install.packages("magrittr") install.packages("ggmap") install.packages("geosphere") install.packages("httr")

In [1]:
```
install.packages("rpart.plot")
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' ... done
```

# King County Housing Prices ¶

***Sruthi Jogi, Jonathan McFadden, Angela Zhao***

**TCSS-551: Big Data Analytics** -:- ***Autumn 2017***

# Introduction

## Overview

For our final project, we have chosen to analyze data covering housing sales in King County. To do this, we are using the data from the **Kaggle** *King County House Sales Prediction* page at

https://www.kaggle.com/harlfoxem/housesalesprediction

From this page, we sign-up for an account (*free, but required for downloading*) and then download the *zip* file containing the *CSV* file with the data.

Our goal is to use this data to create models for home sales in King County based on the feature information provided in the obtained data file. Our eventual goal is two-fold. First, we wish to create a model or models which will enable us to quantitatively predict house sale prices, using this data set as the basis for our model or models. Our other goal is to determine, based on the obtained data, which features are most important to the sale price of a house.

## Data File

Our first task is to import, examine, and then give an overall description of the data. We are especially interested in the size and descriptive contents of the data file. Specifially, we want to know the number of sales contined within the data file and, especially, what parameters the data file uses to describe each house sale. Furthermore, we want to check the import to ensure that the data was initially complete, that it was then imported correctly, and that ***R*** is interpreting the imported data properly.

### Import and First-Look

We begin by importing the data file into the '***houseDFo***( )' data frame. This data frame will serve as an intial data-frame, not the working one. This is because we may need an initial frame to reload as a we clean the data, allowing us to avoid having to reimport the CSV file over ang over again. Thus, we now import the CSV file into this initial data frame.

In [2]:
```
houseDFo <- read.csv("../houseData.csv")
```

We are now interested in the number of data-points contined within the data file. Thus, we want to see how many row ***R*** has imported.

In [3]:
```
nrow(houseDFo)
```

21613

We also want to see how many descriptors the imported data uses to describe each house sale. Thus we want to see how many columns ***R*** has imported.

In [4]:
```
ncol(houseDFo)
```

21

In addition, we want to see what the labels for those columns are and what type of values the elements of each column have (*interger, numeric, string, etc.*)

In [5]:
```
sapply(houseDFo, class)
```

| | |
|---|---|
| **id** | 'numeric' |
| **date** | 'factor' |
| **price** | 'numeric' |
| **bedrooms** | 'integer' |
| **bathrooms** | 'numeric' |
| **sqft_living** | 'integer' |
| **sqft_lot** | 'integer' |
| **floors** | 'numeric' |
| **waterfront** | 'integer' |
| **view** | 'integer' |
| **condition** | 'integer' |
| **grade** | 'integer' |
| **sqft_above** | 'integer' |
| **sqft_basement** | 'integer' |
| **yr_built** | 'integer' |
| **yr_renovated** | 'integer' |
| **zipcode** | 'integer' |
| **lat** | 'numeric' |
| **long** | 'numeric' |
| **sqft_living15** | 'integer' |
| **sqft_lot15** | 'integer' |

From above, it is clear that the **date** column did not import as a *date*, instead importing as a *factor*. Therefore, we will now examine the first few rows of the imported data to see what may have caused the issues with imporation.

In [6]:
```
head(houseDFo)
```

| id | date | price | bedrooms | bathrooms | sqft_living | sqft_lot | floor |
|---|---|---|---|---|---|---|---|
| 7129300520 | 20141013T000000 | 221900 | 3 | 1.00 | 1180 | 5650 | 1 |
| 6414100192 | 20141209T000000 | 538000 | 3 | 2.25 | 2570 | 7242 | 2 |
| 5631500400 | 20150225T000000 | 180000 | 2 | 1.00 | 770 | 10000 | 1 |
| 2487200875 | 20141209T000000 | 604000 | 4 | 3.00 | 1960 | 5000 | 1 |
| 1954400510 | 20150218T000000 | 510000 | 3 | 2.00 | 1680 | 8080 | 1 |
| 7237550310 | 20140512T000000 | 1225000 | 4 | 4.50 | 5420 | 101930 | 1 |

Clearly, some elements of the data file did not import correctly; therefore, we must clean the data before we can proceed to analysis.

## Clean the Data

### Missing Data

First, we will check to see if there are any missing data points.

