

Californian Median Housing Value Predictive Modeling Using Multiple Linear Regression

Introduction

This study examines the multiple factors on the median housing value of Californian properties, with a focus on the geographical and regional information, in order to understand and to make accurate prediction of median housing values.

Materials and Methods

Variable Selection

Applying the power transformation on the predictors and response first, we find their most appropriate transformations. Afterwards, comparison between models determines if the transformation improves the model fitness. Predictors in the fully transformed model, that don't pass the individual significance test, are also compared to models without the particular predictors.

The four selection criteria, adjusted R-squared, Akaike's Information Criterion(AIC), corrected AIC, and Bayesian Information Criterion(BIC), are used to compare model fitness. They measure how well the model fits the dataset, with increasing penalty on model complexity to balance the interpretability of a model with its predictability.

Variance Inflation Factor(VIF) and the correlation coefficients of variables are used to identify and solve multicollinearity issue. We build models with all possible combinations of highly correlated variables, starting with the ones that require the least number of variables to be taken out of the model. From these models that pass the multicollinearity test, we compare the model performance using selection criteria to find the best performing combination of variables.

To decide whether the main effect and interaction terms for the indicator variables, *near_bay*, *near_ocean*, and *oneh_ocean*, should be included in the model, we compare the simple linear models from different categories, specifically the slopes and intercepts.

In order to identify bad leverage points, we apply $4/n$ as a cut-off of the hat matrix to identify high leverage points, then use a cut off of $|\text{standardized residue}| \geq 4$ to identify the outliers among them, which are bad leverage points.

Finally, the individual significance is verified again with the final model. For not individually significant variable, models without the particular variable are compared to a full model using ANOVA test to determine if the variable should be included.

Model Validation

1000 observations are randomly selected from the dataset. It is split by 70/30; the training dataset has 700 observations, while the test dataset has 300 observations. The model is validated with the test dataset by using the same plots for model diagnostics to verify the additional conditions and assumptions.

Model Diagnostics

First, it is to check the two additional conditions on the linearity of the response against the fitted values and linearity between the predictors by appropriate plots. The four assumptions are checked using the residual plots, standardized residual plots, and modified residual plots. Failing the assumptions can greatly affect the credibility and predictability of our model.

Results

Data Description

The sample has 14 variables containing information about 1000 homes in California, randomly selected from a large dataset. From the histograms and Sample Locations plot, most observations are scattered near the San Francisco Bay area or in Los Angeles. The Sample Price plot demonstrates how much the locations of Californian properties determine the median housing prices, with the expensive properties mainly located near ocean, mostly around San Francisco or LA. From the histograms, we can see similar distribution between *total_rooms*, *total_bedrooms*, *populations*, and *households*.

