Data visualizations and Price prediction of Houses Sold in King County:

About this Dataset:

This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015.

```
housing <- read.csv(file.path("C:", "housing.csv"))#read data
housing
str(housing)
$ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
 $ date         : Factor w/ 372 levels "20140502T000000",..: 165 221 291 221 284 11 57 252 340 306 ...
 $ price        : num  221900 538000 180000 604000 510000 ...
 $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
 $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
 $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
 $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
 $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
 $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
 $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
 $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
 $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
 $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
 $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
 $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
 $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
 $ long         : num  -122 -122 -122 -122 -122 ...
 $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
 $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

summary(housing):

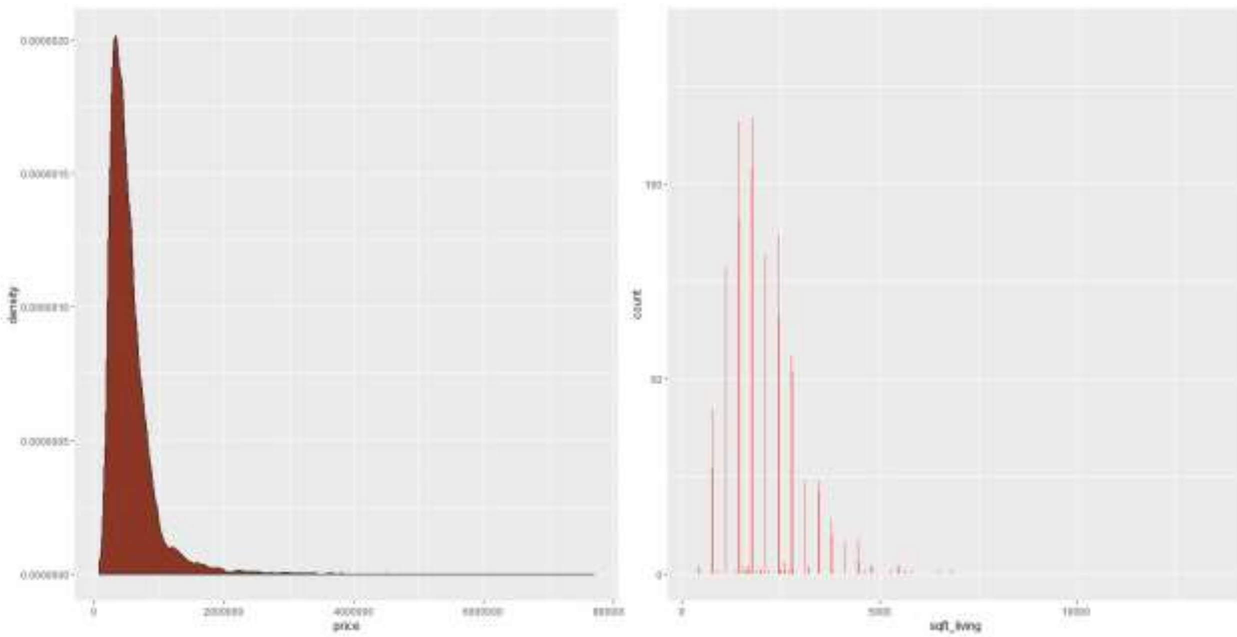
#summary of each variable

```
> summary(housing)#summary of each variable
       id                        date           price            bedroom s
 Min.   :1.000e+06   20140623T000000:  142   Min.   :  75000   Min.   : 0.000
 1st Qu.:2.123e+09   20140625T000000:  131   1st Qu.: 321950   1st Qu.: 3.000
 Median :3.905e+09   20140626T000000:  131   Median : 450000   Median : 3.000
 Mean   :4.580e+09   20140708T000000:  127   Mean   : 540088   Mean   : 3.371
 3rd Qu.:7.309e+09   20150427T000000:  126   3rd Qu.: 645000   3rd Qu.: 4.000
 Max.   :9.900e+09   20150325T000000:  123   Max.   :7700000   Max.   :33.000
                     (Other)        :20833
   bathrooms      sqft_living      sqft_lot           floors
 Min.   :0.000   Min.   :  290   Min.   :    520   Min.   :1.000
 1st Qu.:1.750   1st Qu.: 1427   1st Qu.:   5040   1st Qu.:1.000
 Median :2.250   Median : 1910   Median :   7618   Median :1.500
 Mean   :2.115   Mean   : 2080   Mean   :  15107   Mean   :1.494
 3rd Qu.:2.500   3rd Qu.: 2550   3rd Qu.:  10688   3rd Qu.:2.000
 Max.   :8.000   Max.   :13540   Max.   :1651359   Max.   :3.500

   waterfront            view          condition         grade
 Min.   :0.000000   Min.   :0.0000   Min.   :1.000   Min.   : 1.000
 1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.: 7.000
 Median :0.000000   Median :0.0000   Median :3.000   Median : 7.000
 Mean   :0.007542   Mean   :0.2343   Mean   :3.409   Mean   : 7.657
 3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.: 8.000
 Max.   :1.000000   Max.   :4.0000   Max.   :5.000   Max.   :13.000

   sqft_above    sqft_basement       yr_built     yr_renovated       zipcode
 Min.   : 290   Min.   :   0.0   Min.   :1900   Min.   :   0.0   Min.   :98001
 1st Qu.:1190   1st Qu.:   0.0   1st Qu.:1951   1st Qu.:   0.0   1st Qu.:98033
 Median :1560   Median :   0.0   Median :1975   Median :   0.0   Median :98065
 Mean   :1788   Mean   : 291.5   Mean   :1971   Mean   :  84.4   Mean   :98078
 3rd Qu.:2210   3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:   0.0   3rd Qu.:98118
 Max.   :9410   Max.   :4820.0   Max.   :2015   Max.   :2015.0   Max.   :98199
```

Exploratory Data Analysis:

