UNIVERSITY OF PADUA

Department of Mathematics "Tullio Levi-Civita"

Master's Degree in Data Science

Statistical learning project (Mod B)

House Sales in King County dataset analysis

Students

Bianca Andreea Ciuche, Matteo Reddavide, Arghavan Shafiee

10 September - Academic Year 2020/2021

Contents

1 Objectives of the study

In this study, our goal is to identify the most important variables in determining the price of houses, and to model data based on these variables in order to predict the selling price of houses in the coming years. We start with a simple linear regression model and then we will use multiple and polynomial regressions to find a proper model to predict the price.

2 Preparation of the Dataset

2.1 Data Collection

This dataset consists of sale prices of 21611 houses sold between May 2014 and May 2015 in King County, Washington, USA, which includes the greater Seattle metropolitan area. Added to the price, we have information on 18 house features, Date of sale, and Id of sale. So, the dataset has 21611 rows and 21 columns.

In the code, as first steps we started importing some libraries and the dataset.

```
library(lubridate)
library(GGally)
library(ggplot2)
library(viridis)
library(leaps)
library(leaflet)
library(sf)
library(knitr)
library(kableExtra)
library(corrplot)
library(boot)
library(Metrics)
```

```
#### IMPORTATION ####
house = 'kc_house_data.csv'
kc_housing <- read.csv(house)
attach(kc_housing)
```

We made a first inspection on the dataset.

```
str(kc_housing)
```

```
## 'data.frame':    21611 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ 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 ...
```

Next, there is a description of variables of the dataset:

- *Id* : A unique identifier for each house sold;
- *Date* : Date column, indicating when the house was sold;
- *Price* : Price of each home sold;
- *Bedrooms* : Number of bedrooms in a house;
- *Bathrooms* : Number of bathrooms in a house.
  - 1 stands for a "full bathroom" containing bathtub, shower, toilet, and sink,
  - 0.75 stands for a bathroom with toilet, sink, and shower (or bathtub),
  - 0.5 stands for one containing just toilet and sink,
  - 0.25 stands for a bathroom that has either a sink, a shower, toilet or a bathtub.

  The total number will be the sum of these numbers for each house;

- *Sqft_living* : Square footage of the house interior living space;
- *Sqft_lot* : Square footage of the land space;
- *Floors* : Total number of floors;
- *Waterfront* : House which has view to a waterfront;
- *View* : An index from 0 to 4 of how good the view of the property was;
- *Condition* : How good the condition of house is (Overall). 1 indicates worn out property and 5 excellent;
- *Grade* : Overall grade given to the housing unit, based on King County grading system. 1 is assigned to a poor property, 13 for an excellent one;
- *Sqft_above* : Square footage of house apart from basement;
- *Sqft_basement* : Square footage of the basement;
- *Yr_built* : The year the house was initially built;
- *Yr_renovated* : The year of the house's last renovation;
- *Zipcode* : What zipcode area the house is in;
- *Lat* : Latitude coordinate;
- *Long* : Longitude coordinate;
- *Sqft_living15* : Average size of interior housing living space for the nearest 15 neighbors, in square feet;
- *Sqft_lot15* : Average size of land lots for the nearest 15 neighbors, in square feet;

2.2 Preprocessing

For the preprocessing part, we first analyzed the dimension of the dataset and checked for the presence of null values. There are no null values.

```
# dimension of the dataset
dim(kc_housing)
```

```
## [1] 21611    21
```

```
# check of null values
sum(is.na(kc_housing))
```

```
## [1] 0
```

After we inspect the response variable *price* and make a log transformation.

```
# check on the response variable
summary(kc_housing$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   75000  321725  450000  539799  645000 7700000
```

```
# logarithmic transformation of the response variable price
kc_housing$ log10_price = log10(price)
```

We converted the variable *date* as datetype, then extracted and added two new columns: Month and Year, and we converted these last two to factors.

```
# Convert date
kc_housing$date<-(substr(kc_housing$date, 1, 8))
kc_housing$date<- ymd(kc_housing$date)

# Month
kc_housing$month_sold<-format((as.POSIXlt( c(kc_housing$date), format="%d/%m/%Y")),"%m")
kc_housing$month_sold <- factor(kc_housing$month, labels = month.abb)

# Year
kc_housing$year_sold<-format((as.POSIXlt( c(kc_housing$date), format="%d/%m/%Y")),"%Y")
kc_housing$year_sold <- factor(kc_housing$year_sold)
```

There are some illogical rows that we excluded:

- 10 rows with zero values in Bathroom column.
- 1 row with 33 bedrooms in 1620 square feet with 1.75 bathrooms.

```
# Removing some values
kc_housing <- subset(kc_housing, subset=(bathrooms!=0))
kc_housing <- subset(kc_housing, subset=(bedrooms!=33))
```

Then, we converted all the categorical variables to factors.

```
# convert bedrooms  to a factor
kc_housing$bedrooms = as.factor(kc_housing$bedrooms)

# convert bathrooms to a factor
kc_housing$bathrooms = as.factor(kc_housing$bathrooms)
```

```r
# convert floors to a factor
kc_housing$floors = as.factor(kc_housing$floors)

# convert waterfront to a factor
kc_housing$waterfront= as.factor(as.logical(kc_housing$waterfront))

# convert view to a factor
kc_housing$view = as.factor(kc_housing$view)

# convert condition to a factor
kc_housing$condition = as.factor(kc_housing$condition)

# convert grade to a factor
kc_housing$grade = as.factor(kc_housing$grade)

# convert zip code to a factor
kc_housing$zipcode = as.factor(kc_housing$zipcode)
```

Moreover, we added two new columns which indicate whether the house has been renovated at some point and whether the house has a basement or not.

```r
#Add a new column renovated indicating whether a house was renovated at some point
kc_housing$renovated <- as.factor(kc_housing$yr_renovated != 0)

#Add a new column has_basement indicating whether a house has a basement
kc_housing$has_basement = as.factor(kc_housing$sqft_basement!=0)
```

Since the variable *id* does not provide any useful information we removed it.

```r
# Drop the id column which provides no information about the house prices
kc_housing$id = NULL
```

In the end we have a dataset with 21600 instances 25 variables.

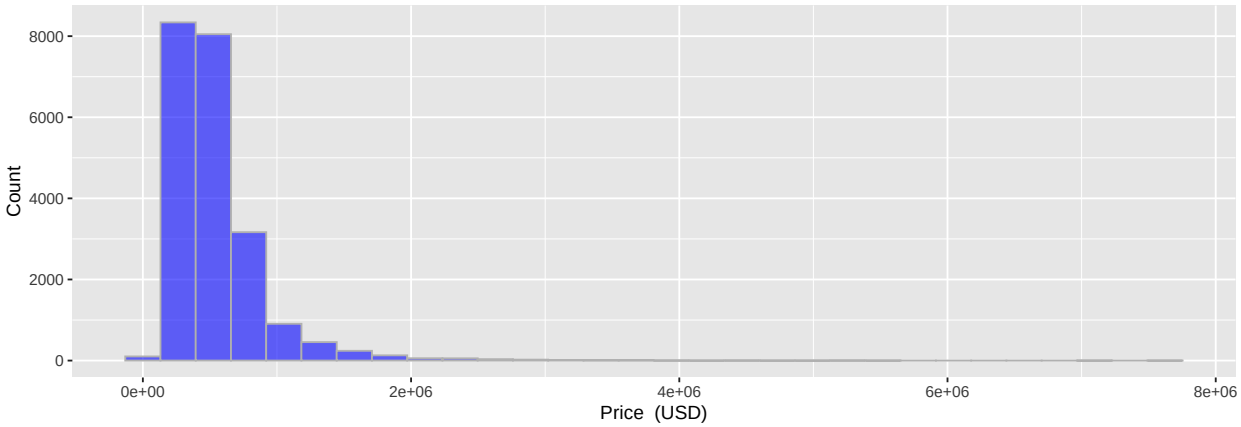
```r
dim(kc_housing)
```

```
## [1] 21600    25
```

2.3 Exploratory and Data Analysis

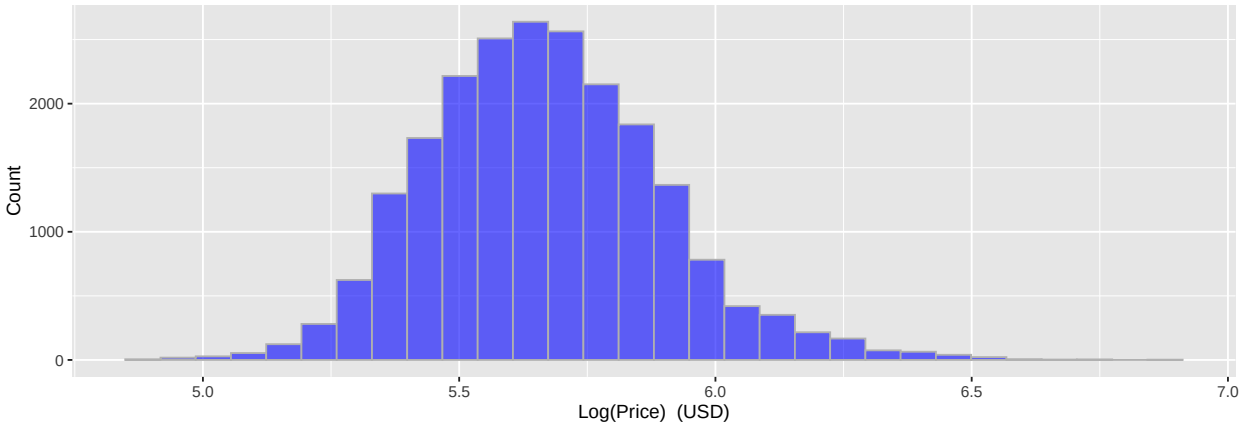
We start with plotting the distribution of some main variables:

```r
ggplot(kc_housing, aes(x= price)) +
  geom_histogram(fill="blue", color="grey", alpha=0.6) +
  labs(x ="Price  (USD)", y="Count")
```



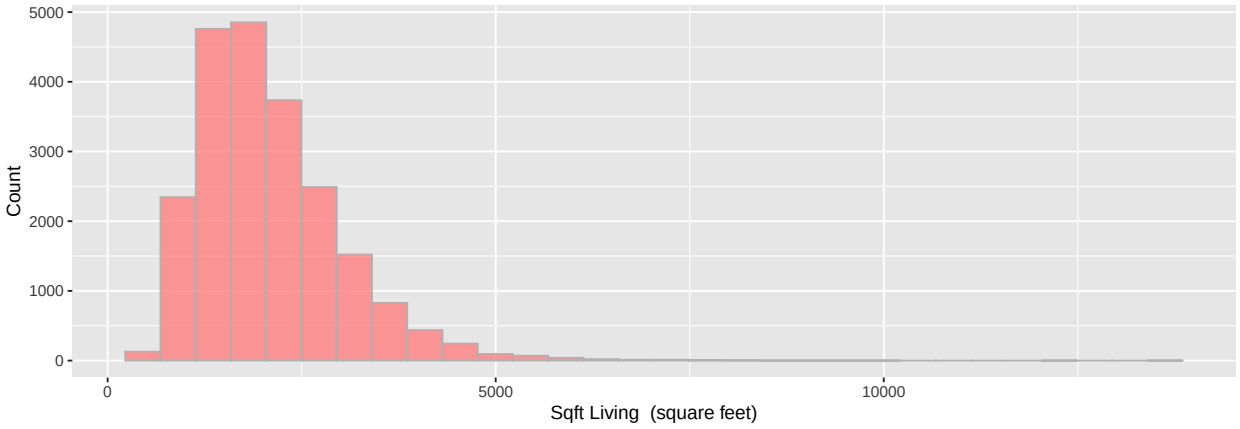
From the plot above we can observe that the selling price of a majority of houses is less than two million dollars. Also it is clear that the distribution of the target variable *price* is right-skewed.

```
ggplot(kc_housing, aes(x= log10_price)) +
  geom_histogram(fill="blue", color="grey", alpha=0.6) +
  labs(x ="Log(Price)  (USD)", y="Count")
```



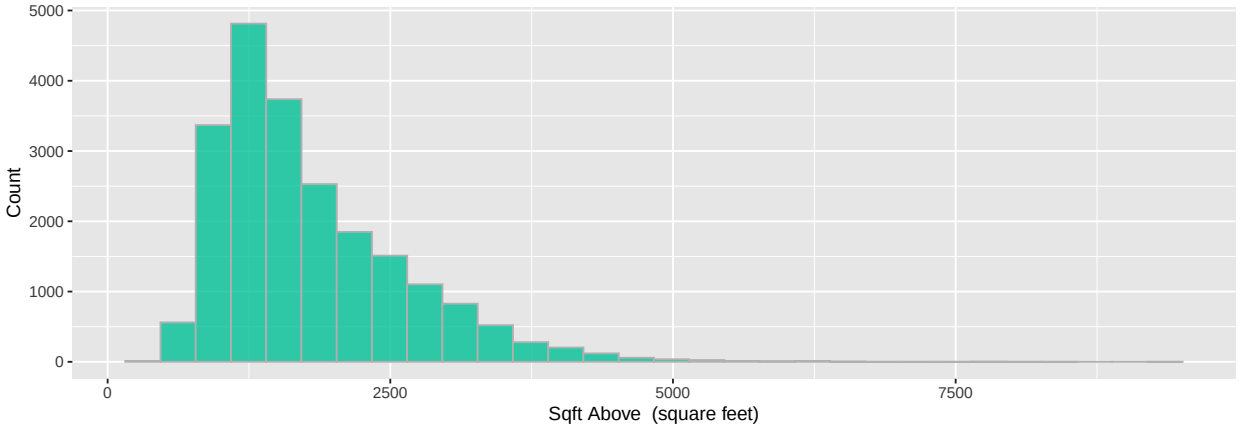
Since our response variable (*price*) is right-skewed, we decided to apply a logarithm transformation on it. As we can observe, the distribution of the logarithmic transformation of *price* becomes bell-shaped.

```
ggplot(kc_housing, aes(x= sqft_living)) +
  geom_histogram(fill="#FF8080", color="grey",alpha=0.8) +
  labs(x ="Sqft Living  (square feet)",y="Count")
```



In the above histogram, we inspected the distribution of the variable *sqft_living* and we can observe that most of the houses appear to have less than 5000 square feet of living space. Moreover, this variable is right-skewed.

```
ggplot(kc_housing, aes(x=sqft_above))+
  geom_histogram(fill="#00C196", color="grey",alpha=0.8) +
  labs(x ="Sqft Above  (square feet)",y="Count")
```

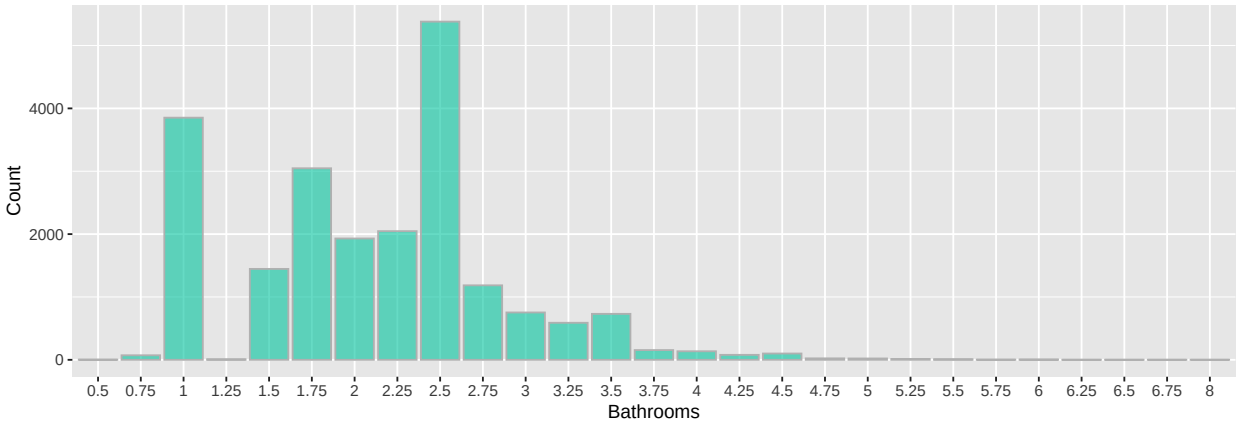


The plot above shows the distribution of the variable *sqft_above*. As we can see the greatest part of the properties have a value lower than 5000 square feet and the distribution is again clearly right-skewed.

```
ggplot(kc_housing, aes(x=sqft_living15))+
  geom_histogram(fill="#FFBF18", color="grey",alpha=0.8) +
  labs(x ="Sqft Living 15  (square feet)",y="Count")
```

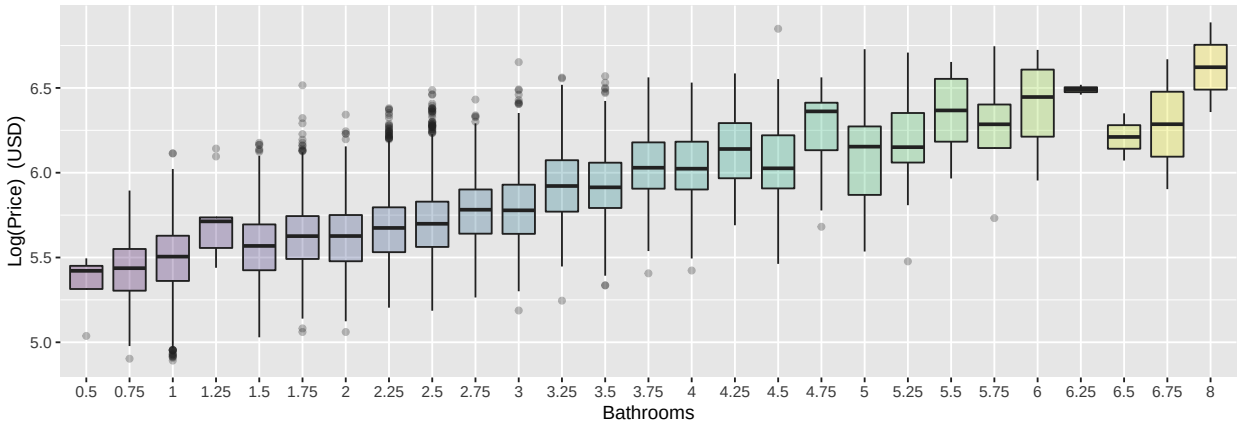
In this histogram, we studied the distribution of the variable *sqft_living15* and we can observe that most of the houses (of the closest 15 neighbors) seem to have an average of less than 4000 square feet of living space. This variable is right-skewed.

```
ggplot(kc_housing, aes(x = bathrooms))+
  geom_bar(fill="#00c39e", color="grey",alpha=0.6) +
  labs(x ="Bathrooms",y="Count")
```



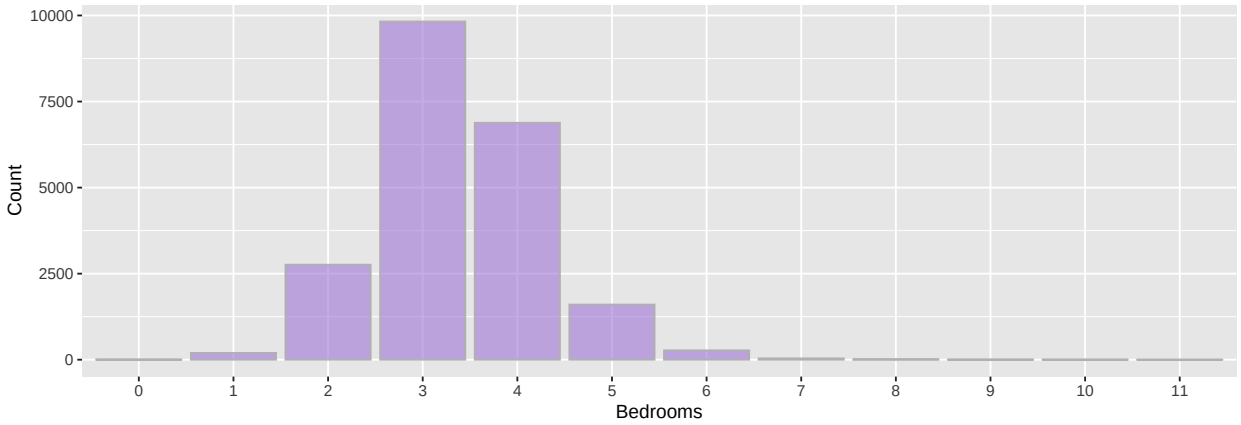
The bar plot above shows that the most frequent values for the variable bathroom is the value 2.5. This means that the house has 2 bathrooms and a room with a toilet but no shower. We can observe a peak for the houses with a single bathroom and in general we can notice that most of the houses appear to have less than 4.5 bathrooms.

```
ggplot(kc_housing, aes(x=bathrooms, y=log10_price, fill=factor(bathrooms)))+
  geom_boxplot(alpha=0.3)+
  theme(legend.position="none") +
  scale_fill_viridis(discrete = TRUE,
                     option = "D") +
  labs(x = 'Bathrooms', y = 'Log(Price)  (USD)')
```



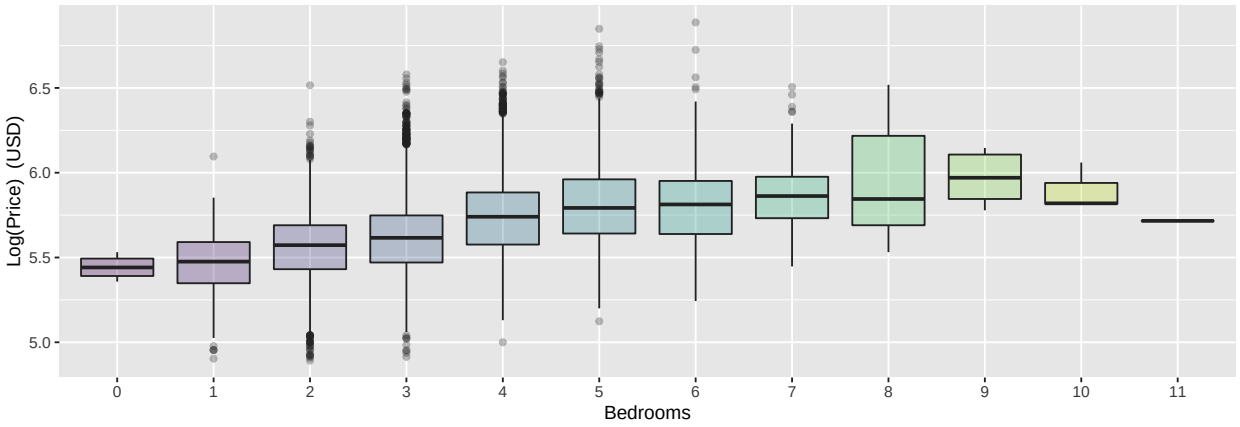
The box-plot above inspects the price of the houses in relation to the number of bathrooms. As we expect, with the increase in the number of bathrooms, the price of houses also increases. Houses with less than 3.5 bathrooms have a price between 740 thousand dollars and one million. Instead, houses with more than 3.5 bathrooms have a price between one million and two million.

```
ggplot(kc_housing, aes(x = bedrooms))+
  geom_bar(fill="#a176d6", color="grey",alpha=0.6) +
  labs(x ="Bedrooms",y="Count")
```



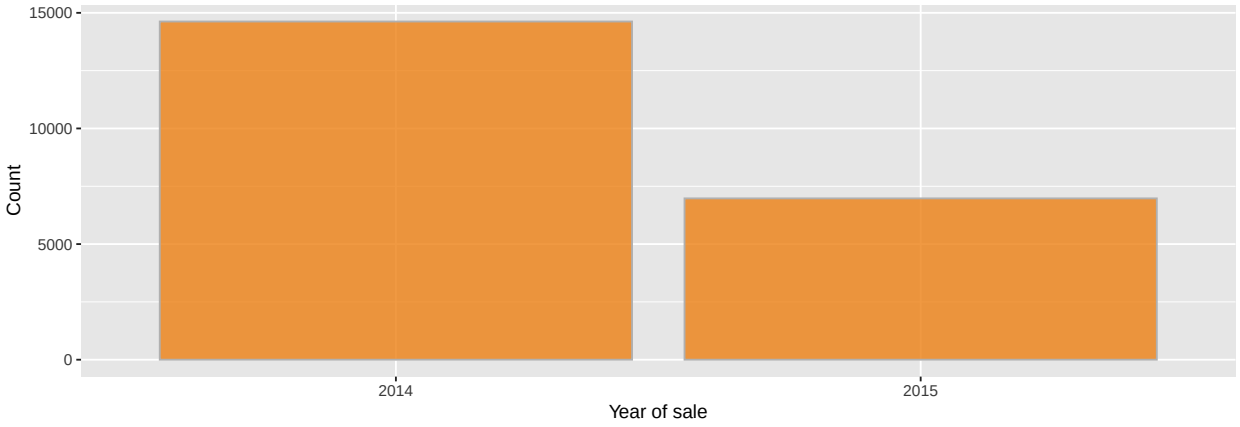
From the bar plot above we can see that the majority of houses sold, has 3 or 4 bedrooms.

```
ggplot(kc_housing, aes(x=bedrooms, y=log10_price, fill=factor(bedrooms)))+
  geom_boxplot(alpha=0.3)+
  theme(legend.position="none") +
  scale_fill_viridis(discrete = TRUE,
                     option = "D") +
  labs(x = 'Bedrooms', y = 'Log(Price)  (USD)')
```



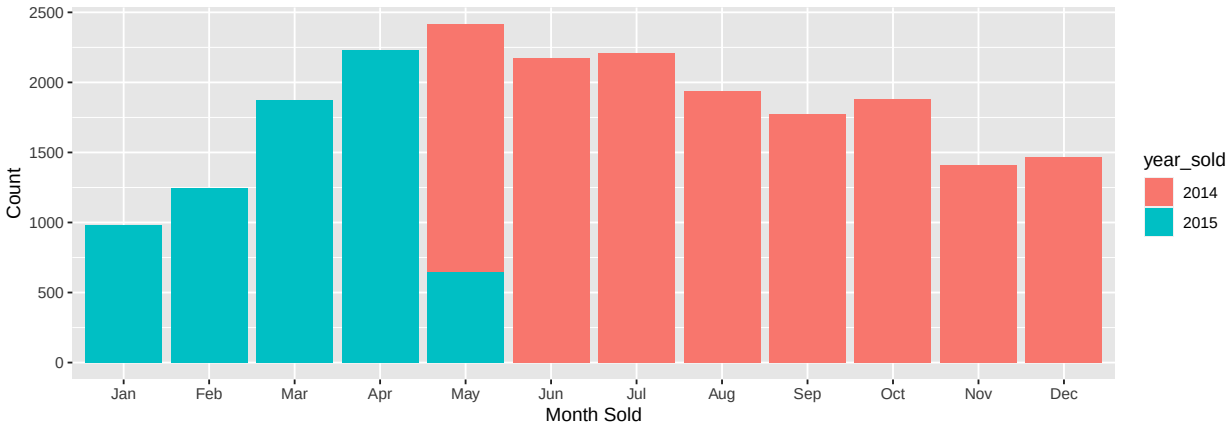
The box-plot above shows that the as the number of bedrooms increase also the the price increase. When we have no bedroom or only 1 the price is 251 thousand dollars, when the number of bedroom is 5 or 6 the price is around 398 thousand dollars, instead with 9 bedrooms the price is around 934 thousand dollars. There is a significant difference in relation with bedrooms.

```
ggplot(kc_housing, aes(year_sold))+
  geom_bar(fill="#ed8013", color="grey",alpha=0.8) +
  labs(x ="Year of sale",y="Count")
```



