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 |standardized residue| >= 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.

<u>Results</u>

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*.

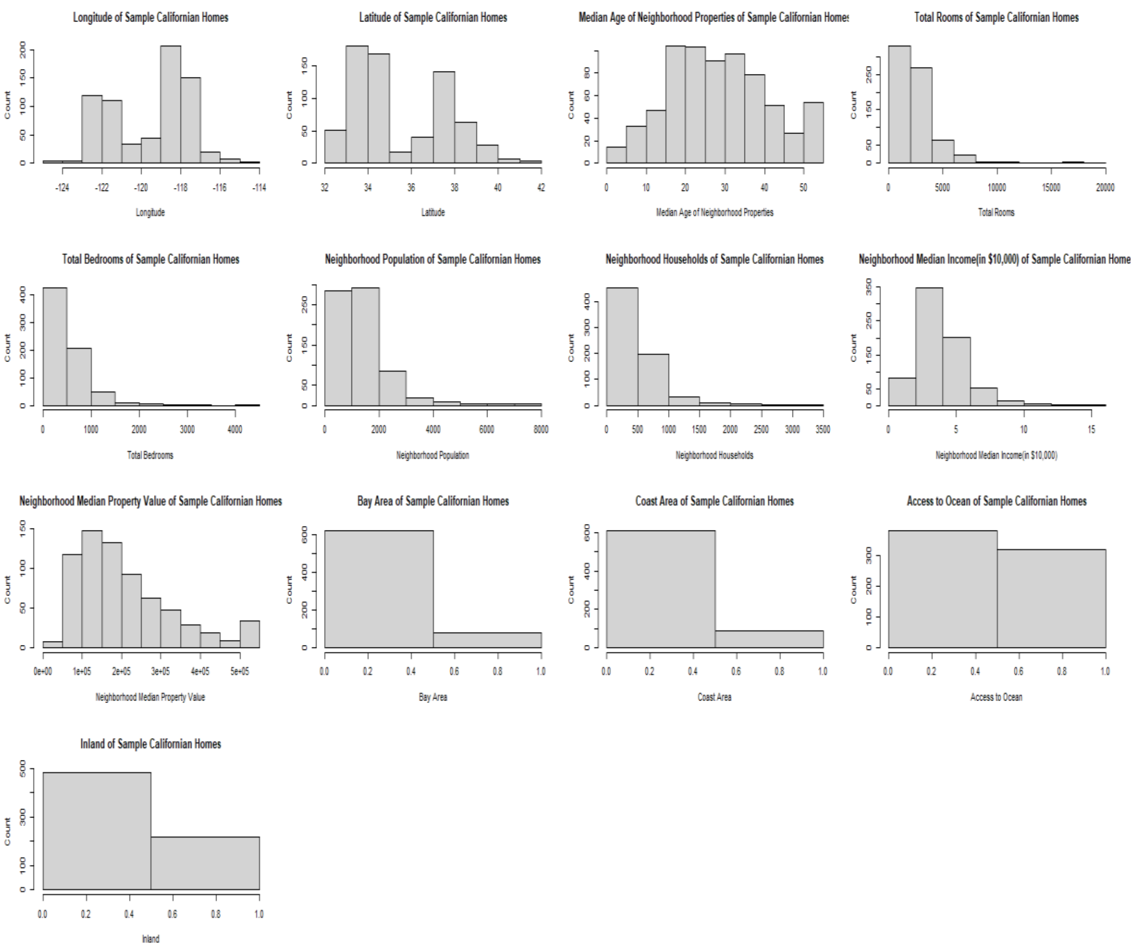

Figure 1 Variable Histograms

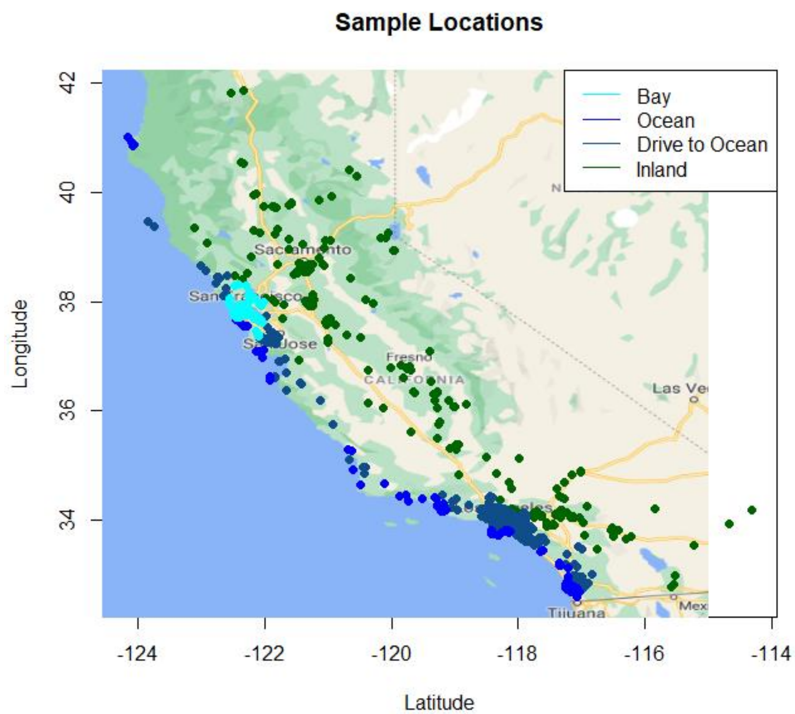

Figure 3 Sample Locations

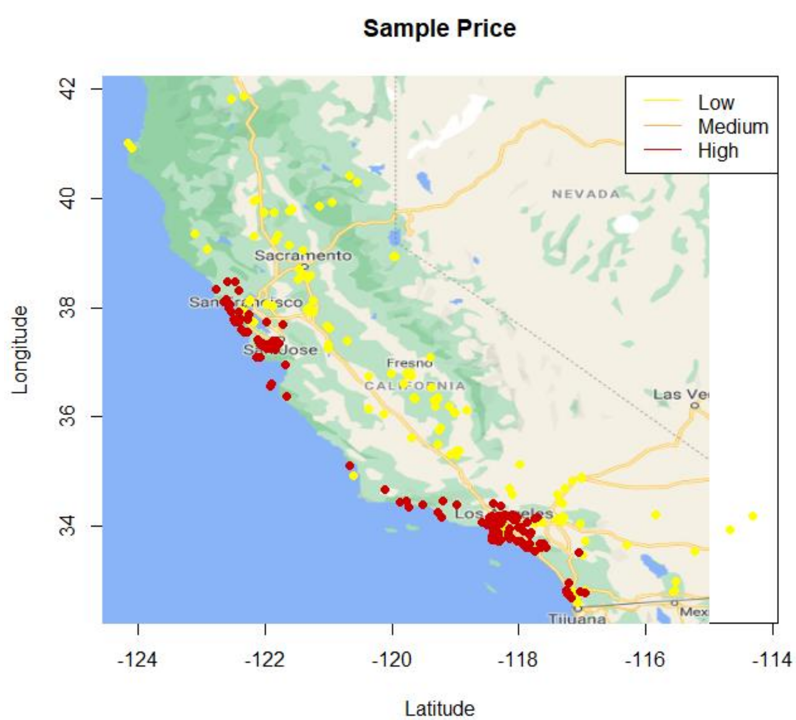

Figure 2 Sample Price

Process of Obtaining the Final Model

Variable	Suggested Power from powerTransform
latitude	-6.2103873
longitude	-5.1456697
median_house_value	0.147994
housing_median_age	0.7533555
total_rooms	0.1445191
total_bedrooms	0.1540388
population	0.1613027
households	0.1747819
median_income	0.161507
near_bay	-1.9938737
near_ocean	-1.7186720
oneh_ocean	-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
longitude, population	0.489	-1204.186	-1204.099	-1177.430
longitude, total_bedrooms	0.473	-1182.664	-1182.577	-1155.908
longitude, total_rooms	0.458	-1162.899	-1162.813	-1136.144
latitude, population	0.473	-1182.425	-1182.339	-1155.669
latitude, total_bedrooms	0.458	-1163.364	-1163.278	-1136.608
latitude, total_rooms	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 *near_bay*	0.660	-1483.025	-1482.572	-1424.412
Untransformed *households*	0.660	-1481.027	-1480.498	-1417.863
Untransformed *near_ocean*	0.660	-1483.025	-1482.572	-1424.412
near_bay dropped	0.659	-1480.828	-1480.445	-1426.766
households dropped	0.660	-1483.025	-1482.572	-1424.412
near_ocean dropped	0.657	-1476.410	-1476.028	-1422.348
Untransformed *longitude*	0.657	-1474.956	-1474.427	-1411.792
Untransformed *median_income*	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 *near_bay* and interaction terms for *near_ocean and oneh_ocean*	0.592	-1357.08	-1356.826	-1312.127
Interaction terms for near_bay, near_ocean, oneh_ocean	0.591	-1356.630	-1356.370	-1311.670
Interaction term for *near_ocean*, main effect terms for *near_bay* and *oneh_ocean*	0.593	-1359.790	-1359.530	-1314.830
Interaction term for *oneh_ocean*, main effect terms for *near_bay* and *near_ocean*	0.592	-1357.443	-1357.183	-1312.483
Interaction term for *near_bay* and main effect term for *near_ocean* and *oneh_ocean*	0.594	-1360.603	-1360.343	-1315.643
Main effect terms for *near_bay*, *near_ocean*, and *oneh_ocean*	0.594	-1361.105	-1360.845	-1316.145
Main effect term for *near_bay* and interaction terms for *near_bay*, *near_ocean*, and *oneh_ocean*	0.602	-1374.095	-1373.776	-1324.584

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

R Code

```r
setwd(dir = 'C:/Users/i5/Downloads/STA302 Data Analysis I/Final Project')
data <- read.csv('C:/Users/i5/Downloads/STA302 Data Analysis I/Final Project/housing.csv', header=TRUE)
set.seed(1004081030)
rows <- sample(1:nrow(data), 1000, replace=FALSE)
train <- data[rows[1:700],]
test <- data[rows[701:1000],]
pred <- c("Longitude", "Latitude", "Median Age of Neighborhood Properties",
          "Total Rooms","Total Bedrooms","Neighborhood Population",
          "Neighborhood Households", "Neighborhood Median Income(in $10,000)",
          "Neighborhood Median Property Value", "Bay Area","Coast Area",
          "Access to Ocean","Inland")
library(car)
library(MASS)
library(leaps)
select_criteria = function(model, n)
{
  SSres <- sum(model$residuals^2)
  Rsq_adj <- summary(model)$adj.r.squared
  p <- length(model$coefficients) - 1
  AIC <- n*log(SSres/n) + 2*p
  AICc <- AIC + (2*(p+2)*(p+3)/(n-p-1))
  BIC <- n*log(SSres/n) + (p+2)*log(n)
  res <- c(SSres, Rsq_adj, AIC, AICc, BIC)
  names(res) <- c("SSres", "Rsq_adj", "AIC", "AIC_c", "BIC")
  return(res)
}
# full model
mod_train <- lm(median_house_value ~ latitude + longitude + housing_median_age
                + total_rooms + total_bedrooms + population + households +
                  median_income + near_bay + near_ocean + oneh_ocean + inland, data = train)
summary(mod_train) # shows total rooms, households, and near bay to be significant
anova(mod_train) # shows housing_median_age and near_bay to be significant

# delete inland given multicollinearity issue
mdl1 <- lm(median_house_value ~ latitude + longitude + housing_median_age
           + total_rooms + total_bedrooms + population + households +
             median_income + near_bay + near_ocean + oneh_ocean, data = train)
summary(mdl1)
anova(mdl1) # suggest latitude, longitude, total rooms, total bedrooms, population,
# households, median income, next near_ocean, and oneh_ocean; w/o median_age and near_bay

################################################################################
# model with transformed values to run individual significance test #
################################################################################
# check with power transformation
powerTransform(lm(train$median_house_value ~ 1)) # 0.147994
powerTransform(lm(cbind(train$median_house_value, train$housing_median_age) ~ 1)) # suggest 0.7533555
powerTransform(lm(cbind(train$median_house_value, train$total_rooms) ~ 1)) # suggest power 0.1445191
powerTransform(lm(cbind(train$median_house_value, train$total_bedrooms) ~ 1)) # suggest power 0.1540388
powerTransform(lm(cbind(train$median_house_value, train$population) ~ 1)) # suggest power 0.1613027
```


