**R Project 1 Linear Regression**

Because people take much care of the house price when they live, and this kind of dataset has the advantage of easily access and well trust. In consideration of the availability, size, reasonable number of predictor columns, thus we selected the California Housing Dataset (https://www.kaggle.com/camnugent/california-housing-prices) as our further R linear regression learning project.

**Step 1. Observing the data**

Import the data.

> View(data)

> data<-housing

> colnames(data)

According to the observation of the data, we have found the names of columns in order: "longitude", "latitude", "housing\_median\_age" "total\_rooms", "total\_bedrooms", "population", "households", "median\_income", "median\_house\_value", "ocean\_proximity" with 10 columns. And the data type of the last column is “character”, which is highly different from the rest of columns those are double data types, and we should take care of this feature. Evaluating the column of “ocean\_proximity”, we have observed there are 5 kinds classifications: “INLAND”, ”ISLAND”, ”NEAR BAY”, “NEAR OCEAN”, ”1H OCEAN”, and we will cater to the lowest common denominator and do the splitting in step 2 of this report.

In order to acquire the general idea of this dataset, we use summary function to produce various model fitting functions such as Min, Mean, Q1, Q3, and Max of each column.

> summary(data)

What we need to pay great attention to is “NA” values in data. Counting the number of “NA”, and ‘NA’ totally locates at “total\_bedrooms” that needed to be addressed. We cannot just simply remove the “NA” values, and the way of handling “NA” will be explained in detail in the step 2.

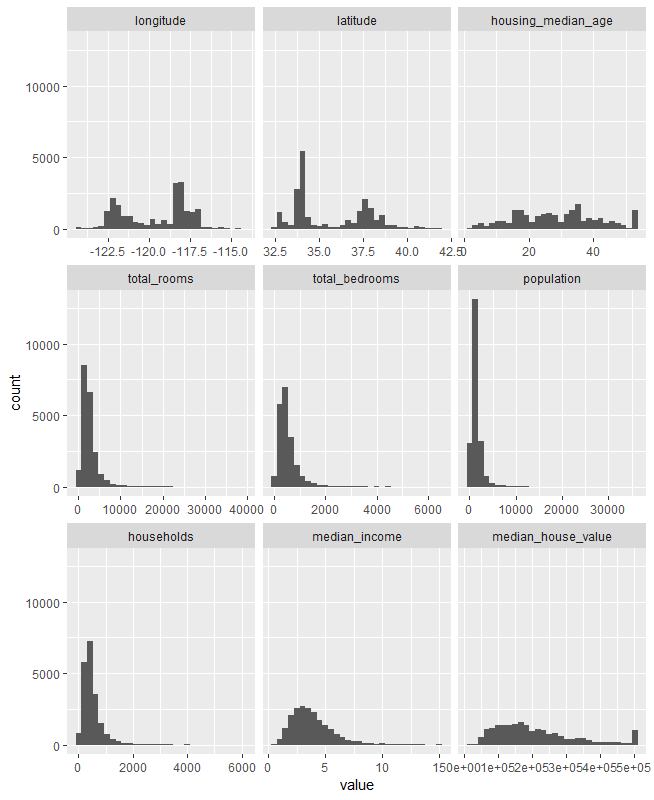
> sum(is.na(data))

[1] 207

Let us make a bird view of the variables.

> library("ggplot2")

> ggplot(data=melt(data),mapping = aes(x=value))+geom\_histogram(bins=30)+facet\_wrap(~variable,scales='free\_x')



What we can extract general information from this figure?

1. We should standardize the scale of this data as some of the variables range from 0-10 while others go up to 50,000.

2. It seems that some variables are dependent on each other that we need to further analyze their correlation between each other.

3. It is so wired that the median house value has some strange cap applied to it causing there to be a blip at the rightmost point on the histogram.

**Step 2. Clean the data**

As we previous said, there are ‘NA’ values in and only in the “total\_bedrooms” column, but the values of relative “total\_rooms” are not ‘NA’. So filling median for “total\_bedrooms” which is a better way with missing values. Utilizing median instead of mean value as it is less influenced by the extreme outliers.

> data$total\_bedrooms[is.na(data$total\_bedrooms)]=median(data$total\_bedrooms,na.rm = TRUE)

And check if there still exists “NA” values.