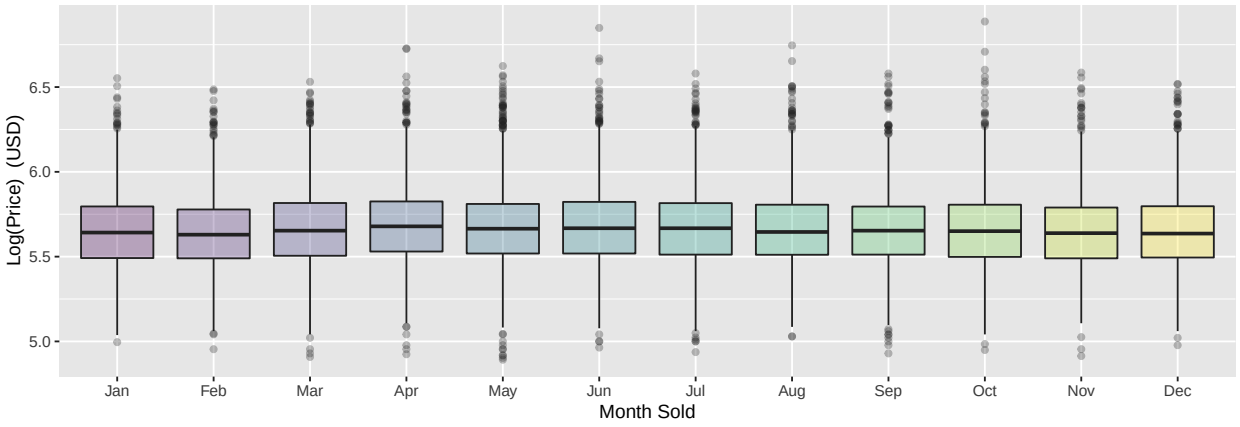
From the bar-plot above we can see that the majority houses in dataset have been sold in year 2014.

```
ggplot(kc_housing, aes(month_sold, fill=year_sold))+
  geom_bar() +
  labs(x = 'Month Sold', y = 'Count')
```



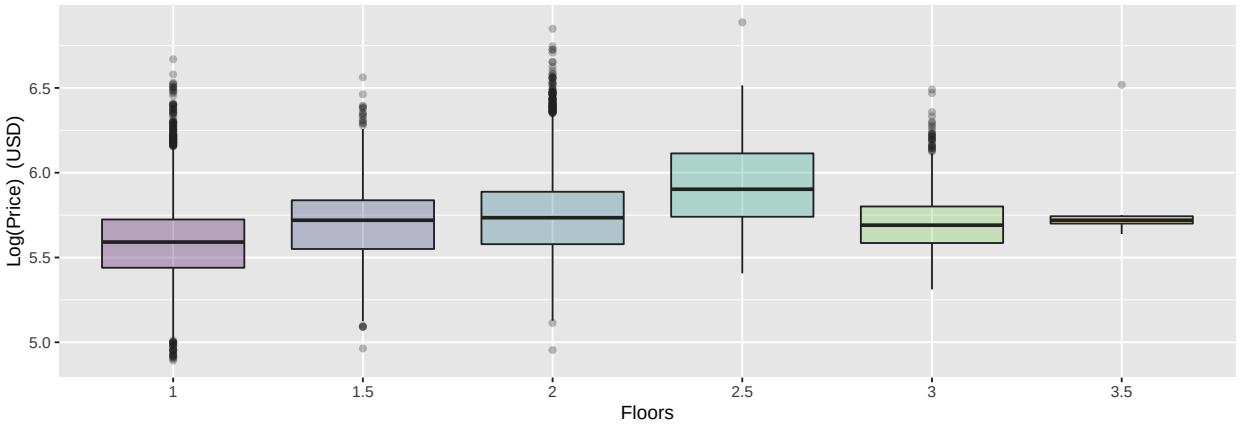
The plot above shows that in the warmer months there are more sales.

```
ggplot(kc_housing, aes(x=month_sold, y=log10_price, fill=month_sold))+
  geom_boxplot(alpha=0.3)+
  theme(legend.position="none") +
  scale_fill_viridis(discrete = TRUE,
                     option = "D")+
  labs(x = 'Month Sold', y = 'Log(Price)  (USD)')
```



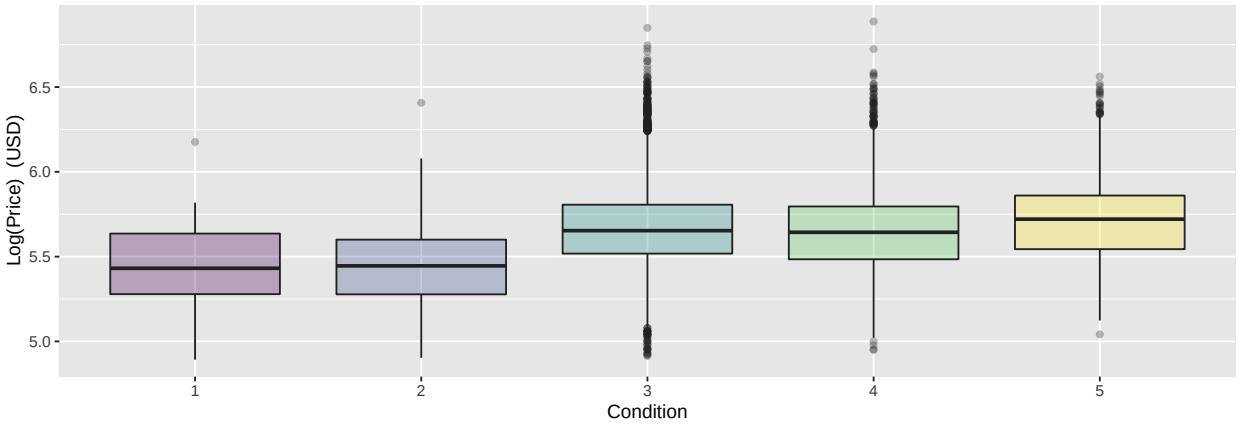
From the box-plot above we can see that the prices of the houses are the same in different months, therefore it is possible to deduce that in this period of time no event occurred that had an effect on prices. Also, considering the previous plot, we can say seasons only change the numbers of sales, not the price.

```
ggplot(kc_housing, aes(x=floors, y=log10_price, fill=floors))+
  geom_boxplot(alpha=0.3)+
  theme(legend.position="none") +
  scale_fill_viridis(discrete = TRUE,
                     option = "D")+
  labs(x = 'Floors', y = 'Log(Price)  (USD)')
```



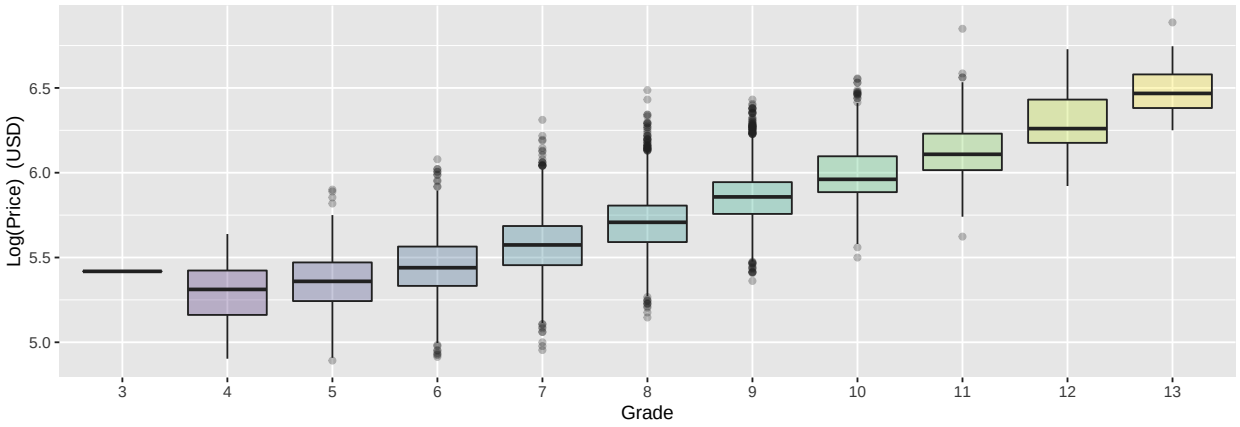
It seems clear that price of houses increases by increase in the number of *floors*.

```
ggplot(kc_housing, aes(x=condition, y=log10_price, fill=condition))+
  geom_boxplot(alpha=0.3)+
  theme(legend.position="none") +
  scale_fill_viridis(discrete = TRUE,
                     option = "D")+
  labs(x = 'Condition', y = 'Log(Price)  (USD)')
```



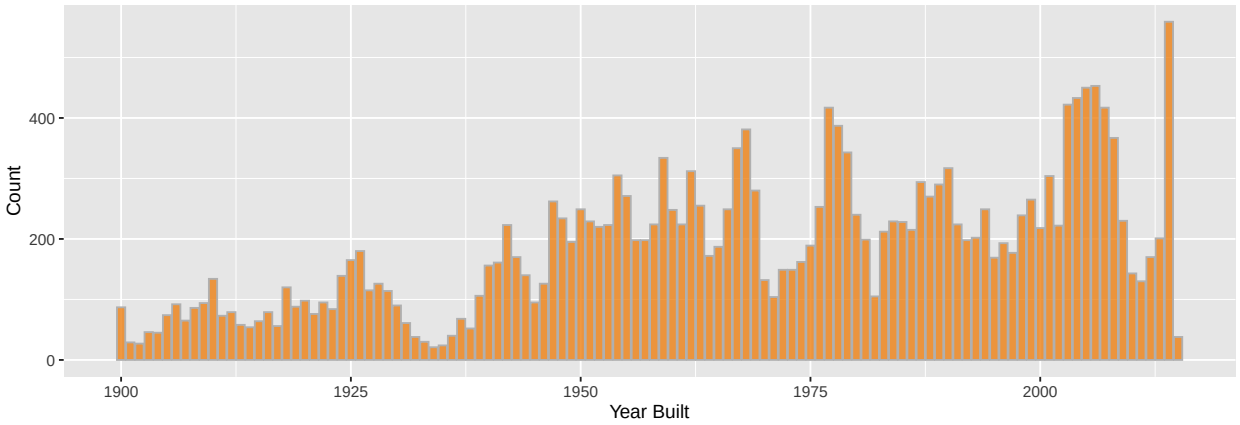
The *price* seems to increase with a higher condition. Between the first and the last there is a clear difference in the mean of price.

```
ggplot(kc_housing, aes(x=grade, y=log10_price, fill=grade))+
  geom_boxplot(alpha=0.3)+
  theme(legend.position="none") +
  scale_fill_viridis(discrete = TRUE,
                     option = "D")+
  labs(x = 'Grade', y = 'Log(Price)  (USD)')
```



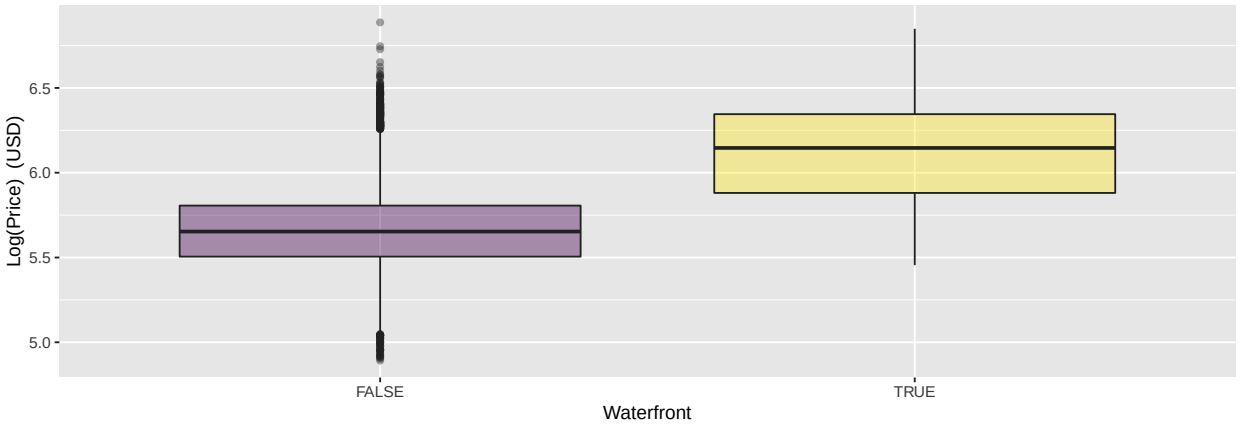
In this box-plot, we can see even more clearly an almost linear dependence. Also, the price seems to be strongly related to the grade.

```
ggplot(kc_housing, aes( yr_built))+
  geom_bar(fill="#ed8013", color="grey",alpha=0.8) +
  labs(x ="Year Built",y="Count")
```



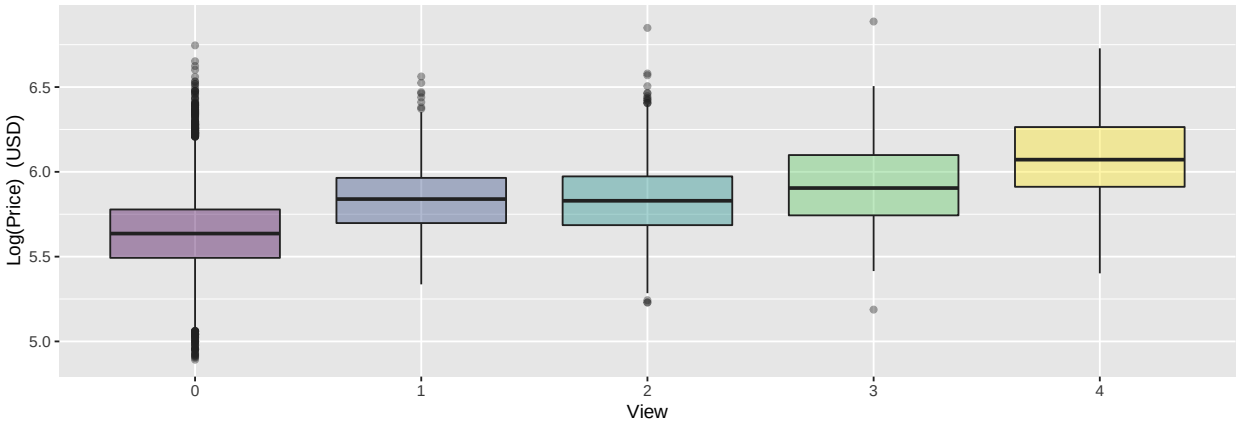
From this plot, we see that most of the houses in this dataset were built recently.

```
ggplot(kc_housing, aes(x=waterfront, y=log10_price, fill=factor(waterfront)))+
  geom_boxplot(alpha=0.4)+
  theme(legend.position="none")+
  scale_fill_viridis(discrete = TRUE, option = "D")+
  labs(x = 'Waterfront', y = 'Log(Price)  (USD)')
```



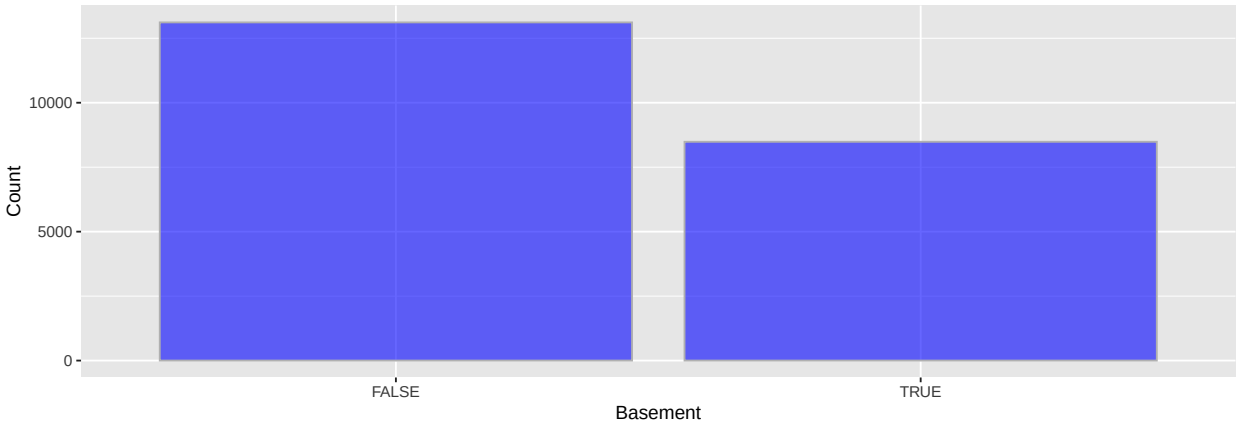
There is a clear difference between the price of houses with and without waterfront.

```
ggplot(kc_housing, aes(x=view, y=log10_price, fill=factor(view)))+
  geom_boxplot(alpha=0.4)+
  theme(legend.position="none")+
  scale_fill_viridis(discrete = TRUE,
                     option = "D")+
  labs(x = 'View', y = 'Log(Price)  (USD)')
```



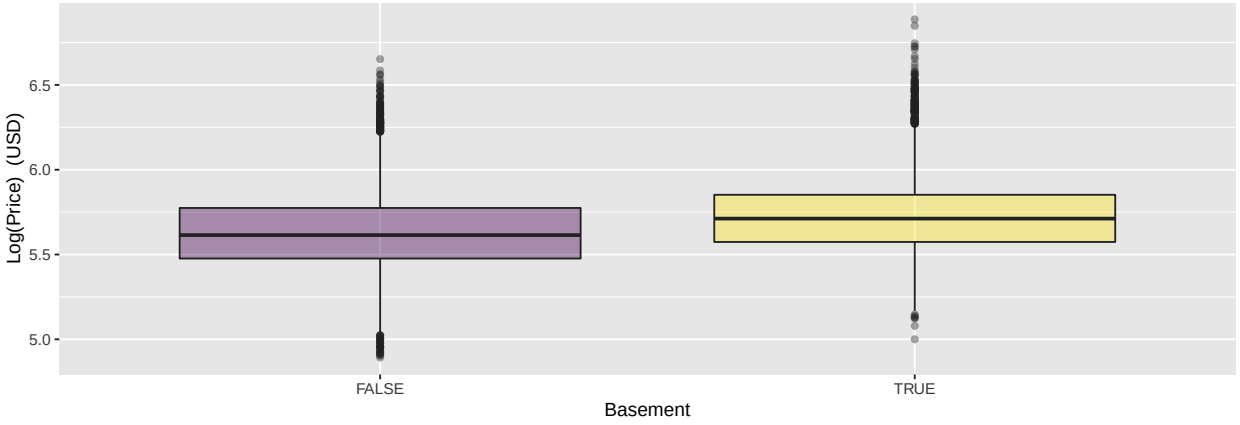
The price seems to increase with a better view. Between the first and the last there is a difference on the mean of prices which means *view* can be an effective variable.

```
#BASEMENT
ggplot(kc_housing, aes(has_basement))+
  geom_bar(fill="blue", color="grey", alpha=0.6)+
  labs(x = 'Basement', y = 'Count')
```



The plot represents the number of houses without a basement compared to the number of houses with a basement.

```
ggplot(kc_housing, aes(x=has_basement, y=log10_price, fill=factor(has_basement)))+
  geom_boxplot(alpha=0.4)+
  theme(legend.position="none")+
  scale_fill_viridis(discrete = TRUE,
                     option = "D")+
  labs(x = 'Basement', y = 'Log(Price)  (USD)')
```



It seems that the presence of a basement influence the price with a higher price for the properties with a basement.

3 Model & Data Analysis

The following section is divided in three parts:

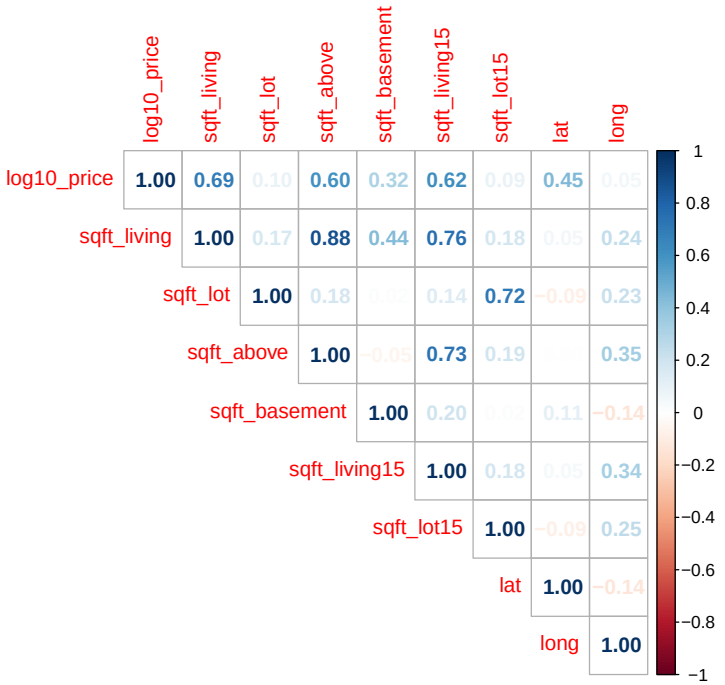
1. We started with a research of the best model with only the numerical variables;
2. Then we did a research with only the categorical variables;
3. In the end we merged the two models.

3.1 Numerical Variables

First we checked the correlation matrix of the numerical variables plus *price*, this was helpful to get a starting point for our study.

```
numerical <- subset(kc_housing, select = c(log10_price, sqft_living, sqft_lot, sqft_above,
                                    sqft_basement, sqft_living15, sqft_lot15, lat, long))

correlation <- cor(numerical)
```

```
corrplot(correlation, method = 'number', type = 'upper')
```



As we can see, the variable *sqft_living* has a higher correlation value with the logarithm of the price. Then, we have plotted them to see their behavior and if a frist degree linear regression is a sufficient interpolation.

```
# linear model
mod.num1 <- lm(data=kc_housing, log10_price ~ sqft_living)
summary(mod.num1)
```

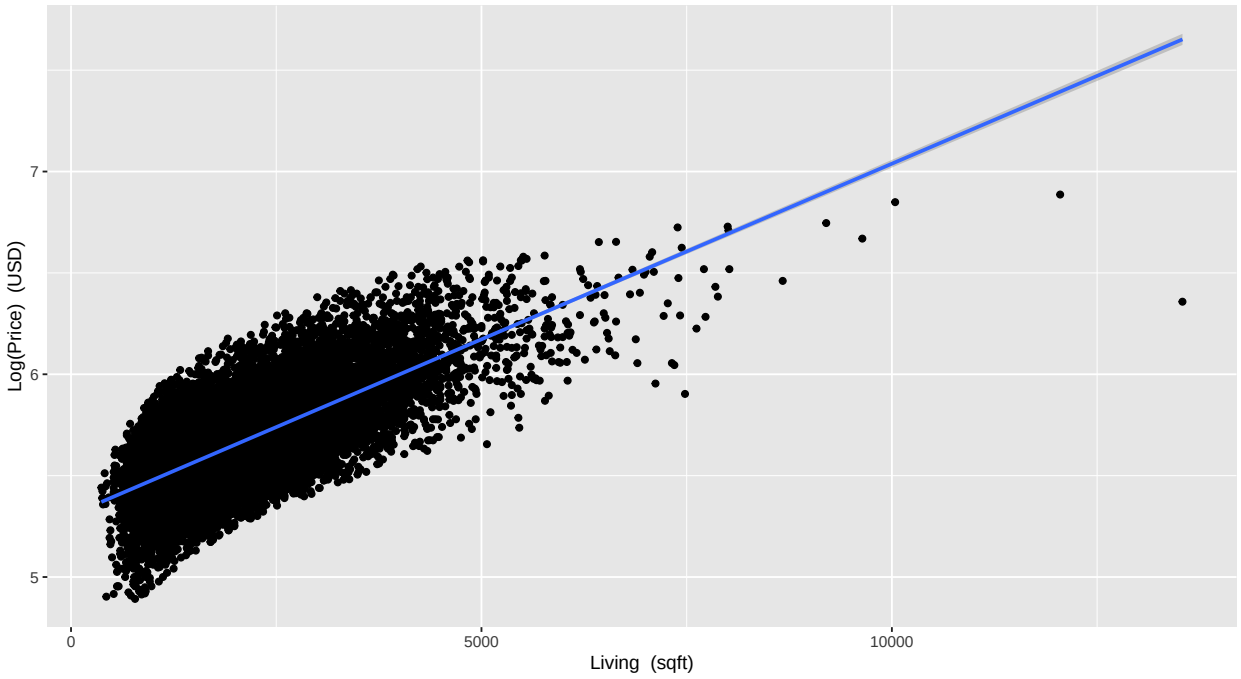
```
##
## Call:
## lm(formula = log10_price ~ sqft_living, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29390 -0.12394  0.00638  0.11316  0.55417
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 5.306e+00  2.772e-03    1914   <2e-16 ***
## sqft_living 1.732e-04  1.220e-06     142   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1643 on 21598 degrees of freedom
## Multiple R-squared:  0.4828, Adjusted R-squared:  0.4828
## F-statistic: 2.016e+04 on 1 and 21598 DF,  p-value: < 2.2e-16
```

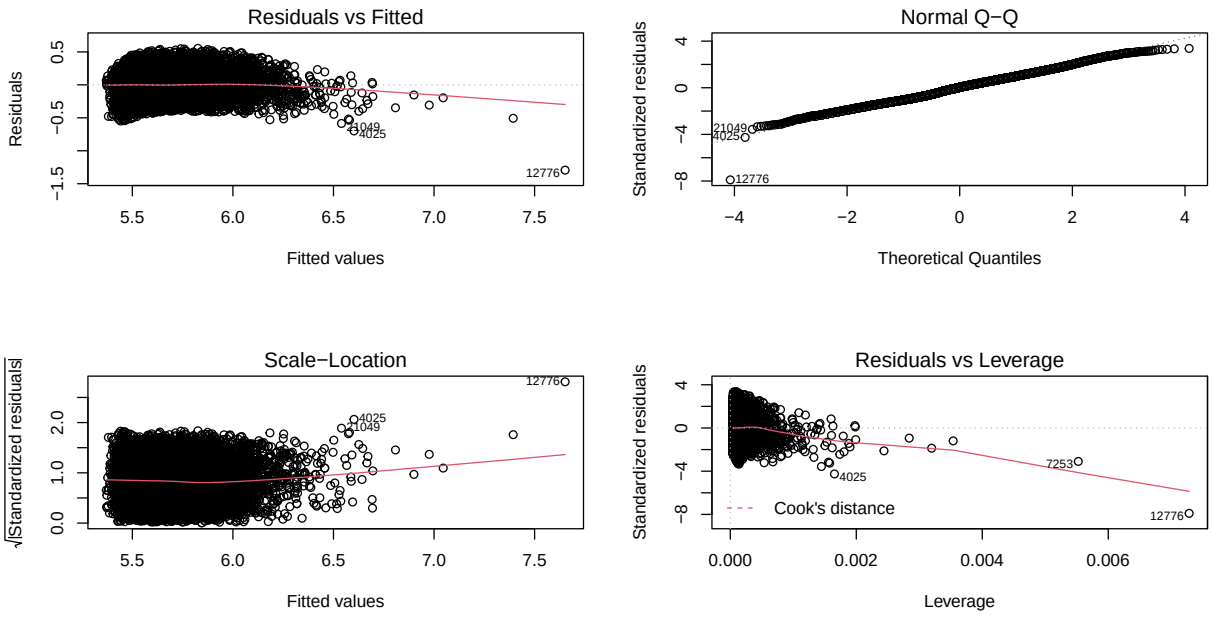
```
BIC(mod.num1)
```

```
## [1] -16688.99
```

```
# Model plot
ggplot(kc_housing, aes(x = sqft_living, y = log10_price)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x) +
  labs(x = 'Living  (sqft)', y = 'Log(Price)  (USD)')
```




```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num1)
```




```
par(mfrow=c(1,1))
```

As we can see a first degree polynomial regression line is not sufficient to interpolate the data. This fact can also be seen in the plot of the residues which are not equally random distributed, this means that some behaviour is not captured by the model. Moreover, since the line on the bottom left plot is not flat, the errors are not homoscedastic.

