## medianIncome                                      0.291028
## bedroomsPerRoom                        0.00000767882042247 ***
## incomePerRoom                          0.00000005278953201 ***
## distToLA                             < 0.0000000000000002 ***
## distToSF                               0.00000000000069740 ***
## cosDirToLA                           < 0.0000000000000002 ***
## sinDirToLA                           < 0.0000000000000002 ***
## cosDirToSF                           < 0.0000000000000002 ***
## sinDirToSF                                        0.011423 *
## cityProximityScore                                0.428837
## distToCenter                           0.00000953331448664 ***
## I(medianIncome^2)                    < 0.0000000000000002 ***
## I(medianIncome^3)                    < 0.0000000000000002 ***
## I(housingMedianAge^2)                             0.000106 ***
## I(medianIncome/housingMedianAge)                  0.060419 .
## I(latitude^2)                                     0.047626 *
## I(longitude^2)                         0.00004732621765700 ***
## I(latitude^3)                                     0.008167 **
## I(longitude^3)                         0.00005938170691358 ***
## log(population)                      < 0.0000000000000002 ***
## log(incomePerRoom)                                0.000232 ***
## log(medianIncome/distToCenter)         0.00000490263236826 ***
## oceanProximityINLAND                   0.00000094379180381 ***
## oceanProximityISLAND                              0.067200 .
## oceanProximityNEAR BAY                            0.863069
## oceanProximityNEAR OCEAN                          0.004632 **
## longitude:latitude                     0.00000000000000182 ***
```

```
## medianIncome:bedroomsPerRoom          < 0.0000000000000002 ***
## housingMedianAge:medianIncome                    0.001194 **
## medianIncome:distToCenter                        0.142541
## latitude:incomePerRoom                           0.006459 **
## bedroomsPerRoom:distToLA                         0.012333 *
## medianIncome:log(population)           0.00000000000003893 ***
## medianIncome:cosDirToLA                          0.003777 **
## medianIncome:sinDirToLA               < 0.0000000000000002 ***
## medianIncome:cosDirToSF                0.00000000489202233 ***
## medianIncome:sinDirToSF                          0.208285
## housingMedianAge:cityProximityScore    0.00000292365690275 ***
## bedroomsPerRoom:cityProximityScore               0.052659 .
## housingMedianAge:incomePerRoom         0.00000000044335998 ***
## distToCenter:oceanProximityINLAND      0.00000027298631852 ***
## distToCenter:oceanProximityISLAND                0.024272 *
## distToCenter:oceanProximityNEAR BAY              0.870880
## distToCenter:oceanProximityNEAR OCEAN            0.972020
## medianIncome:oceanProximityINLAND      0.00003543258237127 ***
## medianIncome:oceanProximityISLAND                0.023161 *
## medianIncome:oceanProximityNEAR BAY              0.092343 .
## medianIncome:oceanProximityNEAR OCEAN            0.865332
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63160 on 18398 degrees of freedom
## Multiple R-squared:  0.7029, Adjusted R-squared:  0.7021
## F-statistic: 853.4 on 51 and 18398 DF,  p-value: < 0.00000000000000022
```

```
par(mfrow = c(2,2))
plot(model.best)
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```




```r
# High leverage points relate to ISLAND
data[data$oceanProximity == "ISLAND",]
```

```
##        id longitude latitude housingMedianAge aveRooms aveBedrooms population
## 7418 8316   -118.33    33.34               52 5.473318    1.371230       1100
## 7419 8317   -118.32    33.33               52 7.385417    1.777778        733
## 7420 8318   -118.32    33.34               52 6.225000    1.650000        341
## 7421 8319   -118.48    33.43               29 4.138728    1.236994        422
##      medianIncome medianHouseValue oceanProximity bedroomsPerRoom incomePerRoom
## 7418       2.8333           414700         ISLAND       0.2505299     0.5176568
## 7419       3.3906           300000         ISLAND       0.2407146     0.4590939
## 7420       2.7361           450000         ISLAND       0.2650602     0.4395341
## 7421       2.6042           287500         ISLAND       0.2988827     0.6292271
##      distToLA distToSF cosDirToLA sinDirToLA cosDirToSF sinDirToSF
## 7418 79.19181 615.5536 -0.1257543 -0.9920614  0.6783490 -0.7347399
## 7419 80.20382 616.9979 -0.1104315 -0.9938837  0.6784175 -0.7346766
## 7420 79.09950 616.0994 -0.1119675 -0.9937119  0.6792428 -0.7339136
## 7421 72.27561 599.3084 -0.3609941 -0.9325681  0.6721632 -0.7404031
##      cityProximityScore distToCenter
## 7418         0.01409202     2.605027
## 7419         0.01393282     2.618585
## 7420         0.01410496     2.609803
## 7421         0.01531292     2.456086
```