> sum(is.na(data)) # check the number of NA

[1] 0

Fixing the total columns.

> data$mean\_bedrooms=data$total\_bedrooms/data$households

> data$mean\_rooms=data$total\_rooms/data$households

> drop=c("total\_bedrooms","total\_rooms")

> data=data[,!(names(data) %in% drop)]

> #View(data)

Take categorical into Booleans

Remember that the “ocean\_proximity” column with 5 kinds of classifications. What we desire to do is make a new empty data frame which each category is its own column. Using a loop to populate the new data frame.

> categories=unique(data$ocean\_proximity)

#split the category

> cat\_housing=data.frame(ocean\_proximity=data$ocean\_proximity)

> for(cat in categories)

{

cat\_housing[,cat]=rep(0,times=nrow(cat\_housing))

}

> for(i in 1:length(cat\_housing$ocean\_proximity))

{

cat=as.character(cat\_housing$ocean\_proximity[i])

cat\_housing[,cat][i]=1

}

> cat\_columns=names(cat\_housing)

> keep\_columns=cat\_columns[cat\_columns!='ocean\_proximity']

> cat\_housing=select(cat\_housing,one\_of(keep\_columns))

Furthermore, we desire to obtain all data are numeric and we plan to scale every one of the numerical except for the “median\_value”, which is we will be working on it. And also, we need to get the unscaled data for some data visualization.

> drops=c("ocean\_proximity","median\_house\_value")

> data\_num=data[,!(names(data) %in% drops)]

> View(data\_num)

> unscaled\_housing=cbind(cat\_housing,data\_num,data$median\_house\_value)

> View(unscaled\_housing)

> scaled\_housing\_num=scale(data\_num)

> cleaned\_housing=cbind(cat\_housing,scaled\_housing\_num,median\_house\_value=data$median\_house\_value)

> View(cleaned\_housing)

**Step 3. Linear Regression**

It is important to emphasize that we should make sure that there are more than 10 columns as features to determine, fit and predict the adorable values of “median\_house\_value”.

The critical statistics and information regarding our regression, such as R2, the F statistic, confidence intervals for the coefficients, residuals, the ANOVA table, and so forth can be obtained by the built-in function in R in the following.

> # performing Multiple Linear Regression

> model<-lm(cleaned\_housing$median\_house\_value~cleaned\_housing$`NEAR BAY`+cleaned\_housing$INLAND+cleaned\_housing$`NEAR OCEAN`+cleaned\_housing$ISLAND+cleaned\_housing$longitude+cleaned\_housing$latitude+cleaned\_housing$housing\_median\_age+cleaned\_housing$population+cleaned\_housing$households+cleaned\_housing$median\_income+cleaned\_housing$mean\_bedrooms+cleaned\_housing$mean\_rooms)