We proceed with a second plot:

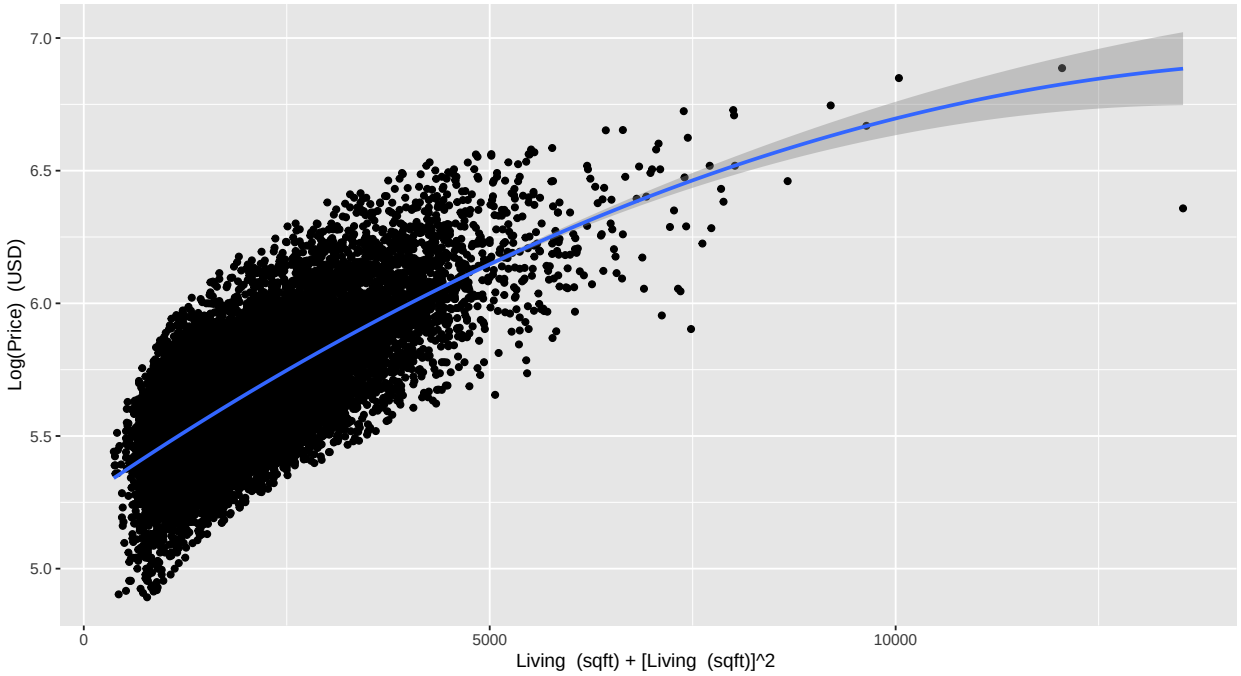
```
# log(price) vs poly(sqft_living) of deg=2
mod.num2 <- lm(data=kc_housing, log10_price ~ sqft_living + I(sqft_living^2))
summary(mod.num2)
```

```
##
## Call:
## lm(formula = log10_price ~ sqft_living + I(sqft_living^2), data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55829 -0.12354  0.00576  0.11235  0.54606
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.265e+00  4.642e-03 1134.16   <2e-16 ***
## sqft_living       2.098e-04  3.493e-06   60.07   <2e-16 ***
## I(sqft_living^2) -6.659e-09  5.962e-10  -11.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1639 on 21597 degrees of freedom
## Multiple R-squared:  0.4858, Adjusted R-squared:  0.4858
## F-statistic: 1.02e+04 on 2 and 21597 DF,  p-value: < 2.2e-16
```

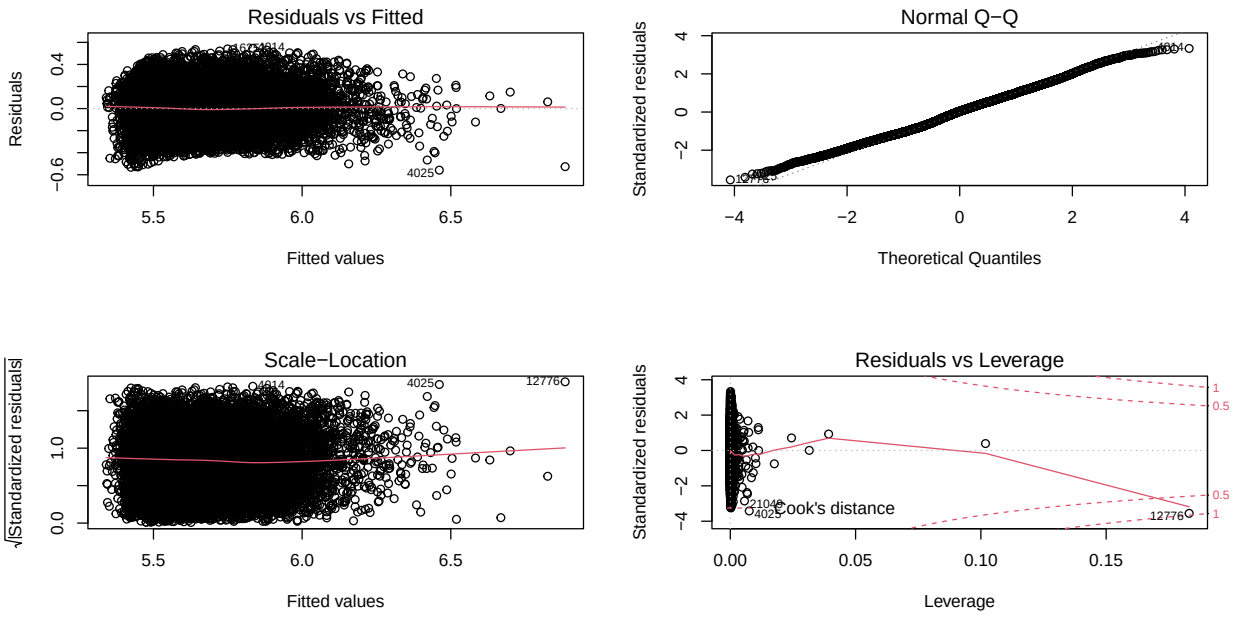
```
BIC(mod.num2)
```

```
## [1] -16803.39
```

```
# Model plot
ggplot(kc_housing, aes(x = sqft_living, y = log10_price)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x + I(x^2)) +
  labs(x = 'Living  (sqft) + [Living  (sqft)]^2 ', y = 'Log(Price)  (USD)')
```




```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num2)
```




```
par(mfrow=c(1,1))
```

Now the situation is clearly improved. The regression line seems to fit better our data and also the residuals are more well distributed in comparison to the previous model. Still, the residuals are not equally distributed, with a higher density in the left side, but there is an improvement of the homoscedasticity in the bottom-left plot. This is our simplest model for the numerical variables.

Our next step is to choose which others numerical variables can be added to our model. To do this, we used the function *regsubsets* which compares models with the same number of variables and gives as output the best one. Since the number of variables is low we decided to run an exhaustive search and not a nested one.

```
regfit.num <- regsubsets(log10_price ~ sqft_living + sqft_above + sqft_lot
                         + sqft_living15 + sqft_lot15 + lat + long, data=kc_housing)

reg.summary <- summary(regfit.num)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(log10_price ~ sqft_living + sqft_above + sqft_lot +
##     sqft_living15 + sqft_lot15 + lat + long, data = kc_housing)
## 7 Variables  (and intercept)
##               Forced in Forced out
## sqft_living       FALSE      FALSE
## sqft_above        FALSE      FALSE
## sqft_lot          FALSE      FALSE
## sqft_living15     FALSE      FALSE
## sqft_lot15        FALSE      FALSE
## lat               FALSE      FALSE
## long              FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
```

```
##          sqft_living sqft_above sqft_lot sqft_living15 sqft_lot15 lat long
## 1  ( 1 ) "*"         " "         " "      " "            " "        " " " "
## 2  ( 1 ) "*"         " "         " "      " "            " "        "*" " "
## 3  ( 1 ) "*"         " "         " "      "*"            " "        "*" " "
## 4  ( 1 ) "*"         " "         " "      "*"            " "        "*" "*"
## 5  ( 1 ) "*"         " "         "*"      "*"            " "        "*" "*"
## 6  ( 1 ) "*"         "*"         "*"      "*"            " "        "*" "*"
## 7  ( 1 ) "*"         "*"         "*"      "*"            "*"        "*" "*"
```

Thus, we get the best seven models and we compared the values of RSS, R^2, C_p and BIC:

```
par(mfrow=c(2,2))
# residual sum of squares
plot(reg.summary$rss, xlab="Number of Variables", ylab="RSS", type="l")

# adjusted-R^2 with its largest value
plot(reg.summary$adjr2, xlab="Number of Variables", ylab="Adjusted RSq", type="l")
which.max(reg.summary$adjr2)
```

```
## [1] 7
```

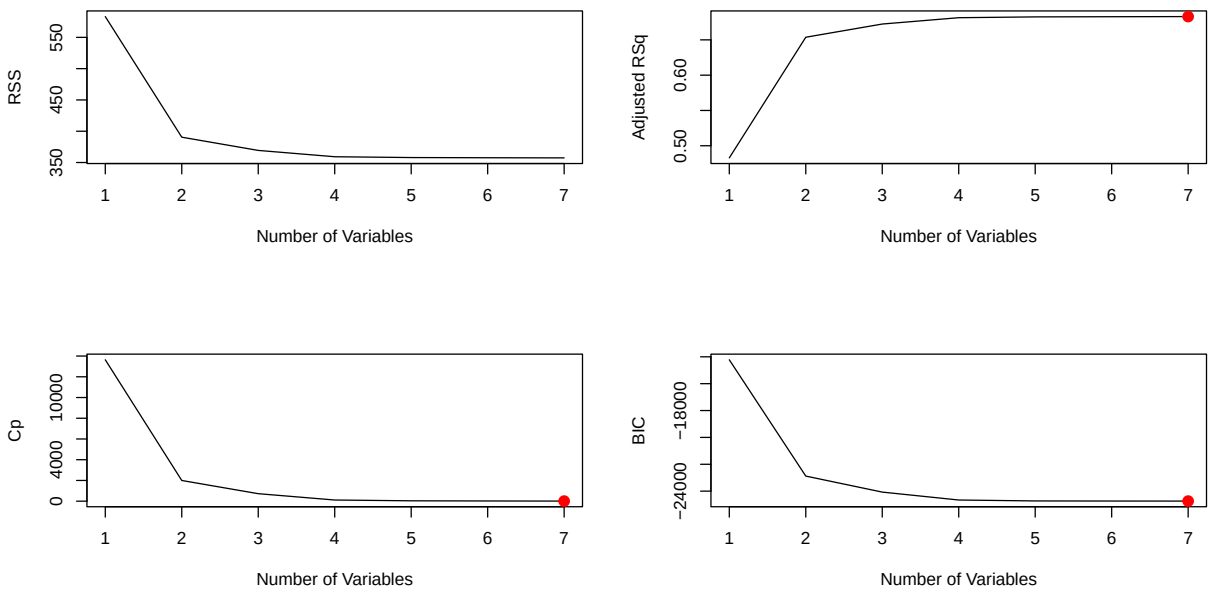
```
points(7, reg.summary$adjr2[7], col="red",cex=2,pch=20)

# Mallow's Cp with its smallest value
plot(reg.summary$cp, xlab="Number of Variables", ylab="Cp", type='l')
which.min(reg.summary$cp)
```

```
## [1] 7
```

```
points(7, reg.summary$cp[7], col="red", cex=2, pch=20)

# BIC with its smallest value
plot(reg.summary$bic, xlab="Number of Variables", ylab="BIC", type='l')
which.min(reg.summary$bic)
```

```
## [1] 7
```

```
points(7, reg.summary$bic[7], col="red", cex=2, pch=20)
```




```
par(mfrow=c(1,1))
```

The plots of the RSS and Adjusted R^2 are as we expected from the theory,the RSS decreases as the number of variables increases while the Adjusted R^2, which is dependent of the RSS by the relationship:

$$AdR^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

It increases while the number of variables increases, with highest value for the full model.

Our choice is to use the BIC for the evaluation of the models because we would like to obtain a model with a lower number of variables. From the plots above and the following overview table we can visibly observe that a model with 4 variables and a model with 7 does not differs so much in terms of performance. So we decided to proceed with a lower number of variables to keep the model as simple as possible.

Our model will contain the following variables:

1. *sqft_living*
2. *sqft_living15*
3. *lat*
4. *long*

```
label1 <- c('1 Variable', '2 Variables', '3 Variables', '4 Variables',
            '5 Variables', '6 Variables', '7 Variables')
com <- data.frame(label1, reg.summary$adjr2, reg.summary$bic)
kable(com, digits = 4, align = "ccc", booktabs = T, "latex",
      col.names = c("Number of variables", "Adjusted R2", "BIC"))%>%
  kable_styling(latex_options="striped")
```

Number of variables	Adjusted R2	BIC
1 Variable	0.4828	-14222.77
2 Variables	0.6537	-22878.06
3 Variables	0.6724	-24070.13
4 Variables	0.6814	-24660.14
5 Variables	0.6825	-24728.76
6 Variables	0.6828	-24736.91
7 Variables	0.6830	-24740.76

Checking the degree of each variable

Next, we did a check on the variables of the fourth model, like we did with the variable *sqft_living*, to see if an higher grade is needed for a good fit of the regression line in the plot of each variable against the logarithm of the *price*:

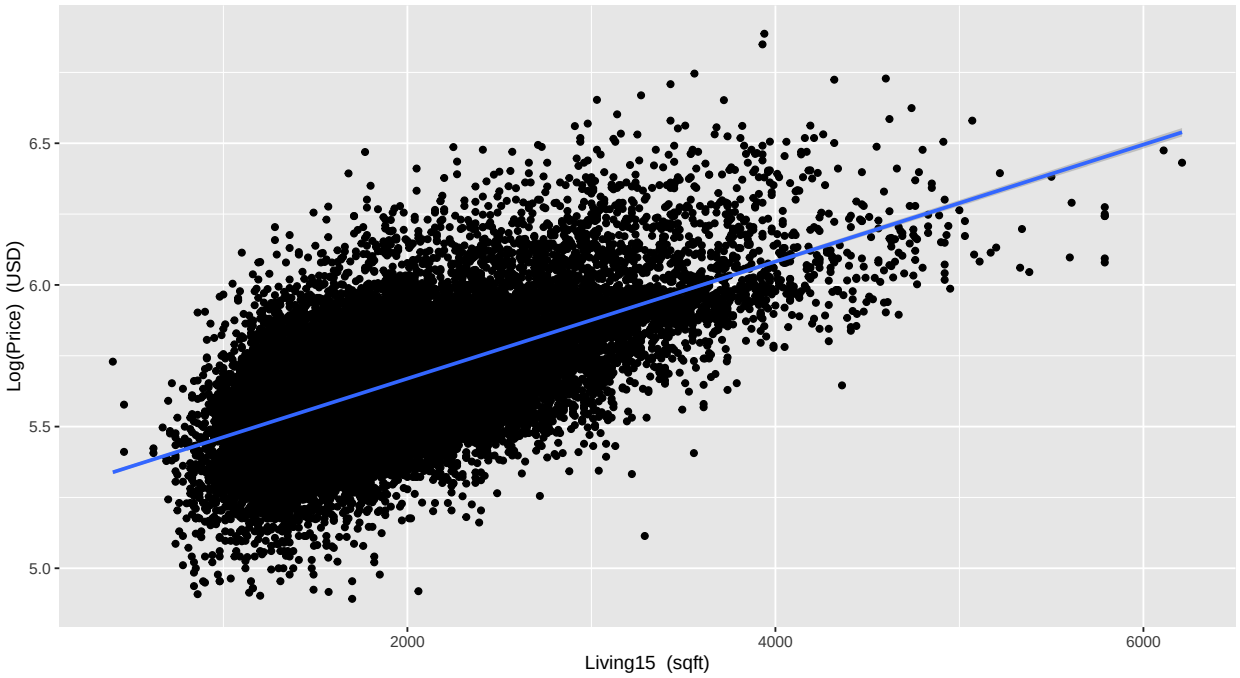
```
# log(price) vs sqft_living15
mod.living15 <- lm(data=kc_housing, log10_price ~ sqft_living15)
summary(mod.living15)
```

```
##
## Call:
## lm(formula = log10_price ~ sqft_living15, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82181 -0.12967 -0.00329  0.11312  0.84718
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.256e+00  3.746e-03  1403.1   <2e-16 ***
## sqft_living15 2.065e-04  1.783e-06   115.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1795 on 21598 degrees of freedom
## Multiple R-squared:  0.3831, Adjusted R-squared:  0.3831
## F-statistic: 1.341e+04 on 1 and 21598 DF,  p-value: < 2.2e-16
```

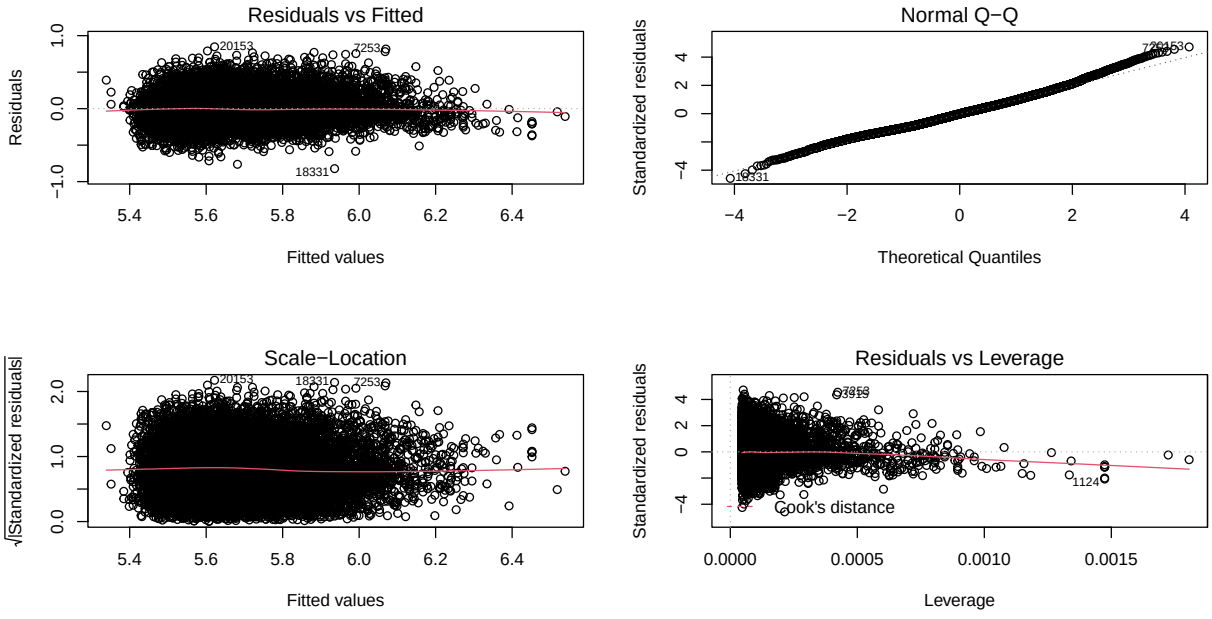
```
BIC(mod.living15)
```

```
## [1] -12881.06
```

```
ggplot(kc_housing, aes(x = sqft_living15, y = log10_price)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x) +
  labs(x = 'Living15  (sqft)', y = 'Log(Price)  (USD)')
```




```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.living15)
```




```
par(mfrow=c(1,1))
```

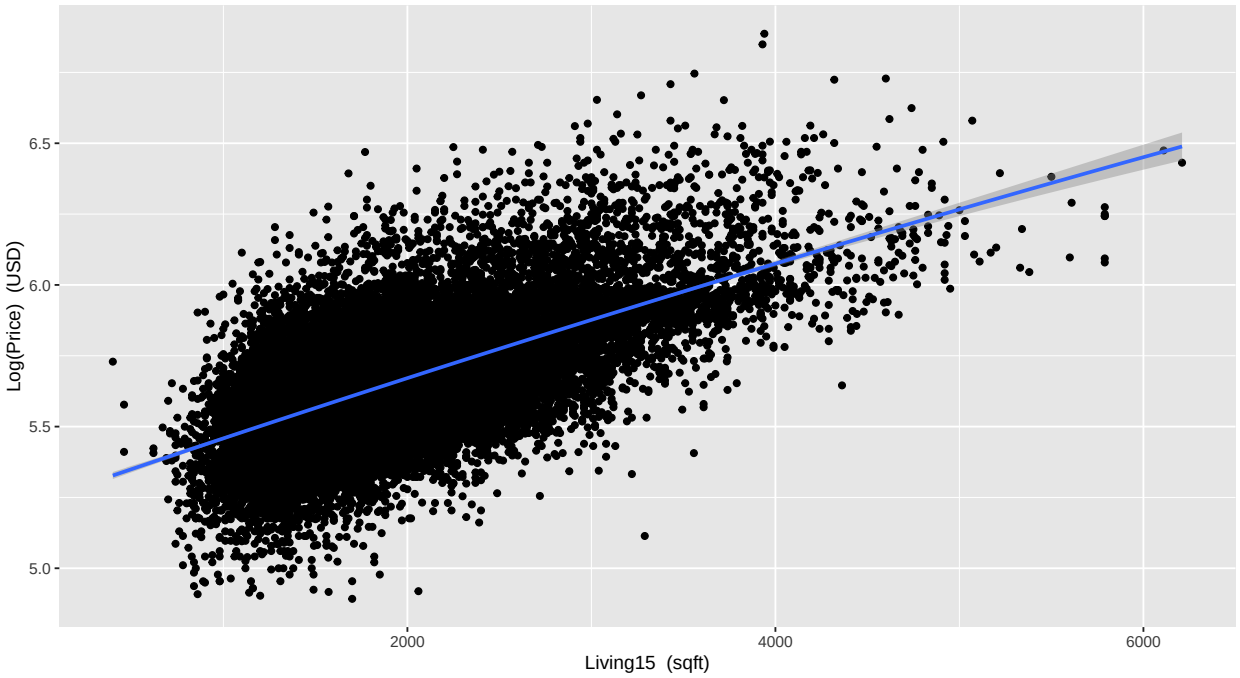
```r
# log(price) vs sqft_living15 +  sqft_living15^2

mod.living152 <- lm(data=kc_housing, log10_price ~ sqft_living15 + I(sqft_living15^2))
summary(mod.living152)
```

```
##
## Call:
## lm(formula = log10_price ~ sqft_living15 + I(sqft_living15^2),
##     data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82096 -0.12981 -0.00338  0.11330  0.84627
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.239e+00  9.224e-03 567.934   <2e-16 ***
## sqft_living15        2.232e-04  8.200e-06  27.219   <2e-16 ***
## I(sqft_living15^2) -3.534e-09  1.692e-09  -2.089   0.0367 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1795 on 21597 degrees of freedom
## Multiple R-squared:  0.3833, Adjusted R-squared:  0.3832
## F-statistic:  6710 on 2 and 21597 DF,  p-value: < 2.2e-16
```

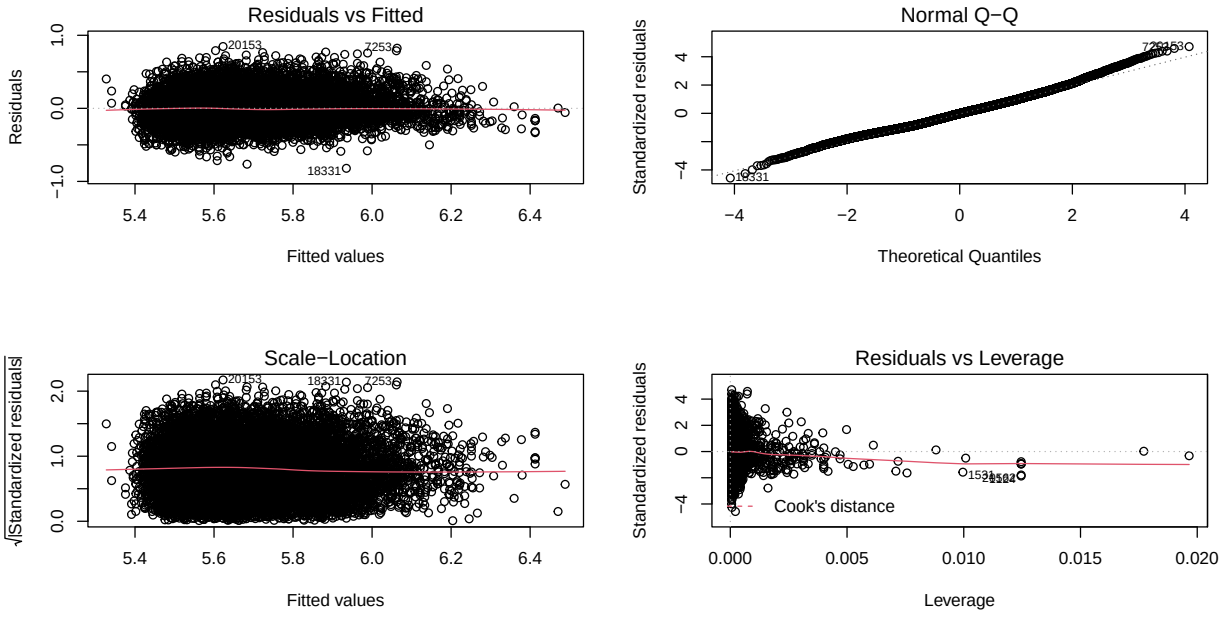
```r
BIC(mod.living152)
```

```
## [1] -12875.44
```

```r
ggplot(kc_housing, aes(x = sqft_living15, y = log10_price)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ poly(x,2)) +
  labs(x = 'Living15  (sqft)', y = 'Log(Price)  (USD)')
```




```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.living152)
```




```
par(mfrow=c(1,1))
```

If we compare these plots we can see that adding a degree does not improve the interpolation of the data.