```
library(ggplot2)
library(tidyverse)
(library(ggplot2))
library(readr)
library(RColorBrewer)
library(dplyr)
library(gridExtra)
library(corrplot)
library(ggmap)
library(tidyverse)
```

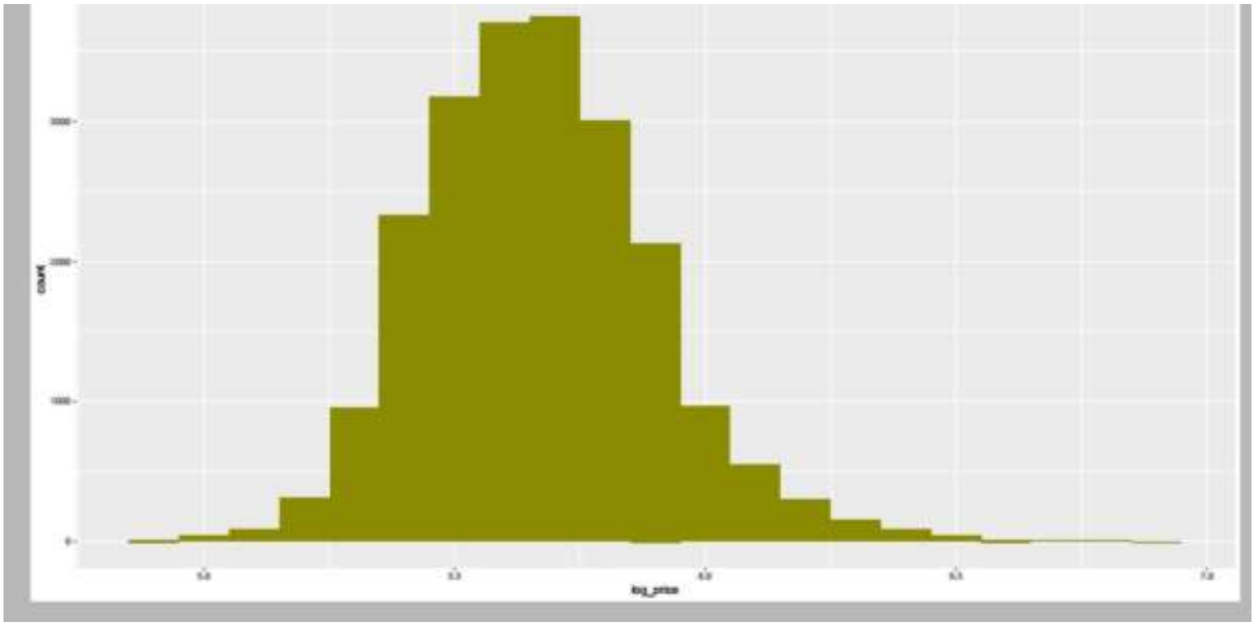
```
g1<-ggplot(housing,aes(x=price))+geom_density(fill="tomato4")
g2<-ggplot(housing,aes(x=sqft_living))+geom_histogram(binwidth=1,fill="tomato")
grid.arrange(g1,g2,nrow=1,ncol=2)
# house price and Qft living are rightly skewed so we do log tranformation and
examine their distributions
```

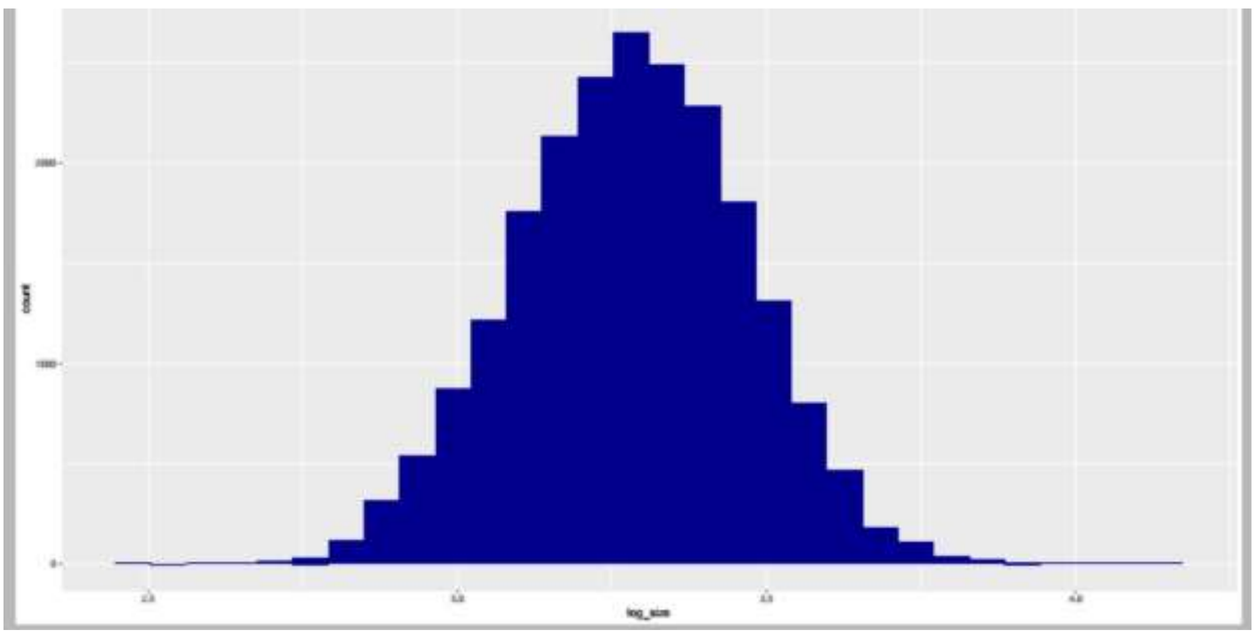


House Price Distribution:

Distribution of house prices & Sqft living was right skewed, so lets apply log() and then plot the distribution

```
housing<-housing %>%
            mutate(log_price=log10(price))
ggplot(housing,aes(x=log_price))+geom_histogram(fill="yellow4",binwidth=0.10)