```r
powerTransform(lm(cbind(train$median_house_value, train$population) ~ 1)) # suggest power 0.1613027
```


```r
powerTransform(lm(cbind(train$median_house_value, train$households) ~ 1)) # suggest power 0.1747819
powerTransform(lm(cbind(train$median_house_value, train$median_income) ~ 1)) # suggest power 0.161507

#   the other five variables are not strictly positive, so must make adjustment
powerTransform(lm(cbind(train$median_house_value, train$latitude) ~ 1)) # suggest power -6.2103873
powerTransform(lm(cbind(train$median_house_value, I(train$near_bay + 0.01)) ~ 1)) # suggest power -1.9938737
powerTransform(lm(cbind(train$median_house_value, I(train$near_ocean + 0.01)) ~ 1)) # suggest power -1.7186720
powerTransform(lm(cbind(train$median_house_value, I(train$oneh_ocean + 0.01)) ~ 1)) # suggest power -0.1156945
powerTransform(lm(cbind(train$median_house_value, abs(train$longitude)) ~ 1)) # suggest power -5.1456697

# all transformed
mdl2 <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
             I(log(housing_median_age)) + I(log(total_rooms)) +
             I(log(total_bedrooms)) + I(log(population)) + I(log(households)) +
             I(log(median_income)) + I((near_bay + 0.01) ^(-2)) +
             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
           data = train)
anova(mdl2) # latitude, total_rooms, median_age, total_rooms and oneh_ocean's individual
# significance has improved; households, near_bay near_ocean are worse;
# households, near_bay, near_ocean are not significant
# same significance: longitude, total_bedroom, population, median_income

# near_bay is insignificant with or without transformation; check if drop near_bay
mdl2a <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) + I(log(housing_median_age))
           + I(log(total_rooms)) + I(log(total_bedrooms)) + I(log(population)) + I(log(households)) +
             I(log(median_income)) + I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)), data = train)
anova(mdl2a, mdl2) # shouldn't drop near_bay

# decide if households should be transformed
# transformed near_bay, transformed near_ocean, original households
```

```
mdl2d <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
             I(log(housing_median_age)) + I(log(total_rooms)) +
             I(log(total_bedrooms)) + I(log(population)) + households +
             I(log(median_income)) + I((near_bay + 0.01) ^(-2)) +
             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
           data = train)
anova(mdl2d) # household significance 0.367657 compared to 0.066101 as transformed;
# keep households as transformed

# compare transformed households vs dropping households
mdl2e <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
             I(log(housing_median_age)) + I(log(total_rooms)) +
             I(log(total_bedrooms)) + I(log(population)) +
             I(log(median_income)) + I((near_bay + 0.01) ^(-2)) +
             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
           data = train)
anova(mdl2e, mdl2) # value 0.7357; should drop

# decide if near_bay should be transformed
mdl2f <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
             I(log(housing_median_age)) + I(log(total_rooms)) +
             I(log(total_bedrooms)) + I(log(population))  +
             I(log(median_income)) + near_bay +
             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
           data = train)
anova(mdl2f) # 0.296095
mdl2g <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
             I(log(housing_median_age)) + I(log(total_rooms)) +
             I(log(total_bedrooms)) + I(log(population)) +
             I(log(median_income)) + I((near_bay + 0.01) ^(-2)) +
             I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
           data = train)
anova(mdl2g) # 0.296095  as transformed
# since it doesn't make a difference, will use original for model simplicity

mdl2h <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(total_bedrooms)) + I(log(population))  +
              I(log(median_income)) +
              I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
            data = train) # near_bay dropped
anova(mdl2h, mdl2f) # 0.04218 shouldn't drop near_bay

# decide if near_ocean should be transformed
mdl2i <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(total_bedrooms)) + I(log(population))  +
              I(log(median_income)) + near_bay +
              near_ocean + I(log(oneh_ocean + 0.01)),
            data = train)
anova(mdl2i)
mdl2j <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(total_bedrooms)) + I(log(population))  +
              I(log(median_income)) + near_bay +
              I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
            data = train)
anova(mdl2j) # same p value; thus use a simplified version
mdl2k <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(total_bedrooms)) + I(log(population))  +
              I(log(median_income)) + near_bay +
               + I(log(oneh_ocean + 0.01)),
            data = train)
anova(mdl2k, mdl2i) # 0.003605 shouldn't drop near_ocean
```

```
mdl2o <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + longitude +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(total_bedrooms)) + I(log(population)) + I(log(households)) +