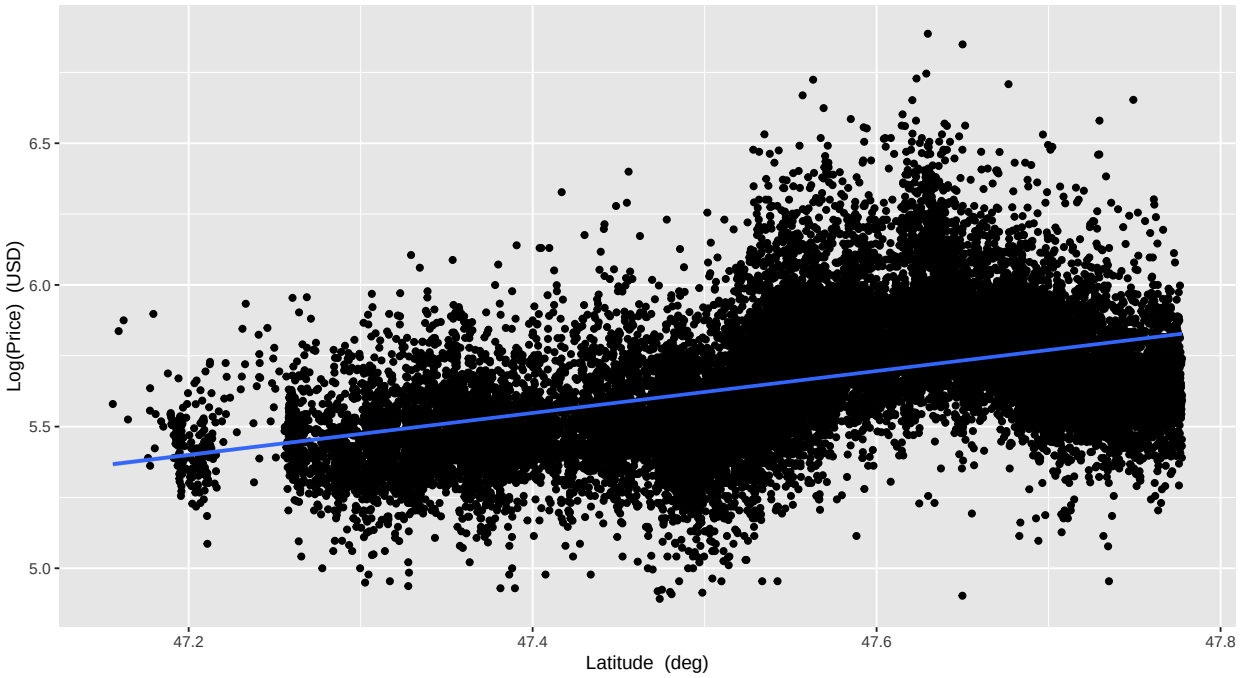
```r
#log(price) vs lat
#Grade 1
mod.lat <- lm(data=kc_housing, log10_price ~ lat)
summary(mod.lat)
```

```
##
## Call:
## lm(formula = log10_price ~ lat, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84199 -0.13455 -0.01857  0.11641  1.16823
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -29.55260    0.47684  -61.98   <2e-16 ***
## lat           0.74052    0.01003   73.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2042 on 21598 degrees of freedom
## Multiple R-squared:  0.2017, Adjusted R-squared:  0.2016
## F-statistic:  5455 on 1 and 21598 DF,  p-value: < 2.2e-16
```

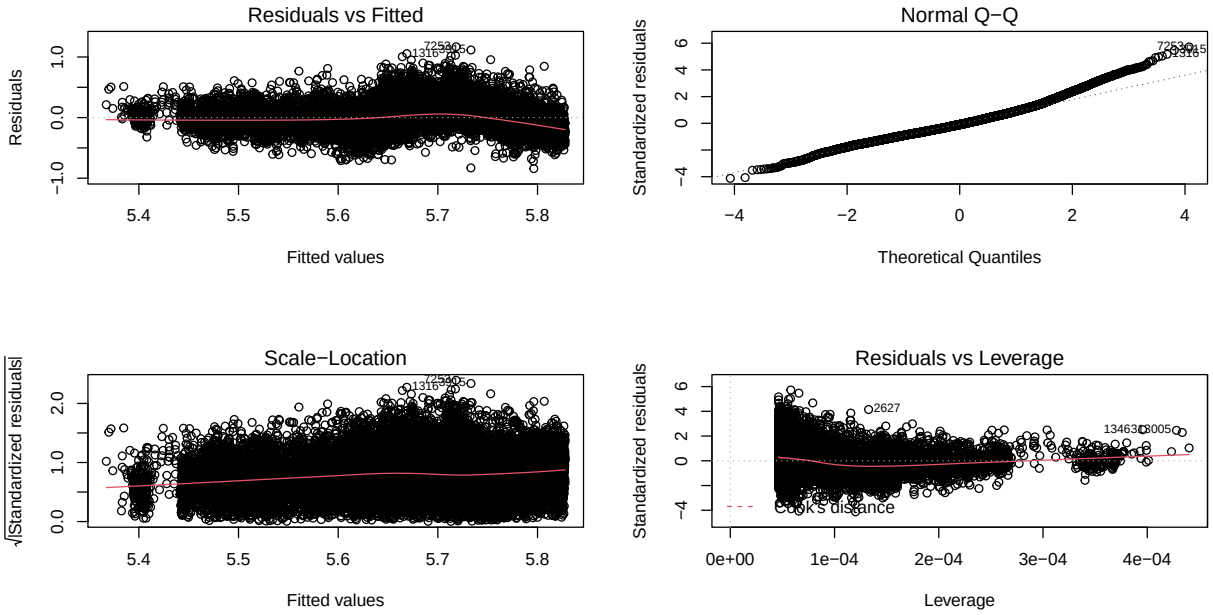
```r
BIC(mod.lat)
```

```
## [1] -7310.799
```

```r
ggplot(kc_housing, aes(x = lat, y = log10_price)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x) +
  labs(x = 'Latitude  (deg)', y = 'Log(Price)  (USD)')
```




```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.lat)
```




```
par(mfrow=c(1,1))
```

It is clear that between the variable *lat* and *price* there is a relationship more complex that a simple linear one.

```
mod.latgrade <- lm(data=kc_housing, log10_price ~ lat + I(lat^2) + I(lat^3) + I(lat^4)
                                                  + I(lat^5))
#summary(mod.lat2)
anova(mod.latgrade)
```

```
## Analysis of Variance Table
##
## Response: log10_price
##              Df Sum Sq Mean Sq F value    Pr(>F)
## lat           1 227.40 227.402  6461.4 < 2.2e-16 ***
## I(lat^2)      1  43.13  43.133  1225.6 < 2.2e-16 ***
## I(lat^4)      1  97.12  97.118  2759.5 < 2.2e-16 ***
## Residuals 21596 760.04   0.035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We inspected the first 5 degrees of the polinomial and thanks to the ANOVA function we observed that we need to plot a model with polynomial degree equal to 4.

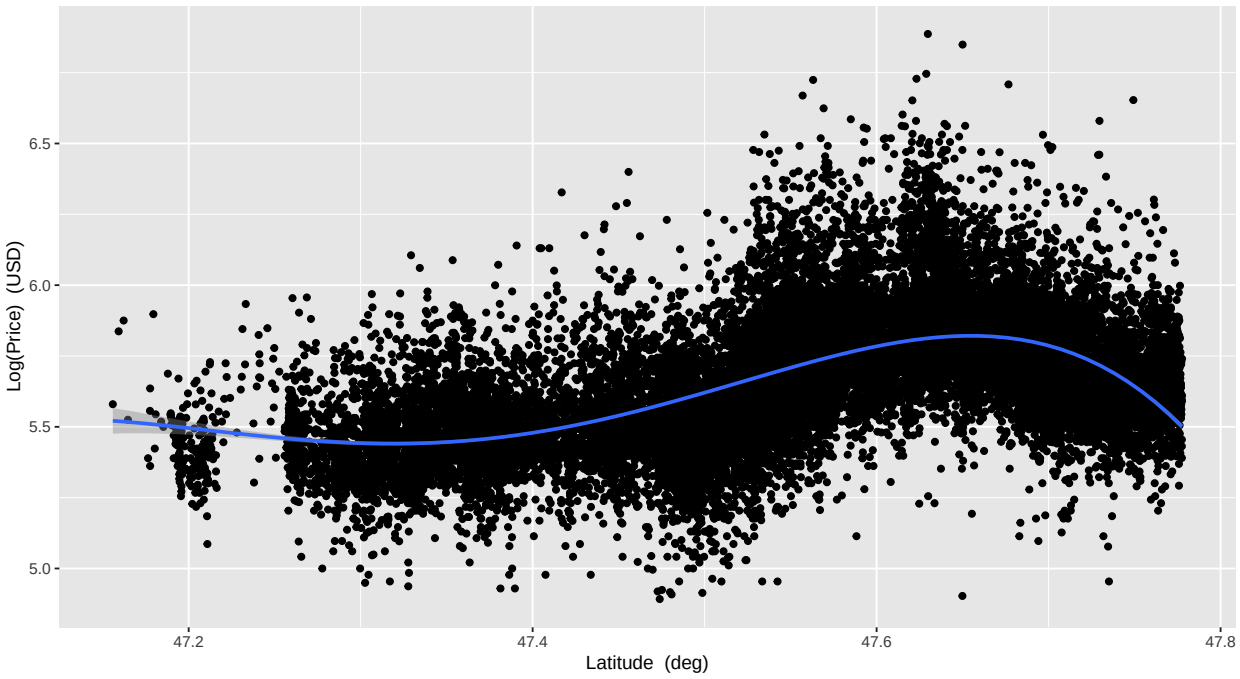
```
#GRADE 4
mod.lat2 <- lm(data=kc_housing, log10_price ~ poly(lat, 4))
summary(mod.lat2)
```

```
##
## Call:
## lm(formula = log10_price ~ poly(lat, 4), data = kc_housing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9178 -0.1199 -0.0158  0.1035  1.0738
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.666614   0.001269 4466.89   <2e-16 ***
## poly(lat, 4)1 15.079843   0.186443   80.88   <2e-16 ***
## poly(lat, 4)2 -6.567556   0.186443  -35.23   <2e-16 ***
## poly(lat, 4)3 -9.852119   0.186443  -52.84   <2e-16 ***
## poly(lat, 4)4 -3.071914   0.186443  -16.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1864 on 21595 degrees of freedom
## Multiple R-squared:  0.3343, Adjusted R-squared:  0.3342
## F-statistic:  2712 on 4 and 21595 DF,  p-value: < 2.2e-16
```

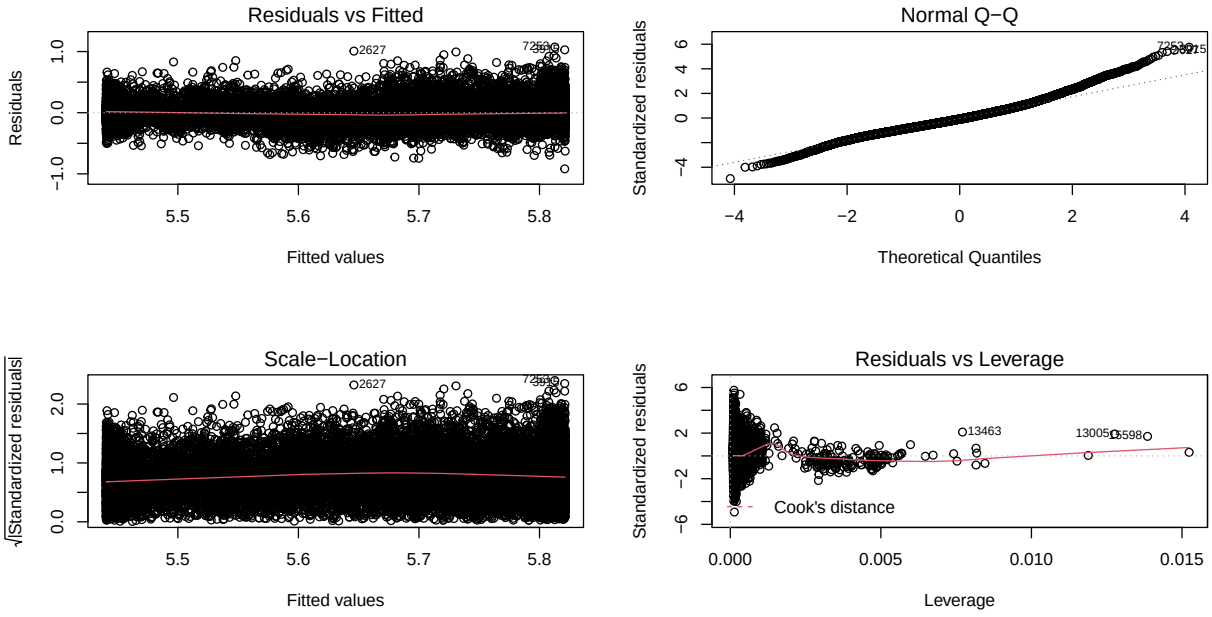
```
BIC(mod.lat2)
```

```
## [1] -11207.05
```

```
ggplot(kc_housing, aes(x = lat, y = log10_price)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ poly(x, 4)) +
  labs(x = 'Latitude  (deg)', y = 'Log(Price)  (USD)')
```




```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.lat2)
```



```
par(mfrow=c(1,1))
```

From the first two plots we can see that a first degree polynomial seems to not fit very well the data but from the residual plot even if not perfect is quite good. If we try to plot a polynomial with degree 4 the data are perfectly fitted by the regression line and also from the residuals plot we can see that there is a equal distribution around the value zero.

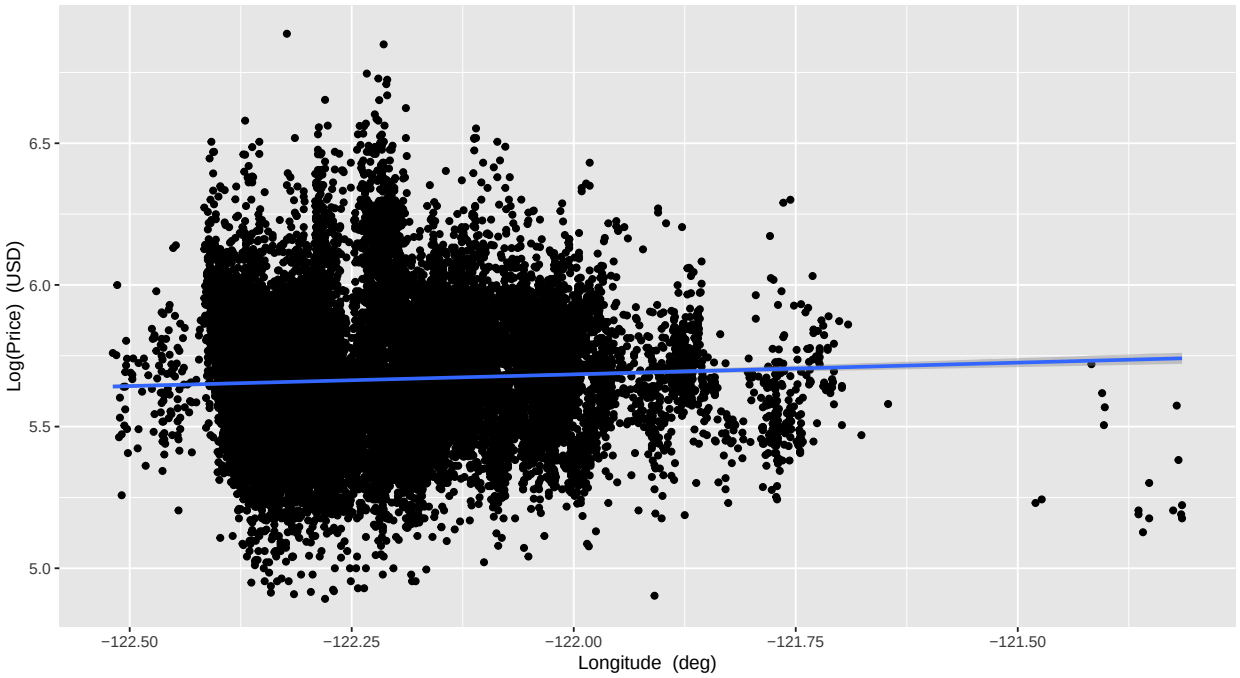
```
#log(price) vs long
#Grade 1
mod.long <- lm(data=kc_housing, log10_price ~ long)
summary(mod.long)
```

```
##
## Call:
## lm(formula = log10_price ~ long, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78874 -0.15921 -0.01191  0.14115  1.22889
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.77334    1.34825  11.699  < 2e-16 ***
## long         0.08270    0.01103   7.496 6.82e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2282 on 21598 degrees of freedom
## Multiple R-squared:  0.002595,   Adjusted R-squared:  0.002549
## F-statistic: 56.19 on 1 and 21598 DF,  p-value: 6.822e-14
```

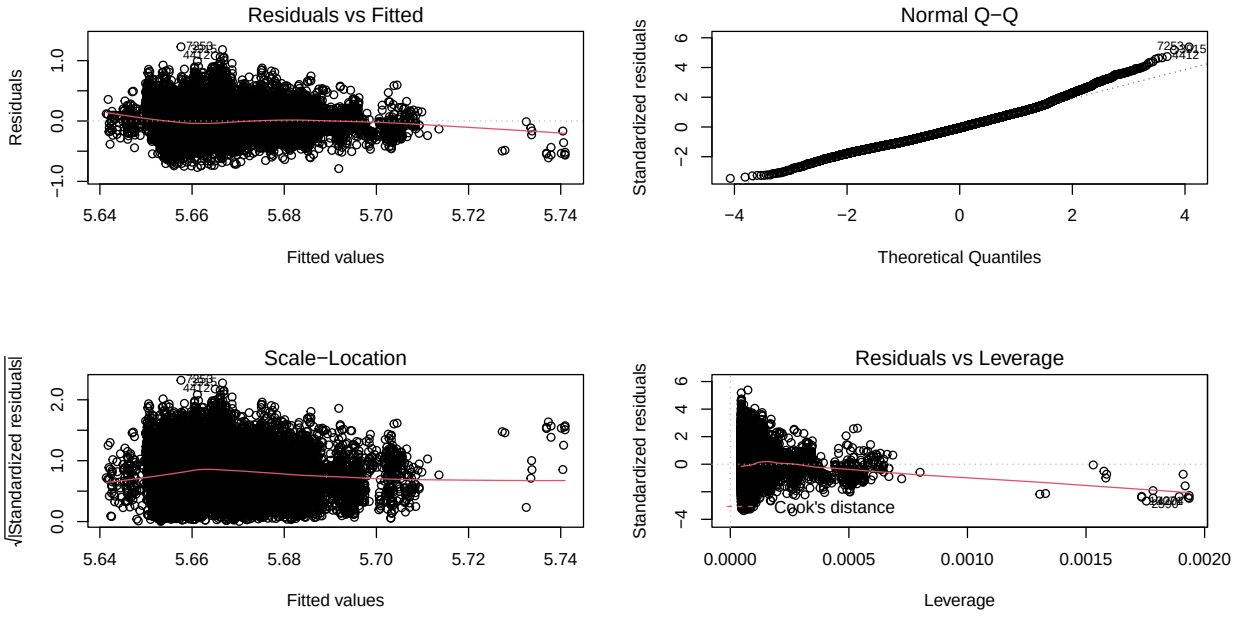
```
BIC(mod.long)
```

```
## [1] -2502.389
```

```
ggplot(kc_housing, aes(x = long, y = log10_price)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x) +
  labs(x = 'Longitude  (deg)', y = 'Log(Price)  (USD)')
```




```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.long)
```




```
par(mfrow=c(1,1))
```

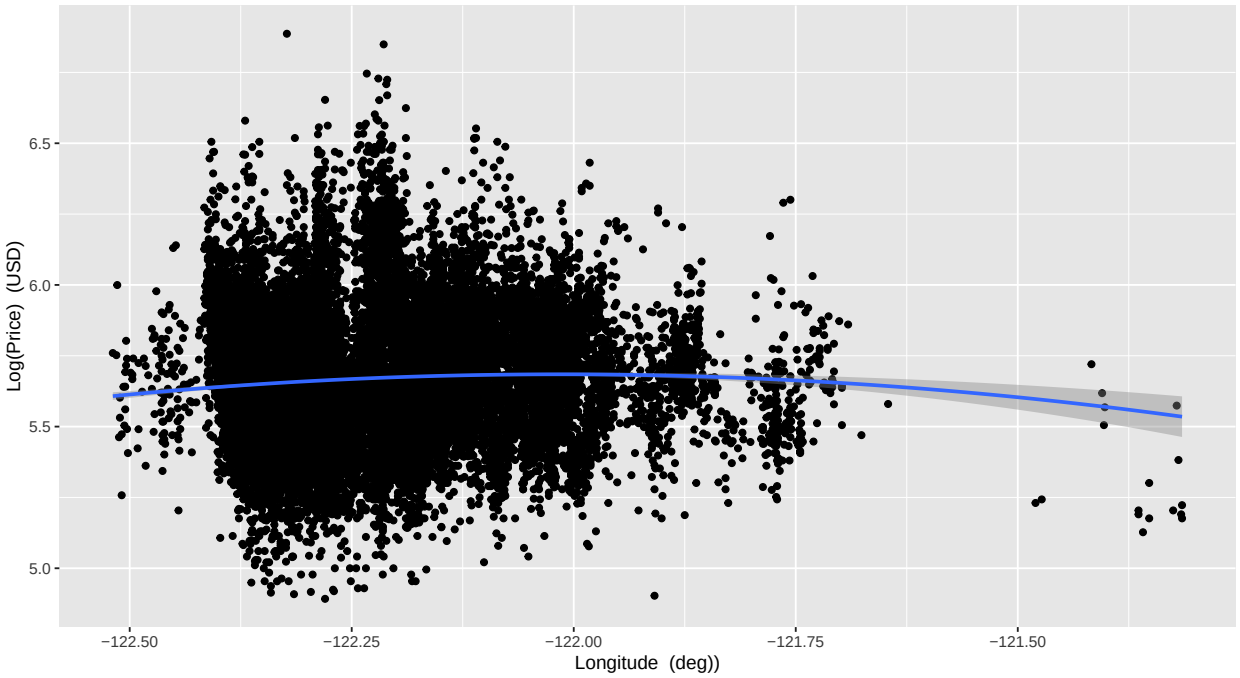
```r
#Grade 2
mod.long2 <- lm(data=kc_housing, log10_price ~ long + I(long^2))
summary(mod.long2)
```

```
##
## Call:
## lm(formula = log10_price ~ long + I(long^2), data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77802 -0.15959 -0.01068  0.14133  1.23061
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4526.4371   775.9175  -5.834 5.50e-09 ***
## long          -74.2874    12.7042  -5.847 5.06e-09 ***
## I(long^2)      -0.3044     0.0520  -5.854 4.87e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.228 on 21597 degrees of freedom
## Multiple R-squared:  0.004175,   Adjusted R-squared:  0.004083
## F-statistic: 45.27 on 2 and 21597 DF,  p-value: < 2.2e-16
```

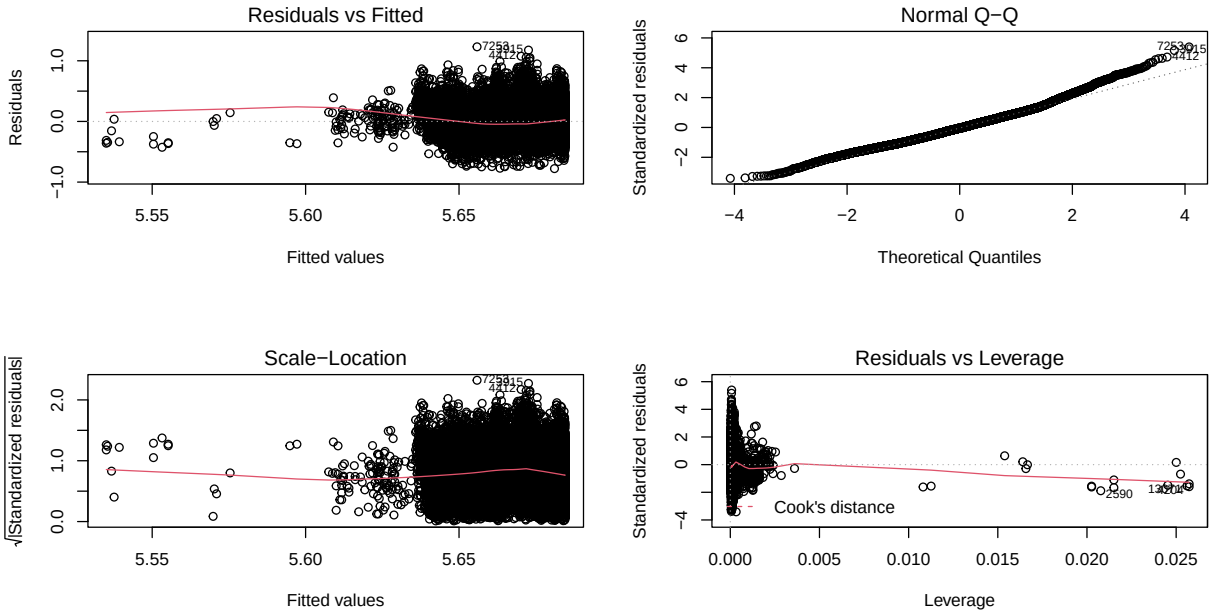
```r
BIC(mod.long2)
```

```
## [1] -2526.655
```

```r
ggplot(kc_housing, aes(x = long, y = log10_price)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x + I(x^2)) +
  labs(x = 'Longitude  (deg))', y = 'Log(Price)  (USD)')
```




```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.long2)
```




```
par(mfrow=c(1,1))
```

For what concerns the variable *long* a polynomial of grade 2 does not improve the fit so much. Moreover, comparing the residuals plots, the first one seems to be better.

In conclusion, after this analysis, we decided to inspect both the models with polynomial degree of 1 and 4 for the variable *lat*.

Comparison of the numerical models

After the analysis that we did we extracted the following numerical models,which all have as response variable the variable *log(price)*:

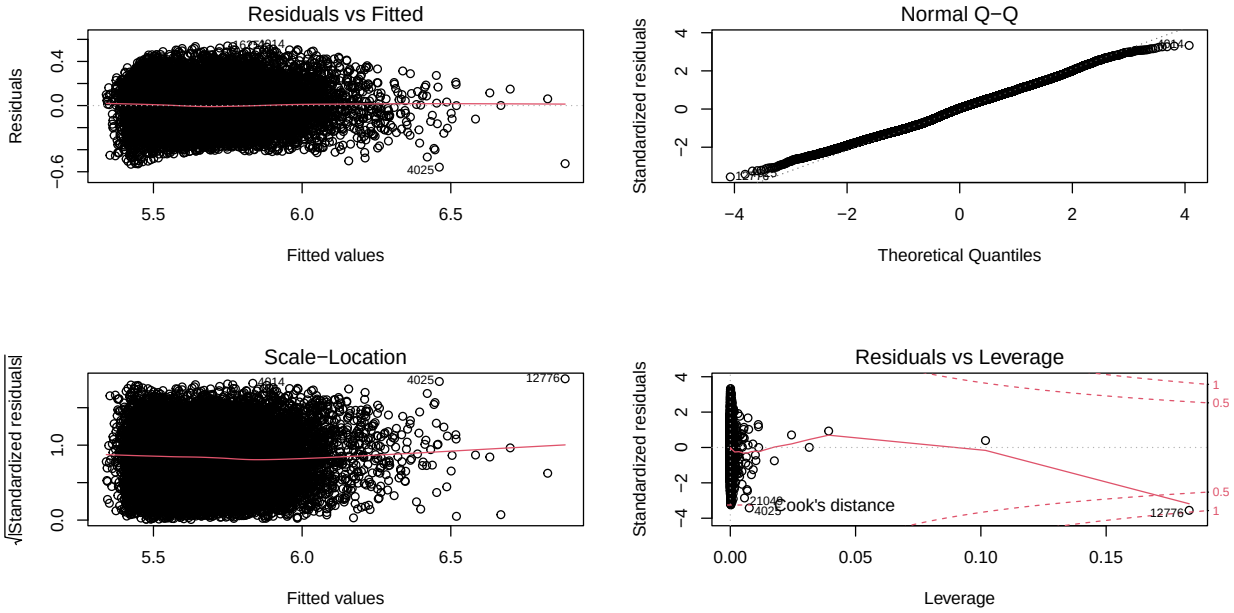
1. A simple model with a first and a second degree component of the variable *sqft_living*
2. A model with four variables:
    - *sqft_living*, with a second degree polynomial
    - *sqft_living15*
    - *lat*
    - *long*
3. One like the second but with a polynomial with degree 4 for the variable *lat*

We can now study them closely and try to evaluate which one is the best for our goals:

```
# linear model of poly(sqft_living) of deg=2
# Diagnostic
summary(mod.num2)
```

```
##
## Call:
## lm(formula = log10_price ~ sqft_living + I(sqft_living^2), data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55829 -0.12354  0.00576  0.11235  0.54606
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.265e+00  4.642e-03 1134.16   <2e-16 ***
## sqft_living       2.098e-04  3.493e-06   60.07   <2e-16 ***
## I(sqft_living^2) -6.659e-09  5.962e-10  -11.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1639 on 21597 degrees of freedom
## Multiple R-squared:  0.4858, Adjusted R-squared:  0.4858
## F-statistic: 1.02e+04 on 2 and 21597 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(mod.num2)
```

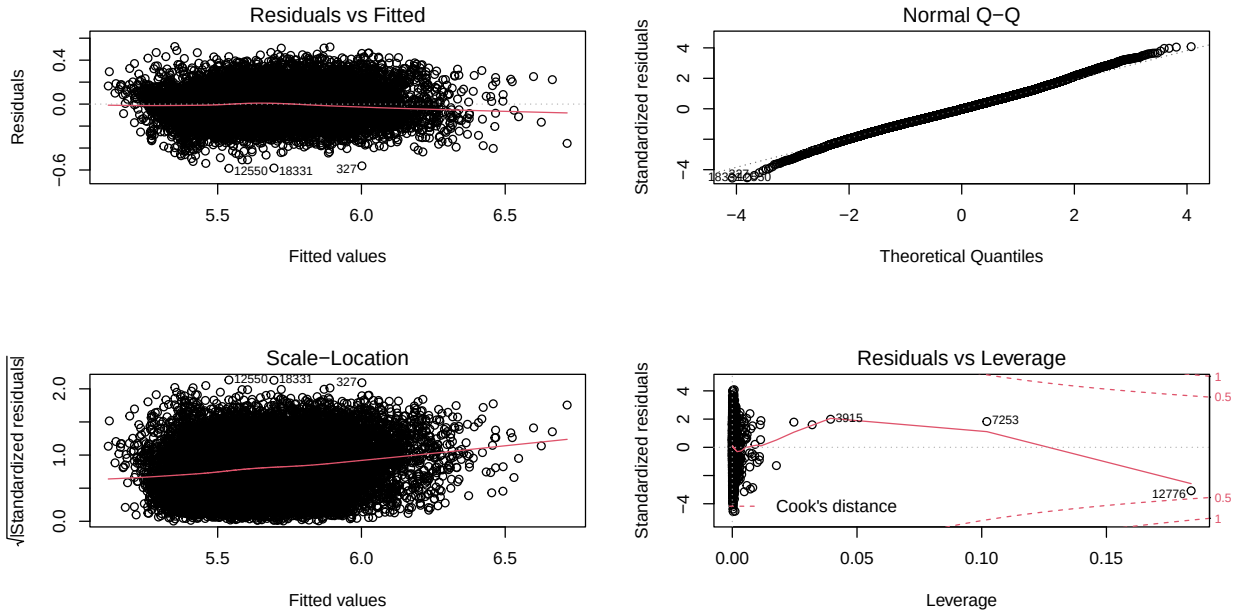