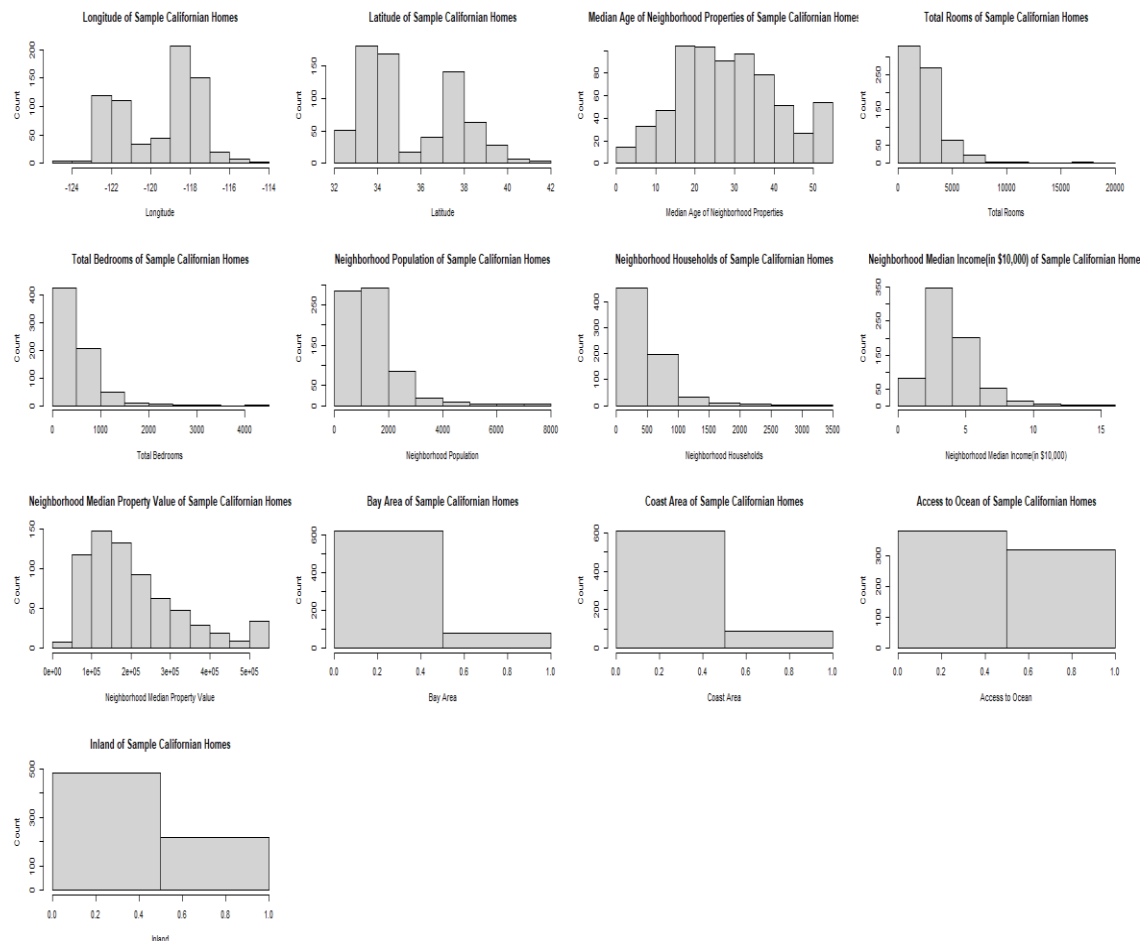


Figure 1 Variable Histograms

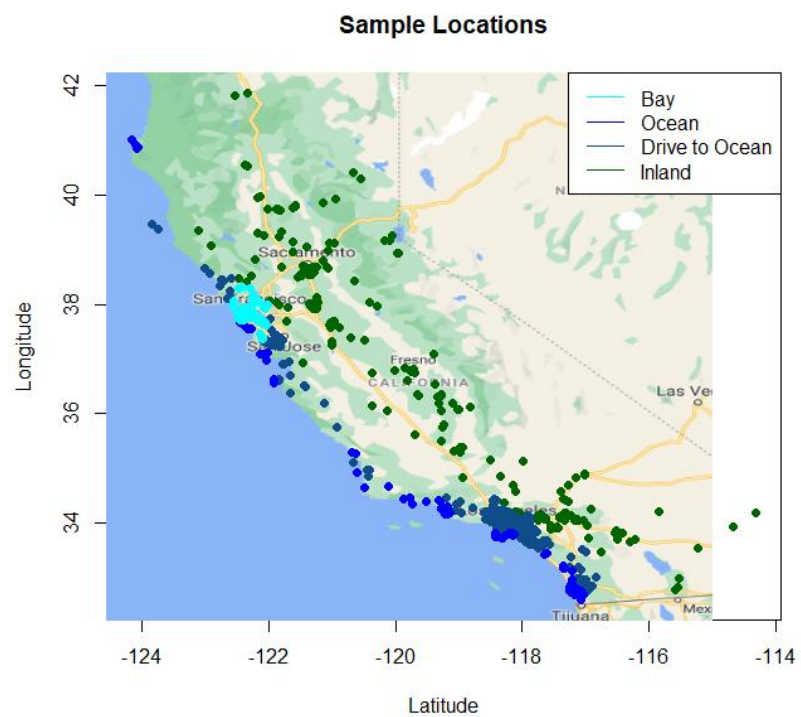


Figure 3 Sample Locations

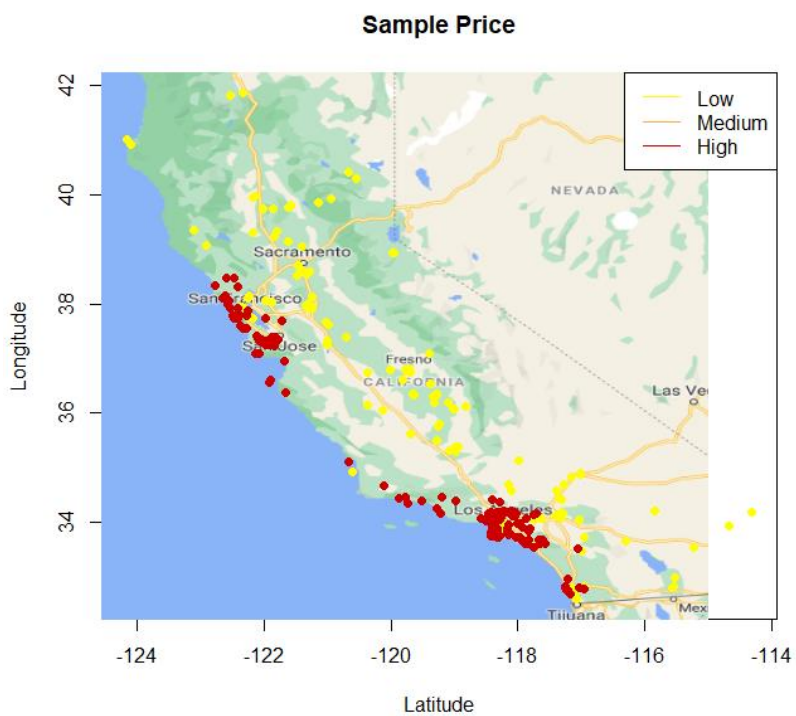


Figure 2 Sample Price

Process of Obtaining the Final Model

Variable	Suggested Power from powerTransform
<i>latitude</i>	-6.2103873
<i>longitude</i>	-5.1456697
<i>median_house_value</i>	0.147994
<i>housing_median_age</i>	0.7533555
<i>total_rooms</i>	0.1445191
<i>total_bedrooms</i>	0.1540388
<i>population</i>	0.1613027
<i>households</i>	0.1747819
<i>median_income</i>	0.161507
<i>near_bay</i>	-1.9938737
<i>near_ocean</i>	-1.7186720
<i>oneh_ocean</i>	-0.1156945

Table 1 Power Transformation Result

The results of power transformation is shown in Table 1. The powers of predictors, *latitude*, *longitude*, and *near_bay* are rounded off to -6, -5, -2 respectively. Because the other predictors and the response have small powers, logarithm transformation is applied instead for interpretability purpose. We add 0.01 to *near_bay*, *near_ocean*, and *oneh_ocean* since they contain zeros.

The results from Table 3 show that predictors *near_bay* and *near_ocean* are best untransformed, while *longitude* and *median_income* should be transformed. Predictor *households* is best to be dropped, because it doesn't pass the significance test.

Model with Highly Correlated Predictors	Adjusted R ²	AIC	Corrected AIC	BIC
<i>longitude, population</i>	0.489	-1204.186	-1204.099	-1177.430
<i>longitude, total_bedrooms</i>	0.473	-1182.664	-1182.577	-1155.908
<i>longitude, total_rooms</i>	0.458	-1162.899	-1162.813	-1136.144
<i>latitude, population</i>	0.473	-1182.425	-1182.339	-1155.669
<i>latitude, total_bedrooms</i>	0.458	-1163.364	-1163.278	-1136.608
<i>latitude, total_rooms</i>	0.447	-1149.563	-1149.477	-1122.808

Table 2 Selection Criteria for Models with Multicollinear Variables

Regarding multicollinearity, *inland* is removed due to perfect. One highly correlated group is *latitude* and *longitude*. The second group is *total_rooms*, *total_bedrooms*, *households*, and *population*. Keeping *longitude* and *total_rooms* results in the best non-collinear model from Table 2. Note that the criteria are from models with appropriately transformed variables; the transformation is not included in the table for simplicity purpose.

The analysis on main effect and interaction term from Table 4 indicates that it is best to keep these interaction terms: *longitude* and *near_bay*, *longitude* and *near_ocean*, and *longitude* and

oneh_ocean; it suggests to keep the main effect term *near_bay*. However, the final model doesn't perform better than the previous model, so the changes aren't implemented in the final model.

The filters on bad leverage points show one data point to be a bad leverage point. This point has the maximum *housing_median_age*, high *median_income*, and the maximum *median_house_value*. The property with this specific set of coordinates is situated in the same neighborhood with well known studios. Contextually, it is best to be evaluated separately as an individual case, so it is removed from the train dataset.

From analysis on model conditions and assumptions, *median_income* and *total_rooms* don't satisfy the constant variance assumption. The assumption violation is fixed by applying square root transformation to the response and the two predictors while removing their prior logarithm transformation.

While there are clusters of residuals, they are not separated from the rest of the data, so the errors are independent. From the modified residual plots, the variances of the errors are constant. The points in normal quantile-quantile plot indicates a straight linear relationship, with no noticeable deviation or other pattern; this tells us that the assumption of normally distributed error is satisfied. Therefore, all four assumptions are met for this model.

Goodness of Final Model

Upon inspection, the train and test datasets are fairly similar in distribution. Our final model performs better using the test dataset compared to using the train dataset, with an adjusted *r*-squared value of 0.667 instead of 0.5905. The model also passes the two additional conditions, upon evaluating the plot from the response against the fitted values and scatterplots of all the predictors. The standardized residual plots and residual plots indicate that the model using the test dataset satisfies the linearity, normality of errors, and independent error assumptions. The modified standard plot show some moderate non-constant variance issues.

Discussion

Final Model Interpretation and Importance

The final model shows that the longitude of the property and its general geographical position in the state of California, and the median age, total number of rooms, median income of the neighborhood housing are significant in relation to the median housing value in a given area. We identify the significant factors and build appropriate model that captures the relationships between the significant factors and median housing value and provides good predictive power.

Limitations of Analysis

The final model doesn't have the best selection criteria compared to some other models we use, for example, logarithm transformation for *median_housing_value*, *total_rooms*, and *median_income* fits the dataset better than square-root transformation. However, models with logarithm transformation don't satisfy the constant variance assumption. Likewise, including more predictors like *latitude* and *total_bedrooms* can improve the predictive power of the model but they cause multicollinearity issue.