              I(log(median_income)) + I((near_bay + 0.01) ^(-2)) +
              I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
            data = train)
mdl2p <- lm(I(log(median_house_value)) ~ I(latitude^(-6)) + I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(total_bedrooms)) + I(log(population)) + I(log(households)) +
              median_income + I((near_bay + 0.01) ^(-2)) +
              I((near_ocean + 0.01)^(-2)) + I(log(oneh_ocean + 0.01)),
            data = train)

# check result with four criterias
resultsa <- round(rbind(
  select_criteria(mdl2, n=nrow(train)),
  select_criteria(mdl2f, n=nrow(train)),
  select_criteria(mdl2d, n=nrow(train)),
  select_criteria(mdl2i, n=nrow(train)),
  select_criteria(mdl2h, n=nrow(train)),
  select_criteria(mdl2e, n=nrow(train)),
  select_criteria(mdl2k, n=nrow(train)),
  select_criteria(mdl2o, n=nrow(train)),
  select_criteria(mdl2p, n=nrow(train))
),3)
rownames(resultsa)<-c("1", "2", "3", "4", "5", "6", "7", "8", "9")
# 1:all transformed 2: near_bay original 3: households original 4: near_ocean original
# 5: near_bay dropped 6: households dropped 7: near_ocean dropped 8: longitude original
# 9: median_income original
resultsa
# drop households; near_ocean and near_bay original; longitude and median_income
# transformed

# decide if to use original version of longitude, median_income
mdl2l <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
             I(log(housing_median_age)) + I(log(total_rooms)) +
             I(log(median_income)) + near_bay +
             near_ocean + I(log(oneh_ocean + 0.01)),
           data = train) # reflect result on near_bay, households, and near_ocean
mdl2o <- lm(I(log(median_house_value)) ~ longitude +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(median_income)) + near_bay +
              near_ocean + I(log(oneh_ocean + 0.01)),
            data = train)
mdl2p <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              median_income + near_bay +
              near_ocean + I(log(oneh_ocean + 0.01)),
            data = train)
mdl2q <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(median_income)) + near_bay +
              near_ocean + I(log(oneh_ocean + 0.01)),
             data = train)
resultsb <- round(rbind(
  select_criteria(mdl2l, n=nrow(train)),
  select_criteria(mdl2o, n=nrow(train)),
  select_criteria(mdl2p, n=nrow(train)),
  select_criteria(mdl2q, n=nrow(train))
),3)
rownames(resultsb)<-c("1", "2", "3", "4")
# 1:all transformed 2: original longitude 3. original median_income
# 4: 1/median_income instead of log
resultsb # 1/median income is bad compared to log or original
# result shows that keep longitude and median_income transformed
```

```r
##############################################################################
# preconditions check #
##############################################################################
plot(I(log(train$median_house_value))~fitted(mdl2))
abline(a=0,b=1)
lines(lowess(log(train$median_house_value)~fitted(mdl2)), col="blue") # condition 1 holds
ttrain <- data.frame(train$longitude^(-5),log(train$housing_median_age),
                     log(train$total_rooms), log(train$median_income), train$near_bay,
                     train$near_ocean, log(train$oneh_ocean + 0.01))
pairs(ttrain) # conditions hold

##############################################################################
# assumptions check based on residual plots#
##############################################################################
par(mfrow=c(3,3))
plot(rstandard(mdl2)~fitted(mdl2), xlab="fitted", ylab="Residuals")
for(i in 1:7){
  plot(rstandard(mdl2)~ttrain[,i], xlab=names(ttrain)[i], ylab="Residuals")
}
qqnorm(rstandard(mdl2))
qqline(rstandard(mdl2))
plot(I(log(train$median_house_value))~fitted(mdl2))
abline(a=0,b=1)
lines(lowess(log(train$median_house_value)~fitted(mdl2)), col="blue")
# standardized residue for linearity assumption check
# linearity ok; normality ok

par(mfrow=c(3,3))
plot(mdl2$residuals~fitted(mdl2), xlab="fitted", ylab="Residuals")
for(i in 1:7){
  plot(mdl2$residuals ~ ttrain[,i], xlab=names(ttrain)[i], ylab="Residuals")
}
qqnorm(residuals(mdl2))
qqline(residuals(mdl2))
# regular residue vs predictor to check independent errors and constant variance
# constant variance might be violated; independent error is ok, since the clusters
# are not separated from the other data; to be sure, we will wait for model validation

# use modified residue plots to check constant variance again
par(mfrow=c(3,3))
for(i in 1:7){
  plot(sqrt(abs(rstandard(mdl2))) ~ ttrain[,i], xlab=names(ttrain)[i],
       ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
  m <- lm(sqrt(abs(rstandard(mdl2))) ~ ttrain[,i])
  abline(a = m$coefficients[1], b = m$coefficients[2])
  }
# constant variance assumption is violated for total_rooms and median_income

#
ttrain1 <- data.frame(train$longitude^(-5),log(train$housing_median_age),
                     log(train$total_rooms), log(train$median_income), train$near_bay,
                     train$near_ocean, log(train$oneh_ocean + 0.01))
newtrain <- ttrain1