```
par(mfrow=c(1,1))
```

```
# First polinomial model
mod.p1 <- lm(log10_price ~ poly(sqft_living, 2) + sqft_living15 + lat + long,
                          data=kc_housing)
summary(mod.p1)
```

```
##
## Call:
## lm(formula = log10_price ~ poly(sqft_living, 2) + sqft_living15 +
##     lat + long, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58481 -0.08431 -0.00286  0.08170  0.52416
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -4.645e+01  8.268e-01  -56.18   <2e-16 ***
## poly(sqft_living, 2)1  1.749e+01  1.977e-01   88.47   <2e-16 ***
## poly(sqft_living, 2)2 -1.661e+00  1.301e-01  -12.77   <2e-16 ***
## sqft_living15          7.874e-05  2.030e-06   38.78   <2e-16 ***
## lat                    6.551e-01  6.403e-03  102.31   <2e-16 ***
## long                  -1.702e-01  6.693e-03  -25.43   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1285 on 21594 degrees of freedom
## Multiple R-squared:  0.6838, Adjusted R-squared:  0.6838
## F-statistic:  9341 on 5 and 21594 DF,  p-value: < 2.2e-16
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.p1)
```



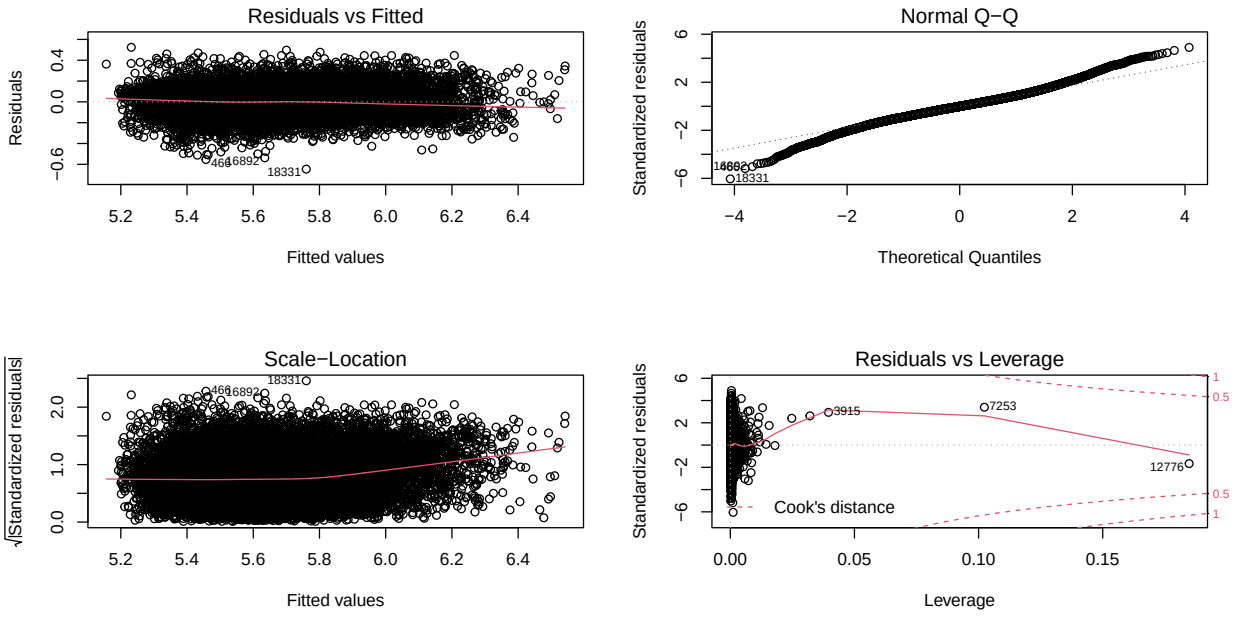

```
par(mfrow=c(1,1))
```

```
mod.p2 <- lm(log10_price ~ poly(sqft_living, 2) + sqft_living15 + poly(lat, 4) + long,
                           data=kc_housing)
summary(mod.p2)
```

```
##
## Call:
## lm(formula = log10_price ~ poly(sqft_living, 2) + sqft_living15 +
##     poly(lat, 4) + long, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64623 -0.06365 -0.00068  0.06138  0.52420
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1.354e+01  6.896e-01  -19.63   <2e-16 ***
## poly(sqft_living, 2)1    1.686e+01  1.648e-01  102.33   <2e-16 ***
## poly(sqft_living, 2)2 -2.140e+00  1.085e-01  -19.72   <2e-16 ***
## sqft_living15            7.538e-05  1.691e-06   44.57   <2e-16 ***
## poly(lat, 4)1            1.344e+01  1.086e-01  123.78   <2e-16 ***
## poly(lat, 4)2           -5.190e+00  1.073e-01  -48.35   <2e-16 ***
## poly(lat, 4)3           -8.752e+00  1.072e-01  -81.64   <2e-16 ***
```

```
## poly(lat, 4)4         -2.628e+00  1.082e-01  -24.29   <2e-16 ***
## long                  -1.559e-01  5.636e-03  -27.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.107 on 21591 degrees of freedom
## Multiple R-squared:  0.7809, Adjusted R-squared:  0.7808
## F-statistic:  9619 on 8 and 21591 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(mod.p2)
```




```
par(mfrow=c(1,1))
```

```
BIC(mod.num2)
```

```
## [1] -16803.39
```

```
BIC(mod.p1)
```

```
## [1] -27278.79
```

```
BIC(mod.p2)
```

```
## [1] -35169.62
```

From this process we can then say that the best model is the third one, with a BIC of -35169.62.
If we want to reduce the number of variables in our model we can use the second model with the price of an higher value of the BIC. For the moment we decide to keep them both and see in the final step which is the better option, when we will merge the numerical model with the categorical's.

3.2 Categorical Variables

In this section we applied a backward selection method to select the optimal model for the categorical variables.

First of all we studied the full model and step by step we removed the non-significant variables. Then we manually tried to remove one variable at a time and select the model with the higher value for the Adjusted R^2. At the end, we selected the model with lowest number of variables and an Adjusted R^2 value not so different from the one of the full model.

The first model containing all the categorical variables is the following one.

```
mod.cat0 <- lm(log10_price ~ bedrooms + bathrooms + floors + waterfront +
                condition + grade + view + yr_built + month_sold + year_sold +
                renovated + has_basement + zipcode, data=kc_housing)
summary(mod.cat0)
```

```
##
## Call:
## lm(formula = log10_price ~ bedrooms + bathrooms + floors + waterfront +
##     condition + grade + view + yr_built + month_sold + year_sold +
##     renovated + has_basement + zipcode, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59326 -0.05099  0.00086  0.05125  0.51775
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.466e+00  1.323e-01  48.888  < 2e-16 ***
## bedrooms1        -4.665e-02  3.735e-02  -1.249 0.211700
## bedrooms2        -2.294e-02  3.693e-02  -0.621 0.534458
## bedrooms3         1.644e-04  3.692e-02   0.004 0.996447
## bedrooms4         2.810e-02  3.693e-02   0.761 0.446768
## bedrooms5         3.388e-02  3.700e-02   0.916 0.359920
## bedrooms6         3.308e-02  3.736e-02   0.885 0.375936
## bedrooms7        -3.697e-03  3.983e-02  -0.093 0.926046
## bedrooms8         3.550e-02  4.472e-02   0.794 0.427346
## bedrooms9         1.292e-02  5.477e-02   0.236 0.813596
## bedrooms10       -2.162e-02  6.400e-02  -0.338 0.735463
## bedrooms11        1.183e-01  9.673e-02   1.223 0.221447
## bathrooms0.75     6.462e-02  4.601e-02   1.404 0.160197
## bathrooms1        7.552e-02  4.469e-02   1.690 0.091015 .
## bathrooms1.25     6.896e-02  5.370e-02   1.284 0.199146
## bathrooms1.5      9.580e-02  4.475e-02   2.141 0.032293 *
## bathrooms1.75     1.209e-01  4.472e-02   2.703 0.006885 **
## bathrooms2        1.212e-01  4.474e-02   2.710 0.006732 **
## bathrooms2.25     1.403e-01  4.475e-02   3.136 0.001715 **
## bathrooms2.5      1.583e-01  4.473e-02   3.539 0.000402 ***
## bathrooms2.75     1.699e-01  4.479e-02   3.794 0.000149 ***
## bathrooms3        1.752e-01  4.484e-02   3.908 9.33e-05 ***
## bathrooms3.25     1.947e-01  4.490e-02   4.336 1.46e-05 ***
## bathrooms3.5      1.989e-01  4.488e-02   4.431 9.43e-06 ***
## bathrooms3.75     2.258e-01  4.534e-02   4.981 6.39e-07 ***
## bathrooms4        2.292e-01  4.544e-02   5.044 4.60e-07 ***
```

```
## bathrooms4.25     2.593e-01  4.593e-02   5.646 1.67e-08 ***
## bathrooms4.5      2.389e-01  4.570e-02   5.227 1.74e-07 ***
## bathrooms4.75     2.941e-01  4.862e-02   6.048 1.49e-09 ***
## bathrooms5        2.705e-01  4.895e-02   5.527 3.29e-08 ***
## bathrooms5.25     2.578e-01  5.146e-02   5.010 5.49e-07 ***
## bathrooms5.5      3.309e-01  5.340e-02   6.197 5.86e-10 ***
## bathrooms5.75     2.489e-01  6.407e-02   3.885 0.000103 ***
## bathrooms6        2.842e-01  5.857e-02   4.853 1.22e-06 ***
## bathrooms6.25     4.262e-01  7.877e-02   5.411 6.33e-08 ***
## bathrooms6.5      3.156e-01  7.769e-02   4.062 4.88e-05 ***
## bathrooms6.75     1.722e-01  7.808e-02   2.205 0.027454 *
## bathrooms8        4.170e-01  7.959e-02   5.240 1.62e-07 ***
## floors1.5         1.908e-02  2.450e-03   7.788 7.13e-15 ***
## floors2           9.397e-03  2.043e-03   4.599 4.27e-06 ***
## floors2.5         2.229e-02  7.394e-03   3.014 0.002580 **
## floors3          -5.533e-02  4.659e-03 -11.876  < 2e-16 ***
## floors3.5        -4.261e-02  3.411e-02  -1.249 0.211553
## waterfrontTRUE    1.931e-01  8.969e-03  21.529  < 2e-16 ***
## condition2        3.546e-02  1.801e-02   1.969 0.048964 *
## condition3        8.684e-02  1.677e-02   5.179 2.25e-07 ***
## condition4        1.059e-01  1.677e-02   6.316 2.74e-10 ***
## condition5        1.297e-01  1.687e-02   7.683 1.62e-14 ***
## grade4           -1.863e-01  9.118e-02  -2.043 0.041066 *
## grade5           -1.652e-01  9.027e-02  -1.830 0.067329 .
## grade6           -1.031e-01  9.020e-02  -1.143 0.253080
## grade7           -3.382e-02  9.022e-02  -0.375 0.707757
## grade8            4.071e-02  9.024e-02   0.451 0.651898
## grade9            1.411e-01  9.026e-02   1.564 0.117918
## grade10           2.162e-01  9.029e-02   2.394 0.016673 *
## grade11           2.943e-01  9.039e-02   3.256 0.001131 **
## grade12           3.875e-01  9.085e-02   4.265 2.00e-05 ***
## grade13           5.031e-01  9.456e-02   5.321 1.04e-07 ***
## view1             7.077e-02  5.049e-03  14.015  < 2e-16 ***
## view2             6.374e-02  3.076e-03  20.725  < 2e-16 ***
## view3             9.903e-02  4.192e-03  23.626  < 2e-16 ***
## view4             1.481e-01  6.495e-03  22.798  < 2e-16 ***
## yr_built         -6.712e-04  3.911e-05 -17.162  < 2e-16 ***
## month_soldFeb     8.723e-03  3.819e-03   2.284 0.022394 *
## month_soldMar     1.979e-02  3.526e-03   5.613 2.02e-08 ***
## month_soldApr     2.944e-02  3.431e-03   8.579  < 2e-16 ***
## month_soldMay     4.123e-02  4.535e-03   9.092  < 2e-16 ***
## month_soldJun     4.981e-02  5.364e-03   9.286  < 2e-16 ***
## month_soldJul     4.848e-02  5.358e-03   9.049  < 2e-16 ***
## month_soldAug     4.991e-02  5.407e-03   9.231  < 2e-16 ***
## month_soldSep     4.620e-02  5.438e-03   8.497  < 2e-16 ***
## month_soldOct     4.698e-02  5.421e-03   8.667  < 2e-16 ***
## month_soldNov     4.552e-02  5.548e-03   8.205 2.43e-16 ***
## month_soldDec     5.121e-02  5.525e-03   9.268  < 2e-16 ***
## year_sold2015     4.968e-02  4.115e-03  12.073  < 2e-16 ***
## renovatedTRUE     2.443e-02  3.301e-03   7.401 1.40e-13 ***
## has_basementTRUE  8.983e-03  1.578e-03   5.692 1.27e-08 ***
## zipcode98002     -2.314e-02  7.906e-03  -2.927 0.003430 **
## zipcode98003     -4.974e-03  7.116e-03  -0.699 0.484581
## zipcode98004      4.871e-01  6.985e-03  69.745  < 2e-16 ***
```

```
## zipcode98005      3.195e-01  8.429e-03  37.897  < 2e-16 ***
## zipcode98006      2.737e-01  6.289e-03  43.514  < 2e-16 ***
## zipcode98007      2.680e-01  8.906e-03  30.093  < 2e-16 ***
## zipcode98008      2.681e-01  7.134e-03  37.574  < 2e-16 ***
## zipcode98010      1.344e-01  1.010e-02  13.306  < 2e-16 ***
## zipcode98011      1.999e-01  7.946e-03  25.160  < 2e-16 ***
## zipcode98014      1.712e-01  9.321e-03  18.364  < 2e-16 ***
## zipcode98019      1.630e-01  8.013e-03  20.345  < 2e-16 ***
## zipcode98022      3.510e-02  7.549e-03   4.650 3.33e-06 ***
## zipcode98023     -2.116e-02  6.183e-03  -3.422 0.000623 ***
## zipcode98024      2.271e-01  1.105e-02  20.543  < 2e-16 ***
## zipcode98027      2.274e-01  6.489e-03  35.040  < 2e-16 ***
## zipcode98028      1.804e-01  7.097e-03  25.422  < 2e-16 ***
## zipcode98029      2.335e-01  6.912e-03  33.775  < 2e-16 ***
## zipcode98030      1.805e-02  7.288e-03   2.476 0.013288 *
## zipcode98031      2.339e-02  7.160e-03   3.266 0.001092 **
## zipcode98032     -2.686e-02  9.280e-03  -2.894 0.003804 **
## zipcode98033      3.295e-01  6.396e-03  51.509  < 2e-16 ***
## zipcode98034      2.206e-01  6.070e-03  36.344  < 2e-16 ***
## zipcode98038      8.235e-02  5.988e-03  13.754  < 2e-16 ***
## zipcode98039      5.699e-01  1.381e-02  41.268  < 2e-16 ***
## zipcode98040      3.722e-01  7.261e-03  51.259  < 2e-16 ***
## zipcode98042      2.982e-02  6.055e-03   4.925 8.48e-07 ***
## zipcode98045      1.565e-01  7.641e-03  20.481  < 2e-16 ***
## zipcode98052      2.730e-01  6.033e-03  45.245  < 2e-16 ***
## zipcode98053      2.944e-01  6.527e-03  45.102  < 2e-16 ***
## zipcode98055      4.908e-02  7.208e-03   6.810 1.00e-11 ***
## zipcode98056      1.337e-01  6.479e-03  20.642  < 2e-16 ***
## zipcode98058      6.837e-02  6.297e-03  10.858  < 2e-16 ***
## zipcode98059      1.621e-01  6.271e-03  25.844  < 2e-16 ***
## zipcode98065      2.012e-01  6.957e-03  28.922  < 2e-16 ***
## zipcode98070      1.585e-01  9.656e-03  16.414  < 2e-16 ***
## zipcode98072      2.268e-01  7.183e-03  31.576  < 2e-16 ***
## zipcode98074      2.404e-01  6.408e-03  37.511  < 2e-16 ***
## zipcode98075      2.580e-01  6.767e-03  38.122  < 2e-16 ***
## zipcode98077      2.294e-01  7.967e-03  28.795  < 2e-16 ***
## zipcode98092      1.753e-02  6.701e-03   2.616 0.008909 **
## zipcode98102      3.504e-01  1.019e-02  34.394  < 2e-16 ***
## zipcode98103      3.127e-01  6.246e-03  50.066  < 2e-16 ***
## zipcode98105      3.607e-01  7.783e-03  46.348  < 2e-16 ***
## zipcode98106      9.824e-02  6.841e-03  14.361  < 2e-16 ***
## zipcode98107      3.103e-01  7.426e-03  41.789  < 2e-16 ***
## zipcode98108      1.188e-01  8.119e-03  14.633  < 2e-16 ***
## zipcode98109      3.750e-01  9.957e-03  37.660  < 2e-16 ***
## zipcode98112      4.090e-01  7.500e-03  54.532  < 2e-16 ***
## zipcode98115      3.177e-01  6.141e-03  51.726  < 2e-16 ***
## zipcode98116      2.770e-01  6.952e-03  39.840  < 2e-16 ***
## zipcode98117      3.083e-01  6.213e-03  49.624  < 2e-16 ***
## zipcode98118      1.654e-01  6.265e-03  26.397  < 2e-16 ***
## zipcode98119      3.563e-01  8.328e-03  42.787  < 2e-16 ***
## zipcode98122      2.857e-01  7.253e-03  39.384  < 2e-16 ***
## zipcode98125      2.263e-01  6.521e-03  34.697  < 2e-16 ***
## zipcode98126      1.929e-01  6.791e-03  28.399  < 2e-16 ***
## zipcode98133      1.758e-01  6.244e-03  28.155  < 2e-16 ***
```

```
## zipcode98136      2.492e-01  7.365e-03  33.838  < 2e-16 ***
## zipcode98144      2.422e-01  6.890e-03  35.157  < 2e-16 ***
## zipcode98146      9.743e-02  7.110e-03  13.704  < 2e-16 ***
## zipcode98148      5.651e-02  1.275e-02   4.433 9.35e-06 ***
## zipcode98155      1.692e-01  6.358e-03  26.609  < 2e-16 ***
## zipcode98166      1.266e-01  7.371e-03  17.169  < 2e-16 ***
## zipcode98168      2.239e-02  7.274e-03   3.079 0.002082 **
## zipcode98177      2.458e-01  7.404e-03  33.198  < 2e-16 ***
## zipcode98178      4.416e-02  7.314e-03   6.038 1.59e-09 ***
## zipcode98188      2.723e-02  8.994e-03   3.028 0.002469 **
## zipcode98198      1.108e-02  7.134e-03   1.552 0.120574
## zipcode98199      3.267e-01  7.031e-03  46.470  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08911 on 21454 degrees of freedom
## Multiple R-squared:  0.8489, Adjusted R-squared:  0.8479
## F-statistic: 831.5 on 145 and 21454 DF,  p-value: < 2.2e-16
```

```
summary(mod.cat0)$adj.r.squared
```

```
## [1] 0.8479209
```

```
BIC(mod.cat0)
```

```
## [1] -41835.03
```

```
tabr2 <- c(summary(mod.cat0)$adj.r.squared)
tabbic <- c(BIC(mod.cat0))
```

We decided to remove the variable *bedrooms* because it is not-significant.

```
#FULL MODEL MINUS BEDROOMS
mod.cat1 <- lm(log10_price ~ bathrooms + floors + waterfront + condition +
                 grade + view + yr_built + month_sold + year_sold + renovated +
                 has_basement + zipcode, data=kc_housing)
summary(mod.cat1)
```

```
##
## Call:
## lm(formula = log10_price ~ bathrooms + floors + waterfront +
##     condition + grade + view + yr_built + month_sold + year_sold +
##     renovated + has_basement + zipcode, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.60436 -0.05165  0.00175  0.05285  0.51148
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.666e+00  1.279e-01  52.126  < 2e-16 ***
```

```
## bathrooms0.75     6.140e-02  4.665e-02   1.316 0.188129
## bathrooms1        8.059e-02  4.529e-02   1.779 0.075181 .
## bathrooms1.25     6.702e-02  5.445e-02   1.231 0.218391
## bathrooms1.5      1.083e-01  4.535e-02   2.389 0.016919 *
## bathrooms1.75     1.372e-01  4.532e-02   3.028 0.002466 **
## bathrooms2        1.400e-01  4.534e-02   3.088 0.002015 **
## bathrooms2.25     1.610e-01  4.535e-02   3.550 0.000387 ***
## bathrooms2.5      1.835e-01  4.533e-02   4.049 5.15e-05 ***
## bathrooms2.75     2.007e-01  4.538e-02   4.422 9.84e-06 ***
## bathrooms3        2.070e-01  4.543e-02   4.556 5.24e-06 ***
## bathrooms3.25     2.256e-01  4.548e-02   4.959 7.13e-07 ***
## bathrooms3.5      2.329e-01  4.546e-02   5.124 3.02e-07 ***
## bathrooms3.75     2.623e-01  4.592e-02   5.712 1.13e-08 ***
## bathrooms4        2.652e-01  4.602e-02   5.763 8.39e-09 ***
## bathrooms4.25     2.952e-01  4.652e-02   6.346 2.26e-10 ***
## bathrooms4.5      2.775e-01  4.626e-02   5.999 2.02e-09 ***
## bathrooms4.75     3.304e-01  4.921e-02   6.714 1.94e-11 ***
## bathrooms5        3.111e-01  4.953e-02   6.280 3.45e-10 ***
## bathrooms5.25     2.929e-01  5.195e-02   5.638 1.74e-08 ***
## bathrooms5.5      3.654e-01  5.406e-02   6.758 1.43e-11 ***
## bathrooms5.75     2.793e-01  6.477e-02   4.312 1.63e-05 ***
## bathrooms6        3.260e-01  5.914e-02   5.512 3.59e-08 ***
## bathrooms6.25     4.664e-01  7.981e-02   5.844 5.16e-09 ***
## bathrooms6.5      3.539e-01  7.867e-02   4.498 6.88e-06 ***
## bathrooms6.75     1.962e-01  7.874e-02   2.492 0.012701 *
## bathrooms8        4.452e-01  8.015e-02   5.555 2.81e-08 ***
## floors1.5         2.800e-02  2.452e-03  11.417  < 2e-16 ***
## floors2           1.231e-02  2.066e-03   5.955 2.64e-09 ***
## floors2.5         2.614e-02  7.473e-03   3.498 0.000470 ***
## floors3          -5.690e-02  4.719e-03 -12.057  < 2e-16 ***
## floors3.5        -5.040e-02  3.440e-02  -1.465 0.142901
## waterfrontTRUE    1.839e-01  9.084e-03  20.240  < 2e-16 ***
## condition2        3.677e-02  1.826e-02   2.014 0.044009 *
## condition3        8.865e-02  1.700e-02   5.216 1.85e-07 ***
## condition4        1.084e-01  1.700e-02   6.374 1.89e-10 ***
## condition5        1.322e-01  1.710e-02   7.730 1.12e-14 ***
## grade4           -1.729e-01  9.237e-02  -1.872 0.061218 .
## grade5           -1.420e-01  9.144e-02  -1.553 0.120432
## grade6           -7.315e-02  9.133e-02  -0.801 0.423223
## grade7            2.300e-03  9.135e-02   0.025 0.979909
## grade8            7.787e-02  9.137e-02   0.852 0.394056
## grade9            1.820e-01  9.139e-02   1.992 0.046410 *
## grade10           2.560e-01  9.142e-02   2.800 0.005112 **
## grade11           3.356e-01  9.152e-02   3.667 0.000246 ***
## grade12           4.288e-01  9.198e-02   4.662 3.15e-06 ***
## grade13           5.473e-01  9.574e-02   5.717 1.10e-08 ***
## view1             6.931e-02  5.118e-03  13.541  < 2e-16 ***
## view2             6.210e-02  3.117e-03  19.922  < 2e-16 ***
## view3             9.725e-02  4.249e-03  22.889  < 2e-16 ***
## view4             1.454e-01  6.583e-03  22.092  < 2e-16 ***
## yr_built         -7.960e-04  3.919e-05 -20.310  < 2e-16 ***
## month_soldFeb     8.331e-03  3.872e-03   2.152 0.031444 *
## month_soldMar     1.986e-02  3.575e-03   5.553 2.83e-08 ***
## month_soldApr     2.929e-02  3.479e-03   8.420  < 2e-16 ***
```