In [7]:
```
houseDFo[!complete.cases(houseDFo),]
```

```
Warning message in cbind(parts$left, ellip_h, parts$right, deparse.leve
l = 0L):
“number of rows of result is not a multiple of vector length (arg 2)”Wa
rning message in cbind(parts$left, ellip_h, parts$right, deparse.level
= 0L):
“number of rows of result is not a multiple of vector length (arg 2)”Wa
rning message in cbind(parts$left, ellip_h, parts$right, deparse.level
= 0L):
“number of rows of result is not a multiple of vector length (arg 2)”Wa
rning message in cbind(parts$left, ellip_h, parts$right, deparse.level
= 0L):
“number of rows of result is not a multiple of vector length (arg 2)”
```

| id | date | price | bedrooms | bathrooms | sqft_living | sqft_lot | floors | waterfront | view | ⋯ | gr |
|---|---|---|---|---|---|---|---|---|---|---|---|

Since there are no missing data points, we can move on to the dates.

#### Dates

From the first few rows of the data table seen above, it is clear that we must first strip the "*T000000*" string at the end of every date. To do this, we require the **stringr** library. Thus, we import **stringr**

In [8]:
```
library(stringr)
```

so we can now strip the offending substrings. Before stripping these substrings, we create a copy of our initial data frame, ***houseDFo***( ), so that our initial import data frame will remain untouched, and therefore available for reloading other data frames. Thus, we create the copy and strip the substrings, storing the result in the copied data frame ***houseDFo1***( ).

In [9]:
```
houseDFo1 <- houseDFo
houseDFo1$date = str_replace(houseDFo$date, "T000000", "")
```

We now examine the result of this

In [10]:
```
head(houseDFo1)
```

| id | date | price | bedrooms | bathrooms | sqft_living | sqft_lot | floors | water |
|---|---|---|---|---|---|---|---|---|
| 7129300520 | 20141013 | 221900 | 3 | 1.00 | 1180 | 5650 | 1 | 0 |
| 6414100192 | 20141209 | 538000 | 3 | 2.25 | 2570 | 7242 | 2 | 0 |
| 5631500400 | 20150225 | 180000 | 2 | 1.00 | 770 | 10000 | 1 | 0 |
| 2487200875 | 20141209 | 604000 | 4 | 3.00 | 1960 | 5000 | 1 | 0 |
| 1954400510 | 20150218 | 510000 | 3 | 2.00 | 1680 | 8080 | 1 | 0 |
| 7237550310 | 20140512 | 1225000 | 4 | 4.50 | 5420 | 101930 | 1 | 0 |

The dates are now just strings of numbers with the format '*yyyymmdd*'; therefore, we can use the date conversion method from ***R*** to convert these dates.

In [11]:
```
houseDFo1 <- transform(houseDFo1, date = as.Date(date, "%Y%m%d"))
```

To ensure that the conversion to dates happend properly, we will no check the column data types followed by looking at the first few rows of the data.

In [12]:
```
sapply(houseDFo1, class)
head(houseDFo1)
```

| | |
|---|---|
| **id** | 'numeric' |
| **date** | 'Date' |
| **price** | 'numeric' |
| **bedrooms** | 'integer' |
| **bathrooms** | 'numeric' |
| **sqft_living** | 'integer' |
| **sqft_lot** | 'integer' |
| **floors** | 'numeric' |
| **waterfront** | 'integer' |
| **view** | 'integer' |
| **condition** | 'integer' |
| **grade** | 'integer' |
| **sqft_above** | 'integer' |
| **sqft_basement** | 'integer' |
| **yr_built** | 'integer' |
| **yr_renovated** | 'integer' |
| **zipcode** | 'integer' |
| **lat** | 'numeric' |
| **long** | 'numeric' |
| **sqft_living15** | 'integer' |
| **sqft_lot15** | 'integer' |

| id | date | price | bedrooms | bathrooms | sqft_living | sqft_lot | floors | waterfron |
|---|---|---|---|---|---|---|---|---|
| 7129300520 | 2014-10-13 | 221900 | 3 | 1.00 | 1180 | 5650 | 1 | 0 |
| 6414100192 | 2014-12-09 | 538000 | 3 | 2.25 | 2570 | 7242 | 2 | 0 |
| 5631500400 | 2015-02-25 | 180000 | 2 | 1.00 | 770 | 10000 | 1 | 0 |
| 2487200875 | 2014-12-09 | 604000 | 4 | 3.00 | 1960 | 5000 | 1 | 0 |
| 1954400510 | 2015-02-18 | 510000 | 3 | 2.00 | 1680 | 8080 | 1 | 0 |
| 7237550310 | 2014-05-12 | 1225000 | 4 | 4.50 | 5420 | 101930 | 1 | 0 |

Since the results for the date conversions are as desired, we can now store the data in a final data frame followed by moving on to begining our analysis.