# Getting Regression Statistics

> anova(model) # get anova table

> coefficients(model) # model coefficients

> coef(model) # same as coefficients() function

> confint(model) # confidentce intervals for the regression coefficients

> deviance(model) # residual sum of squares

> effects(model) # Vector of orthogonal effects

> fitted(model) # Vector of fitted y values

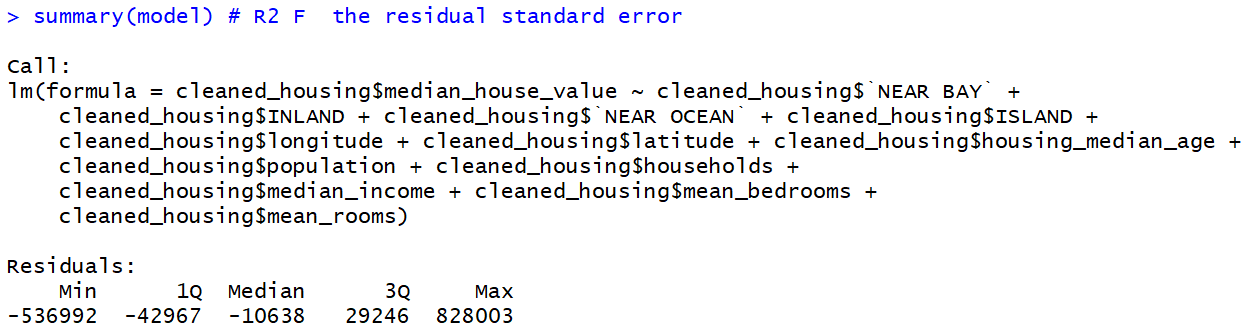
> residuals(model) # Model residules

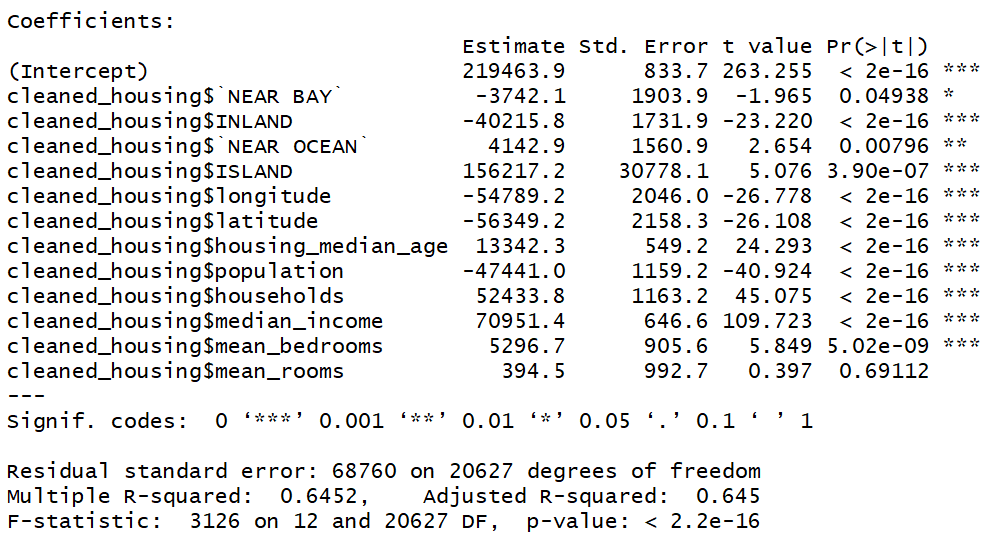
> resid(model) # same as residuals() function

> summary(model) # R2 F the residual standard error

> vcov(model) # variance-covariance matrix of the main parameters

We can extract important information using specialized functions. For example, the summary function shows the estimated coefficients. It shows the critical statistics, such as R2 and the F statistic. It shows an estimate of σ, the standard error of the residuals.





The Min and Max residuals offer a quick way to detect extreme outliers in the data, since extreme outliers produce large residuals.

If the residuals have a nice, bell-shaped distribution, then the first quartile (1Q) and third quartile (3Q) should have about the same magnitude. In this result, the larger magnitude of 1Q versus 3Q (-42967 versus 29246) indicates a slight skew to the left in our data, although the negative median makes the situation less clear-cut.

As we all know that the OLS algorithm is mathematically guaranteed to produce residuals with a mean of zero. Obviously, the median of this model is negative which suggests some skew to the left.

The column labeled Estimate contains the estimated regression coefficients as calculated by ordinary least squares. Theoretically, if a variable’s coefficient is zero then the variable is worthless.

That is the purpose of the t statistics and the p-values, which in the summary are labeled (respectively) t value and Pr(>|t|).

The p-value is a probability. It gauges the likelihood that the coefficient is not significant, so smaller is better. Big is bad because it indicates a high likelihood of insignificance. In this example, the p-value for the “NEAR BAY”, “INLAND”, “ISLAND”, “longitude”, “latitude”, “housing\_median\_age”, “population”, “households”, “median\_income” , “mean\_bedrooms” coefficients are very important, but the p value for “mean\_rooms” is 0.69112 which is over conventional limit of 0.05. Variables with large p-values are candidates for elimination.

Residual standard error: 68760 on 20627 degrees of freedom

This reports the standard error of the residuals (σ)—that is, the sample standard deviation of ε.

R2 (coefficient of determination):

Multiple R-squared: 0.6452, Adjusted R-squared: 0.645.

R2 is a measure of the model’s quality. Bigger is better.