```
## month_soldMay     3.987e-02  4.596e-03   8.674  < 2e-16 ***
## month_soldJun     4.830e-02  5.437e-03   8.884  < 2e-16 ***
## month_soldJul     4.723e-02  5.431e-03   8.696  < 2e-16 ***
## month_soldAug     4.813e-02  5.481e-03   8.782  < 2e-16 ***
## month_soldSep     4.461e-02  5.512e-03   8.093 6.14e-16 ***
## month_soldOct     4.567e-02  5.495e-03   8.311  < 2e-16 ***
## month_soldNov     4.414e-02  5.623e-03   7.849 4.40e-15 ***
## month_soldDec     5.000e-02  5.600e-03   8.929  < 2e-16 ***
## year_sold2015     4.870e-02  4.171e-03  11.674  < 2e-16 ***
## renovatedTRUE     2.067e-02  3.340e-03   6.189 6.16e-10 ***
## has_basementTRUE  1.237e-02  1.593e-03   7.767 8.40e-15 ***
## zipcode98002     -2.331e-02  8.014e-03  -2.908 0.003637 **
## zipcode98003     -7.932e-03  7.214e-03  -1.100 0.271557
## zipcode98004      4.858e-01  7.075e-03  68.669  < 2e-16 ***
## zipcode98005      3.191e-01  8.545e-03  37.343  < 2e-16 ***
## zipcode98006      2.726e-01  6.374e-03  42.773  < 2e-16 ***
## zipcode98007      2.682e-01  9.021e-03  29.729  < 2e-16 ***
## zipcode98008      2.706e-01  7.231e-03  37.427  < 2e-16 ***
## zipcode98010      1.324e-01  1.024e-02  12.930  < 2e-16 ***
## zipcode98011      1.967e-01  8.054e-03  24.427  < 2e-16 ***
## zipcode98014      1.623e-01  9.444e-03  17.189  < 2e-16 ***
## zipcode98019      1.592e-01  8.123e-03  19.598  < 2e-16 ***
## zipcode98022      3.041e-02  7.652e-03   3.974 7.09e-05 ***
## zipcode98023     -2.173e-02  6.269e-03  -3.466 0.000530 ***
## zipcode98024      2.228e-01  1.121e-02  19.878  < 2e-16 ***
## zipcode98027      2.223e-01  6.575e-03  33.807  < 2e-16 ***
## zipcode98028      1.788e-01  7.196e-03  24.847  < 2e-16 ***
## zipcode98029      2.269e-01  7.003e-03  32.403  < 2e-16 ***
## zipcode98030      1.814e-02  7.389e-03   2.455 0.014085 *
## zipcode98031      2.323e-02  7.260e-03   3.200 0.001376 **
## zipcode98032     -2.551e-02  9.409e-03  -2.711 0.006712 **
## zipcode98033      3.271e-01  6.485e-03  50.442  < 2e-16 ***
## zipcode98034      2.187e-01  6.154e-03  35.536  < 2e-16 ***
## zipcode98038      8.005e-02  6.071e-03  13.186  < 2e-16 ***
## zipcode98039      5.678e-01  1.400e-02  40.562  < 2e-16 ***
## zipcode98040      3.721e-01  7.357e-03  50.575  < 2e-16 ***
## zipcode98042      2.894e-02  6.139e-03   4.714 2.44e-06 ***
## zipcode98045      1.519e-01  7.745e-03  19.612  < 2e-16 ***
## zipcode98052      2.712e-01  6.117e-03  44.333  < 2e-16 ***
## zipcode98053      2.858e-01  6.600e-03  43.303  < 2e-16 ***
## zipcode98055      4.472e-02  7.305e-03   6.121 9.46e-10 ***
## zipcode98056      1.337e-01  6.569e-03  20.353  < 2e-16 ***
## zipcode98058      6.886e-02  6.385e-03  10.784  < 2e-16 ***
## zipcode98059      1.645e-01  6.357e-03  25.883  < 2e-16 ***
## zipcode98065      1.982e-01  7.051e-03  28.106  < 2e-16 ***
## zipcode98070      1.469e-01  9.771e-03  15.039  < 2e-16 ***
## zipcode98072      2.234e-01  7.282e-03  30.672  < 2e-16 ***
## zipcode98074      2.380e-01  6.497e-03  36.630  < 2e-16 ***
## zipcode98075      2.587e-01  6.860e-03  37.718  < 2e-16 ***
## zipcode98077      2.271e-01  8.077e-03  28.115  < 2e-16 ***
## zipcode98092      1.687e-02  6.795e-03   2.482 0.013059 *
## zipcode98102      3.325e-01  1.030e-02  32.286  < 2e-16 ***
## zipcode98103      2.991e-01  6.302e-03  47.453  < 2e-16 ***
## zipcode98105      3.485e-01  7.851e-03  44.395  < 2e-16 ***
```

```
## zipcode98106      9.129e-02  6.923e-03  13.186  < 2e-16 ***
## zipcode98107      2.953e-01  7.495e-03  39.396  < 2e-16 ***
## zipcode98108      1.103e-01  8.222e-03  13.413  < 2e-16 ***
## zipcode98109      3.590e-01  1.007e-02  35.632  < 2e-16 ***
## zipcode98112      3.956e-01  7.577e-03  52.212  < 2e-16 ***
## zipcode98115      3.086e-01  6.209e-03  49.700  < 2e-16 ***
## zipcode98116      2.642e-01  7.027e-03  37.593  < 2e-16 ***
## zipcode98117      2.956e-01  6.273e-03  47.123  < 2e-16 ***
## zipcode98118      1.572e-01  6.339e-03  24.793  < 2e-16 ***
## zipcode98119      3.410e-01  8.414e-03  40.520  < 2e-16 ***
## zipcode98122      2.711e-01  7.326e-03  37.000  < 2e-16 ***
## zipcode98125      2.211e-01  6.606e-03  33.469  < 2e-16 ***
## zipcode98126      1.808e-01  6.862e-03  26.340  < 2e-16 ***
## zipcode98133      1.703e-01  6.324e-03  26.935  < 2e-16 ***
## zipcode98136      2.366e-01  7.445e-03  31.778  < 2e-16 ***
## zipcode98144      2.305e-01  6.963e-03  33.101  < 2e-16 ***
## zipcode98146      9.489e-02  7.207e-03  13.166  < 2e-16 ***
## zipcode98148      5.334e-02  1.293e-02   4.126 3.70e-05 ***
## zipcode98155      1.669e-01  6.444e-03  25.899  < 2e-16 ***
## zipcode98166      1.247e-01  7.474e-03  16.691  < 2e-16 ***
## zipcode98168      1.948e-02  7.373e-03   2.643 0.008231 **
## zipcode98177      2.406e-01  7.504e-03  32.058  < 2e-16 ***
## zipcode98178      4.371e-02  7.415e-03   5.895 3.80e-09 ***
## zipcode98188      2.767e-02  9.118e-03   3.035 0.002411 **
## zipcode98198      8.203e-03  7.232e-03   1.134 0.256703
## zipcode98199      3.145e-01  7.109e-03  44.240  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09036 on 21465 degrees of freedom
## Multiple R-squared:  0.8446, Adjusted R-squared:  0.8436
## F-statistic: 870.6 on 134 and 21465 DF,  p-value: < 2.2e-16
```

```
summary(mod.cat1)$adj.r.squared
```

```
## [1] 0.8436285
```

```
BIC(mod.cat1)
```

```
## [1] -41332.54
```

```
tabr2 <- c(tabr2, summary(mod.cat1)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat1))
```

Then we decided to remove the variable *grade* as it is not-significant.

```
#FULL MODEL MINUS BEDROOMS, GRADE
mod.cat2 <- lm(log10_price ~ bathrooms + floors + waterfront + condition +
                 view + yr_built + month_sold + year_sold + renovated +
                 has_basement + zipcode, data=kc_housing)
summary(mod.cat2)
```

```
##
## Call:
## lm(formula = log10_price ~ bathrooms + floors + waterfront +
##     condition + view + yr_built + month_sold + year_sold + renovated +
##     has_basement + zipcode, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68622 -0.06533 -0.00040  0.06395  0.60866
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.405e+00  1.080e-01  50.052  < 2e-16 ***
## bathrooms0.75     3.022e-02  5.663e-02   0.534 0.593549
## bathrooms1        1.039e-01  5.513e-02   1.884 0.059575 .
## bathrooms1.25     1.263e-01  6.628e-02   1.906 0.056721 .
## bathrooms1.5      1.553e-01  5.518e-02   2.815 0.004884 **
## bathrooms1.75     1.928e-01  5.514e-02   3.497 0.000472 ***
## bathrooms2        1.937e-01  5.516e-02   3.511 0.000447 ***
## bathrooms2.25     2.321e-01  5.517e-02   4.207 2.60e-05 ***
## bathrooms2.5      2.728e-01  5.515e-02   4.946 7.64e-07 ***
## bathrooms2.75     2.991e-01  5.522e-02   5.417 6.12e-08 ***
## bathrooms3        3.073e-01  5.527e-02   5.560 2.73e-08 ***
## bathrooms3.25     3.765e-01  5.533e-02   6.804 1.05e-11 ***
## bathrooms3.5      3.865e-01  5.531e-02   6.989 2.85e-12 ***
## bathrooms3.75     4.355e-01  5.586e-02   7.797 6.63e-15 ***
## bathrooms4        4.542e-01  5.596e-02   8.116 5.06e-16 ***
## bathrooms4.25     4.972e-01  5.656e-02   8.792  < 2e-16 ***
## bathrooms4.5      4.632e-01  5.625e-02   8.236  < 2e-16 ***
## bathrooms4.75     5.600e-01  5.978e-02   9.368  < 2e-16 ***
## bathrooms5        5.183e-01  6.019e-02   8.611  < 2e-16 ***
## bathrooms5.25     5.025e-01  6.311e-02   7.961 1.79e-15 ***
## bathrooms5.5      6.899e-01  6.528e-02  10.567  < 2e-16 ***
## bathrooms5.75     5.708e-01  7.809e-02   7.310 2.77e-13 ***
## bathrooms6        5.910e-01  7.128e-02   8.292  < 2e-16 ***
## bathrooms6.25     8.130e-01  9.557e-02   8.507  < 2e-16 ***
## bathrooms6.5      6.830e-01  9.549e-02   7.153 8.76e-13 ***
## bathrooms6.75     5.067e-01  9.561e-02   5.300 1.17e-07 ***
## bathrooms8        8.353e-01  9.583e-02   8.717  < 2e-16 ***
## floors1.5         4.316e-02  2.978e-03  14.496  < 2e-16 ***
## floors2           4.231e-02  2.485e-03  17.027  < 2e-16 ***
## floors2.5         8.965e-02  9.057e-03   9.899  < 2e-16 ***
## floors3          -5.008e-02  5.718e-03  -8.759  < 2e-16 ***
## floors3.5        -2.455e-02  4.184e-02  -0.587 0.557326
## waterfrontTRUE    1.724e-01  1.105e-02  15.603  < 2e-16 ***
## condition2        7.528e-02  2.217e-02   3.395 0.000688 ***
## condition3        1.499e-01  2.059e-02   7.281 3.44e-13 ***
## condition4        1.657e-01  2.059e-02   8.047 8.94e-16 ***
## condition5        1.864e-01  2.072e-02   8.999  < 2e-16 ***
## view1             9.910e-02  6.218e-03  15.937  < 2e-16 ***
## view2             1.053e-01  3.757e-03  28.031  < 2e-16 ***
## view3             1.585e-01  5.119e-03  30.963  < 2e-16 ***
## view4             2.336e-01  7.930e-03  29.462  < 2e-16 ***
## yr_built         -2.149e-04  4.653e-05  -4.619 3.87e-06 ***
```

```
## month_soldFeb     6.403e-03  4.713e-03   1.359 0.174306
## month_soldMar     1.951e-02  4.352e-03   4.482 7.42e-06 ***
## month_soldApr     2.963e-02  4.235e-03   6.996 2.71e-12 ***
## month_soldMay     4.005e-02  5.597e-03   7.156 8.60e-13 ***
## month_soldJun     4.773e-02  6.619e-03   7.210 5.77e-13 ***
## month_soldJul     4.587e-02  6.613e-03   6.937 4.12e-12 ***
## month_soldAug     4.632e-02  6.672e-03   6.943 3.96e-12 ***
## month_soldSep     3.984e-02  6.712e-03   5.937 2.96e-09 ***
## month_soldOct     4.185e-02  6.690e-03   6.256 4.03e-10 ***
## month_soldNov     4.062e-02  6.847e-03   5.932 3.03e-09 ***
## month_soldDec     4.901e-02  6.819e-03   7.188 6.78e-13 ***
## year_sold2015     4.528e-02  5.078e-03   8.918  < 2e-16 ***
## renovatedTRUE     2.132e-02  4.056e-03   5.257 1.48e-07 ***
## has_basementTRUE  9.381e-03  1.932e-03   4.856 1.21e-06 ***
## zipcode98002     -4.966e-02  9.747e-03  -5.095 3.52e-07 ***
## zipcode98003      1.235e-02  8.775e-03   1.407 0.159462
## zipcode98004      5.758e-01  8.544e-03  67.391  < 2e-16 ***
## zipcode98005      4.110e-01  1.034e-02  39.765  < 2e-16 ***
## zipcode98006      3.609e-01  7.680e-03  46.988  < 2e-16 ***
## zipcode98007      3.235e-01  1.096e-02  29.519  < 2e-16 ***
## zipcode98008      3.050e-01  8.790e-03  34.695  < 2e-16 ***
## zipcode98010      1.425e-01  1.246e-02  11.437  < 2e-16 ***
## zipcode98011      2.217e-01  9.799e-03  22.625  < 2e-16 ***
## zipcode98014      1.674e-01  1.148e-02  14.575  < 2e-16 ***
## zipcode98019      1.478e-01  9.888e-03  14.944  < 2e-16 ***
## zipcode98022      2.559e-02  9.302e-03   2.751 0.005941 **
## zipcode98023      2.268e-03  7.621e-03   0.298 0.766010
## zipcode98024      2.436e-01  1.364e-02  17.861  < 2e-16 ***
## zipcode98027      2.596e-01  7.983e-03  32.518  < 2e-16 ***
## zipcode98028      1.984e-01  8.755e-03  22.660  < 2e-16 ***
## zipcode98029      2.620e-01  8.502e-03  30.819  < 2e-16 ***
## zipcode98030      1.822e-02  8.998e-03   2.025 0.042869 *
## zipcode98031      2.384e-02  8.838e-03   2.697 0.007005 **
## zipcode98032     -7.900e-03  1.145e-02  -0.690 0.490296
## zipcode98033      3.709e-01  7.875e-03  47.097  < 2e-16 ***
## zipcode98034      2.346e-01  7.489e-03  31.332  < 2e-16 ***
## zipcode98038      7.022e-02  7.391e-03   9.500  < 2e-16 ***
## zipcode98039      6.906e-01  1.697e-02  40.696  < 2e-16 ***
## zipcode98040      4.656e-01  8.880e-03  52.431  < 2e-16 ***
## zipcode98042      3.038e-02  7.474e-03   4.065 4.81e-05 ***
## zipcode98045      1.654e-01  9.429e-03  17.540  < 2e-16 ***
## zipcode98052      3.224e-01  7.413e-03  43.490  < 2e-16 ***
## zipcode98053      3.269e-01  8.016e-03  40.776  < 2e-16 ***
## zipcode98055      4.188e-02  8.892e-03   4.709 2.50e-06 ***
## zipcode98056      1.392e-01  7.986e-03  17.433  < 2e-16 ***
## zipcode98058      9.146e-02  7.765e-03  11.779  < 2e-16 ***
## zipcode98059      1.874e-01  7.729e-03  24.249  < 2e-16 ***
## zipcode98065      1.888e-01  8.584e-03  21.997  < 2e-16 ***
## zipcode98070      1.431e-01  1.189e-02  12.034  < 2e-16 ***
## zipcode98072      2.750e-01  8.843e-03  31.103  < 2e-16 ***
## zipcode98074      3.172e-01  7.844e-03  40.440  < 2e-16 ***
## zipcode98075      3.413e-01  8.266e-03  41.292  < 2e-16 ***
## zipcode98077      3.220e-01  9.762e-03  32.986  < 2e-16 ***
## zipcode98092      4.018e-02  8.264e-03   4.862 1.17e-06 ***
```

```
## zipcode98102      4.075e-01  1.250e-02  32.609  < 2e-16 ***
## zipcode98103      3.262e-01  7.649e-03  42.650  < 2e-16 ***
## zipcode98105      4.024e-01  9.528e-03  42.229  < 2e-16 ***
## zipcode98106      7.656e-02  8.414e-03   9.099  < 2e-16 ***
## zipcode98107      3.123e-01  9.111e-03  34.274  < 2e-16 ***
## zipcode98108      1.068e-01  1.001e-02  10.671  < 2e-16 ***
## zipcode98109      4.182e-01  1.223e-02  34.182  < 2e-16 ***
## zipcode98112      4.832e-01  9.153e-03  52.793  < 2e-16 ***
## zipcode98115      3.398e-01  7.541e-03  45.063  < 2e-16 ***
## zipcode98116      2.924e-01  8.542e-03  34.234  < 2e-16 ***
## zipcode98117      3.242e-01  7.625e-03  42.522  < 2e-16 ***
## zipcode98118      1.610e-01  7.708e-03  20.885  < 2e-16 ***
## zipcode98119      3.935e-01  1.021e-02  38.544  < 2e-16 ***
## zipcode98122      3.130e-01  8.888e-03  35.216  < 2e-16 ***
## zipcode98125      2.339e-01  8.038e-03  29.101  < 2e-16 ***
## zipcode98126      1.880e-01  8.346e-03  22.528  < 2e-16 ***
## zipcode98133      1.849e-01  7.694e-03  24.034  < 2e-16 ***
## zipcode98136      2.551e-01  9.056e-03  28.170  < 2e-16 ***
## zipcode98144      2.452e-01  8.471e-03  28.943  < 2e-16 ***
## zipcode98146      8.983e-02  8.763e-03  10.251  < 2e-16 ***
## zipcode98148      5.362e-02  1.574e-02   3.407 0.000658 ***
## zipcode98155      1.810e-01  7.841e-03  23.087  < 2e-16 ***
## zipcode98166      1.457e-01  9.088e-03  16.036  < 2e-16 ***
## zipcode98168      1.004e-02  8.957e-03   1.121 0.262321
## zipcode98177      2.920e-01  9.099e-03  32.097  < 2e-16 ***
## zipcode98178      3.125e-02  9.019e-03   3.464 0.000532 ***
## zipcode98188      2.676e-02  1.110e-02   2.410 0.015954 *
## zipcode98198      6.383e-03  8.805e-03   0.725 0.468519
## zipcode98199      3.701e-01  8.623e-03  42.913  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.11 on 21475 degrees of freedom
## Multiple R-squared:  0.7695, Adjusted R-squared:  0.7681
## F-statistic:   578 on 124 and 21475 DF,  p-value: < 2.2e-16
```

```
summary(mod.cat2)$adj.r.squared
```

```
## [1] 0.7681203
```

```
BIC(mod.cat2)
```

```
## [1] -32912.22
```

```
tabr2 <- c(tabr2, summary(mod.cat2)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat2))
```

From now we remove the variable which gives us the model with highest Adjusted R^2:

```
summary(mod.cat3)$adj.r.squared
```

```
## [1] 0.7679007
```

```
BIC(mod.cat3)
```

```
## [1] -32900.75
```

```
tabr2 <- c(tabr2, summary(mod.cat3)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat3))
```

```
summary(mod.cat4)$adj.r.squared
```

```
## [1] 0.7676618
```

```
BIC(mod.cat4)
```

```
## [1] -32887.51
```

```
tabr2 <- c(tabr2, summary(mod.cat4)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat4))
```

```
summary(mod.cat5)$adj.r.squared
```

```
## [1] 0.7671217
```

```
BIC(mod.cat5)
```

```
## [1] -32846.32
```

```
tabr2 <- c(tabr2, summary(mod.cat5)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat5))
```

```
summary(mod.cat6)$adj.r.squared
```

```
## [1] 0.7662779
```

```
BIC(mod.cat6)
```

```
## [1] -32777.18
```

```
tabr2 <- c(tabr2, summary(mod.cat6)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat6))
```

```
summary(mod.cat7)$adj.r.squared
```

```
## [1] 0.7643694
```

```
BIC(mod.cat7)
```

```
## [1] -32700.24
```

```
tabr2 <- c(tabr2, summary(mod.cat7)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat7))
```

```
summary(mod.cat8)$adj.r.squared
```

```
## [1] 0.7615568
```

```
BIC(mod.cat8)#THIS IS THE GOOD ONE!!!!
```

```
## [1] -32452.91
```

```
tabr2 <- c(tabr2, summary(mod.cat8)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat8))
```

```
summary(mod.cat9)$adj.r.squared
```

```
## [1] 0.757967
```

```
BIC(mod.cat9)
```

```
## [1] -32166.05
```

```
tabr2 <- c(tabr2, summary(mod.cat9)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat9))
```

```
summary(mod.cat10)$adj.r.squared
```

```
## [1] 0.7483026
```

```
BIC(mod.cat10)
```

```
## [1] -31365.21
```

```
tabr2 <- c(tabr2, summary(mod.cat10)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat10))
```

```
summary(mod.cat11)$adj.r.squared
```

```
## [1] 0.7047706
```

```
BIC(mod.cat11)
```

```
## [1] -27955.37
```

```
tabr2 <- c(tabr2, summary(mod.cat11)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat11))
```

```
summary(mod.cat12)$adj.r.squared
```

```
## [1] 0.5281227
```

```
BIC(mod.cat12)
```

```
## [1] -18059.09
```

With the following table, we summarize the values of the Adjusted R^2 and BIC obtained with the different categorical models, we can see that the difference between Model 2 (the full model without the two non-significant variables) and Model 8 is:

1. Adjusted R^2: 0.0065
2. BIC: 459.31

```
tabr2 <- c(tabr2, summary(mod.cat12)$adj.r.squared)
tabbic <- c(tabbic, BIC(mod.cat12))
label2 <- c('Full model','Model 1','Model 2','Model 3','Model 4','Model 5',
            'Model 6','Model 7','Model 8','Model 9','Model 10','Model 11',
            'Model 12')

res <- data.frame(label2,tabr2, tabbic)
kable(res, digits = 4, align = "ccc", booktabs = T, "latex",
      col.names = c("Model", "Adjusted R2", "BIC"))%>%
  kable_styling(latex_options="striped")
```

Model	Adjusted R2	BIC
Full model	0.8479	-41835.03
Model 1	0.8436	-41332.54
Model 2	0.7681	-32912.22
Model 3	0.7679	-32900.75
Model 4	0.7677	-32887.51
Model 5	0.7671	-32846.32
Model 6	0.7663	-32777.18
Model 7	0.7644	-32700.24
Model 8	0.7616	-32452.91
Model 9	0.7580	-32166.05
Model 10	0.7483	-31365.21
Model 11	0.7048	-27955.37
Model 12	0.5281	-18059.09

After this categorical analysis we decided to candidate Model 8 as our best categorical model.

```
summary(mod.cat8)
```

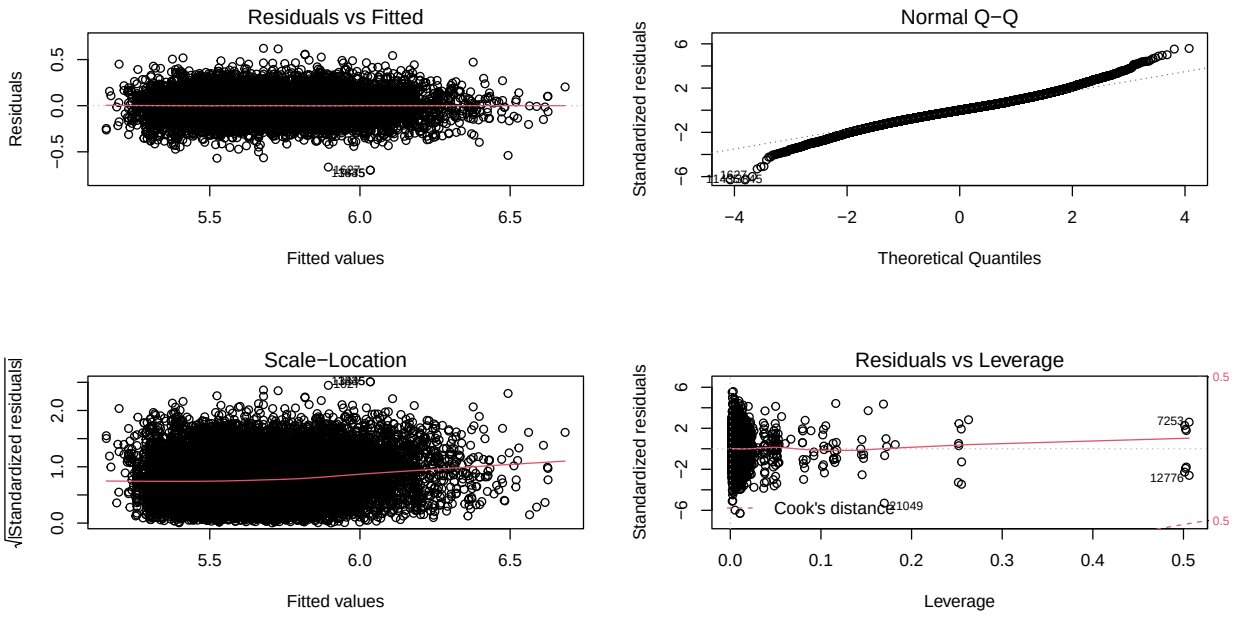
```
##
## Call:
## lm(formula = log10_price ~ bathrooms + floors + condition + view +
##     zipcode, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69875 -0.06648 -0.00035  0.06464  0.62185
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.034627   0.059914  84.030  < 2e-16 ***
## bathrooms0.75  0.042097   0.057406   0.733 0.463376
## bathrooms1     0.105630   0.055880   1.890 0.058729 .
## bathrooms1.25  0.130272   0.067162   1.940 0.052434 .
## bathrooms1.5   0.157149   0.055933   2.810 0.004965 **
## bathrooms1.75  0.196146   0.055894   3.509 0.000450 ***
## bathrooms2     0.196232   0.055915   3.510 0.000450 ***
## bathrooms2.25  0.235511   0.055919   4.212 2.55e-05 ***
## bathrooms2.5   0.273889   0.055891   4.900 9.63e-07 ***
## bathrooms2.75  0.302177   0.055959   5.400 6.74e-08 ***
## bathrooms3     0.311216   0.056011   5.556 2.79e-08 ***
## bathrooms3.25  0.380537   0.056069   6.787 1.18e-11 ***
## bathrooms3.5   0.387879   0.056037   6.922 4.58e-12 ***
## bathrooms3.75  0.438155   0.056601   7.741 1.03e-14 ***
## bathrooms4     0.460650   0.056709   8.123 4.79e-16 ***
## bathrooms4.25  0.498361   0.057304   8.697  < 2e-16 ***
## bathrooms4.5   0.469748   0.056997   8.242  < 2e-16 ***
## bathrooms4.75  0.571183   0.060581   9.428  < 2e-16 ***
## bathrooms5     0.516173   0.060996   8.462  < 2e-16 ***
## bathrooms5.25  0.507572   0.063958   7.936 2.19e-15 ***
## bathrooms5.5   0.699525   0.066152  10.575  < 2e-16 ***
```

```
## bathrooms5.75  0.558087   0.079133   7.053 1.81e-12 ***
## bathrooms6     0.602399   0.072238   8.339  < 2e-16 ***
## bathrooms6.25  0.789783   0.096830   8.156 3.64e-16 ***
## bathrooms6.5   0.690553   0.096767   7.136 9.90e-13 ***
## bathrooms6.75  0.559155   0.096865   5.773 7.92e-09 ***
## bathrooms8     0.811551   0.097091   8.359  < 2e-16 ***
## floors1.5      0.045864   0.002893  15.852  < 2e-16 ***
## floors2        0.035299   0.002248  15.700  < 2e-16 ***
## floors2.5      0.085205   0.009133   9.330  < 2e-16 ***
## floors3       -0.067987   0.005199 -13.078  < 2e-16 ***
## floors3.5     -0.040122   0.042353  -0.947 0.343480
## condition2     0.074234   0.022474   3.303 0.000958 ***
## condition3     0.147252   0.020839   7.066 1.64e-12 ***
## condition4     0.164018   0.020863   7.862 3.97e-15 ***
## condition5     0.184031   0.020995   8.766  < 2e-16 ***
## view1          0.103058   0.006293  16.377  < 2e-16 ***
## view2          0.109845   0.003793  28.963  < 2e-16 ***
## view3          0.168339   0.005159  32.631  < 2e-16 ***
## view4          0.312310   0.006504  48.018  < 2e-16 ***
## zipcode98002  -0.046700   0.009877  -4.728 2.28e-06 ***
## zipcode98003   0.013873   0.008895   1.560 0.118867
## zipcode98004   0.580283   0.008645  67.124  < 2e-16 ***
## zipcode98005   0.413949   0.010469  39.539  < 2e-16 ***
## zipcode98006   0.361314   0.007778  46.451  < 2e-16 ***
## zipcode98007   0.327069   0.011104  29.454  < 2e-16 ***
## zipcode98008   0.308912   0.008902  34.700  < 2e-16 ***
## zipcode98010   0.144339   0.012619  11.438  < 2e-16 ***
## zipcode98011   0.223578   0.009929  22.517  < 2e-16 ***
## zipcode98014   0.166093   0.011642  14.267  < 2e-16 ***
## zipcode98019   0.149805   0.010022  14.947  < 2e-16 ***
## zipcode98022   0.024322   0.009422   2.581 0.009847 **
## zipcode98023   0.004071   0.007724   0.527 0.598153
## zipcode98024   0.247888   0.013816  17.942  < 2e-16 ***
## zipcode98027   0.262617   0.008081  32.497  < 2e-16 ***
## zipcode98028   0.201185   0.008870  22.682  < 2e-16 ***
## zipcode98029   0.261941   0.008618  30.396  < 2e-16 ***
## zipcode98030   0.019822   0.009122   2.173 0.029793 *
## zipcode98031   0.025537   0.008960   2.850 0.004373 **
## zipcode98032  -0.006335   0.011605  -0.546 0.585185
## zipcode98033   0.374769   0.007981  46.960  < 2e-16 ***
## zipcode98034   0.236545   0.007589  31.170  < 2e-16 ***
## zipcode98038   0.070651   0.007486   9.437  < 2e-16 ***
## zipcode98039   0.699431   0.017178  40.716  < 2e-16 ***
## zipcode98040   0.475696   0.008984  52.951  < 2e-16 ***
## zipcode98042   0.031731   0.007573   4.190 2.80e-05 ***
## zipcode98045   0.162915   0.009556  17.048  < 2e-16 ***
## zipcode98052   0.324233   0.007513  43.156  < 2e-16 ***
## zipcode98053   0.323559   0.008117  39.862  < 2e-16 ***
## zipcode98055   0.046224   0.009004   5.134 2.86e-07 ***
## zipcode98056   0.142547   0.008095  17.609  < 2e-16 ***
## zipcode98058   0.094253   0.007873  11.972  < 2e-16 ***
## zipcode98059   0.187555   0.007834  23.942  < 2e-16 ***
## zipcode98065   0.187677   0.008701  21.569  < 2e-16 ***
## zipcode98070   0.170545   0.011926  14.300  < 2e-16 ***
```