housing<-housing %>%
            mutate(log_size=log10(sqft_living))
ggplot(housing,aes(x=log_size))+geom_histogram(fill="blue4")
```

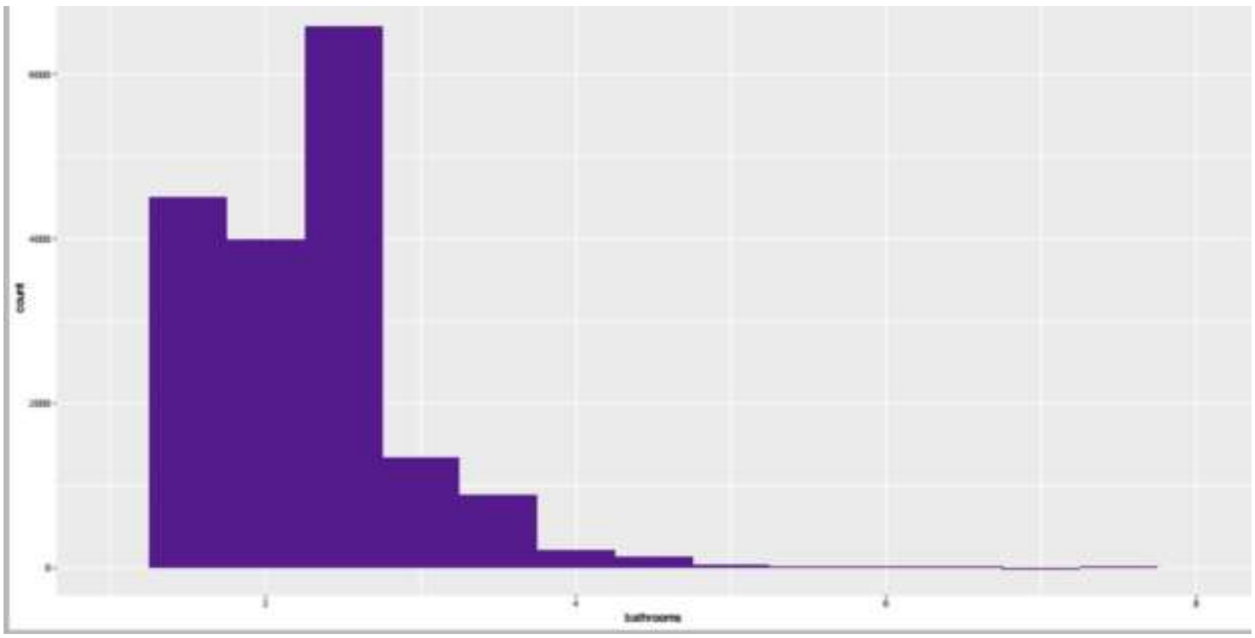




Most of house price lies between 5.4 to 6 million.

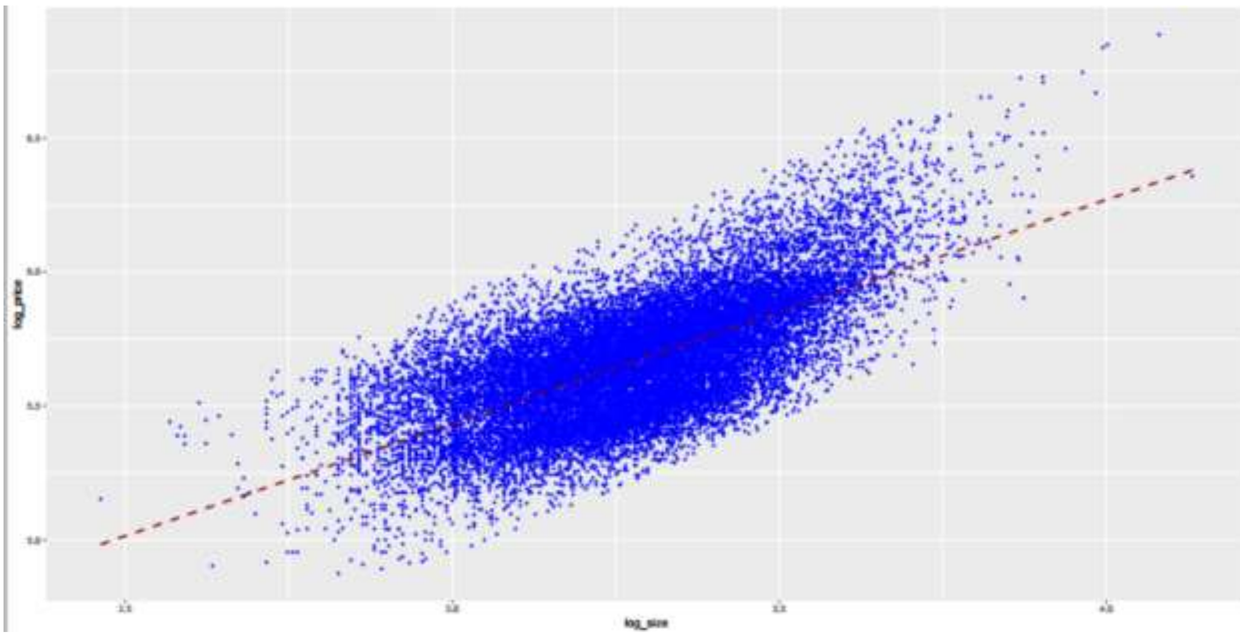
Bathrooms:

```
# examining the variable BATHROOM and plotting a histogram

ggplot(housing,aes(x=bathrooms))+geom_histogram(fill="purple4",binwidth=0.5,size=0.1)+
        scale_x_continuous(limits=c(1,8))
Warning messages:
1: Removed 86 rows containing non-finite values (stat_bin).
2: Removed 2 rows containing missing values (geom_bar).
```



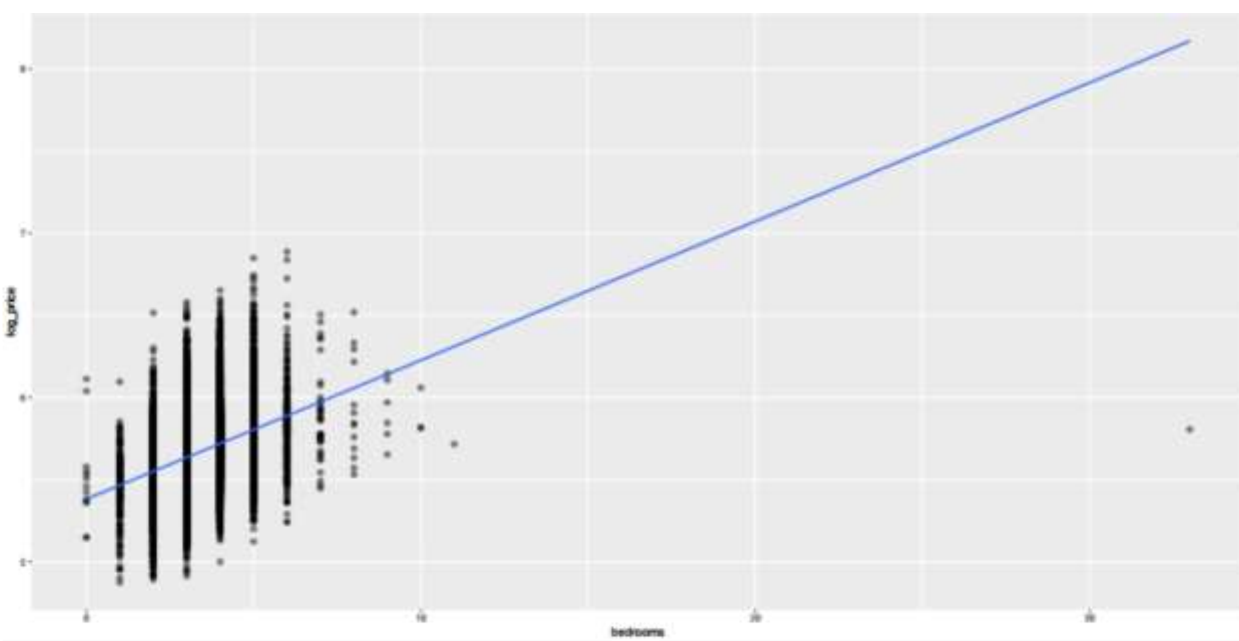
Examining the relationship between log_price and log_size:

```
ggplot(housing, aes(x=log_size,y=log_price))+geom_point(shape=18, color="blue")+
geom_smooth(method=lm, se= FALSE, linetype="dashed", color="darkred")
```



Relationship between bedrooms and log_price:

```
#bedrooms vs log_price
ggplot(housing, aes(x=bedrooms, y= log_price))+geom_point(alpha=0.5,size=2)+
geom_smooth(method="lm",se=F)+
labs("title=Sqft Living vs Price")+theme(legend.position="none")
```



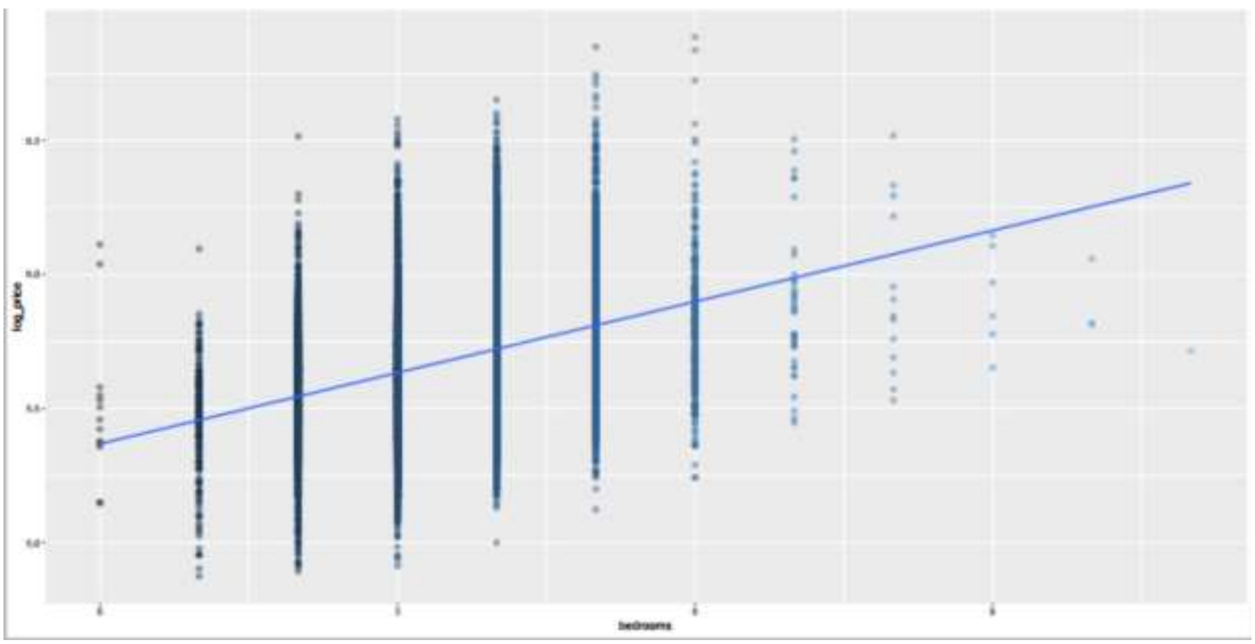
There exists one outlier in the bedrooms variable, which shows house with 33 bedrooms, which may not be the case, so let's remove that row and plot the graph again.

5, 6-bedroom house price seems to high

Treating an outlier in variable bedrooms:

There exists one outlier in the bedrooms variable, which shows house with 33 bedrooms, which may not be the case, so let's remove that row. and plot the graph again. 5, 6-bedroom house price seems to high.

```
# treating one outlier in bedroom variable and removing that row

housing %>% filter(bedrooms<30)%>%
ggplot(aes(x=bedrooms,y=log_price,col=bedrooms))+
geom_point(alpha=0.5,size=2)+
geom_smooth(method="lm",se=F)+
labs("title=Bedrooms vs Price")+theme(legend.position="none")
```



House condition and prices:

table(housing$condition)

1	2	3	4	5
30	172	14031	5679	1701

Number 1 being the worst and 5 being the best condition house and most of the houses are of condition 3 (14031)

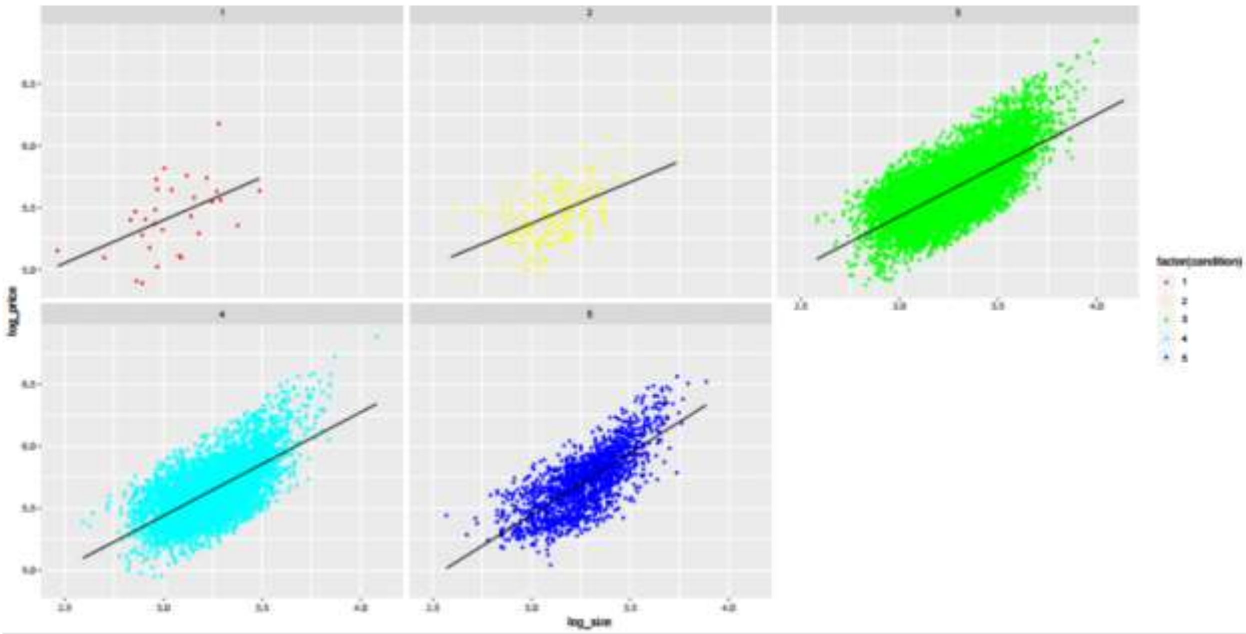
creating a table of relative mean prices of house according to their conditions:

housing
%>%group_by(factor(condition))%>%summarise(mean_price=mean(log_price),sd=sd(log_price),count=n())

Factor condition	Mean_price	sd	count
1	5.42	0.293	30
2	5.45	0.233	172
3	5.76	0.224	14031
4	5.65	0.228	5679
5	5.71	0.244	1704

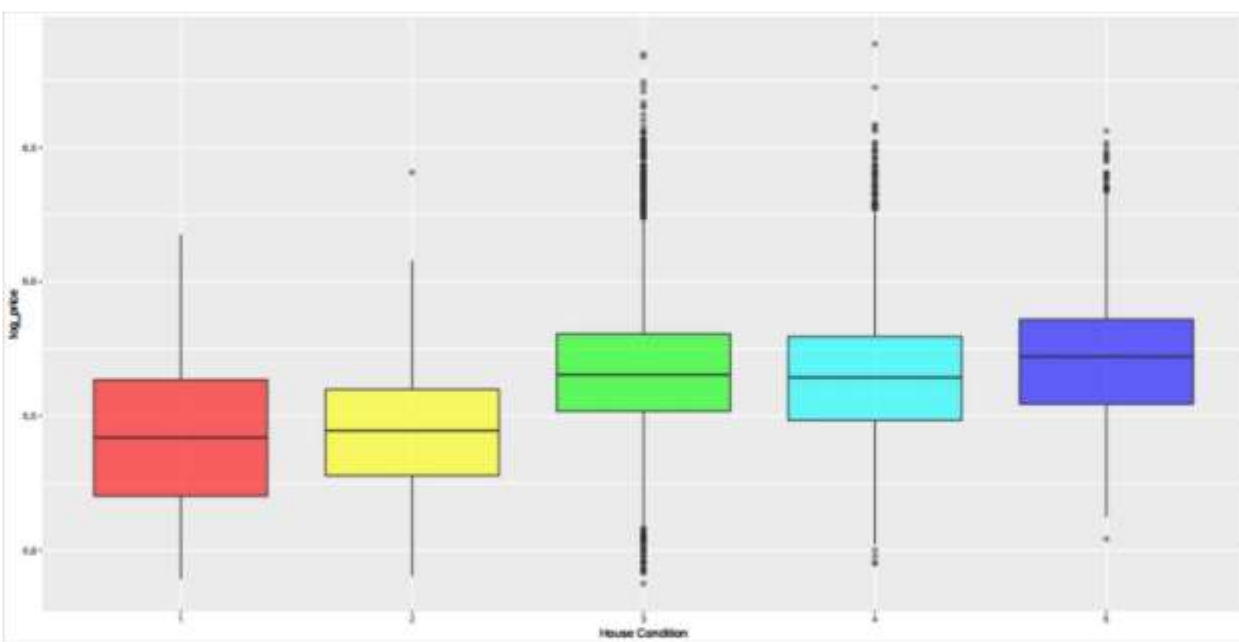
Relationship between sqft_living, price and the condition of the house:

```
#plotting the relationship between log price and log size  accross factors of variable condition(1-5)

options(repr.plot.width=8, repr.plot.height=5)
ggplot(housing,aes(x=log_size,y=log_price,color=factor(condition)))+geom_point(size=0.5)+
geom_smooth(method="lm",se=F,alpha=0.6,size=0.5,color="black")+ scale_color_manual(values =rainbow(n=6))+facet_wrap(~condition)
```



Distribution of house prices according to condition:

```
#Distribution of house prices according to condition

ggplot(housing,aes(factor(condition),log_price,fill=factor(condition)))+
geom_boxplot(alpha=0.6)+scale_fill_manual(values=rainbow(6))+
theme(legend.position="none")+
labs(x="House Condition")
```

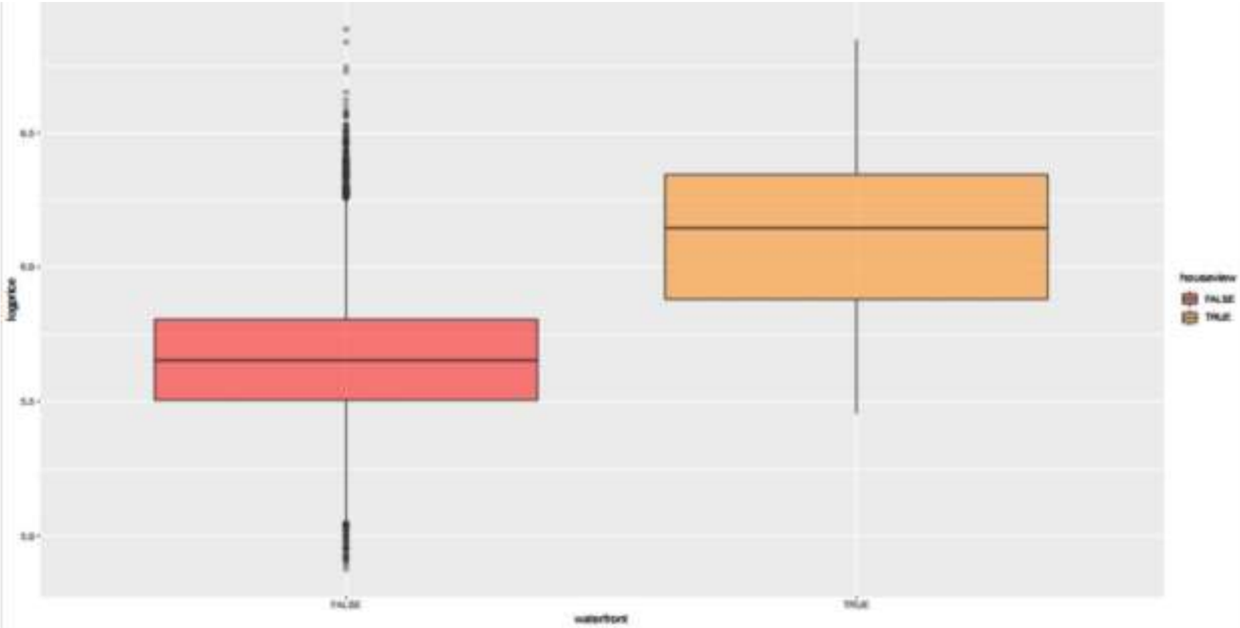


From the above plot its very clear that house prices were high if the condition was good.

Houseview variable creation:

```
#Houseview variable creation
housing$houseview<-ifelse(housing$waterfront==1,TRUE,FALSE)

housing%>%
  select(log_price, houseview) %>%
  glimpse()
#Observations: 21,613
#Variables: 2
#$ log_price <dbl> 5.346157, 5.730782, 5.255273, 5.781037, 5.707570, 6.088136, ...
#$ houseview <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS...
```

```
ggplot(housing, aes(x = houseview, y = log_price,fill=houseview)) +
  geom_boxplot(alpha=0.5) +
  labs(x = "waterfront", y = "logprice")+
scale_fill_manual(values=rainbow(n=12))
```



Houses that have view of waterfront tend to be much expensive than house not having a view. Most of the houses doesn't have water front, see below the price difference between those.