From the output it is clear that the model has improved since the reintroduction of oceanProximity and its respective interaction terms. The Adjusted R-squared has increased from 0.6995 to 0.7021.

From the Residuals vs Fitted plot it is clear that the fit has improved significantly since the original model and the Scale Location plot displays less heteroscedasticity than the previous model. However one cause for concern is the significant leverage (for two rows leverage is 1). However from further investigation it is clear that the the highly leveraged points relate to the four rows where oceanProximity is ISLAND. If they are removed ISLAND can't be used a predictor and hence neither than oceanProximity. This high leverage for a minority of houses is where oceanProximity is ISLAND is a trade-off of the final model. This should not be a big issue for the final test MSE because the predictions will be left and right censored.

2.5 Most significant predictors

```r
# Using best selection
regfit.full=regsubsets(new.formula,data = data,nvmax = 3,really.big = TRUE)
reg.full.summary <- summary(regfit.full)

# Show which variables are included in the best model of size 3
which(reg.full.summary$which[3, ])[-1]  # '3' refers to 3-variable model and [-1] removes intercept
```

```
##                         medianIncome                oceanProximityINLAND
##                                    5                                  27
## housingMedianAge:cityProximityScore
##                                   42
```

```r
summary(model.best)
```

```
##
## Call:
## lm(formula = new.formula, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -408514  -37711   -8488   26111  444487
##
## Coefficients:
##                                          Estimate     Std. Error t value
## (Intercept)                          1221195367.532  308730841.015   3.956
## longitude                              31625764.372    7801341.024   4.054
## latitude                                2575546.922     674208.586   3.820
## housingMedianAge                          -2312.125        246.099  -9.395
## medianIncome                              -7452.574       7058.028  -1.056
## bedroomsPerRoom                          144592.331      32309.499   4.475
## incomePerRoom                           -343692.666      63133.380  -5.444
## distToLA                                   -361.874         36.713  -9.857
## distToSF                                   -324.034         45.098  -7.185
## cosDirToLA                               -23920.844       2528.925  -9.459
## sinDirToLA                                50782.338       3048.435  16.658
## cosDirToSF                               -68159.116       4853.541 -14.043
## sinDirToSF                               -11680.409       4617.229  -2.530
## cityProximityScore                       -44241.047      55916.370  -0.791
## distToCenter                              42674.364       9635.788   4.429
```

```
## I(medianIncome^2)                          6598.428         647.080  10.197
## I(medianIncome^3)                          -394.358          25.434 -15.505
## I(housingMedianAge^2)                        14.117           3.641   3.877
## I(medianIncome/housingMedianAge)          -5921.504        3153.363  -1.878
## I(latitude^2)                            -38946.219       19661.659  -1.981
## I(longitude^2)                           267182.061       65655.990   4.069
## I(latitude^3)                               444.931         168.191   2.645
## I(longitude^3)                              741.101         184.529   4.016
## log(population)                          -13019.223        1369.370  -9.507
## log(incomePerRoom)                        19042.460        5171.140   3.682
## log(medianIncome/distToCenter)           -36045.347        7886.918  -4.570
## oceanProximityINLAND                      41610.139        8483.764   4.905
## oceanProximityISLAND                   -2416681.302     1320271.246  -1.830
## oceanProximityNEAR BAY                    -6038.556       35011.840  -0.172
## oceanProximityNEAR OCEAN                  18980.973        6702.481   2.832
## longitude:latitude                        12863.459        1616.064   7.960
## medianIncome:bedroomsPerRoom             111176.839        9887.108  11.245
## housingMedianAge:medianIncome               122.941          37.936   3.241
## medianIncome:distToCenter                   771.541         526.126   1.466
## latitude:incomePerRoom                     4836.782        1775.712   2.724
## bedroomsPerRoom:distToLA                   -120.748          48.247  -2.503
## medianIncome:log(population)               2290.172         302.512   7.571
## medianIncome:cosDirToLA                   -1553.312         536.264  -2.897
## medianIncome:sinDirToLA                   -7518.266         705.836 -10.652
## medianIncome:cosDirToSF                    5639.558         963.433   5.854
## medianIncome:sinDirToSF                   -1121.863         891.543  -1.258
## housingMedianAge:cityProximityScore        3904.937         834.823   4.678
## bedroomsPerRoom:cityProximityScore      -198869.663      102624.752  -1.938
## housingMedianAge:incomePerRoom             1614.170         258.626   6.241
## distToCenter:oceanProximityINLAND        -12188.084        2369.785  -5.143
## distToCenter:oceanProximityISLAND       1327063.411      589026.767   2.253
## distToCenter:oceanProximityNEAR BAY        1615.530        9939.055   0.163
## distToCenter:oceanProximityNEAR OCEAN        60.914        1736.678   0.035
## medianIncome:oceanProximityINLAND         -4167.721        1007.566  -4.136
## medianIncome:oceanProximityISLAND       -300389.435      132274.194  -2.271
## medianIncome:oceanProximityNEAR BAY        1815.726        1078.701   1.683
## medianIncome:oceanProximityNEAR OCEAN      -143.401         845.553  -0.170
##                                                     Pr(>|t|)
## (Intercept)                            0.00007665005233137 ***
## longitude                              0.00005058050980433 ***
## latitude                                          0.000134 ***
## housingMedianAge                     < 0.0000000000000002 ***
## medianIncome                                      0.291028
## bedroomsPerRoom                        0.00000767882042247 ***
## incomePerRoom                          0.00000005278953201 ***
## distToLA                             < 0.0000000000000002 ***
## distToSF                               0.00000000000069740 ***
## cosDirToLA                           < 0.0000000000000002 ***
## sinDirToLA                           < 0.0000000000000002 ***
## cosDirToSF                           < 0.0000000000000002 ***
## sinDirToSF                                        0.011423 *
## cityProximityScore                                0.428837
## distToCenter                           0.00000953331448664 ***
## I(medianIncome^2)                    < 0.0000000000000002 ***
```

```
## I(medianIncome^3)                     < 0.0000000000000002 ***
## I(housingMedianAge^2)                             0.000106 ***
## I(medianIncome/housingMedianAge)                  0.060419 .
## I(latitude^2)                                     0.047626 *
## I(longitude^2)                         0.00004732621765700 ***
## I(latitude^3)                                     0.008167 **
## I(longitude^3)                         0.00005938170691358 ***
## log(population)                       < 0.0000000000000002 ***
## log(incomePerRoom)                                0.000232 ***
## log(medianIncome/distToCenter)         0.00000490263236826 ***
## oceanProximityINLAND                   0.00000094379180381 ***
## oceanProximityISLAND                              0.067200 .
## oceanProximityNEAR BAY                            0.863069
## oceanProximityNEAR OCEAN                          0.004632 **
## longitude:latitude                     0.00000000000000182 ***
## medianIncome:bedroomsPerRoom          < 0.0000000000000002 ***
## housingMedianAge:medianIncome                     0.001194 **
## medianIncome:distToCenter                         0.142541
## latitude:incomePerRoom                            0.006459 **
## bedroomsPerRoom:distToLA                          0.012333 *
## medianIncome:log(population)           0.00000000000003893 ***
## medianIncome:cosDirToLA                           0.003777 **
## medianIncome:sinDirToLA               < 0.0000000000000002 ***
## medianIncome:cosDirToSF                0.00000000489202233 ***
## medianIncome:sinDirToSF                           0.208285
## housingMedianAge:cityProximityScore    0.00000292365690275 ***
## bedroomsPerRoom:cityProximityScore                0.052659 .
## housingMedianAge:incomePerRoom         0.00000000044335998 ***
## distToCenter:oceanProximityINLAND      0.00000027298631852 ***
## distToCenter:oceanProximityISLAND                 0.024272 *
## distToCenter:oceanProximityNEAR BAY               0.870880
## distToCenter:oceanProximityNEAR OCEAN             0.972020
## medianIncome:oceanProximityINLAND      0.00003543258237127 ***
## medianIncome:oceanProximityISLAND                 0.023161 *
## medianIncome:oceanProximityNEAR BAY               0.092343 .
## medianIncome:oceanProximityNEAR OCEAN             0.865332
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63160 on 18398 degrees of freedom
## Multiple R-squared:  0.7029, Adjusted R-squared:  0.7021
## F-statistic: 853.4 on 51 and 18398 DF,  p-value: < 0.00000000000000022
```