References

Supplementary Tables

Model	Adjusted R ²	AIC	Corrected AIC	BIC
All Variables Transformed	0.660	-1481.141	-1480.612	-1417.977
Untransformed <i>near_bay</i>	0.660	-1483.025	-1482.572	-1424.412
Untransformed <i>households</i>	0.660	-1481.027	-1480.498	-1417.863
Untransformed <i>near_ocean</i>	0.660	-1483.025	-1482.572	-1424.412
<i>near_bay</i> dropped	0.659	-1480.828	-1480.445	-1426.766
<i>households</i> dropped	0.660	-1483.025	-1482.572	-1424.412
<i>near_ocean</i> dropped	0.657	-1476.410	-1476.028	-1422.348
Untransformed <i>longitude</i>	0.657	-1474.956	-1474.427	-1411.792
Untransformed <i>median_income</i>	0.665	-1492.173	-1491.644	-1429.009

Table 3 Selection Criteria for Models with Varying Transformations

Model	Adjusted R-squared	AIC	Corrected AIC	BIC
Main effect term for <i>near_bay</i> and interaction terms for <i>near_ocean</i> and <i>oneh_ocean</i>	0.592	-1357.08	-1356.826	-1312.127
Interaction terms for <i>near_bay</i> , <i>near_ocean</i> , and <i>oneh_ocean</i>	0.591	-1356.630	-1356.370	-1311.670
Interaction term for <i>near_ocean</i> , main effect terms for <i>near_bay</i> and <i>oneh_ocean</i>	0.593	-1359.790	-1359.530	-1314.830
Interaction term for <i>oneh_ocean</i> , main effect terms for <i>near_bay</i> and <i>near_ocean</i>	0.592	-1357.443	-1357.183	-1312.483
Interaction term for <i>near_bay</i> and main effect term for <i>near_ocean</i> and <i>oneh_ocean</i>	0.594	-1360.603	-1360.343	-1315.643
Main effect terms for <i>near_bay</i> , <i>near_ocean</i> , and <i>oneh_ocean</i>	0.594	-1361.105	-1360.845	-1316.145
Main effect term for <i>near_bay</i> and interaction terms for <i>near_bay</i> , <i>near_ocean</i> , and <i>oneh_ocean</i>	0.602	-1374.095	-1373.776	-1324.584

Table 4 Selection Criteria for Models with Varying Main Effect and Interaction Terms

R Code

```
1 setwd(dir = 'C:/Users/i5/Downloads/STA302 Data Analysis I/Final Project')
2 data <- read.csv('C:/Users/i5/Downloads/STA302 Data Analysis I/Final Project/housing.csv', header=TRUE)
3 set.seed(1004081030)
4 rows <- sample(1:nrow(data), 1000, replace=FALSE)
5 train <- data[rows[1:700],]
6 test <- data[rows[701:1000],]
7 pred <- c("Longitude", "Latitude", "Median Age of Neighborhood Properties",
8           "Total Rooms", "Total Bedrooms", "Neighborhood Population",
9           "Neighborhood Households", "Neighborhood Median Income(in $10,000)",
10          "Neighborhood Median Property Value", "Bay Area", "Coast Area",
11          "Access to ocean", "Inland")
12 library(car)
13 library(MASS)
14 library(leaps)
15 select_criteria = function(model, n)
16 {
17   SSres <- sum(model$residuals^2)
18   Rsq_adj <- summary(model)$adj.r.squared
19   p <- length(model$coefficients) - 1
20   AIC <- n*log(SSres/n) + 2*p
21   AICC <- AIC + (2*(p+2)*(p+3))/(n-p-1)
22   BIC <- n*log(SSres/n) + (p+2)*log(n)
23   res <- c(SSres, Rsq_adj, AIC, AICC, BIC)
24   names(res) <- c("SSres", "Rsq_adj", "AIC", "AIC_c", "BIC")
25   return(res)
26 }
27 # full model
28 mod_train <- lm(median_house_value ~ latitude + longitude + housing_median_age
29               + total_rooms + total_bedrooms + population + households +
30               median_income + near_bay + near_ocean + oneh_ocean + inland, data = train)
31 summary(mod_train) # shows total rooms, households, and near bay to be significant
32 anova(mod_train) # shows housing_median_age and near_bay to be significant
33
34 # delete inland given multicollinearity issue
35 mdl1 <- lm(median_house_value ~ latitude + longitude + housing_median_age
36           + total_rooms + total_bedrooms + population + households +
37           median_income + near_bay + near_ocean + oneh_ocean, data = train)
38 summary(mdl1)
39 anova(mdl1) # suggest latitude, longitude, total rooms, total bedrooms, population,
40 # households, median income, next near_ocean, and oneh_ocean; w/o median_age and near_bay
41
42 #####
43 # model with transformed values to run individual significance test #
44 #####
45 # check with power transformation
46 powerTransform(lm(train$median_house_value ~ 1)) # 0.147994
47 powerTransform(lm(cbind(train$median_house_value, train$housing_median_age) ~ 1)) # suggest 0.7533555
48 powerTransform(lm(cbind(train$median_house_value, train$total_rooms) ~ 1)) # suggest power 0.1445191
49 powerTransform(lm(cbind(train$median_house_value, train$total_bedrooms) ~ 1)) # suggest power 0.1540388
50 powerTransform(lm(cbind(train$median_house_value, train$population) ~ 1)) # suggest power 0.1613027
51 powerTransform(lm(cbind(train$median_house_value, train$population) ~ 1)) # suggest power 0.1613027
52 powerTransform(lm(cbind(train$median_house_value, train$households) ~ 1)) # suggest power 0.1747819
53 powerTransform(lm(cbind(train$median_house_value, train$median_income) ~ 1)) # suggest power 0.161507
54
55 # the other five variables are not strictly positive, so must make adjustment
56 powerTransform(lm(cbind(train$median_house_value, train$latitude) ~ 1)) # suggest power -6.2103873
57 powerTransform(lm(cbind(train$median_house_value, I(train$near_bay + 0.01)) ~ 1)) # suggest power -1.9938737
58 powerTransform(lm(cbind(train$median_house_value, I(train$near_ocean + 0.01)) ~ 1)) # suggest power -1.7186720
59 powerTransform(lm(cbind(train$median_house_value, I(train$oneh_ocean + 0.01)) ~ 1)) # suggest power -0.1156945
60 powerTransform(lm(cbind(train$median_house_value, abs(train$longitude)) ~ 1)) # suggest power -5.1456697
61
62 # all transformed
63 mdl2 <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
64           I(log(housing_median_age)) + I(log(total_rooms)) +
65           I(log(total_bedrooms)) + I(log(population)) + I(log(households)) +
66           I(log(median_income)) + I((near_bay + 0.01)^(-2)) +
67           I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
68           data = train)
69 anova(mdl2) # latitude, total_rooms, median_age, total_rooms and oneh_ocean's individual
70 # significance has improved; households, near_bay near_ocean are worse;
71 # households, near_bay, near_ocean are not significant
72 # same significance: longitude, total_bedroom, population, median_income
73
74 # near_bay is insignificant with or without transformation; check if drop near_bay
75 mdl2a <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) + I(log(housing_median_age))
76           + I(log(total_rooms)) + I(log(total_bedrooms)) + I(log(population)) + I(log(households)) +
77           I(log(median_income)) + I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)), data = train)
78 anova(mdl2a, mdl2) # shouldn't drop near_bay
79
80 # decide if households should be transformed
81 # transformed near_bay, transformed near_ocean, original households
```



```

81 md12d <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
82             I(log(housing_median_age)) + I(log(total_rooms)) +
83             I(log(total_bedrooms)) + I(log(population)) + households +
84             I(log(median_income)) + I((near_bay + 0.01)^(-2)) +
85             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
86             data = train)
87 anova(md12d) # household significance 0.367657 compared to 0.066101 as transformed;
88 # keep households as transformed
89
90 # compare transformed households vs dropping households
91 md12e <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
92             I(log(housing_median_age)) + I(log(total_rooms)) +
93             I(log(total_bedrooms)) + I(log(population)) +
94             I(log(median_income)) + I((near_bay + 0.01)^(-2)) +
95             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
96             data = train)
97 anova(md12e, md12) # value 0.7357; should drop
98
99 # decide if near_bay should be transformed
100 md12f <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
101             I(log(housing_median_age)) + I(log(total_rooms)) +
102             I(log(total_bedrooms)) + I(log(population)) +
103             I(log(median_income)) + near_bay +
104             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
105             data = train)
106 anova(md12f) # 0.296095
107 md12g <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
108             I(log(housing_median_age)) + I(log(total_rooms)) +
109             I(log(total_bedrooms)) + I(log(population)) +
110             I(log(median_income)) + I((near_bay + 0.01)^(-2)) +
111             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
112             data = train)
113 anova(md12g) # 0.296095 as transformed
114 # since it doesn't make a difference, will use original for model simplicity
115
116 md12h <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
117             I(log(housing_median_age)) + I(log(total_rooms)) +
118             I(log(total_bedrooms)) + I(log(population)) +
119             I(log(median_income)) +
120             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
121             data = train) # near_bay dropped
122 anova(md12h, md12f) # 0.04218 shouldn't drop near_bay
123
124 # decide if near_ocean should be transformed
125 md12i <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
126             I(log(housing_median_age)) + I(log(total_rooms)) +
127             I(log(total_bedrooms)) + I(log(population)) +
128             I(log(median_income)) + near_bay +
129             near_ocean + I(log(oneh_ocean + 0.01)),
130             data = train)
131 anova(md12i)
132 md12j <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
133             I(log(housing_median_age)) + I(log(total_rooms)) +
134             I(log(total_bedrooms)) + I(log(population)) +
135             I(log(median_income)) + near_bay +
136             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
137             data = train)
138 anova(md12j) # same p value; thus use a simplified version
139 md12k <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
140             I(log(housing_median_age)) + I(log(total_rooms)) +
141             I(log(total_bedrooms)) + I(log(population)) +
142             I(log(median_income)) + near_bay +
143             + I(log(oneh_ocean + 0.01)),
144             data = train)
145 anova(md12k, md12i) # 0.003605 shouldn't drop near_ocean