```
#count of number houses having waterfront view and their mean prices


housing %>%
  group_by(houseview) %>%
  summarize(mean_price = mean(log_price), house_count = n())
# houseview mean_price house_count
 # <lgl>          <dbl>       <int>
#1 FALSE           5.66       21450
#2 TRUE            6.12         163
```

Grade vs price:

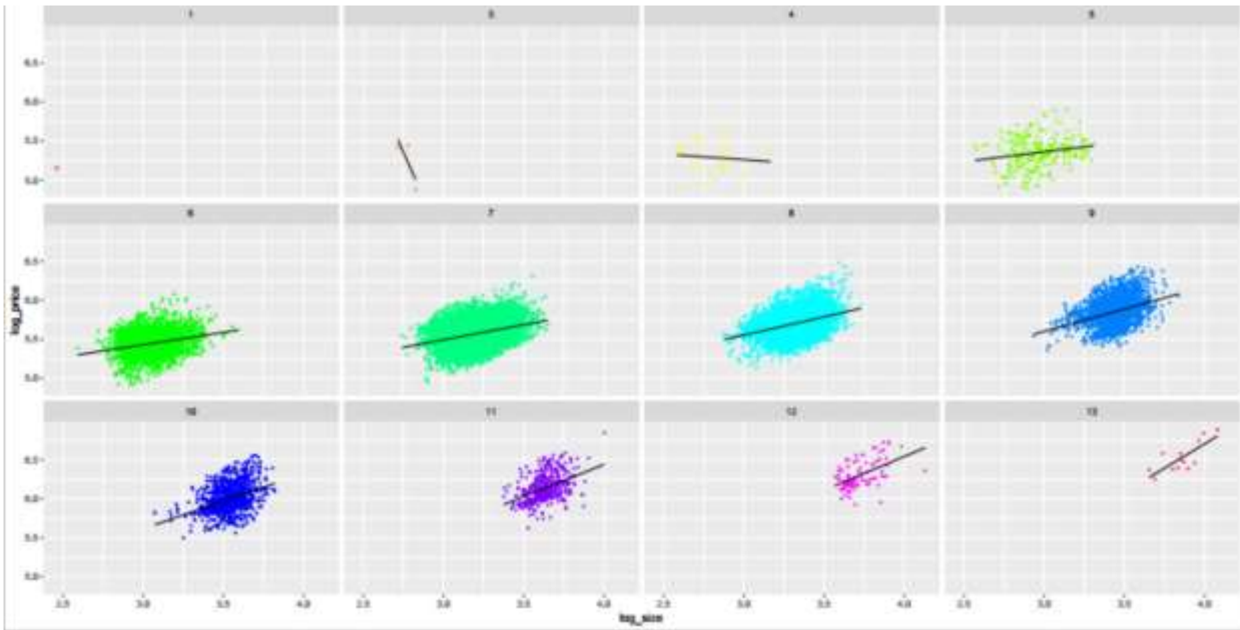
table(housing$grade)

1	3	4	5	6	7	8	9	10	11	12	13
1	3	29	242	2038	8981	6068	2615	1134	399	90	1

comparison of log price and log size accross the factors of variable Grade:

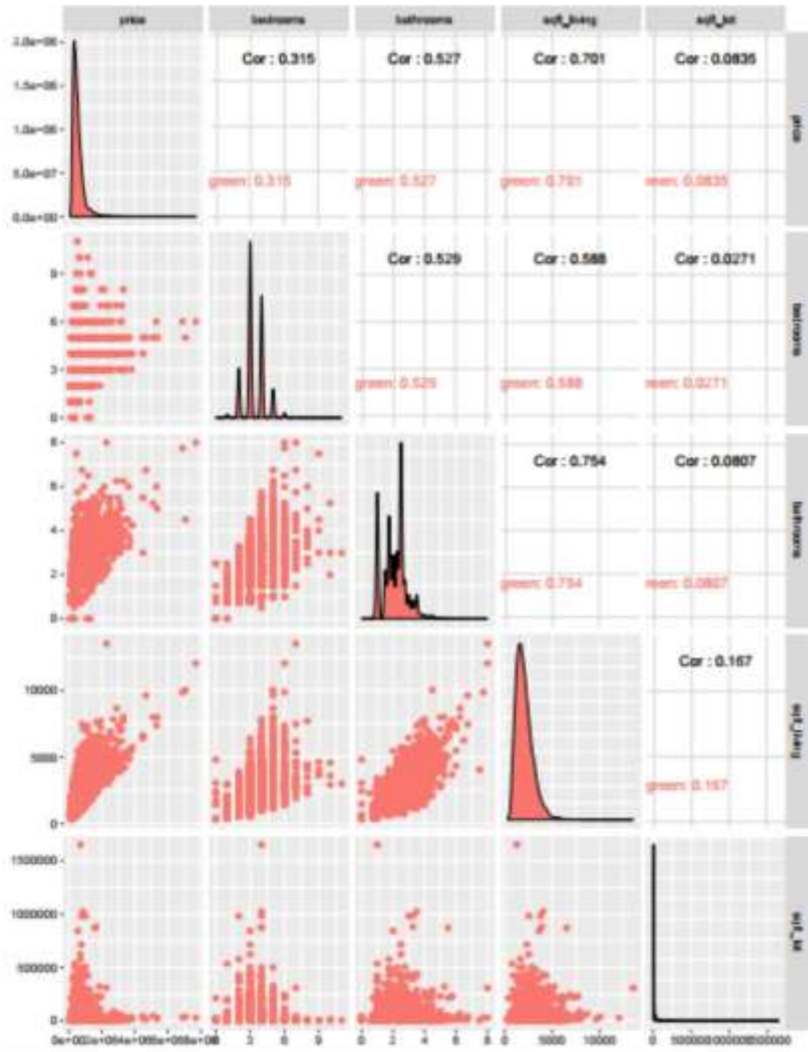
```
#comparison of log price and log size accross the factors of variable Grade
```