- Median Income
  - Median income is positively correlated with median house value, this holds true because as house buyers have a higher income they will have a greater tendency to purchase a house with a higher value to match what they can afford. Areas with higher income levels usually have residents with greater purchasing power. This translates into a higher demand for better and more expensive housing, which in turn drives up property values. The strong statistical significance of median income validates its role as an predictor of house values. Additionally, the p-value of this predictor in the model is essentially 0 (< 2e-16) signifying its statistical significance in the model at all levels of significance.
- Ocean Proximity - Inland

  - This variable reflects the classification of a households ocean proximity. The other classifications are closer to the ocean and generally, as a result, have higher house prices due to desirability of ocean views and proximity to recreational activities with the ocean. In contrast, "inland" ocean proximity does not typically benefit from these types of coastal premiums. This difference is what makes this variable particularly significant in our model when classifying houses and determining their house values. Additionally, the p-value of this predictor in the model is essentially 0 (9.44e-07), signifying its statistical significance in the model at all levels of significance.

- Interaction between Housing Median Age and City Proximity Score
  - City proximity score is positively correlated with median house value, and weakly so with housing median age. Their interaction tests whether proximity to major cities boosts or dampens the value of older housing stock. Their interaction tests whether proximity to major cities boosts or dampens the value of older housing stock. In other words, the interaction term enables us to explore whether the effect of a house's age on its value depends on how close the property is to a major city. Additionally, the p-value of this predictor in the model is essentially 0 (2.92e-06), signifying its statistical significance in the model at all levels of significance.

---

3 Assessing the model performance

3.1 Training MSE

```r
# Compute training MSE
# Initial model
# Need to reload data since new predictor columns have been added
initial_data <- read.csv("Assignt1_data.csv")
initial_data <- na.omit(initial_data)
initial_fit <- lm(medianHouseValue ~ . - id, data = initial_data)
initial_predictions <- predict(initial_fit, newdata = initial_data)
initial_train_mse <- mean((initial_data$medianHouseValue - initial_predictions)^2)
initial_train_mse
```

```
## [1] 5129801167
```

```r
# Final Model
final_predictions <- predict(model.best, newdata=data)
final_train_mse <- mean((data$medianHouseValue - final_predictions)^2)
final_train_mse
```

```
## [1] 3977659474
```

3.2 Validation set MSE 80-20 split

```r
# Set seed required for this approach to get the 80-20 split
set.seed(123)
```

```r
# Split data - 80% for training and 20% for validation
n <- nrow(data)
train_indicies <- sample(1:n, size=floor(0.8 * n))
train_set <- data[train_indicies, ]
validation_set <- data[-train_indicies, ]

train_set_init <- initial_data[train_indicies, ]
validation_set_init <- initial_data[-train_indicies, ]

# Initial model
initial_fit_train <- lm(medianHouseValue ~ . - id, data=train_set_init)
initial_val_predictions <- predict(initial_fit_train, newdata=validation_set_init)
initial_val_mse <- mean((validation_set$medianHouseValue - initial_val_predictions)^2)
initial_val_mse
```

```
## [1] 5211492438
```

```r
# Final model
final_fit_train <- lm(new.formula, data=train_set)
final_val_predictions <- predict(final_fit_train, newdata=validation_set)
final_val_mse <- mean((validation_set$medianHouseValue - final_val_predictions)^2)
final_val_mse
```

```
## [1] 3895087989
```