```



```

146 md12o <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + longitude +
147           I(log(housing_median_age)) + I(log(total_rooms)) +
148           I(log(total_bedrooms)) + I(log(population)) + I(log(households)) +
149           I(log(median_income)) + I((near_bay + 0.01)^(-2)) +
150           I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
151           data = train)
152 md12p <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
153           I(log(housing_median_age)) + I(log(total_rooms)) +
154           I(log(total_bedrooms)) + I(log(population)) + I(log(households)) +
155           median_income + I((near_bay + 0.01)^(-2)) +
156           I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
157           data = train)
158
159 # check result with four criterias
160 resultsa <- round(rbind(
161   select_criteria(md12, n=nrow(train)),
162   select_criteria(md12f, n=nrow(train)),
163   select_criteria(md12d, n=nrow(train)),
164   select_criteria(md12i, n=nrow(train)),
165   select_criteria(md12h, n=nrow(train)),
166   select_criteria(md12e, n=nrow(train)),
167   select_criteria(md12k, n=nrow(train)),
168   select_criteria(md12o, n=nrow(train)),
169   select_criteria(md12p, n=nrow(train))
170 ),3)
171 rownames(resultsa)<-c("1", "2", "3", "4", "5", "6", "7", "8", "9")
172 # 1:all transformed 2: near_bay original 3: households original 4: near_ocean original
173 # 5: near_bay dropped 6: households dropped 7: near_ocean dropped 8: longitude original
174 # 9: median_income original
175 resultsa
176 # drop households; near_ocean and near_bay original; longitude and median_income
177 # transformed
178
179 # decide if to use original version of longitude, median_income
180 md12l <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
181           I(log(housing_median_age)) + I(log(total_rooms)) +
182           I(log(median_income)) + near_bay +
183           near_ocean + I(log(oneh_ocean + 0.01)),
184           data = train) # reflect result on near_bay, households, and near_ocean
185 md12o <- lm(I(log(median_house_value)) ~ longitude +
186           I(log(housing_median_age)) + I(log(total_rooms)) +
187           I(log(median_income)) + near_bay +
188           near_ocean + I(log(oneh_ocean + 0.01)),
189           data = train)
190 md12p <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
191           I(log(housing_median_age)) + I(log(total_rooms)) +
192           median_income + near_bay +
193           near_ocean + I(log(oneh_ocean + 0.01)),
194           data = train)
195 md12q <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
196           I(log(housing_median_age)) + I(log(total_rooms)) +
197           I(log(median_income)) + near_bay +
198           near_ocean + I(log(oneh_ocean + 0.01)),
199           data = train)
200 resultsb <- round(rbind(
201   select_criteria(md12l, n=nrow(train)),
202   select_criteria(md12o, n=nrow(train)),
203   select_criteria(md12p, n=nrow(train)),
204   select_criteria(md12q, n=nrow(train))
205 ),3)
206 rownames(resultsb)<-c("1", "2", "3", "4")
207 # 1:all transformed 2: original longitude 3. original median_income
208 # 4: 1/median_income instead of log
209 resultsb # 1/median income is bad compared to log or original
210 # result shows that keep longitude and median_income transformed
211