```
ggplot(housing,aes(x=log_size,y=log_price,color=factor(grade)))+geom_point(size=0.3)+
geom_smooth(method="lm",se=F,alpha=0.6,size=0.5,color="black")+ scale_color_manual(values =rainbow(n=12))+
facet_wrap(~grade)+
theme(legend.position="none")
```



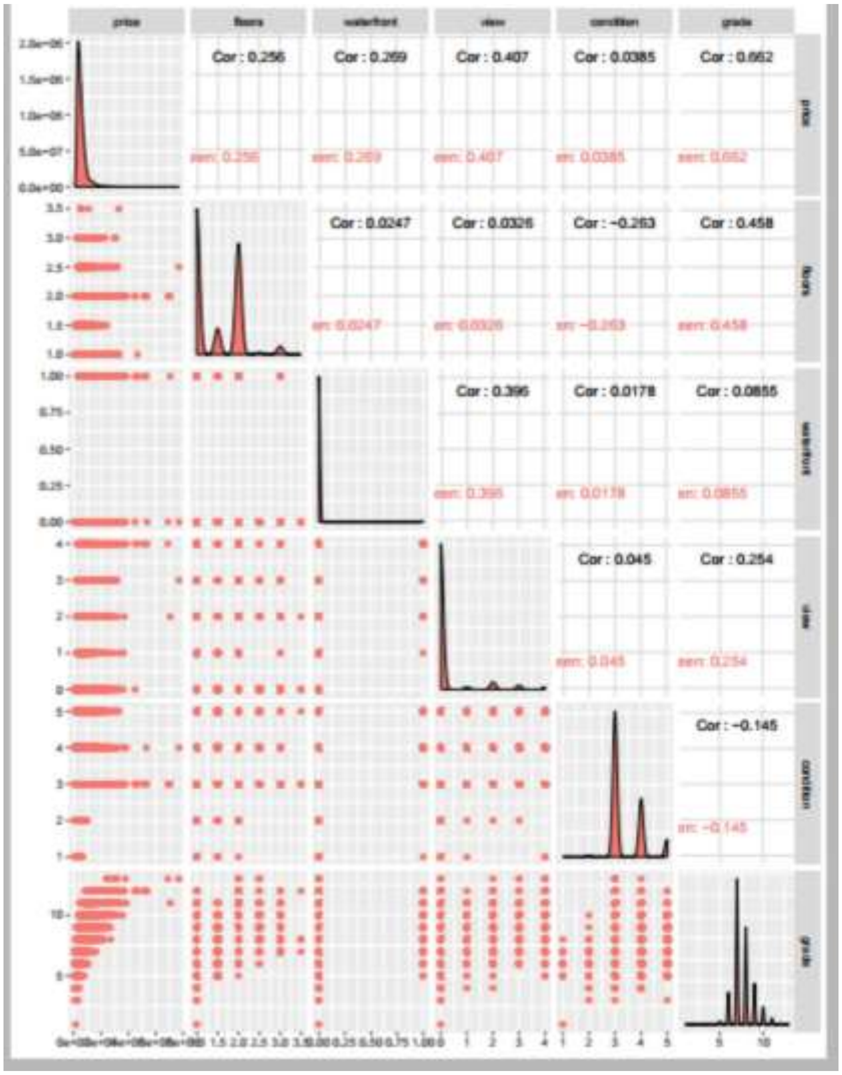
Correlation Matrix Among Variables:

```
plot1<-ggpairs(data=train, columns=3:7,
               mapping = aes(color = "dark green"),
               axisLabels="show")
plot1
```



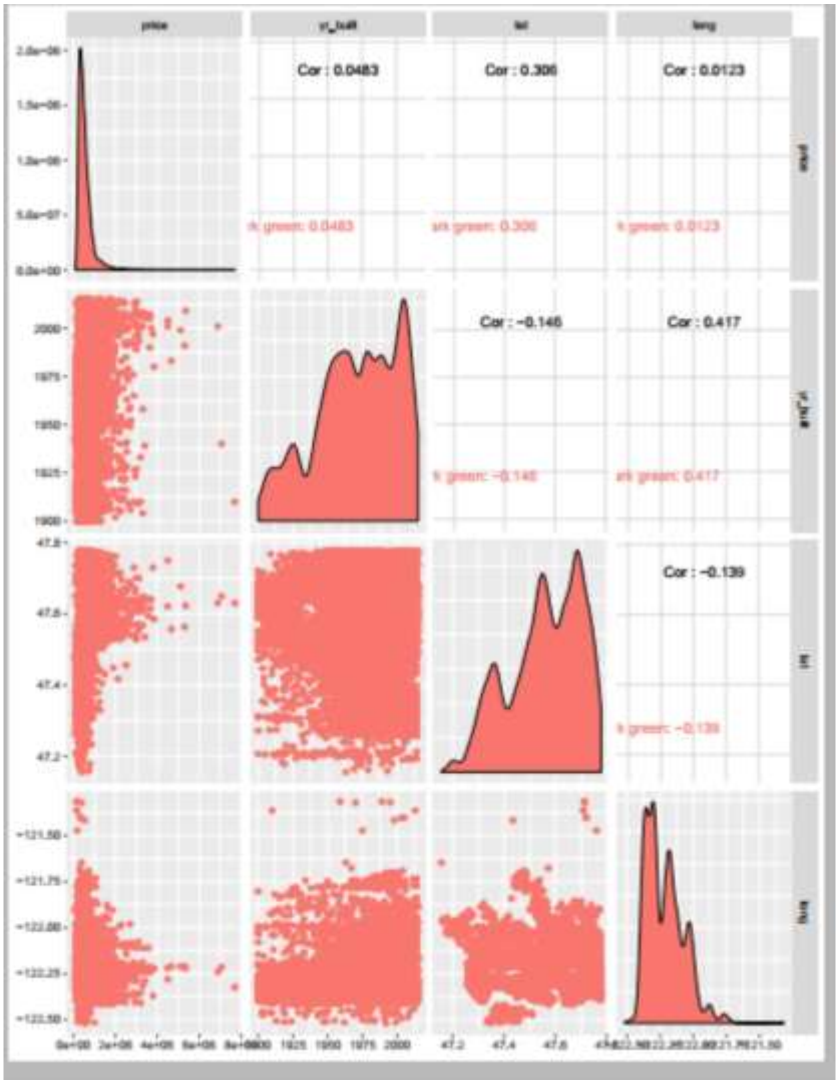
<u>Checking Relationship between price, floors, waterfront, view, condition and grade:</u>

```
## Checking Relationship between price, floors, waterfront, view, condition and grade
plot2<-ggpairs(data=train, columns=c(3,8:12),
               mapping = aes(color = "dark green"),
               axisLabels="show")
plot2
```



Checking Relationship between price, yr built, lat and long:

```
## Checking Relationship between price, yr built, lat and long
plot3=ggpairs(data=train, columns=c(3,15,18,19),
              mapping = aes(color = "dark green"),
              axisLabels="show")
plot3
```



Splitting the data into train and test data sets with 70/30 ratio:

```
  # split the data set into 70/30 ratio
set.seed(1234)
id<-sample(2,nrow(housing),prob=c(0.70,0.30),replace=TRUE)
train<-housing[id==1,]
test<-housing[id==2,]
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':	15127 obs. of  23 variables:
 $ id           : chr  "7129300520" "6414100192" "5631500400" "2487200875" ...
 $ date         : POSIXct, format: "2014-10-13" "2014-12-09" ...
 $ price        : num  221900 538000 180000 604000 1225000 ...
 $ bedrooms     : int  3 3 2 4 4 3 3 3 3 3 ...
 $ bathrooms    : num  1 2.25 1 3 4.5 2.25 1.5 1 2.5 2.5 ...
 $ sqft_living  : int  1180 2570 770 1960 5420 1715 1060 1780 1890 3560 ...
 $ sqft_lot     : int  5650 7242 10000 5000 101930 6819 9711 7470 6560 9796 ...
 $ floors       : num  1 2 1 1 1 2 1 1 2 1 ...
 $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
 $ grade        : int  7 7 6 7 11 7 7 7 7 8 ...
 $ sqft_above   : int  1180 2170 770 1050 3890 1715 1060 1050 1890 1860 ...
 $ sqft_basement: int  0 400 0 910 1530 0 0 730 0 1700 ...
 $ yr_built     : int  1955 1951 1933 1965 2001 1995 1963 1960 2003 1965 ...
 $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
 $ zipcode      : int  98178 98125 98028 98136 98053 98003 98198 98146 98038 98007 ...
 $ lat          : num  47.5 47.7 47.7 47.5 47.7 ...
 $ long         : num  -122 -122 -122 -122 -122 ...
 $ sqft_living15: int  1340 1690 2720 1360 4760 2238 1650 1780 2390 2210 ...
 $ sqft_lot15   : int  5650 7639 8062 5000 101930 6819 9711 8113 7570 8925 ...
 $ log_price    : num  5.35 5.73 5.26 5.78 6.09 ...
 $ log_size     : num  3.07 3.41 2.89 3.29 3.73 ...
```

Simple Regression Model:

Let's build a simple regression model with the variable having highest correlation with Price which is sqft_living.

```
Model2 <- lm(data=train,log(price)~log(sqft_living))
summary(Model2)
```

By looking at the regression output, we can see the values of Rsq and adjusted Rsq. The values indicate that the model is not a perfect fit which is normally not an easy task.

```
Call:
lm(formula = log(price) ~ log(sqft_living), data = train)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.10740 -0.29196  0.01142  0.25654  1.33632 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      6.758687   0.056481   119.7   <2e-16 ***
log(sqft_living) 0.832974   0.007469   111.5   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3887 on 15125 degrees of freedom
Multiple R-squared:  0.4512,	Adjusted R-squared:  0.4512 
F-statistic: 1.244e+04 on 1 and 15125 DF,  p-value: < 2.2e-16
```

Multiple regression Model with two independent variables:

Now let's build a multiple regression model by including another independent variable Bedrooms that is associated with the Price variable.

```
model_price_2 <- lm(log_price ~ log_size + bedrooms,
                    data = housing)
summary(model_price_2)
##Coefficients:
             Estimate Std. Error t value            Pr(>|t|)    
(Intercept)  2.714600   0.022945  118.31 <0.0000000000000002 ***
log_size     0.931302   0.007871  118.32 <0.0000000000000002 ***
bedrooms    -0.030202   0.001561  -19.34 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1673 on 21610 degrees of freedom
Multiple R-squared:  0.4648,	Adjusted R-squared:  0.4647 
F-statistic:  9383 on 2 and 21610 DF,  p-value: < 0.00000000000000022
```

The above regression output shows that the value of adjusted Rsq is slightly higher than the simple regression model but not significantly higher. The Estimate of Bedrooms -0.030 shows that for every unit increase in bedrooms the Price will decline by 0.030 units.

Multiple regression Model 2:

In the previous section I used a simple linear regression and found a poor fit. In order to improve this model, I am planning to add more features, but we should be careful about the overfit which can be seen by the difference between the training and test evaluation metrics. When we have more than one feature in a linear regression, it is defined as multiple regression.

Another important thing is correlation, if there is very high correlation between two features, keeping both of them is not a good idea most of the time. For instance, sqt_above and sqt_living is highly correlated. This can be estimated when you look at the definitions at the dataset and check to be sure by looking at the correlation matrix.

Now, let's include few more variables that have significant association with the variable Price. This model includes Sqft_living, Bedrooms, Bathrooms, View and Grade.

```
> Model3 <- lm(log(price)~log(sqft_living)+bedrooms+bathrooms+view+grade,data=train)
> summary(Model3)

Call:
lm(formula = log(price) ~ log(sqft_living) + bedrooms + bathrooms +
    view + grade, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.14565 -0.24897  0.00393  0.22799  1.31664

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       8.419044   0.076548 109.984  < 2e-16 ***
log(sqft_living)  0.428788   0.013227  32.418  < 2e-16 ***
bedrooms         -0.020357   0.004116  -4.946 7.68e-07 ***
bathrooms        -0.003404   0.005786  -0.588    0.556
view              0.112377   0.003768  29.823  < 2e-16 ***
grade             0.188200   0.003790  49.658  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3444 on 15121 degrees of freedom
Multiple R-squared:  0.5695,	Adjusted R-squared:  0.5693
F-statistic:  4000 on 5 and 15121 DF,  p-value: < 2.2e-16
```

The value of adjusted Rsq is higher than the previous models so this model might be better.

Final Model:

Now, lets include the variables ***sqft living, sqft lot, bedrroms,bathrooms,floors,waterfront,view, grade, yr buit, zipcode***.

```
model_final<-lm(log(price)~log(sqft_living)+bedrooms+bathrooms+sqft_lot+floors+waterfront+view+grade+yr_built+zipcode,data=train)
summary(model_final)
```

Output:

```
Call:
lm(formula = log(price) ~ log(sqft_living) + bedrooms + bathrooms +
    sqft_lot + floors + waterfront + view + grade + yr_built +
    zipcode, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.30741 -0.21023  0.01527  0.20905  1.40234

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -2.056e+00  5.219e+00  -0.394    0.694
log(sqft_living)  3.847e-01  1.223e-02  31.453  < 2e-16 ***
bedrooms         -3.712e-02  3.786e-03  -9.803  < 2e-16 ***
bathrooms         8.840e-02  5.811e-03  15.213  < 2e-16 ***
sqft_lot          3.117e-08  6.367e-08   0.490    0.624
floors            6.744e-02  5.933e-03  11.366  < 2e-16 ***
waterfront        3.242e-01  3.197e-02  10.142  < 2e-16 ***
view              5.788e-02  3.818e-03  15.157  < 2e-16 ***
grade             2.250e-01  3.588e-03  62.703  < 2e-16 ***
yr_built         -5.904e-03  1.154e-04 -51.154  < 2e-16 ***
zipcode           2.236e-04  5.237e-05   4.270 1.96e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.313 on 15116 degrees of freedom
Multiple R-squared:  0.6443,    Adjusted R-squared:  0.6441
F-statistic:  2739 on 10 and 15116 DF,  p-value: < 2.2e-16
```

Examining the Results of Final Model:

By looking at the output, It is clear that the adjusted R sq is the highest among all the models created. This model might be a Better choice among all.

Predicted prices from test data:

Checking the predicted prices from selected model:

```
pricehat_finalmodel <- round(exp(predict(model_final,newdata=test)),0)
output <- (cbind('id'=test$id, 'original price'= test$price, 'predicted price'=pricehat_finalmodel))
write.csv(output,file='test data originalpricevspredicted.csv', row.names=F)
```

Plotting predicted values vs Actual values:

```
ggplot(test,aes(x=price,y=price_hat_Model3))+geom_point()+geom_abline(color="blue")
```