```
## zipcode98072   0.277640   0.008963  30.975  < 2e-16 ***
## zipcode98074   0.318269   0.007949  40.037  < 2e-16 ***
## zipcode98075   0.343835   0.008376  41.051  < 2e-16 ***
## zipcode98077   0.322564   0.009895  32.600  < 2e-16 ***
## zipcode98092   0.040464   0.008376   4.831 1.37e-06 ***
## zipcode98102   0.419372   0.012508  33.529  < 2e-16 ***
## zipcode98103   0.336728   0.007602  44.293  < 2e-16 ***
## zipcode98105   0.416671   0.009489  43.912  < 2e-16 ***
## zipcode98106   0.084993   0.008488  10.013  < 2e-16 ***
## zipcode98107   0.322244   0.009131  35.289  < 2e-16 ***
## zipcode98108   0.113619   0.010091  11.260  < 2e-16 ***
## zipcode98109   0.429448   0.012260  35.028  < 2e-16 ***
## zipcode98112   0.497366   0.009074  54.810  < 2e-16 ***
## zipcode98115   0.350385   0.007518  46.609  < 2e-16 ***
## zipcode98116   0.299471   0.008555  35.003  < 2e-16 ***
## zipcode98117   0.333210   0.007602  43.831  < 2e-16 ***
## zipcode98118   0.171839   0.007722  22.253  < 2e-16 ***
## zipcode98119   0.406040   0.010176  39.901  < 2e-16 ***
## zipcode98122   0.324702   0.008860  36.649  < 2e-16 ***
## zipcode98125   0.242191   0.008101  29.898  < 2e-16 ***
## zipcode98126   0.195727   0.008393  23.320  < 2e-16 ***
## zipcode98133   0.191659   0.007768  24.673  < 2e-16 ***
## zipcode98136   0.265487   0.009105  29.157  < 2e-16 ***
## zipcode98144   0.256729   0.008475  30.293  < 2e-16 ***
## zipcode98146   0.098294   0.008854  11.102  < 2e-16 ***
## zipcode98148   0.057895   0.015941   3.632 0.000282 ***
## zipcode98155   0.187368   0.007926  23.640  < 2e-16 ***
## zipcode98166   0.157003   0.009182  17.099  < 2e-16 ***
## zipcode98168   0.015212   0.009044   1.682 0.092596 .
## zipcode98177   0.291566   0.009182  31.755  < 2e-16 ***
## zipcode98178   0.038998   0.009102   4.285 1.84e-05 ***
## zipcode98188   0.031566   0.011247   2.807 0.005010 **
## zipcode98198   0.011974   0.008922   1.342 0.179588
## zipcode98199   0.381229   0.008635  44.149  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1116 on 21491 degrees of freedom
## Multiple R-squared:  0.7627, Adjusted R-squared:  0.7616
## F-statistic: 639.7 on 108 and 21491 DF,  p-value: < 2.2e-16
```

Now we check the plots of the model to see if the results are good:

```
par(mfrow=c(2,2))
plot(mod.cat8)
```




```
par(mfrow=c(1,1))
```

The residuals plot shows an almost perfect straight line at the zero value as we like. The QQ plot is pretty good too, there is some distance on both ends but is acceptable. For what concerns the leverage points it seems that there are a few but with a not so high influence so we can keep them.

From this residual analysis we can confidently choose Model 8 as our best model for the categorical variables.

4 Final models

Merging our best numerical models with our the best categorical model we obtained three final models. The first model is composed by four numerical variables (*sqft_living*, *sqft_living15*, *lat*, *long*) and five categorical variables (*bathrooms*, *floors*, *condition*, *view* and *zipcode*). With this model we obtained an Adjusted R^2 of 0.8584 and a BIC value of -43679.89.

```
mod.A <- lm(log10_price ~ sqft_living + sqft_living15 + lat + long + bathrooms + floors +
                          condition + view + zipcode, data=kc_housing)
summary(mod.A)
```

```
##
## Call:
## lm(formula = log10_price ~ sqft_living + sqft_living15 + lat +
##     long + bathrooms + floors + condition + view + zipcode, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54631 -0.04691  0.00169  0.04777  0.47238
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.374e+01  3.277e+00  -7.246 4.44e-13 ***
## sqft_living    1.030e-04  1.301e-06  79.180  < 2e-16 ***
## sqft_living15  5.306e-05  1.497e-06  35.443  < 2e-16 ***
## lat            2.355e-01  3.395e-02   6.938 4.09e-12 ***
## long          -1.425e-01  2.431e-02  -5.863 4.61e-09 ***
## bathrooms0.75  9.905e-02  4.424e-02   2.239 0.025154 *
## bathrooms1     1.248e-01  4.306e-02   2.899 0.003751 **
## bathrooms1.25  1.414e-01  5.175e-02   2.731 0.006310 **
## bathrooms1.5   1.369e-01  4.310e-02   3.177 0.001489 **
## bathrooms1.75  1.514e-01  4.307e-02   3.516 0.000439 ***
## bathrooms2     1.493e-01  4.309e-02   3.464 0.000533 ***
## bathrooms2.25  1.597e-01  4.309e-02   3.707 0.000210 ***
## bathrooms2.5   1.647e-01  4.308e-02   3.823 0.000132 ***
## bathrooms2.75  1.625e-01  4.314e-02   3.767 0.000166 ***
## bathrooms3     1.620e-01  4.318e-02   3.752 0.000176 ***
## bathrooms3.25  1.773e-01  4.324e-02   4.101 4.13e-05 ***
## bathrooms3.5   1.701e-01  4.322e-02   3.935 8.35e-05 ***
## bathrooms3.75  1.829e-01  4.368e-02   4.188 2.82e-05 ***
## bathrooms4     1.653e-01  4.378e-02   3.775 0.000160 ***
## bathrooms4.25  1.625e-01  4.426e-02   3.672 0.000241 ***
## bathrooms4.5   1.453e-01  4.403e-02   3.300 0.000968 ***
## bathrooms4.75  1.677e-01  4.684e-02   3.580 0.000344 ***
## bathrooms5     1.539e-01  4.712e-02   3.267 0.001090 **
## bathrooms5.25  1.446e-01  4.941e-02   2.927 0.003426 **
## bathrooms5.5   1.650e-01  5.124e-02   3.221 0.001279 **
## bathrooms5.75  3.444e-02  6.128e-02   0.562 0.574088
## bathrooms6     1.046e-01  5.588e-02   1.872 0.061266 .
## bathrooms6.25  5.270e-02  7.496e-02   0.703 0.482067
## bathrooms6.5   1.761e-01  7.485e-02   2.352 0.018658 *
## bathrooms6.75 -1.297e-01  7.507e-02  -1.728 0.084014 .
## bathrooms8    -3.960e-01  7.588e-02  -5.219 1.82e-07 ***
## floors1.5      1.849e-02  2.246e-03   8.231  < 2e-16 ***
## floors2        1.306e-02  1.743e-03   7.492 7.03e-14 ***
## floors2.5      2.612e-02  7.061e-03   3.699 0.000217 ***
## floors3       -1.838e-02  4.028e-03  -4.564 5.05e-06 ***
## floors3.5     -2.619e-02  3.264e-02  -0.802 0.422302
## condition2     8.025e-02  1.732e-02   4.633 3.62e-06 ***
## condition3     1.435e-01  1.606e-02   8.937  < 2e-16 ***
## condition4     1.581e-01  1.608e-02   9.830  < 2e-16 ***
## condition5     1.794e-01  1.619e-02  11.084  < 2e-16 ***
## view1          4.781e-02  4.877e-03   9.803  < 2e-16 ***
## view2          5.135e-02  2.972e-03  17.277  < 2e-16 ***
## view3          8.676e-02  4.047e-03  21.438  < 2e-16 ***
## view4          2.095e-01  5.095e-03  41.126  < 2e-16 ***
## zipcode98002  -9.633e-03  7.743e-03  -1.244 0.213464
## zipcode98003   5.742e-03  6.921e-03   0.830 0.406763
## zipcode98004   4.304e-01  1.258e-02  34.199  < 2e-16 ***
## zipcode98005   2.774e-01  1.345e-02  20.631  < 2e-16 ***
## zipcode98006   2.387e-01  1.099e-02  21.719  < 2e-16 ***
## zipcode98007   2.428e-01  1.387e-02  17.504  < 2e-16 ***
## zipcode98008   2.382e-01  1.317e-02  18.082  < 2e-16 ***
## zipcode98010   1.508e-01  1.180e-02  12.780  < 2e-16 ***
## zipcode98011   9.444e-02  1.715e-02   5.508 3.66e-08 ***
```

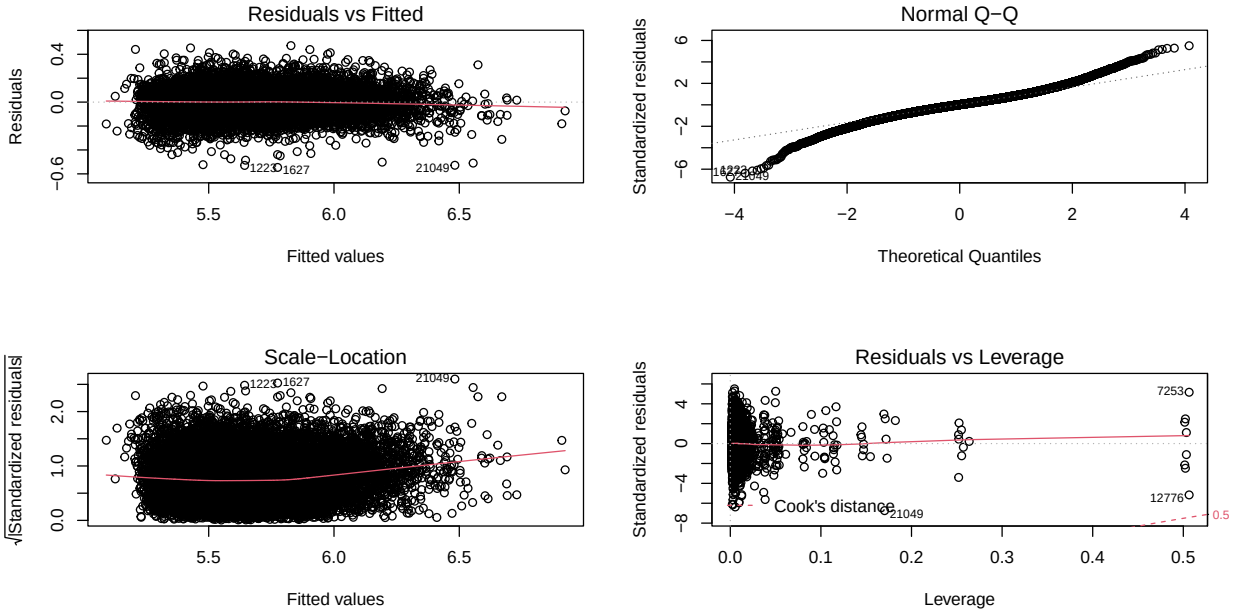
```
## zipcode98014   1.254e-01  1.882e-02   6.665 2.70e-11 ***
## zipcode98019   8.201e-02  1.858e-02   4.415 1.02e-05 ***
## zipcode98022   1.007e-01  1.026e-02   9.815  < 2e-16 ***
## zipcode98023  -2.158e-02  6.373e-03  -3.387 0.000709 ***
## zipcode98024   2.037e-01  1.656e-02  12.306  < 2e-16 ***
## zipcode98027   2.062e-01  1.128e-02  18.275  < 2e-16 ***
## zipcode98028   7.731e-02  1.665e-02   4.643 3.46e-06 ***
## zipcode98029   2.462e-01  1.288e-02  19.111  < 2e-16 ***
## zipcode98030   1.828e-02  7.610e-03   2.402 0.016295 *
## zipcode98031   1.834e-02  7.929e-03   2.313 0.020730 *
## zipcode98032  -2.746e-02  9.207e-03  -2.983 0.002857 **
## zipcode98033   2.716e-01  1.429e-02  19.005  < 2e-16 ***
## zipcode98034   1.476e-01  1.532e-02   9.634  < 2e-16 ***
## zipcode98038   8.640e-02  8.554e-03  10.101  < 2e-16 ***
## zipcode98039   5.081e-01  1.714e-02  29.645  < 2e-16 ***
## zipcode98040   3.384e-01  1.113e-02  30.414  < 2e-16 ***
## zipcode98042   3.556e-02  7.286e-03   4.881 1.06e-06 ***
## zipcode98045   1.924e-01  1.580e-02  12.184  < 2e-16 ***
## zipcode98052   2.185e-01  1.458e-02  14.982  < 2e-16 ***
## zipcode98053   2.111e-01  1.563e-02  13.500  < 2e-16 ***
## zipcode98055   2.997e-02  8.821e-03   3.397 0.000682 ***
## zipcode98056   9.962e-02  9.596e-03  10.381  < 2e-16 ***
## zipcode98058   6.204e-02  8.340e-03   7.439 1.05e-13 ***
## zipcode98059   1.194e-01  9.410e-03  12.691  < 2e-16 ***
## zipcode98065   1.569e-01  1.454e-02  10.786  < 2e-16 ***
## zipcode98070   1.435e-01  1.086e-02  13.208  < 2e-16 ***
## zipcode98072   1.393e-01  1.706e-02   8.163 3.44e-16 ***
## zipcode98074   2.139e-01  1.380e-02  15.499  < 2e-16 ***
## zipcode98075   2.230e-01  1.327e-02  16.799  < 2e-16 ***
## zipcode98077   1.491e-01  1.775e-02   8.398  < 2e-16 ***
## zipcode98092   3.703e-02  6.922e-03   5.349 8.93e-08 ***
## zipcode98102   3.424e-01  1.467e-02  23.345  < 2e-16 ***
## zipcode98103   2.658e-01  1.379e-02  19.278  < 2e-16 ***
## zipcode98105   3.221e-01  1.411e-02  22.824  < 2e-16 ***
## zipcode98106   6.973e-02  1.023e-02   6.816 9.62e-12 ***
## zipcode98107   2.690e-01  1.422e-02  18.914  < 2e-16 ***
## zipcode98108   8.073e-02  1.128e-02   7.160 8.30e-13 ***
## zipcode98109   3.494e-01  1.460e-02  23.934  < 2e-16 ***
## zipcode98112   3.815e-01  1.293e-02  29.514  < 2e-16 ***
## zipcode98115   2.607e-01  1.401e-02  18.612  < 2e-16 ***
## zipcode98116   2.467e-01  1.140e-02  21.646  < 2e-16 ***
## zipcode98117   2.479e-01  1.419e-02  17.465  < 2e-16 ***
## zipcode98118   1.335e-01  9.946e-03  13.425  < 2e-16 ***
## zipcode98119   3.404e-01  1.378e-02  24.700  < 2e-16 ***
## zipcode98122   2.814e-01  1.228e-02  22.909  < 2e-16 ***
## zipcode98125   1.471e-01  1.516e-02   9.701  < 2e-16 ***
## zipcode98126   1.646e-01  1.048e-02  15.705  < 2e-16 ***
## zipcode98133   9.139e-02  1.566e-02   5.836 5.44e-09 ***
## zipcode98136   2.244e-01  1.074e-02  20.895  < 2e-16 ***
## zipcode98144   2.088e-01  1.145e-02  18.228  < 2e-16 ***
## zipcode98146   5.782e-02  9.600e-03   6.023 1.74e-09 ***
## zipcode98148   3.422e-02  1.307e-02   2.618 0.008857 **
## zipcode98155   7.882e-02  1.628e-02   4.841 1.30e-06 ***
## zipcode98166   9.823e-02  8.784e-03  11.183  < 2e-16 ***
```

```
## zipcode98168  -2.462e-02  9.283e-03  -2.652 0.008005 **
## zipcode98177   1.423e-01  1.635e-02   8.705  < 2e-16 ***
## zipcode98178   9.332e-03  9.573e-03   0.975 0.329661
## zipcode98188   4.586e-03  9.848e-03   0.466 0.641461
## zipcode98198   1.583e-03  7.469e-03   0.212 0.832197
## zipcode98199   2.805e-01  1.347e-02  20.817  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08597 on 21487 degrees of freedom
## Multiple R-squared:  0.8592, Adjusted R-squared:  0.8584
## F-statistic:  1170 on 112 and 21487 DF,  p-value: < 2.2e-16
```

```
BIC(mod.A)
```

```
## [1] -43679.89
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.A)
```




```
par(mfrow=c(1,1))
```

The second models is like the previous one plus a polynomial component of second grade for the sqft_living variable, this model reaches an Adjusted R^2 of 0.8603 and a BIC value equals to -43950.11.

```
mod.B <- lm(log10_price ~ poly(sqft_living, 2) + sqft_living15 + lat + long + bathrooms +
                          floors + condition + view + zipcode, data=kc_housing)
summary(mod.B)
```

```
##
## Call:
## lm(formula = log10_price ~ poly(sqft_living, 2) + sqft_living15 +
##     lat + long + bathrooms + floors + condition + view + zipcode,
##     data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53580 -0.04721  0.00129  0.04747  0.47856
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -2.257e+01  3.256e+00  -6.932 4.27e-12 ***
## poly(sqft_living, 2)1  1.393e+01  1.742e-01  79.959  < 2e-16 ***
## poly(sqft_living, 2)2 -2.297e+00  1.371e-01 -16.749  < 2e-16 ***
## sqft_living15          5.214e-05  1.489e-06  35.026  < 2e-16 ***
## lat                    2.356e-01  3.373e-02   6.986 2.92e-12 ***
## long                  -1.348e-01  2.416e-02  -5.580 2.44e-08 ***
## bathrooms0.75          1.061e-01  4.395e-02   2.413 0.015832 *
## bathrooms1             1.242e-01  4.278e-02   2.902 0.003710 **
## bathrooms1.25          1.328e-01  5.142e-02   2.583 0.009813 **
## bathrooms1.5           1.283e-01  4.282e-02   2.996 0.002743 **
## bathrooms1.75          1.388e-01  4.280e-02   3.242 0.001188 **
## bathrooms2             1.366e-01  4.282e-02   3.190 0.001426 **
## bathrooms2.25          1.435e-01  4.283e-02   3.351 0.000808 ***
## bathrooms2.5           1.465e-01  4.281e-02   3.422 0.000623 ***
## bathrooms2.75          1.437e-01  4.287e-02   3.352 0.000805 ***
## bathrooms3             1.443e-01  4.292e-02   3.362 0.000774 ***
## bathrooms3.25          1.651e-01  4.297e-02   3.842 0.000122 ***
## bathrooms3.5           1.576e-01  4.295e-02   3.670 0.000243 ***
## bathrooms3.75          1.762e-01  4.340e-02   4.061 4.91e-05 ***
## bathrooms4             1.677e-01  4.349e-02   3.855 0.000116 ***
## bathrooms4.25          1.764e-01  4.399e-02   4.009 6.11e-05 ***
## bathrooms4.5           1.542e-01  4.375e-02   3.526 0.000423 ***
## bathrooms4.75          2.054e-01  4.659e-02   4.408 1.05e-05 ***
## bathrooms5             1.832e-01  4.685e-02   3.910 9.27e-05 ***
## bathrooms5.25          1.786e-01  4.914e-02   3.635 0.000278 ***
## bathrooms5.5           2.630e-01  5.124e-02   5.133 2.88e-07 ***
## bathrooms5.75          1.906e-01  6.160e-02   3.094 0.001979 **
## bathrooms6             2.031e-01  5.583e-02   3.638 0.000276 ***
## bathrooms6.25          2.902e-01  7.581e-02   3.828 0.000130 ***
## bathrooms6.5           2.894e-01  7.468e-02   3.876 0.000107 ***
## bathrooms6.75          1.363e-01  7.626e-02   1.787 0.073898 .
## bathrooms8             4.203e-01  8.977e-02   4.683 2.85e-06 ***
## floors1.5              1.502e-02  2.241e-03   6.703 2.09e-11 ***
## floors2                1.303e-02  1.732e-03   7.521 5.65e-14 ***
## floors2.5              2.852e-02  7.017e-03   4.064 4.84e-05 ***
## floors3               -1.331e-02  4.013e-03  -3.316 0.000915 ***
## floors3.5              5.583e-03  3.248e-02   0.172 0.863543
## condition2             8.037e-02  1.721e-02   4.670 3.03e-06 ***
## condition3             1.410e-01  1.596e-02   8.835  < 2e-16 ***
## condition4             1.548e-01  1.598e-02   9.685  < 2e-16 ***
## condition5             1.765e-01  1.608e-02  10.973  < 2e-16 ***
## view1                  4.671e-02  4.846e-03   9.639  < 2e-16 ***
```

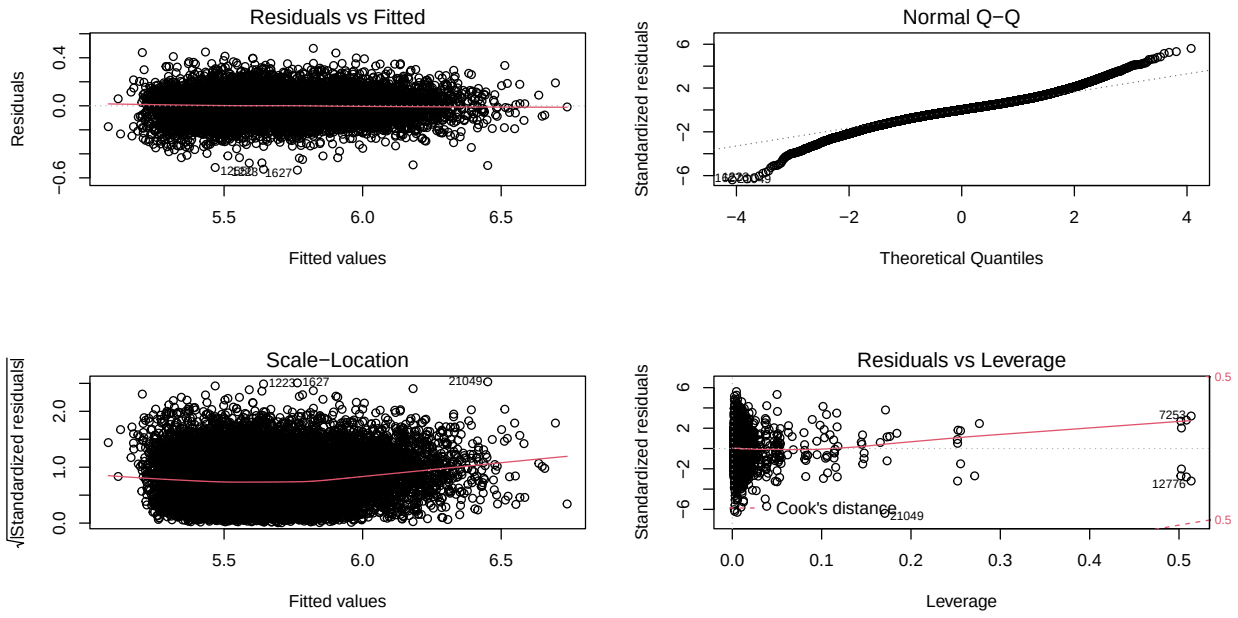
```
## view2                5.143e-02  2.953e-03  17.416  < 2e-16 ***
## view3                8.752e-02  4.021e-03  21.767  < 2e-16 ***
## view4                2.150e-01  5.072e-03  42.382  < 2e-16 ***
## zipcode98002        -9.342e-03  7.693e-03  -1.214 0.224624
## zipcode98003         6.564e-03  6.876e-03   0.955 0.339841
## zipcode98004         4.328e-01  1.250e-02  34.612  < 2e-16 ***
## zipcode98005         2.762e-01  1.336e-02  20.672  < 2e-16 ***
## zipcode98006         2.395e-01  1.092e-02  21.934  < 2e-16 ***
## zipcode98007         2.405e-01  1.378e-02  17.449  < 2e-16 ***
## zipcode98008         2.370e-01  1.309e-02  18.104  < 2e-16 ***
## zipcode98010         1.494e-01  1.173e-02  12.739  < 2e-16 ***
## zipcode98011         9.311e-02  1.704e-02   5.466 4.66e-08 ***
## zipcode98014         1.235e-01  1.870e-02   6.606 4.05e-11 ***
## zipcode98019         7.934e-02  1.846e-02   4.298 1.73e-05 ***
## zipcode98022         9.931e-02  1.019e-02   9.745  < 2e-16 ***
## zipcode98023        -2.086e-02  6.332e-03  -3.294 0.000989 ***
## zipcode98024         2.026e-01  1.645e-02  12.316  < 2e-16 ***
## zipcode98027         2.051e-01  1.121e-02  18.298  < 2e-16 ***
## zipcode98028         7.568e-02  1.655e-02   4.574 4.82e-06 ***
## zipcode98029         2.448e-01  1.280e-02  19.129  < 2e-16 ***
## zipcode98030         1.735e-02  7.561e-03   2.295 0.021742 *
## zipcode98031         1.814e-02  7.878e-03   2.302 0.021317 *
## zipcode98032        -2.720e-02  9.147e-03  -2.974 0.002943 **
## zipcode98033         2.710e-01  1.420e-02  19.087  < 2e-16 ***
## zipcode98034         1.474e-01  1.522e-02   9.684  < 2e-16 ***
## zipcode98038         8.536e-02  8.499e-03  10.043  < 2e-16 ***
## zipcode98039         5.139e-01  1.703e-02  30.175  < 2e-16 ***
## zipcode98040         3.394e-01  1.106e-02  30.700  < 2e-16 ***
## zipcode98042         3.545e-02  7.239e-03   4.897 9.80e-07 ***
## zipcode98045         1.899e-01  1.569e-02  12.098  < 2e-16 ***
## zipcode98052         2.160e-01  1.449e-02  14.905  < 2e-16 ***
## zipcode98053         2.110e-01  1.553e-02  13.584  < 2e-16 ***
## zipcode98055         2.945e-02  8.764e-03   3.361 0.000778 ***
## zipcode98056         9.843e-02  9.535e-03  10.323  < 2e-16 ***
## zipcode98058         6.081e-02  8.287e-03   7.339 2.23e-13 ***
## zipcode98059         1.182e-01  9.350e-03  12.647  < 2e-16 ***
## zipcode98065         1.536e-01  1.445e-02  10.631  < 2e-16 ***
## zipcode98070         1.437e-01  1.079e-02  13.316  < 2e-16 ***
## zipcode98072         1.381e-01  1.695e-02   8.145 4.01e-16 ***
## zipcode98074         2.115e-01  1.371e-02  15.425  < 2e-16 ***
## zipcode98075         2.200e-01  1.319e-02  16.683  < 2e-16 ***
## zipcode98077         1.485e-01  1.763e-02   8.423  < 2e-16 ***
## zipcode98092         3.588e-02  6.878e-03   5.217 1.84e-07 ***
## zipcode98102         3.445e-01  1.457e-02  23.644  < 2e-16 ***
## zipcode98103         2.665e-01  1.370e-02  19.450  < 2e-16 ***
## zipcode98105         3.229e-01  1.402e-02  23.025  < 2e-16 ***
## zipcode98106         7.360e-02  1.017e-02   7.238 4.70e-13 ***
## zipcode98107         2.721e-01  1.413e-02  19.255  < 2e-16 ***
## zipcode98108         8.140e-02  1.120e-02   7.267 3.81e-13 ***
## zipcode98109         3.495e-01  1.451e-02  24.096  < 2e-16 ***
## zipcode98112         3.815e-01  1.284e-02  29.706  < 2e-16 ***
## zipcode98115         2.602e-01  1.392e-02  18.693  < 2e-16 ***
## zipcode98116         2.474e-01  1.132e-02  21.851  < 2e-16 ***
## zipcode98117         2.494e-01  1.410e-02  17.689  < 2e-16 ***
```

```
## zipcode98118           1.340e-01  9.882e-03  13.563  < 2e-16 ***
## zipcode98119           3.401e-01  1.369e-02  24.834  < 2e-16 ***
## zipcode98122           2.824e-01  1.220e-02  23.136  < 2e-16 ***
## zipcode98125           1.465e-01  1.506e-02   9.729  < 2e-16 ***
## zipcode98126           1.668e-01  1.041e-02  16.024  < 2e-16 ***
## zipcode98133           9.205e-02  1.556e-02   5.916 3.35e-09 ***
## zipcode98136           2.259e-01  1.067e-02  21.174  < 2e-16 ***
## zipcode98144           2.108e-01  1.138e-02  18.522  < 2e-16 ***
## zipcode98146           5.822e-02  9.538e-03   6.105 1.05e-09 ***
## zipcode98148           3.520e-02  1.299e-02   2.710 0.006726 **
## zipcode98155           7.848e-02  1.618e-02   4.851 1.23e-06 ***
## zipcode98166           9.712e-02  8.728e-03  11.128  < 2e-16 ***
## zipcode98168          -2.439e-02  9.223e-03  -2.644 0.008194 **
## zipcode98177           1.418e-01  1.624e-02   8.732  < 2e-16 ***
## zipcode98178           8.499e-03  9.511e-03   0.894 0.371566
## zipcode98188           3.870e-03  9.785e-03   0.395 0.692490
## zipcode98198           2.076e-03  7.421e-03   0.280 0.779681
## zipcode98199           2.808e-01  1.339e-02  20.981  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08542 on 21486 degrees of freedom
## Multiple R-squared:  0.861,  Adjusted R-squared:  0.8603
## F-statistic:  1178 on 113 and 21486 DF,  p-value: < 2.2e-16
```

```
BIC(mod.B)
```

```
## [1] -43950.11
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.B)
```




```
par(mfrow=c(1,1))
```

The third model, differently from the previously model has a 4th polynomial component for the categorical variable *lat*. The Adjusted R^2 has a value of 0.8634 and a BIC values of -44416.35, so this two indicators increase.