3.3 5 fold CV MSE

```r
set.seed(123)

# Define number of folds
k <- 5
n <- nrow(data)
folds <- sample(1:k, n, replace=TRUE)

# Initial model
# Vector to hold MSE for each fold
initial_cv_errors <- numeric(k)
final_cv_errors <- numeric(k)


for (i in 1:k) {

  train_indicies <- which(folds != i)
  test_indicies <- which(folds == i)

  train_data <- data[train_indicies, ]
  test_data <- data[test_indicies, ]

  train_data_init <- initial_data[train_indicies, ]
  test_data_init <- initial_data[test_indicies, ]
```

```r
  # Initial Model
  initial_cv_model <- lm(medianHouseValue ~ . - id, data=train_data_init)
  initial_predictions <- predict(initial_cv_model, newdata=test_data_init)
  initial_cv_errors[i] <- mean((test_data_init$medianHouseValue - initial_predictions)^2)

  # Final Model
  final_cv_model <- lm(new.formula, data=train_data)
  final_predictions <- predict(final_cv_model, newdata=test_data)
  final_cv_errors[i] <- mean((test_data$medianHouseValue - final_predictions)^2)

}

initial_cv_5fold_mse <- mean(initial_cv_errors)
initial_cv_5fold_mse
```

```
## [1] 5153080811
```

```r
final_cv_5fold_mse <- mean(final_cv_errors)
final_cv_5fold_mse
```

```
## [1] 4017021575
```

3.4 LOOCV MSE

```r
set.seed(123)

# Initial Model
train_control <- trainControl(method = "LOOCV")
initial_loocv_model <- train(
  medianHouseValue ~ . - id,
  data = initial_data,
  method = "lm",
  trControl = train_control
)

# Extract RMSE
initial_loocv_rmse <- initial_loocv_model$results$RMSE


# Final Model
train_control <- trainControl(method = "LOOCV")
final_loocv_model <- train(
  new.formula,
  data = data,
  method = "lm",
  trControl = train_control
)
final_loocv_rmse <- final_loocv_model$results$RMSE

# Calculate MSE from RMSE
```

```r
initial_loocv_mse <- initial_loocv_rmse^2
initial_loocv_mse
```

```
## [1] 5163153328
```

```r
final_loocv_mse <- final_loocv_rmse^2
final_loocv_mse
```

```
## [1] 4013601583
```

3.5 Comparison

```r
initial_fit <- lm(medianHouseValue ~ . - id, data = data)
final_fit   <- lm(model.best, data = data)

AIC_initial <- AIC(initial_fit)
AIC_final   <- AIC(final_fit)

BIC_initial <- BIC(initial_fit)
BIC_final   <- BIC(final_fit)

cat("Initial Model: AIC =", AIC_initial, ", BIC =", BIC_initial, "\n")
```

```
## Initial Model: AIC = 462483.2 , BIC = 462663.1
```

```r
cat("Final Model:   AIC =", AIC_final, ", BIC =", BIC_final, "\n")
```

```
## Final Model:   AIC = 460282.9 , BIC = 460697.5
```

The final MLR model is better than the initial MLR model. This is clearly evident by the train mean squared error and the test error rates, under all of the methods covered above, being lower for the final MLR model when compared to the initial MLR model.

A well-performing model must strike a balance between bias (error from assumptions in the model) and variance (error from sensitivity from the data set), and this bias-variance trade-off was an important considerations when determining which model was more effective than the other. The initial MLR model was too simple and failed to capture important relationships present in the data, whereas the final model-by including additional predictors, interaction terms, and by refining the variable selection we reduced the variance of the model without introducing excessive bias. This improved balance lowered our test error rates as calculated above.