Mathematically, it is the fraction of the variance of y that is explained by the regression model. The remaining variance is not explained by the model, so it must be due to other factors. In this case, the model explains 0.6452 (64.52%) of the variance of y, and the remaining 0.3548 (35.48%) is unexplained. The adjusted value accounts for the number of variables in your model and so is a more realistic assessment of its effectiveness.

F statistic

F-statistic: 3126 on 12 and 20627 DF, p-value is less than 2.2e-16. The F statistic tells you whether the model is significant or insignificant. The model is significant if any of the coefficients are nonzero. It is insignificant if all coefficients are zero (β1 = β2 = … = βn = 0).

Conventionally, a p-value of less than 0.05 indicates that the model is likely significant (one or more βi are nonzero) whereas values exceeding 0.05 indicate that the model is likely not significant.

**Selecting the best regression variables**

Because we have the luxury of many regression variables within limit computation resource, this suggests us to select the best subset of those variables but not all variables.

The step function can perform stepwise regression, either forward or backward. Backward stepwise regression starts with many variables and removes the underperformers:

> full\_model<-lm(cleaned\_housing$median\_house\_value~cleaned\_housing$`NEAR BAY`+cleaned\_housing$INLAND+cleaned\_housing$`NEAR OCEAN`+cleaned\_housing$ISLAND+cleaned\_housing$longitude+cleaned\_housing$latitude+cleaned\_housing$housing\_median\_age+cleaned\_housing$population+cleaned\_housing$households+cleaned\_housing$median\_income+cleaned\_housing$mean\_bedrooms+cleaned\_housing$mean\_rooms)

> summary(full\_model)

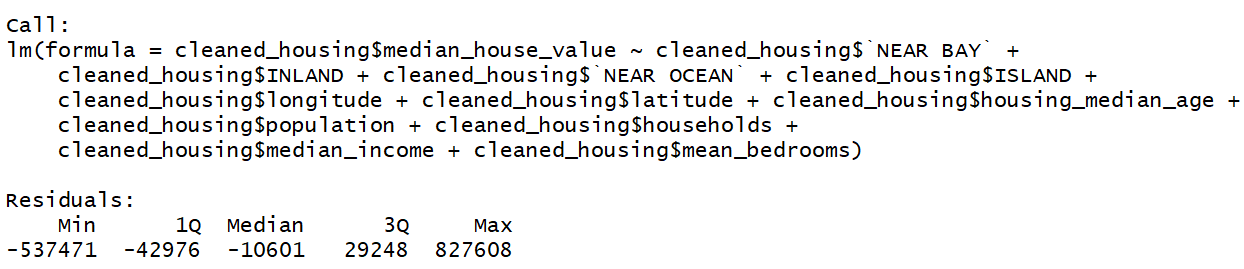
> reduced\_mode<-step(full\_model,direction = "backward")

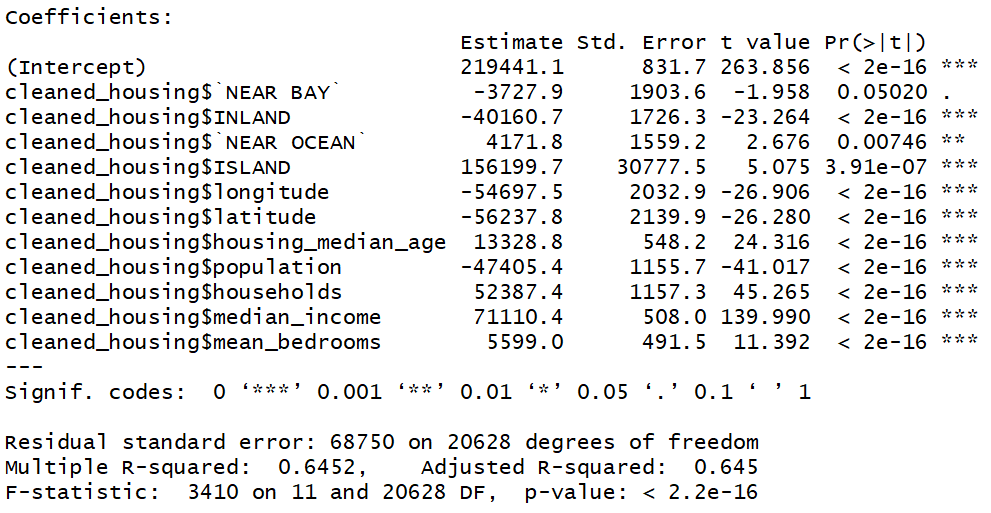
> summary(reduced\_mode)

The step function automates that search. Backward stepwise regression is the easiest approach. Start with a model that includes all the predictors. We call that the full model.