mdl2r <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms)) +
              I(log(median_income)) + near_bay +
              near_ocean + I(log(oneh_ocean)),
            data = train)
par(mfrow=c(3,3))
for(i in 1:7){
  plot(sqrt(abs(rstandard(mdl2r))) ~ ttrain1[,i], xlab=names(ttrain1)[i],
       ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
  m <- lm(sqrt(abs(rstandard(mdl2r))) ~ ttrain1[,i])
  abline(a = m$coefficients[1], b = m$coefficients[2])
}
# constant variance is much improved
round(select_criteria(mdl2r, n=nrow(train))) # 0.568  6192.757  6193.017  6237.716 worst
summary(mdl2r)
```

```r
mdl2s <- lm(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
              I(log(housing_median_age)) + I(sqrt(total_rooms)) +
              I(sqrt(median_income)) + near_bay +
              near_ocean + I(log(oneh_ocean)),
            data = train)
round(select_criteria(mdl2s, n=nrow(train)),3)
# 0.593    6151.591    6151.851    6196.551 better than mdl2r

par(mfrow=c(3,3))
plot(rstandard(mdl2s)~fitted(mdl2s), xlab="fitted", ylab="Residuals")
for(i in 1:7){
  plot(rstandard(mdl2s)~ttrain[,i], xlab=names(ttrain)[i], ylab="Residuals")
}
qqnorm(rstandard(mdl2s))
qqline(rstandard(mdl2s))
plot(I(log(train$median_house_value))~fitted(mdl2s))
abline(a=0,b=1)
lines(lowess(log(train$median_house_value)~fitted(mdl2s)), col="blue")

par(mfrow=c(3,3))
plot(mdl2s$residuals~fitted(mdl2s), xlab="fitted", ylab="Residuals")
for(i in 1:7){
  plot(mdl2s$residuals ~ ttrain[,i], xlab=names(ttrain)[i], ylab="Residuals")
}
qqnorm(residuals(mdl2s))
qqline(residuals(mdl2s))
par(mfrow=c(3,3))
for(i in 1:7){
  plot(sqrt(abs(rstandard(mdl2s))) ~ ttrain[,i], xlab=names(ttrain)[i],
       ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
  m <- lm(sqrt(abs(rstandard(mdl2s))) ~ ttrain[,i])
  abline(a = m$coefficients[1], b = m$coefficients[2])
}
# mdl2r and mdl2s satisfy assumptions


################################################################################
# variable selection #
################################################################################
# multicolinearity check #
vif(mdl2) # given the threshold is 5, latitude, longitude, total_rooms,
# total_bedrooms, and households don't pass the multicolinearity check;
# correlation of variables with latitude
cors <- NULL
for (i in 1:13){
  cors <- c(cors, cor(train$latitude, train[, (i+1)]))
}
cdf <- data.frame("Correlation" = cors, "Predictors" = pred)
cdf[order(-abs(cdf$Correlation)),] # latitude strongly correlate with longitude
# by -0.91787592, second is -0.42341103   Access to Ocean

# correlation of variables with total_rooms
cors <- NULL
for (i in 1:13){
  cors <- c(cors, cor(train$total_rooms, train[, (i+1)]))
}
cdf <- data.frame("Correlation" = cors, "Predictors" = pred)
cdf[order(-abs(cdf$Correlation)),] # Total Bedrooms: 0.932037674;
# Neighborhood Households: 0.931694831; Neighborhood Population: 0.883482788

# model versions: 1. latitude, no longitude 2. longitude, no latitude
# 3. one of total_rooms, total_bedrooms, population
# 4. two of total_rooms, total_bedrooms, population
t1 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,8)])
vif(t1) # problematic vif with latitude and longitude
t2 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,2,8)])
vif(t2) # latitude < 5 w/o longitude
t3 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,8)])
vif(t3) # longitude < 5 w/o latitude
# need to remove one from total_rooms,total_bedrooms,and population,
```

```r
# since vif is still a problem after removing households
t4 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,5,8)]) # w/o total_rooms
vif(t4) # bedrooms and population still problematic
t5 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,6,8)]) # w/o total_bedrooms
vif(t5) # total_rooms still problematic
t6 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,7,8)]) # w/o population
vif(t6) # total_rooms and total_bedrooms stil problematic
# remove another variable
t7 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,5,6,8)])
# w/o total_rooms and total_bedrooms
vif(t7)
t8 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,5,7,8)])
# w/o total_rooms and population
vif(t8)
t9 <- lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,6,7,8)])
# w/o population and total_bedrooms
vif(t9)
# t7, t8, t9 both pass; same as t2, t3
# models: one of
# 1. latitude, longitude 2. total_rooms, total_bedrooms, population
######################compare six models above################################
results <- round(rbind(
  select_criteria(lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,5,6,8)]), n=nrow(newtrain)),
  select_criteria(lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,5,7,8)]), n=nrow(newtrain)),
  select_criteria(lm(log(train$median_house_value)~., data=newtrain[,-c(1,3,6,7,8)]), n=nrow(newtrain)),
  select_criteria(lm(log(train$median_house_value)~., data=newtrain[,-c(1,2,5,6,8)]), n=nrow(newtrain)),
  select_criteria(lm(log(train$median_house_value)~., data=newtrain[,-c(1,2,5,7,8)]), n=nrow(newtrain)),
  select_criteria(lm(log(train$median_house_value)~., data=newtrain[,-c(1,2,6,7,8)]), n=nrow(newtrain))
),3)
rownames(results)<-c("1", "2", "3", "4", "5", "6")