```

```

212 #####
213 # preconditions check #
214 #####
215 plot(I(log(train$median_house_value))~fitted(md12))
216 abline(a=0,b=1)
217 lines(lowess(log(train$median_house_value)~fitted(md12)), col="blue") # condition 1 holds
218 ttrain <- data.frame(train$longitude^(-5), log(train$housing_median_age),
219                     log(train$total_rooms), log(train$median_income), train$near_bay,
220                     train$near_ocean, log(train$oneh_ocean + 0.01))
221 pairs(ttrain) # conditions hold
222
223 #####
224 # assumptions check based on residual plots#
225 #####
226 par(mfrow=c(3,3))
227 plot(rstandard(md12)~fitted(md12), xlab="fitted", ylab="Residuals")
228 for(i in 1:7){
229   plot(rstandard(md12)~ttrain[,i], xlab=names(ttrain)[i], ylab="Residuals")
230 }
231 qqnorm(rstandard(md12))
232 qqline(rstandard(md12))
233 plot(I(log(train$median_house_value))~fitted(md12))
234 abline(a=0,b=1)
235 lines(lowess(log(train$median_house_value)~fitted(md12)), col="blue")
236 # standardized residue for linearity assumption check
237 # linearity ok; normality ok
238
239 par(mfrow=c(3,3))
240 plot(md12$residuals~fitted(md12), xlab="fitted", ylab="Residuals")
241 for(i in 1:7){
242   plot(md12$residuals ~ ttrain[,i], xlab=names(ttrain)[i], ylab="Residuals")
243 }
244 qqnorm(residuals(md12))
245 qqline(residuals(md12))
246 # regular residue vs predictor to check independent errors and constant variance
247 # constant variance might be violated; independent error is ok, since the clusters
248 # are not separated from the other data; to be sure, we will wait for model validation
249
250 # use modified residue plots to check constant variance again
251 par(mfrow=c(3,3))
252 for(i in 1:7){
253   plot(sqrt(abs(rstandard(md12))) ~ ttrain[,i], xlab=names(ttrain)[i],
254        ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
255   m <- lm(sqrt(abs(rstandard(md12))) ~ ttrain[,i])
256   abline(a = m$coefficients[1], b = m$coefficients[2])
257 }
258 # constant variance assumption is violated for total_rooms and median_income
259
260 #
261 ttrain1 <- data.frame(train$longitude^(-5), log(train$housing_median_age),
262                      log(train$total_rooms), log(train$median_income), train$near_bay,
263                      train$near_ocean, log(train$oneh_ocean + 0.01))
264 newtrain <- ttrain1
265 md12r <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
266            I(log(housing_median_age)) + I(log(total_rooms)) +
267            I(log(median_income)) + near_bay +
268            near_ocean + I(log(oneh_ocean)),
269            data = train)
270 par(mfrow=c(3,3))
271 for(i in 1:7){
272   plot(sqrt(abs(rstandard(md12r))) ~ ttrain1[,i], xlab=names(ttrain1)[i],
273        ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
274   m <- lm(sqrt(abs(rstandard(md12r))) ~ ttrain1[,i])
275   abline(a = m$coefficients[1], b = m$coefficients[2])
276 }
277 # constant variance is much improved
278 round(select_criteria(md12r, n=nrow(train))) # 0.568 6192.757 6193.017 6237.716 worst
279 summary(md12r)
280

```

```

281 md12s <- lm(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
282             I(log(housing_median_age)) + I(sqrt(total_rooms)) +
283             I(sqrt(median_income)) + near_bay +
284             near_ocean + I(log(oneh_ocean)),
285             data = train)
286 round(select_criteria(md12s, n=nrow(train)),3)
287 # 0.593    6151.591    6151.851    6196.551 better than md12r
288
289 par(mfrow=c(3,3))
290 plot(rstandard(md12s)~fitted(md12s), xlab="fitted", ylab="Residuals")
291 for(i in 1:7){
292   plot(rstandard(md12s)~ttrain[,i], xlab=names(ttrain)[i], ylab="Residuals")
293 }
294 qqnorm(rstandard(md12s))
295 qqline(rstandard(md12s))
296 plot(I(log(train$median_house_value))~fitted(md12s))
297 abline(a=0,b=1)
298 lines(lowess(log(train$median_house_value)~fitted(md12s)), col="blue")
299
300 par(mfrow=c(3,3))
301 plot(md12s$residuals~fitted(md12s), xlab="fitted", ylab="Residuals")
302 for(i in 1:7){
303   plot(md12s$residuals ~ ttrain[,i], xlab=names(ttrain)[i], ylab="Residuals")
304 }
305 qqnorm(residuals(md12s))
306 qqline(residuals(md12s))
307 par(mfrow=c(3,3))
308 for(i in 1:7){
309   plot(sqrt(abs(rstandard(md12s))) ~ ttrain[,i], xlab=names(ttrain)[i],
310         ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
311   m <- lm(sqrt(abs(rstandard(md12s))) ~ ttrain[,i])
312   abline(a = m$coefficients[1], b = m$coefficients[2])
313 }
314 # md12r and md12s satisfy assumptions
315
316 #####
317 # variable selection #
318 #####
319 # multicollinearity check #
320 vif(md12) # given the threshold is 5, latitude, longitude, total_rooms,
321 # total_bedrooms, and households don't pass the multicollinearity check;
322 # correlation of variables with latitude
323 cors <- NULL
324 for (i in 1:13){
325   cors <- c(cors, cor(train$latitude, train[, (i+1)]))
326 }
327 cdf <- data.frame("Correlation" = cors, "Predictors" = pred)
328 cdf[order(-abs(cdf$Correlation)),] # latitude strongly correlate with longitude
329 # by -0.91787592, second is -0.42341103 Access to Ocean
330
331 # correlation of variables with total_rooms
332 cors <- NULL
333 for (i in 1:13){
334   cors <- c(cors, cor(train$total_rooms, train[, (i+1)]))
335 }
336 cdf <- data.frame("Correlation" = cors, "Predictors" = pred)
337 cdf[order(-abs(cdf$Correlation)),] # Total Bedrooms: 0.932037674;
338 # Neighborhood Households: 0.931694831; Neighborhood Population: 0.883482788
339
340 # model versions: 1. latitude, no longitude 2. longitude, no latitude
341 # 3. one of total_rooms, total_bedrooms, population
342 # 4. two of total_rooms, total_bedrooms, population
343 t1 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,8)])
344 vif(t1) # problematic vif with latitude and longitude
345 t2 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,2,8)])
346 vif(t2) # latitude < 5 w/o longitude
347 t3 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,8)])
348 vif(t3) # longitude < 5 w/o latitude
349 # need to remove one from total_rooms,total_bedrooms,and population,
350