We want to eliminate the insignificant variables, so we use step to incrementally eliminate the underperformers. The result is called the reduced model.

The output from step shows the sequence of models that it explored. In this case, step function only left “NEAR BAY”, “INLAND”, “NEAR OCEAN”, “ISLAND”, “longitude”, “latitude”, “housing\_median\_age”, “population”, “households”, “median\_income”, “mean\_bedrooms” in the final (reduced) model. The summary of the reduced model shows that it contains only significant predictors:



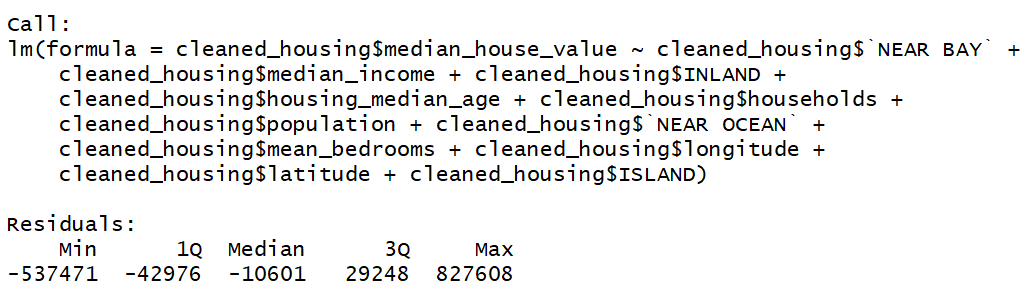


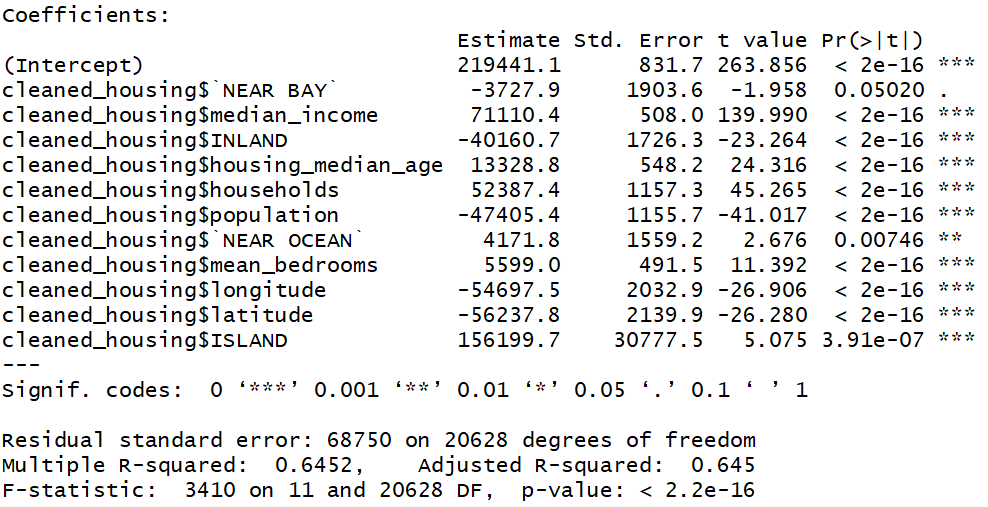
The step-forward algorithm reached the same model as the step-backward model by:

|  |
| --- |
| > min\_model<-lm(cleaned\_housing$median\_house\_value~cleaned\_housing$`NEAR BAY`) |
|  |
| |  | | --- | |  | |

> fwd\_model<-step(min\_model,direction="forward",scope = (~cleaned\_housing$`NEAR BAY`+cleaned\_housing$INLAND+cleaned\_housing$`NEAR OCEAN`+cleaned\_housing$ISLAND+cleaned\_housing$longitude+cleaned\_housing$latitude+cleaned\_housing$housing\_median\_age+cleaned\_housing$population+cleaned\_housing$households+cleaned\_housing$median\_income+cleaned\_housing$mean\_bedrooms+cleaned\_housing$mean\_rooms))