# 1:longitude&population 2: longitude&total_bedrooms 3: longitude$total_rooms
# 4: latitude&population 5: latitude&total_bedrooms 6:latitude&total_rooms
results # mdl 3 is best with highest adjusted r and lowest other values
# chosen model: longitude, median_age, total_rooms, median_income;
# near_bay, near_ocean, oneh_ocean
# w/o latitude, total_bedrooms, population, inland
# from individual significance check earlier, drop households, use simplified
# near_bay and near_ocean
md16 <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
             I(log(housing_median_age)) + I(log(total_rooms))
           + I(log(median_income)) + near_bay + near_ocean + oneh_ocean,
           data = train)
summary(md16)
anova(md16) # longitude, housing_median_age are insignificant

# recheck transformation
mult <- lm(cbind(train$median_house_value, train$housing_median_age, train$total_rooms,
                 train$median_income, train$near_bay,
                 train$near_ocean, train$oneh_ocean) ~ 1)
pow <- powerTransform(mult, family="bcnPower")
pow # suggest log for median_house_value, median_age, near_bay, near_ocean, oneh_ocean
# suggest 1205 for total_rooms, 1.6 for median_income

# check if dropping longitude
md16a <- lm(I(log(median_house_value)) ~  I(log(housing_median_age)) +
              I(log(total_rooms))  + I(1/median_income) +
              near_bay + near_ocean + oneh_ocean,
            data = train)
anova(md16a,md16) # shouldn't drop

# check if dropping housing_median_age
md16b <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
               I(log(total_rooms))
            + I(1/median_income) +
              near_bay + near_ocean + oneh_ocean,
            data = train)
anova(md16b, md16) # 0.007716 shouldn't drop

# check if dropping near_bay
md16c <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms))
            + I(1/median_income) + near_ocean + oneh_ocean,
            data = train)
anova(md16c, md16) # shouldn't drop

 # stepwise selection using AIC
stepAIC(lm(log(median_house_value) ~ I(longitude^(-5)) +
              I(log(housing_median_age)) + I(log(total_rooms))
            + I(log(median_income)) + near_bay + near_ocean +
              log(oneh_ocean + 0.01), data=train, direction = "both", k=2))
# suggest dropping nothing
```

```r
##################################################################################
# Analysis of Covariance #
##################################################################################
# 12 models
# near_bay
mod1a <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$near_bay==0),])
mod1b <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$near_bay==1),])
mod1a$coefficients # 1.187153e+01     -4.486685e+09
mod1b$coefficients # 4.631502e+01      9.331775e+11
# different intercept and slope --> near_bay main effect and interaction to longitude

mod2a <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$near_bay==0),])
mod2b <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$near_bay==1),])
mod2a$coefficients # 12.02524635                 0.01059813
mod2b$coefficients # 12.220974293               -0.001511231
# similar intercept, slightly different slope

mod3a <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$near_bay==0),])
mod3b <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$near_bay==1),])
mod3a$coefficients # 10.8942132           0.1517004
mod3b$coefficients # 8.6754424          0.4732542
# slightly different intercept, similar slope

mod4a <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$near_bay==0),])
mod4b <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$near_bay==1),])
mod4a$coefficients # 10.9670588          0.8834415
mod4b$coefficients # 11.1663574          0.8770492
# similar intercept and slope

# near_ocean
mod5a <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$near_ocean==0),])
mod5b <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$near_ocean==1),])
mod5a$coefficients # 1.215589e+01      2.780397e+09
mod5b$coefficients # 1.338566e+01      2.549884e+10
# similar intercept, different slope --> near_ocean*longitude

mod6a <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$near_ocean==0),])
mod6b <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$near_ocean==1),])
mod6a$coefficients # 11.9801013                 0.0190077
mod6b$coefficients # 11.99110614                0.09843133
# similar intercept and slope

mod7a <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$near_ocean==0),])
mod7b <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$near_ocean==1),])
mod7a$coefficients # 10.781933           0.164615
mod7b$coefficients # 10.586101           0.224867
# similar intercept and slope

mod8a <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$near_ocean==0),])
mod8b <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$near_ocean==1),])
mod8a$coefficients # 10.9354817         0.8995935
mod8b$coefficients # 11.449247           0.694835
# similar intercept and slope
# consider drop near_ocean, test near_ocean and longitude as interact

# oneh_ocean
mod9a <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$oneh_ocean==0),])
mod9b <- lm(log(median_house_value) ~ I(longitude^(-5)), data=train[which(train$oneh_ocean==1),])