```
mod.C <- lm(log10_price ~ poly(sqft_living, 2) + sqft_living15 + poly(lat, 4) + long +
                          bathrooms + floors + condition + view + zipcode,
                          data=kc_housing)
summary(mod.C)
```

```
##
## Call:
## lm(formula = log10_price ~ poly(sqft_living, 2) + sqft_living15 +
##     poly(lat, 4) + long + bathrooms + floors + condition + view +
##     zipcode, data = kc_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53359 -0.04600  0.00167  0.04744  0.46884
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -4.965e+00  2.938e+00  -1.690 0.091069 .
## poly(sqft_living, 2)1    1.395e+01  1.722e-01  81.016  < 2e-16 ***
## poly(sqft_living, 2)2 -2.385e+00  1.357e-01 -17.581  < 2e-16 ***
## sqft_living15            5.172e-05  1.472e-06  35.133  < 2e-16 ***
## poly(lat, 4)1            1.116e+01  7.660e-01  14.570  < 2e-16 ***
## poly(lat, 4)2           -2.921e+00  3.650e-01  -8.002 1.29e-15 ***
## poly(lat, 4)3           -4.039e+00  2.385e-01 -16.937  < 2e-16 ***
```

```
## poly(lat, 4)4        -1.452e+00  2.159e-01  -6.728 1.76e-11 ***
## long                 -8.326e-02  2.402e-02  -3.467 0.000528 ***
## bathrooms0.75         9.679e-02  4.346e-02   2.227 0.025942 *
## bathrooms1            1.157e-01  4.230e-02   2.736 0.006232 **
## bathrooms1.25         1.306e-01  5.084e-02   2.569 0.010198 *
## bathrooms1.5          1.202e-01  4.234e-02   2.839 0.004524 **
## bathrooms1.75         1.314e-01  4.232e-02   3.105 0.001904 **
## bathrooms2            1.292e-01  4.233e-02   3.051 0.002284 **
## bathrooms2.25         1.364e-01  4.234e-02   3.222 0.001276 **
## bathrooms2.5          1.394e-01  4.233e-02   3.293 0.000992 ***
## bathrooms2.75         1.359e-01  4.239e-02   3.206 0.001347 **
## bathrooms3            1.366e-01  4.243e-02   3.219 0.001289 **
## bathrooms3.25         1.566e-01  4.248e-02   3.686 0.000229 ***
## bathrooms3.5          1.500e-01  4.247e-02   3.531 0.000415 ***
## bathrooms3.75         1.661e-01  4.291e-02   3.870 0.000109 ***
## bathrooms4            1.588e-01  4.300e-02   3.692 0.000223 ***
## bathrooms4.25         1.665e-01  4.349e-02   3.829 0.000129 ***
## bathrooms4.5          1.481e-01  4.325e-02   3.425 0.000616 ***
## bathrooms4.75         1.996e-01  4.606e-02   4.334 1.47e-05 ***
## bathrooms5            1.746e-01  4.632e-02   3.769 0.000164 ***
## bathrooms5.25         1.693e-01  4.858e-02   3.485 0.000493 ***
## bathrooms5.5          2.595e-01  5.066e-02   5.123 3.04e-07 ***
## bathrooms5.75         1.911e-01  6.090e-02   3.137 0.001706 **
## bathrooms6            2.003e-01  5.520e-02   3.629 0.000285 ***
## bathrooms6.25         2.755e-01  7.496e-02   3.675 0.000238 ***
## bathrooms6.5          2.773e-01  7.384e-02   3.756 0.000173 ***
## bathrooms6.75         1.364e-01  7.539e-02   1.810 0.070361 .
## bathrooms8            4.400e-01  8.875e-02   4.957 7.20e-07 ***
## floors1.5             1.312e-02  2.218e-03   5.917 3.33e-09 ***
## floors2               1.134e-02  1.715e-03   6.612 3.88e-11 ***
## floors2.5             2.591e-02  6.939e-03   3.734 0.000189 ***
## floors3              -1.718e-02  3.971e-03  -4.327 1.52e-05 ***
## floors3.5             1.651e-03  3.212e-02   0.051 0.959004
## condition2            8.002e-02  1.701e-02   4.703 2.58e-06 ***
## condition3            1.409e-01  1.578e-02   8.928  < 2e-16 ***
## condition4            1.546e-01  1.580e-02   9.785  < 2e-16 ***
## condition5            1.760e-01  1.590e-02  11.071  < 2e-16 ***
## view1                 4.672e-02  4.792e-03   9.749  < 2e-16 ***
## view2                 5.049e-02  2.920e-03  17.291  < 2e-16 ***
## view3                 8.740e-02  3.975e-03  21.984  < 2e-16 ***
## view4                 2.164e-01  5.016e-03  43.147  < 2e-16 ***
## zipcode98002         -1.142e-02  7.610e-03  -1.500 0.133577
## zipcode98003          8.247e-03  6.806e-03   1.212 0.225683
## zipcode98004          2.806e-01  1.490e-02  18.831  < 2e-16 ***
## zipcode98005          1.237e-01  1.567e-02   7.897 2.99e-15 ***
## zipcode98006          1.164e-01  1.373e-02   8.478  < 2e-16 ***
## zipcode98007          8.677e-02  1.608e-02   5.398 6.81e-08 ***
## zipcode98008          8.003e-02  1.554e-02   5.149 2.64e-07 ***
## zipcode98010          1.353e-01  1.168e-02  11.580  < 2e-16 ***
## zipcode98011          2.005e-02  1.767e-02   1.135 0.256353
## zipcode98014         -4.931e-02  2.040e-02  -2.417 0.015663 *
## zipcode98019         -3.833e-02  1.940e-02  -1.976 0.048205 *
## zipcode98022          7.979e-02  1.439e-02   5.545 2.97e-08 ***
## zipcode98023         -1.506e-02  6.287e-03  -2.396 0.016585 *
```

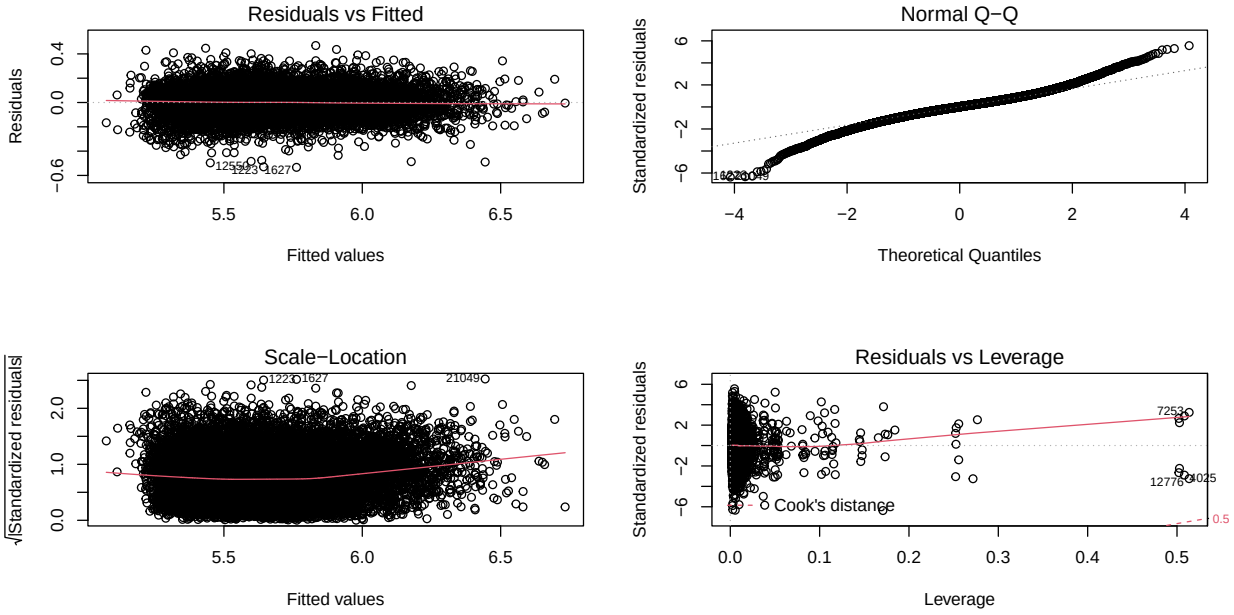
```
## zipcode98024          6.720e-02  1.858e-02   3.617 0.000299 ***
## zipcode98027          1.031e-01  1.374e-02   7.503 6.49e-14 ***
## zipcode98028          4.540e-03  1.718e-02   0.264 0.791578
## zipcode98029          1.137e-01  1.543e-02   7.372 1.75e-13 ***
## zipcode98030          6.983e-03  8.019e-03   0.871 0.383868
## zipcode98031         -3.871e-03  8.956e-03  -0.432 0.665539
## zipcode98032         -3.406e-02  9.531e-03  -3.573 0.000353 ***
## zipcode98033          1.139e-01  1.606e-02   7.090 1.38e-12 ***
## zipcode98034          1.940e-02  1.649e-02   1.177 0.239356
## zipcode98038          6.589e-02  8.958e-03   7.355 1.98e-13 ***
## zipcode98039          3.590e-01  1.875e-02  19.146  < 2e-16 ***
## zipcode98040          2.198e-01  1.369e-02  16.059  < 2e-16 ***
## zipcode98042          2.156e-02  7.683e-03   2.807 0.005009 **
## zipcode98045          1.077e-01  1.734e-02   6.213 5.28e-10 ***
## zipcode98052          5.901e-02  1.634e-02   3.612 0.000304 ***
## zipcode98053          4.887e-02  1.736e-02   2.815 0.004875 **
## zipcode98055         -2.073e-02  1.081e-02  -1.918 0.055094 .
## zipcode98056          1.242e-02  1.225e-02   1.014 0.310383
## zipcode98058          1.547e-02  1.025e-02   1.509 0.131220
## zipcode98059          4.234e-02  1.193e-02   3.549 0.000387 ***
## zipcode98065          3.534e-02  1.682e-02   2.101 0.035676 *
## zipcode98070          1.268e-01  1.153e-02  10.992  < 2e-16 ***
## zipcode98072          5.322e-02  1.768e-02   3.010 0.002615 **
## zipcode98074          4.725e-02  1.613e-02   2.929 0.003404 **
## zipcode98075          7.180e-02  1.577e-02   4.553 5.32e-06 ***
## zipcode98077          5.319e-02  1.842e-02   2.888 0.003878 **
## zipcode98092          3.108e-02  6.813e-03   4.561 5.11e-06 ***
## zipcode98102          1.924e-01  1.646e-02  11.688  < 2e-16 ***
## zipcode98103          1.174e-01  1.545e-02   7.599 3.12e-14 ***
## zipcode98105          1.665e-01  1.590e-02  10.470  < 2e-16 ***
## zipcode98106         -2.268e-02  1.274e-02  -1.780 0.075145 .
## zipcode98107          1.212e-01  1.586e-02   7.640 2.26e-14 ***
## zipcode98108         -2.749e-02  1.370e-02  -2.007 0.044755 *
## zipcode98109          1.986e-01  1.636e-02  12.143  < 2e-16 ***
## zipcode98112          2.289e-01  1.504e-02  15.218  < 2e-16 ***
## zipcode98115          1.101e-01  1.565e-02   7.032 2.10e-12 ***
## zipcode98116          1.265e-01  1.374e-02   9.209  < 2e-16 ***
## zipcode98117          1.039e-01  1.571e-02   6.612 3.88e-11 ***
## zipcode98118          2.799e-02  1.267e-02   2.209 0.027157 *
## zipcode98119          1.885e-01  1.564e-02  12.056  < 2e-16 ***
## zipcode98122          1.367e-01  1.456e-02   9.387  < 2e-16 ***
## zipcode98125          1.983e-02  1.630e-02   1.216 0.223868
## zipcode98126          6.560e-02  1.293e-02   5.073 3.94e-07 ***
## zipcode98133         -3.770e-03  1.640e-02  -0.230 0.818215
## zipcode98136          1.294e-01  1.312e-02   9.862  < 2e-16 ***
## zipcode98144          7.772e-02  1.391e-02   5.586 2.36e-08 ***
## zipcode98146         -1.315e-02  1.197e-02  -1.098 0.272154
## zipcode98148          4.308e-03  1.400e-02   0.308 0.758322
## zipcode98155          9.320e-03  1.680e-02   0.555 0.579154
## zipcode98166          5.676e-02  1.058e-02   5.362 8.31e-08 ***
## zipcode98168         -8.931e-02  1.161e-02  -7.692 1.51e-14 ***
## zipcode98177          6.092e-02  1.692e-02   3.600 0.000319 ***
## zipcode98178         -6.650e-02  1.205e-02  -5.519 3.45e-08 ***
## zipcode98188         -3.511e-02  1.138e-02  -3.085 0.002041 **
```

```
## zipcode98198          -9.072e-03  8.206e-03  -1.106 0.268919
## zipcode98199           1.292e-01  1.532e-02   8.437  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08445 on 21483 degrees of freedom
## Multiple R-squared:  0.8641, Adjusted R-squared:  0.8634
## F-statistic:  1178 on 116 and 21483 DF,  p-value: < 2.2e-16
```

```
BIC(mod.C)
```

```
## [1] -44416.35
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.C)
```




```
par(mfrow=c(1,1))
```

From a comparison between this three final models we have chosen the second one, mod.B. This model have high performances in comparison with mod.A and more significant estimated coefficients with respect to mod.C. Furthermore, looking at the residuals of mod.B and mod.C we can see that the plots are rather the same, there are not big differences.

4.1 Cross Validation of the final model

In order to check the presence of the overfitting effect of our model we calculate the MSE of our final model with a 10-fold cross validation and without cross-validation. We observed that the results obtained are comparable and therefore they don't look alarming.

```r
best.fit <- lm(log10_price ~ sqft_living + I(sqft_living^2) + sqft_living15 + lat +
                             long + floors + bathrooms + condition + view + zipcode,
                             data=kc_housing)

model.glm <- glm(formula = log10_price ~ sqft_living + I(sqft_living^2) + sqft_living15 +
                                         lat + long + floors + bathrooms + condition +
                                         view + zipcode, data=kc_housing)
summary(model.glm)
```

```
##
## Call:
## glm(formula = log10_price ~ sqft_living + I(sqft_living^2) +
##     sqft_living15 + lat + long + floors + bathrooms + condition +
##     view + zipcode, data = kc_housing)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.53580  -0.04721   0.00129   0.04747   0.47856
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2.284e+01  3.256e+00  -7.014 2.38e-12 ***
## sqft_living       1.493e-04  3.049e-06  48.964  < 2e-16 ***
## I(sqft_living^2) -8.358e-09  4.990e-10 -16.749  < 2e-16 ***
## sqft_living15     5.214e-05  1.489e-06  35.026  < 2e-16 ***
## lat               2.356e-01  3.373e-02   6.986 2.92e-12 ***
## long             -1.348e-01  2.416e-02  -5.580 2.44e-08 ***
## floors1.5         1.502e-02  2.241e-03   6.703 2.09e-11 ***
## floors2           1.303e-02  1.732e-03   7.521 5.65e-14 ***
## floors2.5         2.852e-02  7.017e-03   4.064 4.84e-05 ***
## floors3          -1.331e-02  4.013e-03  -3.316 0.000915 ***
## floors3.5         5.583e-03  3.248e-02   0.172 0.863543
## bathrooms0.75     1.061e-01  4.395e-02   2.413 0.015832 *
## bathrooms1        1.242e-01  4.278e-02   2.902 0.003710 **
## bathrooms1.25     1.328e-01  5.142e-02   2.583 0.009813 **
## bathrooms1.5      1.283e-01  4.282e-02   2.996 0.002743 **
## bathrooms1.75     1.388e-01  4.280e-02   3.242 0.001188 **
## bathrooms2        1.366e-01  4.282e-02   3.190 0.001426 **
## bathrooms2.25     1.435e-01  4.283e-02   3.351 0.000808 ***
## bathrooms2.5      1.465e-01  4.281e-02   3.422 0.000623 ***
## bathrooms2.75     1.437e-01  4.287e-02   3.352 0.000805 ***
## bathrooms3        1.443e-01  4.292e-02   3.362 0.000774 ***
## bathrooms3.25     1.651e-01  4.297e-02   3.842 0.000122 ***
## bathrooms3.5      1.576e-01  4.295e-02   3.670 0.000243 ***
## bathrooms3.75     1.762e-01  4.340e-02   4.061 4.91e-05 ***
## bathrooms4        1.677e-01  4.349e-02   3.855 0.000116 ***
## bathrooms4.25     1.764e-01  4.399e-02   4.009 6.11e-05 ***
## bathrooms4.5      1.542e-01  4.375e-02   3.526 0.000423 ***
## bathrooms4.75     2.054e-01  4.659e-02   4.408 1.05e-05 ***
## bathrooms5        1.832e-01  4.685e-02   3.910 9.27e-05 ***
## bathrooms5.25     1.786e-01  4.914e-02   3.635 0.000278 ***
## bathrooms5.5      2.630e-01  5.124e-02   5.133 2.88e-07 ***
## bathrooms5.75     1.906e-01  6.160e-02   3.094 0.001979 **
```

```
## bathrooms6       2.031e-01  5.583e-02   3.638 0.000276 ***
## bathrooms6.25    2.902e-01  7.581e-02   3.828 0.000130 ***
## bathrooms6.5     2.894e-01  7.468e-02   3.876 0.000107 ***
## bathrooms6.75    1.363e-01  7.626e-02   1.787 0.073898 .
## bathrooms8       4.203e-01  8.977e-02   4.683 2.85e-06 ***
## condition2       8.037e-02  1.721e-02   4.670 3.03e-06 ***
## condition3       1.410e-01  1.596e-02   8.835  < 2e-16 ***
## condition4       1.548e-01  1.598e-02   9.685  < 2e-16 ***
## condition5       1.765e-01  1.608e-02  10.973  < 2e-16 ***
## view1            4.671e-02  4.846e-03   9.639  < 2e-16 ***
## view2            5.143e-02  2.953e-03  17.416  < 2e-16 ***
## view3            8.752e-02  4.021e-03  21.767  < 2e-16 ***
## view4            2.150e-01  5.072e-03  42.382  < 2e-16 ***
## zipcode98002    -9.342e-03  7.693e-03  -1.214 0.224624
## zipcode98003     6.564e-03  6.876e-03   0.955 0.339841
## zipcode98004     4.328e-01  1.250e-02  34.612  < 2e-16 ***
## zipcode98005     2.762e-01  1.336e-02  20.672  < 2e-16 ***
## zipcode98006     2.395e-01  1.092e-02  21.934  < 2e-16 ***
## zipcode98007     2.405e-01  1.378e-02  17.449  < 2e-16 ***
## zipcode98008     2.370e-01  1.309e-02  18.104  < 2e-16 ***
## zipcode98010     1.494e-01  1.173e-02  12.739  < 2e-16 ***
## zipcode98011     9.311e-02  1.704e-02   5.466 4.66e-08 ***
## zipcode98014     1.235e-01  1.870e-02   6.606 4.05e-11 ***
## zipcode98019     7.934e-02  1.846e-02   4.298 1.73e-05 ***
## zipcode98022     9.931e-02  1.019e-02   9.745  < 2e-16 ***
## zipcode98023    -2.086e-02  6.332e-03  -3.294 0.000989 ***
## zipcode98024     2.026e-01  1.645e-02  12.316  < 2e-16 ***
## zipcode98027     2.051e-01  1.121e-02  18.298  < 2e-16 ***
## zipcode98028     7.568e-02  1.655e-02   4.574 4.82e-06 ***
## zipcode98029     2.448e-01  1.280e-02  19.129  < 2e-16 ***
## zipcode98030     1.735e-02  7.561e-03   2.295 0.021742 *
## zipcode98031     1.814e-02  7.878e-03   2.302 0.021317 *
## zipcode98032    -2.720e-02  9.147e-03  -2.974 0.002943 **
## zipcode98033     2.710e-01  1.420e-02  19.087  < 2e-16 ***
## zipcode98034     1.474e-01  1.522e-02   9.684  < 2e-16 ***
## zipcode98038     8.536e-02  8.499e-03  10.043  < 2e-16 ***
## zipcode98039     5.139e-01  1.703e-02  30.175  < 2e-16 ***
## zipcode98040     3.394e-01  1.106e-02  30.700  < 2e-16 ***
## zipcode98042     3.545e-02  7.239e-03   4.897 9.80e-07 ***
## zipcode98045     1.899e-01  1.569e-02  12.098  < 2e-16 ***
## zipcode98052     2.160e-01  1.449e-02  14.905  < 2e-16 ***
## zipcode98053     2.110e-01  1.553e-02  13.584  < 2e-16 ***
## zipcode98055     2.945e-02  8.764e-03   3.361 0.000778 ***
## zipcode98056     9.843e-02  9.535e-03  10.323  < 2e-16 ***
## zipcode98058     6.081e-02  8.287e-03   7.339 2.23e-13 ***
## zipcode98059     1.182e-01  9.350e-03  12.647  < 2e-16 ***
## zipcode98065     1.536e-01  1.445e-02  10.631  < 2e-16 ***
## zipcode98070     1.437e-01  1.079e-02  13.316  < 2e-16 ***
## zipcode98072     1.381e-01  1.695e-02   8.145 4.01e-16 ***
## zipcode98074     2.115e-01  1.371e-02  15.425  < 2e-16 ***
## zipcode98075     2.200e-01  1.319e-02  16.683  < 2e-16 ***
## zipcode98077     1.485e-01  1.763e-02   8.423  < 2e-16 ***
## zipcode98092     3.588e-02  6.878e-03   5.217 1.84e-07 ***
## zipcode98102     3.445e-01  1.457e-02  23.644  < 2e-16 ***
```

```
## zipcode98103      2.665e-01  1.370e-02  19.450  < 2e-16 ***
## zipcode98105      3.229e-01  1.402e-02  23.025  < 2e-16 ***
## zipcode98106      7.360e-02  1.017e-02   7.238 4.70e-13 ***
## zipcode98107      2.721e-01  1.413e-02  19.255  < 2e-16 ***
## zipcode98108      8.140e-02  1.120e-02   7.267 3.81e-13 ***
## zipcode98109      3.495e-01  1.451e-02  24.096  < 2e-16 ***
## zipcode98112      3.815e-01  1.284e-02  29.706  < 2e-16 ***
## zipcode98115      2.602e-01  1.392e-02  18.693  < 2e-16 ***
## zipcode98116      2.474e-01  1.132e-02  21.851  < 2e-16 ***
## zipcode98117      2.494e-01  1.410e-02  17.689  < 2e-16 ***
## zipcode98118      1.340e-01  9.882e-03  13.563  < 2e-16 ***
## zipcode98119      3.401e-01  1.369e-02  24.834  < 2e-16 ***
## zipcode98122      2.824e-01  1.220e-02  23.136  < 2e-16 ***
## zipcode98125      1.465e-01  1.506e-02   9.729  < 2e-16 ***
## zipcode98126      1.668e-01  1.041e-02  16.024  < 2e-16 ***
## zipcode98133      9.205e-02  1.556e-02   5.916 3.35e-09 ***
## zipcode98136      2.259e-01  1.067e-02  21.174  < 2e-16 ***
## zipcode98144      2.108e-01  1.138e-02  18.522  < 2e-16 ***
## zipcode98146      5.822e-02  9.538e-03   6.105 1.05e-09 ***
## zipcode98148      3.520e-02  1.299e-02   2.710 0.006726 **
## zipcode98155      7.848e-02  1.618e-02   4.851 1.23e-06 ***
## zipcode98166      9.712e-02  8.728e-03  11.128  < 2e-16 ***
## zipcode98168     -2.439e-02  9.223e-03  -2.644 0.008194 **
## zipcode98177      1.418e-01  1.624e-02   8.732  < 2e-16 ***
## zipcode98178      8.499e-03  9.511e-03   0.894 0.371566
## zipcode98188      3.870e-03  9.785e-03   0.395 0.692490
## zipcode98198      2.076e-03  7.421e-03   0.280 0.779681
## zipcode98199      2.808e-01  1.339e-02  20.981  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.007295851)
##
##     Null deviance: 1127.70  on 21599  degrees of freedom
## Residual deviance:  156.76  on 21486  degrees of freedom
## AIC: -44868
##
## Number of Fisher Scoring iterations: 2
```

```
cv.err2 <- cv.glm(kc_housing, model.glm, K=10)
cv.err2$delta[1]
```

```
## [1] 0.007392514
```

```
mse(kc_housing$log10_price, best.fit$fitted.values)
```

```
## [1] 0.007257345
```

After this final check we confirm that this model is the best model obtained given our analysis.

5 Conclusions

In the end, our final model is model mod.B which has as outcome variable the *log10_price*, as numerical predictors the variables *sqft_living*, *sqft_living15*, *lat*, *long* and as categorical predictors the variables *floors*, *bathrooms*, *condition*, *view* and *zipcode*. All the estimated coefficients of the numerical variables are significant, also all the estimated coefficients of the categorical variables are significant, except for some levels which can be ignored.
This model reaches an Adjusted R^2 of 86%, so 86% of the variability of the response variable is explained by the explanatory variables that compose the model. Furthermore, since the final model has a high Adjusted R^2, we have an indicator of an accurate prediction for future values.