> summary(fwd\_model)

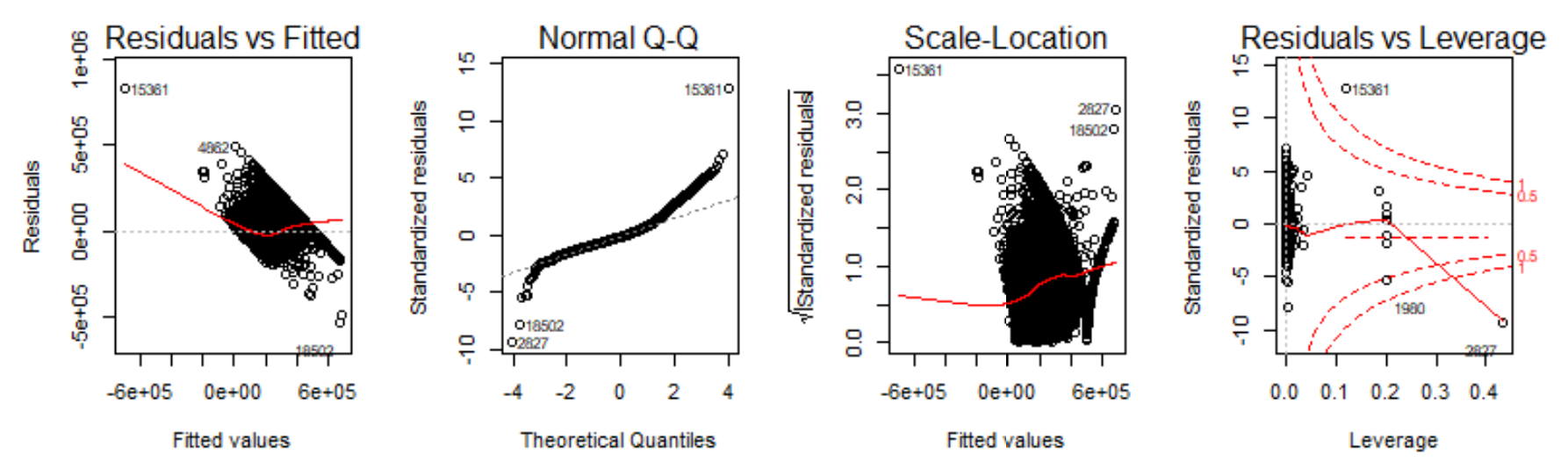




Including “NEAR BAY”, “INLAND”, “NEAR OCEAN”, “ISLAND”, “longitude”, “latitude”, “housing\_median\_age”, “population”, “households”, “median\_income”, “mean\_bedrooms” as the result of the backward regression.

**Forming Confidence Intervals for Regression**

If the model y = β0 + β1(x1)i + β2(x2)i + εi. The confint function returns the confidence intervals for the intercept (β0), the coefficient of x1 (β1), and the coefficient of x2 (β2). By default, confint uses a confidence level of 95%. Use the level parameter to select a different level such as 99.5%.

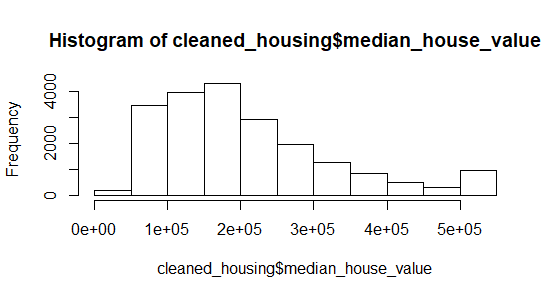


**Identifying influential observations**

You want to identify the observations that are having the most influence on the regression model. This is useful for diagnosing possible problems with the data.

The influence.measures function reports several statistics: DFBETAS, DFFITS, covariance ratio, Cook’s distance, and hat matrix values. If any of these measures indicate that an observation is influential, the function flags that observation with an asterisk (\*) along the righthand side:

And it seems that all the observations are important without any asterisk(\*).



Because the average is much larger than the median, indicating that the distribution of housing costs is right-biased, we can confirm this with a histogram.

Step 4. Data visualization