```

```

351 # since yif is still a problem after removing households
352 t4 <- lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,5,8)]) # w/o total_rooms
353 vif(t4) # bedrooms and population still problematic
354 t5 <- lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,6,8)]) # w/o total_bedrooms
355 vif(t5) # total_rooms still problematic
356 t6 <- lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,7,8)]) # w/o population
357 vif(t6) # total_rooms and total_bedrooms still problematic
358 # remove another variable
359 t7 <- lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,5,6,8)])
360 # w/o total_rooms and total_bedrooms
361 vif(t7) |
362 t8 <- lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,5,7,8)])
363 # w/o total_rooms and population
364 vif(t8)
365 t9 <- lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,6,7,8)])
366 # w/o population and total_bedrooms
367 vif(t9)
368 # t7, t8, t9 both pass; same as t2, t3
369 # models: one of
370 # 1. latitude, longitude 2. total_rooms, total_bedrooms, population
371 #####compare six models above#####
372 results <- round(rbind(
373   select_criteria(lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,5,6,8)]), n=nrow(newtrain)),
374   select_criteria(lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,5,7,8)]), n=nrow(newtrain)),
375   select_criteria(lm(log(train$median_house_value)~., data=newtrain[, -c(1,3,6,7,8)]), n=nrow(newtrain)),
376   select_criteria(lm(log(train$median_house_value)~., data=newtrain[, -c(1,2,5,6,8)]), n=nrow(newtrain)),
377   select_criteria(lm(log(train$median_house_value)~., data=newtrain[, -c(1,2,5,7,8)]), n=nrow(newtrain)),
378   select_criteria(lm(log(train$median_house_value)~., data=newtrain[, -c(1,2,6,7,8)]), n=nrow(newtrain))
379 ),3)
380 rownames(results)<-c("1", "2", "3", "4", "5", "6")
381 # 1:longitude&population 2: longitude&total_bedrooms 3: longitude&total_rooms
382 # 4: latitude&population 5: latitude&total_bedrooms 6:latitude&total_rooms
383 results # mdl 3 is best with highest adjusted r and lowest other values
384 # chosen model: longitude, median_age, total_rooms, median_income;
385 # near_bay, near_ocean, oneh_ocean
386 # w/o latitude, total_bedrooms, population, inland
387 # from individual significance check earlier, drop households, use simplified
388 # near_bay and near_ocean
389 mdl6 <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
390   I(log(housing_median_age)) + I(log(total_rooms))
391   + I(log(median_income)) + near_bay + near_ocean + oneh_ocean,
392   data = train)
393 summary(mdl6)
394 anova(mdl6) # longitude, housing_median_age are insignificant
395
396 # recheck transformation
397 mult <- lm(cbind(train$median_house_value, train$housing_median_age, train$total_rooms,
398   train$median_income, train$near_bay,
399   train$near_ocean, train$oneh_ocean) ~ 1)
400 pow <- powerTransform(mult, family="bcnPower")
401 pow # suggest log for median_house_value, median_age, near_bay, near_ocean, oneh_ocean
402 # suggest 1205 for total_rooms, 1.6 for median_income
403
404 # check if dropping longitude
405 mdl6a <- lm(I(log(median_house_value)) ~ I(log(housing_median_age)) +
406   I(log(total_rooms)) + I(1/median_income) +
407   near_bay + near_ocean + oneh_ocean,
408   data = train)
409 anova(mdl6a, mdl6) # shouldn't drop
410
411 # check if dropping housing_median_age
412 mdl6b <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
413   I(log(total_rooms))
414   + I(1/median_income) +
415   near_bay + near_ocean + oneh_ocean,
416   data = train)
417 anova(mdl6b, mdl6) # 0.007716 shouldn't drop
418
419 # check if dropping near_bay
420 mdl6c <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
421   I(log(housing_median_age)) + I(log(total_rooms))
422   + I(1/median_income) + near_ocean + oneh_ocean,
423   data = train)
424 anova(mdl6c, mdl6) # shouldn't drop
425
426 # stepwise selection using AIC
427 stepAIC(lm(log(median_house_value) ~ I(longitude^(-5)) +
428   I(log(housing_median_age)) + I(log(total_rooms))
429   + I(log(median_income)) + near_bay + near_ocean +
430   log(oneh_ocean + 0.01), data=train, direction = "both", k=2))
431 # suggest dropping nothing

```



```

433 ~ #####
434 # Analysis of Covariance #
435 ~ #####
436 # 12 models
437 # near_bay
438 mod1a <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$near_bay==0),])
439 mod1b <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$near_bay==1),])
440 mod1a$coefficients # 1.187153e+01 -4.486685e+09
441 mod1b$coefficients # 4.631502e+01 9.331775e+11
442 # different intercept and slope --> near_bay main effect and interaction to longitude
443
444 mod2a <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$near_bay==0),])
445 mod2b <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$near_bay==1),])
446 mod2a$coefficients # 12.02524635 0.01059813
447 mod2b$coefficients # 12.220974293 -0.001511231
448 # similar intercept, slightly different slope
449
450 mod3a <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$near_bay==0),])
451 mod3b <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$near_bay==1),])
452 mod3a$coefficients # 10.8942132 0.1517004
453 mod3b$coefficients # 8.6754424 0.4732542
454 # slightly different intercept, similar slope
455
456 mod4a <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$near_bay==0),])
457 mod4b <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$near_bay==1),])
458 mod4a$coefficients # 10.9670588 0.8834415
459 mod4b$coefficients # 11.1663574 0.8770492
460 # similar intercept and slope
461
462 # near_ocean
463 mod5a <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$near_ocean==0),])
464 mod5b <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$near_ocean==1),])
465 mod5a$coefficients # 1.215589e+01 2.780397e+09
466 mod5b$coefficients # 1.338566e+01 2.549884e+10
467 # similar intercept, different slope --> near_ocean*longitude
468
469 mod6a <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$near_ocean==0),])
470 mod6b <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$near_ocean==1),])
471 mod6a$coefficients # 11.9801013 0.0190077
472 mod6b$coefficients # 11.99110614 0.09843133
473 # similar intercept and slope
474
475 mod7a <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$near_ocean==0),])
476 mod7b <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$near_ocean==1),])
477 mod7a$coefficients # 10.781933 0.164615
478 mod7b$coefficients # 10.586101 0.224867
479 # similar intercept and slope
480
481 mod8a <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$near_ocean==0),])
482 mod8b <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$near_ocean==1),])
483 mod8a$coefficients # 10.9354817 0.8995935
484 mod8b$coefficients # 11.449247 0.694835
485 # similar intercept and slope
486 # consider drop near_ocean, test near_ocean and longitude as interact
487
488 # oneh_ocean
489 mod9a <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$oneh_ocean==0),])
490 mod9b <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$oneh_ocean==1),])
491 mod9a$coefficients # 1.282667e+01 2.326147e+10
492 mod9b$coefficients # 1.307992e+01 1.843861e+10
493 # similar intercept, different slope --> oneh_ocean*longitude
494
495 mod10a <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$oneh_ocean==0),])
496 mod10b <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$oneh_ocean==1),])
497 mod10a$coefficients # 11.3991593 0.1497339
498 mod10b$coefficients # 12.55622368 -0.07935795
499 # similar intercept, slightly different slope
500
501 mod11a <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$oneh_ocean==0),])
502 mod11b <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$oneh_ocean==1),])
503 mod11a$coefficients # 10.2632509 0.2120744
504 mod11b$coefficients # 11.2553440 0.1366175
505 # similar intercept and slope
506
507 mod12a <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$oneh_ocean==0),])
508 mod12b <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$oneh_ocean==1),])
509 mod12a$coefficients # 10.8223276 0.9436217
510 mod12b$coefficients # 11.3910752 0.6718017
511 # similar intercept and slope
512 # consider drop oneh_ocean, test interaction term between oneh_ocean and longitude
513