mod9a$coefficients # 1.282667e+01      2.326147e+10
mod9b$coefficients # 1.307992e+01      1.843861e+10
# similar intercept, different slope --> oneh_ocean*longitude

mod10a <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$oneh_ocean==0),])
mod10b <- lm(log(median_house_value) ~ I(log(housing_median_age)), data=train[which(train$oneh_ocean==1),])
mod10a$coefficients # 11.3991593                 0.1497339
mod10b$coefficients # 12.55622368               -0.07935795
# similar intercept, slightly different slope

mod11a <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$oneh_ocean==0),])
mod11b <- lm(log(median_house_value) ~ I(log(total_rooms)), data=train[which(train$oneh_ocean==1),])
mod11a$coefficients # 10.2632509           0.2120744
mod11b$coefficients # 11.2553440           0.1366175
# similar intercept and slope

mod12a <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$oneh_ocean==0),])
mod12b <- lm(log(median_house_value) ~ I(log(median_income)), data=train[which(train$oneh_ocean==1),])
mod12a$coefficients # 10.8223276          0.9436217
mod12b$coefficients # 11.3910752          0.6718017
# similar intercept and slope
# consider drop oneh_ocean, test interaction term between oneh_ocean and longitude
```

```r
mdl7 <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) + I(longitude^(-5)*near_bay)
           + I(longitude^(-5)*near_ocean) + I(longitude^(-5)*oneh_ocean) +
                   I(log(housing_median_age)) + I(log(total_rooms))
                 + I(log(median_income)) + near_bay, data = train)
summary(mdl7)

stepAIC(lm(log(median_house_value) ~ I(longitude^(-5)) + I(longitude^(-5)*near_bay)
           + I(longitude^(-5)*near_ocean) + I(longitude^(-5)*oneh_ocean) +
             I(log(housing_median_age)) + I(log(total_rooms))
           + I(log(median_income)) + near_bay + near_ocean + oneh_ocean,
           data=train, direction = 'both', k = 2))
# suggest dropping longitude
mdl7a <- lm(I(log(median_house_value)) ~ I(longitude^(-5)*near_bay)
           + I(longitude^(-5)*near_ocean) + I(longitude^(-5)*oneh_ocean) +
             I(log(housing_median_age)) + I(log(total_rooms))
           + I(log(median_income)) + near_bay, data = train)
anova(mdl7a, mdl7) # shouldn't drop
vif(mdl7a) # longitude and longitude*near_bay too correlated
mdl7b <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) + I(longitude^(-5)*near_ocean) +
              I(longitude^(-5)*oneh_ocean) + I(log(housing_median_age)) +
              I(log(total_rooms)) + I(log(median_income)) + near_bay, data = train)
vif(mdl7b)
mdl7c <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) + I(longitude^(-5)*near_bay)
            + I(longitude^(-5)*near_ocean) + I(longitude^(-5)*oneh_ocean) +
              I(log(housing_median_age)) + I(log(total_rooms))
            + I(log(median_income)), data = train)
vif(mdl7c)
# either delete longitude*near_bay or near_bay makes VIF test pass
mdl6d <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
              I(longitude^(-5)*near_ocean) + I(log(housing_median_age)) +
              I(log(total_rooms)) + I(log(median_income)) + near_bay + oneh_ocean,
            data = train) # interaction term for near_ocean instead of main effect
mdl6e <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
              I(longitude^(-5)*oneh_ocean) + I(log(housing_median_age)) +
              I(log(total_rooms)) + I(log(median_income)) + near_bay + near_ocean,
            data = train) # interaction term for oneh_ocean instead of main effect
mdl6f <- lm(I(log(median_house_value)) ~ I(longitude^(-5)) +
              I(longitude^(-5)*near_bay) +I(log(housing_median_age)) +
              I(log(total_rooms)) + I(log(median_income)) + near_ocean + oneh_ocean,
           data = train) # interaction term for near_bay instead of main effect
# use selection criteria to compare models
results1 <- round(rbind(
  select_criteria(mdl7b, n=nrow(newtrain)),
  select_criteria(mdl7c, n=nrow(newtrain)),
  select_criteria(mdl6d, n=nrow(newtrain)),
  select_criteria(mdl6e, n=nrow(newtrain)),
  select_criteria(mdl6f, n=nrow(newtrain)),
  select_criteria(mdl6, n=nrow(newtrain))
),3)
rownames(results1)<-c("1", "2", "3", "4", "5", "6")
# 1: main effect term for near_bay 2: interaction term for near_bay 3: all main effects
results1 # mdl6 is best

##############################################################################
# Leverage Points #
##############################################################################
# leverage points
hii1 <- hatvalues(mdl2s)
# hlp1 <- which(hii1 > 4/nrow(train))
# hlp1 # high leverage points
# show that there are some high leverage points
#standardized residue
r1 <- rstandard(mdl2s)
# use the residues of leverage points to check for outliers among leverage points
lr1 <- r1[which(hii1 > 4/nrow(ttrain2))]
lrs1 <- which(lr1 >= 4 | lr1 <= -4) # outlier range for large dataset
lrs1
```

```r
# shows point: 478
ttrain2[478,]
summary(ttrain2)
train[478,]
summary(train)
# 478 very old median_age, high median_income, oneh_ocean
# house value 500001, max of housing value

# test out models again
results1 <- round(rbind(
  select_criteria(mdl7b, n=nrow(newtrain)),