The use of the validation techniques (80-20 split validation set approach, 5-fold CV, and LOOCV) provided more robust estimates of the test error than the training mse. The final model's lower error rates in each of the 3 tests mentioned demonstrates our final models predictive performance is pretty reliable. Additionally, the 2 cross validation methods used to derive the test error rates for our model, are known for providing a close to unbiased estimate of the test error which helped when comparing the models. By the final model showing that it consistently achieves a lower test error rates in all of the approaches mentioned it provides even stronger evidence to support the strong predictive accuracy of the final model over the initial model.

The final model also had a lower AIC (Akaike Information Criterion, 460282.9 vs. $462,438.2$) and a lower BIC (Bayesian Information Criterion, 460697.5 vs. $462,663.1$) compared to the initial model, indicating that the final model also achieves a better trade-off between fit and complexity. Although the difference may not seem very large it is significant to note that the final model is a lot more complex than the initial model but it still has a much better fit to the data, without introducing too much bias.

The final model was adapted to our knowledge of the data (such as ocean proximity and the city proximity score) to make sense from an economic and geographical perspective of the prediction task at hand. This supports the idea that the final model not only fits with the data better but it also aligns with theoretical expectations of how house values interact with these economic and geographic factors.

Finally, the final model was also created with use of the best subset selection method helps evaluates the most appropriate predictors for the model. This method evaluated which predictors add the most substantive explanatory power, as these predictors will be the most impactful to the predictive accuracy of our model. The results of this method of evaluating predictors also aligned with both statistical evidence and the knowledge of the data, as mentioned prior, further justifying the superiority of the predictors used in our final MLR over the initial MLR.

---

4 A Prediction Competition

4.1 Test MSE for the final (best) model

```r
test_data <- read.csv("Assignt1_test full.csv")
test_data$bedroomsPerRoom <- test_data $aveBedrooms / test_data $aveRooms
test_data$distToCenter <- sqrt((test_data$latitude - center_lat)^2 +
                                (test_data$longitude - center_lon)^2)
test_data$incomePerRoom = test_data$medianIncome / test_data$aveRoom
# Add distance to LA
test_data$distToLA <- apply(test_data[, c("longitude", "latitude")], 1, function(coord) {
  distGeo(coord, la_coords) / 1000  # convert meters to km
})

#Add distance to SF
test_data$distToSF <- apply(test_data[, c("longitude", "latitude")], 1, function(coord) {
  distGeo(coord, sf_coords) / 1000
})

# Compute direction angles
test_data$dirToLA <- atan2(test_data$latitude - la_coords[2],
                           test_data$longitude - la_coords[1])
test_data$dirToSF <- atan2(test_data$latitude - sf_coords[2],
                           test_data$longitude - sf_coords[1])


# Encode directions using sin and cos
test_data$cosDirToLA <- cos(test_data$dirToLA)
test_data$sinDirToLA <- sin(test_data$dirToLA)

test_data$cosDirToSF <- cos(test_data$dirToSF)
test_data$sinDirToSF <- sin(test_data$dirToSF)

# Remove intermediate angle variables
```

```r
test_data$dirToLA <- NULL
test_data$dirToSF <- NULL

test_data$cityProximityScore <- 1 / (1 + test_data$distToLA) + 1 / (1 + test_data$distToSF)
test_data$incomeFlag <- ifelse(test_data$medianIncome == 15.0001, 1, 0)

actual <- test_data$medianHouseValue

#Best model

pred <- predict(model.best, newdata = test_data  )
pred <- pmax(pmin(pred,500001),14999) # apply censoring
best_mse <- mean((pred - actual)^2)

cat("This is the mean squared error of our final MLR model for the competition \n",best_mse)
```

```
## This is the mean squared error of our final MLR model for the competition
##  3763439107
```

We loaded the test dataset and setup our final MLR model to predict the median housing prices. Firstly, we calculated our new predictors to the new data. Secondly, we used our final MLR model and predicted the median house values using the test data, from which we could easily calculate our mean squared error for the competition as shown below. We also applied censoring on the median housing prices to replicate the style of the dataset.