```

```

514 md17 <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) + I(longitude^(-5)*near_bay)
515 + I(longitude^(-5)*near_ocean) + I(longitude^(-5)*oneh_ocean) +
516 I(log(housing_median_age)) + I(log(total_rooms))
517 + I(log(median_income)) + near_bay, data = train)
518 summary(md17)
519
520 stepAIC(lm(log(median_house_value) ~ I(longitude^(-5)) + I(longitude^(-5)*near_bay)
521 + I(longitude^(-5)*near_ocean) + I(longitude^(-5)*oneh_ocean) +
522 I(log(housing_median_age)) + I(log(total_rooms))
523 + I(log(median_income)) + near_bay + near_ocean + oneh_ocean,
524 data=train, direction = 'both', k = 2))
525 # suggest dropping longitude
526 md17a <- lm(I(log(median_house_value)) ~ I(longitude^(-5)*near_bay)
527 + I(longitude^(-5)*near_ocean) + I(longitude^(-5)*oneh_ocean) +
528 I(log(housing_median_age)) + I(log(total_rooms))
529 + I(log(median_income)) + near_bay, data = train)
530 anova(md17a, md17) # shouldn't drop
531 vif(md17a) # longitude and longitude*near_bay too correlated
532 md17b <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) + I(longitude^(-5)*near_ocean) +
533 I(longitude^(-5)*oneh_ocean) + I(log(housing_median_age)) +
534 I(log(total_rooms)) + I(log(median_income)) + near_bay, data = train)
535 vif(md17b)
536 md17c <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) + I(longitude^(-5)*near_bay)
537 + I(longitude^(-5)*near_ocean) + I(longitude^(-5)*oneh_ocean) +
538 I(log(housing_median_age)) + I(log(total_rooms))
539 + I(log(median_income)), data = train)
540 vif(md17c)
541 # either delete longitude*near_bay or near_bay makes VIF test pass
542 md16d <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
543 I(longitude^(-5)*near_ocean) + I(log(housing_median_age)) +
544 I(log(total_rooms)) + I(log(median_income)) + near_bay + oneh_ocean,
545 data = train) # interaction term for near_ocean instead of main effect
546 md16e <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
547 I(longitude^(-5)*oneh_ocean) + I(log(housing_median_age)) +
548 I(log(total_rooms)) + I(log(median_income)) + near_bay + near_ocean,
549 data = train) # interaction term for oneh_ocean instead of main effect
550 md16f <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
551 I(longitude^(-5)*near_bay) + I(log(housing_median_age)) +
552 I(log(total_rooms)) + I(log(median_income)) + near_ocean + oneh_ocean,
553 data = train) # interaction term for near_bay instead of main effect
554 # use selection criteria to compare models
555 results1 <- round(rbind(
556 select_criteria(md17b, n=nrow(newtrain)),
557 select_criteria(md17c, n=nrow(newtrain)),
558 select_criteria(md16d, n=nrow(newtrain)),
559 select_criteria(md16e, n=nrow(newtrain)),
560 select_criteria(md16f, n=nrow(newtrain)),
561 select_criteria(md16, n=nrow(newtrain))
562 ),3)
563 rownames(results1)<-c("1", "2", "3", "4", "5", "6")
564 # 1: main effect term for near_bay 2: interaction term for near_bay 3: all main effects
565 results1 # md16 is best
566
567 #####
568 # Leverage Points #
569 #####
570 # leverage points
571 hii1 <- hatvalues(md12s)
572 # hlp1 <- which(hii1 > 4/nrow(train))
573 # hlp1 # high leverage points
574 # show that there are some high leverage points
575 #standardized residue
576 r1 <- rstandard(md12s)
577 # use the residues of leverage points to check for outliers among leverage points
578 lr1 <- r1[which(hii1 > 4/nrow(ttrain2))]
579 lrs1 <- which(lr1 >= 4 | lr1 <= -4) # outlier range for large dataset
580 lrs1

```



```

581 # shows point: 478
582 ttrain2[478,]
583 summary(ttrain2)
584 train[478,]
585 summary(train)
586 # 478 very old median_age, high median_income, oneh_ocean
587 # house value 500001, max of housing value
588
589 # test out models again
590 results1 <- round(rbind(
591   select_criteria(md17b, n=nrow(newtrain)),
592   select_criteria(md17c, n=nrow(newtrain)),
593   select_criteria(md16d, n=nrow(newtrain)),
594   select_criteria(md16e, n=nrow(newtrain)),
595   select_criteria(md16f, n=nrow(newtrain)),
596   select_criteria(md11, n=nrow(newtrain)),
597   select_criteria(md12, n=nrow(newtrain)),
598   select_criteria(md13, n=nrow(newtrain)),
599   select_criteria(md16, n=nrow(newtrain)),
600   select_criteria(md17, n=nrow(newtrain))
601 ),3)
602 rownames(results1)<-c("1", "2", "3", "4", "5", "6", "7")
603 # 1: main effect term for near_bay 2: interaction term for near_bay
604 # 3: interaction term for near_ocean 4: interaction term for oneh_ocean
605 # 5: interaction term for near_bay
606 # md16 all main effect; md17 interaction term for near_ocean and oneh_ocean, and
607 # both interaction term and main effect for near_bay
608 results1 # md12 is best, after that md17, then md16; both md12 and 7 have correlation
609 # issue, so md16 is still best
610
611 #####
612 # check residual plot again #
613
614 # update md18 based on bad leverage
615 md18 <- lm(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
616   I(log(housing_median_age)) + I(sqrt(total_rooms)) +
617   I(sqrt(median_income)) + near_bay +
618   near_ocean + I(1/log(oneh_ocean)), data = train[-478,])
619 round(select_criteria(md18, n=nrow(train)),3)
620 # md18 four measures: .520 -1246.194 -1245.933 -1201.234
621 # compared to 0.590 6151.303 6151.563 6196.262 from md12s; improved!
622 anova(md18)
623 # longitude and housing_median_age are not significant
624 best <- regsubsets(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
625   I(log(housing_median_age)) + I(sqrt(total_rooms)) +
626   I(sqrt(median_income)) + near_bay +
627   near_ocean + I(1/log(oneh_ocean)),
628   data = train[-478,], nbest=1)
629 summary(best)
630
631 # let's plot these for easier digestibility
632 subsets(best, statistic="adjr2") # favor keeping all variables
633 # check if dropping longitude
634 md18a <- lm(I(sqrt(median_house_value)) ~
635   I(log(housing_median_age)) + I(sqrt(total_rooms)) +
636   I(sqrt(median_income)) + near_bay +
637   near_ocean + I(1/log(oneh_ocean)), data = train[-478,])
638 anova(md18a,md18) # 0.03162
639 md18b <- lm(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
640   I(sqrt(total_rooms)) +
641   I(sqrt(median_income)) + near_bay +
642   near_ocean + I(1/log(oneh_ocean)), data = train[-478,])
643 anova(md18b,md18) # 0.0004241
644 # shouldn't drop either variable

```

```

646 ~ #####
647 # preconditions recheck #
648 ~ #####
649 plot(I(sqrt(train[-478,]$median_house_value))~fitted(md18))
650 abline(a=0,b=1)
651 lines(lowess(sqrt(train[-478,]$median_house_value)~fitted(md18)), col="blue")
652 # condition 1 holds
653 ttrain2 <- data.frame(train$longitude^(-5),log(train$housing_median_age),
654                       sqrt(train$total_rooms), sqrt(train$median_income), train$near_bay,
655                       train$near_ocean, 1/log(train$oneh_ocean))
656 newtrain <- ttrain2[-478,]
657 pairs(newtrain) # conditions hold
658
659 ~ #####
660 # assumptions recheck based on residual plots#
661 ~ #####
662 par(mfrow=c(3,4))
663 plot(rstandard(md18)~fitted(md18), xlab="fitted", ylab="Residuals")
664 ~ for(i in 1:7){
665   plot(rstandard(md18)~newtrain[,i], xlab=names(newtrain)[i], ylab="Residuals")
666 ~ }
667 qqnorm(rstandard(md18))
668 qqline(rstandard(md18))
669 plot(I(sqrt(train[-478,]$median_house_value))~fitted(md18))
670 abline(a=0,b=1)
671 lines(lowess(log(train$median_house_value)~fitted(md18)), col="blue")
672 # standardized residue for linearity assumption check
673 # linearity ok; normality ok
674
675 par(mfrow=c(3,3))
676 plot(md18$residuals~fitted(md18), xlab="fitted", ylab="Residuals")
677 ~ for(i in 1:7){
678   plot(md18$residuals ~ newtrain[,i], xlab=names(newtrain)[i], ylab="Residuals")
679 ~ }
680 qqnorm(residuals(md18))
681 qqline(residuals(md18))
682 # regular residue vs predictor to check independent errors and constant variance
683 # constant variance might be violated; independent error is ok, since the clusters
684 # are not separated from the other data; to be sure, we will wait for model validation
685
686 # use modified residue plots to check constant variance again
687 par(mfrow=c(3,3))
688 ~ for(i in 1:7){
689   plot(sqrt(abs(rstandard(md18))) ~ newtrain[,i], xlab=names(newtrain)[i],
690         ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
691   m <- lm(sqrt(abs(rstandard(md18))) ~ newtrain[,i])
692   abline(a = m$coefficients[1], b = m$coefficients[2])
693 ~ }
694 # constant variance assumption is satisfied
695 vif(md18)
696
697 ~ #####
698 # model validation #
699 ~ #####
700 summary(train)
701 summary(test)
702 # looks comparable
703
704 # let's fit the same 3 predictor model but using the test data
705 md18_test <- lm(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
706               I(log(housing_median_age)) + I(sqrt(total_rooms)) +
707               I(sqrt(median_income)) + near_bay +
708               near_ocean + I(log(oneh_ocean)),
709               data = test)
710 summary(md18_test)
711 summary(md18)
712 anova(md18_test)
713 # model performed better for test dataset with adjr2 0.667 instead of 0.5905
714 vif(md18_test)