  select_criteria(mdl7c, n=nrow(newtrain)),
  select_criteria(mdl6d, n=nrow(newtrain)),
  select_criteria(mdl6e, n=nrow(newtrain)),
  select_criteria(mdl6f, n=nrow(newtrain)),
  select_criteria(mdl1, n=nrow(newtrain)),
  select_criteria(mdl2, n=nrow(newtrain)),
  select_criteria(mdl3, n=nrow(newtrain)),
  select_criteria(mdl6, n=nrow(newtrain)),
  select_criteria(mdl7, n=nrow(newtrain))
),3)
rownames(results1)<-c("1", "2", "3", "4", "5", "6", "7")
# 1: main effect term for near_bay 2: interaction term for near_bay
# 3: interaction term for near_ocean  4: interaction term for oneh_ocean
# 5: interaction term for near_bay
# mdl6 all main effect; mdl7 interaction term for near_ocean and oneh_ocean, and
# both interaction term and main effect for near_bay
results1 # mdl2 is best, after that mdl7, then mdl6; both mdl2 and 7 have correlation
# issue, so mdl6 is still best


################################################################################
# check residual plot again #

# update mdl8 based on bad leverage
mdl8 <- lm(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
             I(log(housing_median_age)) + I(sqrt(total_rooms)) +
             I(sqrt(median_income)) + near_bay +
             near_ocean + I(1/log(oneh_ocean)), data = train[-478,])
round(select_criteria(mdl8, n=nrow(train)),3)
# mdl8 four measures: .520 -1246.194 -1245.933 -1201.234
# compared to 0.590    6151.303    6151.563    6196.262  from mdl2s; improved!
anova(mdl8)
# longitude and housing_median_age are not significant
best <- regsubsets(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
                     I(log(housing_median_age)) + I(sqrt(total_rooms)) +
                     I(sqrt(median_income)) + near_bay +
                     near_ocean + I(1/log(oneh_ocean)),
                   data = train[-478,], nbest=1)
summary(best)
# let's plot these for easier digestibility
subsets(best, statistic="adjr2") # favor keeping all variables
# check if dropping longitude
mdl8a <- lm(I(sqrt(median_house_value)) ~
              I(log(housing_median_age)) + I(sqrt(total_rooms)) +
              I(sqrt(median_income)) + near_bay +
              near_ocean + I(1/log(oneh_ocean)), data = train[-478,])
anova(mdl8a,mdl8) # 0.03162
mdl8b <- lm(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
               I(sqrt(total_rooms)) +
              I(sqrt(median_income)) + near_bay +
              near_ocean + I(1/log(oneh_ocean)), data = train[-478,])
anova(mdl8b,mdl8) # 0.0004241
# shouldn't drop either variable
```

```
###########################################################################
# preconditions recheck #
###########################################################################
plot(I(sqrt(train[-478,]$median_house_value))~fitted(mdl8))
abline(a=0,b=1)
lines(lowess(sqrt(train[-478,]$median_house_value)~fitted(mdl8)), col="blue")
# condition 1 holds
ttrain2 <- data.frame(train$longitude^(-5),log(train$housing_median_age),
                      sqrt(train$total_rooms), sqrt(train$median_income), train$near_bay,
                      train$near_ocean, 1/log(train$oneh_ocean))
newtrain <- ttrain2[-478,]
pairs(newtrain) # conditions hold

###########################################################################
# assumptions recheck based on residual plots#
###########################################################################
par(mfrow=c(3,4))
plot(rstandard(mdl8)~fitted(mdl8), xlab="fitted", ylab="Residuals")
for(i in 1:7){
  plot(rstandard(mdl8)~newtrain[,i], xlab=names(newtrain)[i], ylab="Residuals")
}
qqnorm(rstandard(mdl8))
qqline(rstandard(mdl8))
plot(I(sqrt(train[-478,]$median_house_value))~fitted(mdl8))
abline(a=0,b=1)
lines(lowess(log(train$median_house_value)~fitted(mdl8)), col="blue")
# standardized residue for linearity assumption check
# linearity ok; normality ok

par(mfrow=c(3,3))
plot(mdl8$residuals~fitted(mdl8), xlab="fitted", ylab="Residuals")
for(i in 1:7){
  plot(mdl8$residuals ~ newtrain[,i], xlab=names(newtrain)[i], ylab="Residuals")
}
qqnorm(residuals(mdl8))
qqline(residuals(mdl8))
# regular residue vs predictor to check independent errors and constant variance
# constant variance might be violated; independent error is ok, since the clusters
# are not separated from the other data; to be sure, we will wait for model validation

# use modified residue plots to check constant variance again
par(mfrow=c(3,3))
for(i in 1:7){
  plot(sqrt(abs(rstandard(mdl8))) ~ newtrain[,i], xlab=names(newtrain)[i],
       ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
  m <- lm(sqrt(abs(rstandard(mdl8))) ~ newtrain[,i])
  abline(a = m$coefficients[1], b = m$coefficients[2])
}
# constant variance assumption is satisfied
vif(mdl8)

###########################################################################
# model validation #
###########################################################################
summary(train)
summary(test)
# looks comparable

# let's fit the same 3 predictor model but using the test data
mdl8_test <- lm(I(sqrt(median_house_value)) ~ I(longitude^(-5)) +
                I(log(housing_median_age)) + I(sqrt(total_rooms)) +
                I(sqrt(median_income)) + near_bay +
                near_ocean + I(log(oneh_ocean)),
              data = test)
summary(mdl8_test)
summary(mdl8)
anova(mdl8_test)
# model performed better for test dataset with adjr2 0.667 instead of 0.5905
vif(mdl8_test)
```

```
# check preconditions
plot(I(sqrt(test$median_house_value))~fitted(mdl8_test))
abline(a=0,b=1)
lines(lowess(sqrt(test$median_house_value)~fitted(mdl8_test)), col="blue")
# condition 1 holds
ttest <- data.frame(test$longitude^(-5),log(test$housing_median_age),
                    sqrt(test$total_rooms), sqrt(test$median_income), test$near_bay,
                    test$near_ocean, log(test$oneh_ocean))
newtest <- ttest
pairs(newtest) # conditions pass

par(mfrow=c(3,4))
plot(rstandard(mdl8_test)~fitted(mdl8_test), xlab="fitted", ylab="Residuals")
for(i in 1:7){
  plot(rstandard(mdl8_test)~newtest[,i], xlab=names(newtest)[i], ylab="Residuals")
}
qqnorm(rstandard(mdl8_test))
qqline(rstandard(mdl8_test))
plot(I(sqrt(train$median_house_value))~fitted(mdl8_test))
abline(a=0,b=1)
lines(lowess(log(train$median_house_value)~fitted(mdl8_test)), col="blue")
# standardized residue for linearity assumption check
# linearity ok; normality ok

par(mfrow=c(3,3))
plot(mdl8_test$residuals~fitted(mdl8_test), xlab="fitted", ylab="Residuals")
for(i in 1:7){
  plot(mdl8_test$residuals ~ newtest[,i], xlab=names(newtest)[i], ylab="Residuals")
}
qqnorm(residuals(mdl8_test))
qqline(residuals(mdl8_test))
# regular residue vs predictor to check independent errors and constant variance
# constant variance might be violated; independent error is ok, since the clusters
# are not separated from the other data; to be sure, we will wait for model validation

# use modified residue plots to check constant variance again
par(mfrow=c(3,3))
for(i in 1:7){
  plot(sqrt(abs(rstandard(mdl8_test))) ~ newtest[,i], xlab=names(newtest)[i],
       ylab="|Standard. Residuals|^0.5", main="|Standard. Residuals|^0.5 vs Predictor",)
  m <- lm(sqrt(abs(rstandard(mdl8_test))) ~ newtest[,i])
  abline(a = m$coefficients[1], b = m$coefficients[2])
}
# constant variance not a horizontal line for several variables

################################################################################
# summary statistics of sample data set #
################################################################################
str(train)
summary(train)
apply(train[,2:14], 2, mean)
apply(train[,2:14], 2, sd)

#histograms
par(mfrow=c(4,4))
for (i in 1:9){
  hist(as.numeric(train[,(i+1)]), breaks=10, main=sprintf(
    "%s of Sample Californian Homes", pred[i]), xlab=pred[i], ylab= "Count")
}
for (i in 10:13){
  hist(as.numeric(train[,(i+1)]), breaks=2, main=sprintf(
    "%s of Sample Californian Homes", pred[i]), xlab=pred[i], ylab = "Count")
}
```

```r
# boxplot 1
par(mfrow=c(3,3))
for (i in 1:9){
  boxplot(as.numeric(train[,(i+1)]), xlab=pred[i], ylab="Value")
}

# scatter plot of longitude and latitude
ocean_pts <- which(train$near_ocean == 1)
bay_pts <- which(train$near_bay == 1)
inland_pts <- which(train$inland == 1)
oneh_pts <- which(train$oneh_ocean == 1)
neither <- which((train$oneh_ocean == 0)&(train$inland == 0))
mix <- sample(train$X, 20, replace = FALSE)

par(mfrow=c(1,1))
library(png)
img <- readPNG('C:/Users/i5/Downloads/STA302 Data Analysis I/Mini project 2/map0.png')
plot(train$longitude, train$latitude, xlab=pred[2], ylab=pred[1],
     main ="Sample Locations", type = "n")
rasterImage(img,xleft=-125, xright=-115, ybottom=32, ytop=42.5)
points(train$longitude[inland_pts],train$latitude[inland_pts], col = "darkgreen",
       pch = 16)
points(train$longitude[oneh_pts],train$latitude[oneh_pts], col = "dodgerblue4", pch = 16)
points(train$longitude[ocean_pts], train$latitude[ocean_pts], col = "blue", pch = 16)
points(train$longitude[bay_pts],train$latitude[bay_pts], col = "cyan", pch = 16)
legend("topright", legend = c("Bay", "Ocean","Drive to Ocean", "Inland"),
       col = c("cyan", "blue","dodgerblue4", "darkgreen"), lty = 1)

#graph in term of price points
low <- which(train$median_house_value <= 110025)
average <- which(110025<train$median_house_value <267525)
high <- which(train$median_house_value >= 267525)
plot(train$longitude, train$latitude, xlab=pred[2], ylab=pred[1],
     main ="Sample Price", type = "n")
rasterImage(img,xleft=-125, xright=-115, ybottom=32, ytop=42.5)
points(train$longitude[low],train$latitude[low], col = "yellow", pch = 16)
points(train$longitude[average],train$latitude[average], col = "tan1", pch = 16)
points(train$longitude[high],train$latitude[high], col = "red3", pch = 16)
legend("topright", legend = c("Low", "Medium", "High"), col = c("yellow", "tan1",
                                                                 "red3"), lty = 1)
# update train based on new dataset

# relationship between each variables
pairs(train[,2:14], lower.panel=NULL)

# relationship with median house values
par(mfrow=c(4,4))
for (i in 1:11){
  plot(train[, (i+1)], train$median_house_value, xlab = pred[i], ylab = pred[9])
}

# correlation table of all relationships
library(xtable)
cors <- NULL
for (i in 1:13){
  cors <- c(cors, cor(train$median_house_value, train[, (i+1)]))
}
cdf <- data.frame("Correlation" = cors, "Predictors" = pred)
cdf[order(-abs(cdf$Correlation)),]
```