```

```

716 # check preconditions
717 plot(I(sqrt(test$median_house_value))~fitted(md18_test))
718 abline(a=0,b=1)
719 lines(lowess(sqrt(test$median_house_value)~fitted(md18_test)), col="blue")
720 # condition 1 holds
721 ttest <- data.frame(test$longitude^(-5),log(test$housing_median_age),
722                     sqrt(test$total_rooms), sqrt(test$median_income), test$near_bay,
723                     test$near_ocean, log(test$oneh_ocean))
724 newtest <- ttest
725 pairs(newtest) # conditions pass
726
727 par(mfrow=c(3,4))
728 plot(rstandard(md18_test)~fitted(md18_test), xlab="fitted", ylab="Residuals")
729 for(i in 1:7){
730   plot(rstandard(md18_test)~newtest[,i], xlab=names(newtest)[i], ylab="Residuals")
731 }
732 qqnorm(rstandard(md18_test))
733 qqline(rstandard(md18_test))
734 plot(I(sqrt(train$median_house_value))~fitted(md18_test))
735 abline(a=0,b=1)
736 lines(lowess(log(train$median_house_value)~fitted(md18_test)), col="blue")
737 # standardized residue for linearity assumption check
738 # linearity ok; normality ok
739
740 par(mfrow=c(3,3))
741 plot(md18_test$residuals~fitted(md18_test), xlab="fitted", ylab="Residuals")
742 for(i in 1:7){
743   plot(md18_test$residuals ~ newtest[,i], xlab=names(newtest)[i], ylab="Residuals")
744 }
745 qqnorm(residuals(md18_test))
746 qqline(residuals(md18_test))
747 # regular residue vs predictor to check independent errors and constant variance
748 # constant variance might be violated; independent error is ok, since the clusters
749 # are not separated from the other data; to be sure, we will wait for model validation
750
751 # use modified residue plots to check constant variance again
752 par(mfrow=c(3,3))
753 for(i in 1:7){
754   plot(sqrt(abs(rstandard(md18_test))) ~ newtest[,i], xlab=names(newtest)[i],
755         ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
756   m <- lm(sqrt(abs(rstandard(md18_test))) ~ newtest[,i])
757   abline(a = m$coefficients[1], b = m$coefficients[2])
758 }
759 # constant variance not a horizontal line for several variables
760
761 #####
762 # summary statistics of sample data set #
763 #####
764 str(train)
765 summary(train)
766 apply(train[,2:14], 2, mean)
767 apply(train[,2:14], 2, sd)
768
769 #histograms
770 par(mfrow=c(4,4))
771 for (i in 1:9){
772   hist(as.numeric(train[, (i+1)]), breaks=10, main=sprintf(
773     "%s of Sample Californian Homes", pred[i]), xlab=pred[i], ylab= "Count")
774 }
775 for (i in 10:13){
776   hist(as.numeric(train[, (i+1)]), breaks=2, main=sprintf(
777     "%s of Sample Californian Homes", pred[i]), xlab=pred[i], ylab = "Count")
778 }
779

```

```

780 # boxplot 1
781 par(mfrow=c(3,3))
782 for (i in 1:9){
783   boxplot(as.numeric(train[, (i+1)]), xlab=pred[i], ylab="Value")
784 }
785
786 # scatter plot of longitude and latitude
787 ocean_pts <- which(train$near_ocean == 1)
788 bay_pts <- which(train$near_bay == 1)
789 inland_pts <- which(train$inland == 1)
790 oneh_pts <- which(train$oneh_ocean == 1)
791 neither <- which((train$oneh_ocean == 0)&(train$inland == 0))
792 mix <- sample(train$X, 20, replace = FALSE)
793
794 par(mfrow=c(1,1))
795 library(png)
796 img <- readPNG('c:/Users/i5/Downloads/STA302 Data Analysis I/Mini project 2/map0.png')
797 plot(train$longitude, train$latitude, xlab=pred[2], ylab=pred[1],
798       main="Sample Locations", type = "n")
799 rasterImage(img,xleft=-125, xright=-115, ybottom=32, ytop=42.5)
800 points(train$longitude[inland_pts],train$latitude[inland_pts], col = "darkgreen",
801        pch = 16)
802 points(train$longitude[oneh_pts],train$latitude[oneh_pts], col = "dodgerblue4", pch = 16)
803 points(train$longitude[ocean_pts], train$latitude[ocean_pts], col = "blue", pch = 16)
804 points(train$longitude[bay_pts],train$latitude[bay_pts], col = "cyan", pch = 16)
805 legend("topright", legend = c("Bay", "Ocean","Drive to Ocean", "Inland"),
806        col = c("cyan", "blue","dodgerblue4", "darkgreen"), lty = 1)
807
808 #graph in term of price points
809 low <- which(train$median_house_value <= 110025)
810 average <- which(110025<train$median_house_value <267525)
811 high <- which(train$median_house_value >= 267525)
812 plot(train$longitude, train$latitude, xlab=pred[2], ylab=pred[1],
813       main="Sample Price", type = "n")
814 rasterImage(img,xleft=-125, xright=-115, ybottom=32, ytop=42.5)
815 points(train$longitude[low],train$latitude[low], col = "yellow", pch = 16)
816 points(train$longitude[average],train$latitude[average], col = "tan1", pch = 16)
817 points(train$longitude[high],train$latitude[high], col = "red3", pch = 16)
818 legend("topright", legend = c("Low", "Medium", "High"), col = c("yellow", "tan1",
819                                                                    "red3"), lty = 1)
820
821 # update train based on new dataset
822
823 # relationship between each variables
824 pairs(train[,2:14], lower.panel=NULL)
825
826 # relationship with median house values
827 par(mfrow=c(4,4))
828 for (i in 1:11){
829   plot(train[, (i+1)], train$median_house_value, xlab = pred[i], ylab = pred[9])
830 }
831
832 # correlation table of all relationships
833 library(xtable)
834 cors <- NULL
835 for (i in 1:13){
836   cors <- c(cors, cor(train$median_house_value, train[, (i+1)]))
837 }
838 cdf <- data.frame("Correlation" = cors, "Predictors" = pred)
839 cdf[order(-abs(cdf$Correlation)),]

```