In [13]:
```
houseDF <- houseDFo1
```

We will also create a version of the data with the **ID** column stripped out.

In [14]:
```
houseDFa <- houseDF[-c(1)]
```

# Initial Analysis

To begin our analysis, we will look at the basic statistics of every column (*except the date*).

In [15]:
```
sapply(houseDFa[-c(1)], function(x) list(mean=mean(x),
                                         stdev=sd(x, na.rm=TRUE)))
```

| | price | bedrooms | bathrooms | sqft_living | sqft_lot | floors | waterfront | vie |
|---|---|---|---|---|---|---|---|---|
| **mean** | 540088.1 | 3.370842 | 2.114757 | 2079.9 | 15106.97 | 1.494309 | 0.007541757 | 0.2 |
| **stdev** | 367127.2 | 0.9300618 | 0.7701632 | 918.4409 | 41420.51 | 0.5399889 | 0.0865172 | 0.7 |

and get a summary of the entire

In [16]:
```
summary(houseDFa)
```

```
      date                price             bedrooms        bathrooms      
 Min.   :2014-05-02   Min.   :  75000   Min.   : 0.000   Min.   :0.000  
 1st Qu.:2014-07-22   1st Qu.: 321950   1st Qu.: 3.000   1st Qu.:1.750  
 Median :2014-10-16   Median : 450000   Median : 3.000   Median :2.250  
 Mean   :2014-10-29   Mean   : 540088   Mean   : 3.371   Mean   :2.115  
 3rd Qu.:2015-02-17   3rd Qu.: 645000   3rd Qu.: 4.000   3rd Qu.:2.500  
 Max.   :2015-05-27   Max.   :7700000   Max.   :33.000   Max.   :8.000  
  sqft_living       sqft_lot           floors        waterfront      
 Min.   :  290   Min.   :    520   Min.   :1.000   Min.   :0.000000  
 1st Qu.: 1427   1st Qu.:   5040   1st Qu.:1.000   1st Qu.:0.000000  
 Median : 1910   Median :   7618   Median :1.500   Median :0.000000  
 Mean   : 2080   Mean   :  15107   Mean   :1.494   Mean   :0.007542  
 3rd Qu.: 2550   3rd Qu.:  10688   3rd Qu.:2.000   3rd Qu.:0.000000  
 Max.   :13540   Max.   :1651359   Max.   :3.500   Max.   :1.000000  
      view          condition         grade          sqft_above    
 Min.   :0.0000   Min.   :1.000   Min.   : 1.000   Min.   : 290  
 1st Qu.:0.0000   1st Qu.:3.000   1st Qu.: 7.000   1st Qu.:1190  
 Median :0.0000   Median :3.000   Median : 7.000   Median :1560  
 Mean   :0.2343   Mean   :3.409   Mean   : 7.657   Mean   :1788  
 3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.: 8.000   3rd Qu.:2210  
 Max.   :4.0000   Max.   :5.000   Max.   :13.000   Max.   :9410  
 sqft_basement       yr_built     yr_renovated       zipcode      
 Min.   :   0.0   Min.   :1900   Min.   :   0.0   Min.   :98001  
 1st Qu.:   0.0   1st Qu.:1951   1st Qu.:   0.0   1st Qu.:98033  
 Median :   0.0   Median :1975   Median :   0.0   Median :98065  
 Mean   : 291.5   Mean   :1971   Mean   :  84.4   Mean   :98078  
 3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:   0.0   3rd Qu.:98118  
 Max.   :4820.0   Max.   :2015   Max.   :2015.0   Max.   :98199  
      lat             long         sqft_living15    sqft_lot15     
 Min.   :47.16   Min.   :-122.5   Min.   : 399   Min.   :   651  
 1st Qu.:47.47   1st Qu.:-122.3   1st Qu.:1490   1st Qu.:  5100  
 Median :47.57   Median :-122.2   Median :1840   Median :  7620  
 Mean   :47.56   Mean   :-122.2   Mean   :1987   Mean   : 12768  
 3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2360   3rd Qu.: 10083  
 Max.   :47.78   Max.   :-121.3   Max.   :6210   Max.   :871200  
```

We also run a simple linear model on the *entire* dataset so that we can see how significant each variable is to determining the price (*basically running a t-Test on all variables*). To do this, we need to **nnet** library, so we load it

In [17]:
```
library(nnet)
```

Then we run the model and display the results.

In [18]:
```
house.lm.tot <- lm(price ~., data=houseDFa)
summary(house.lm.tot)
```

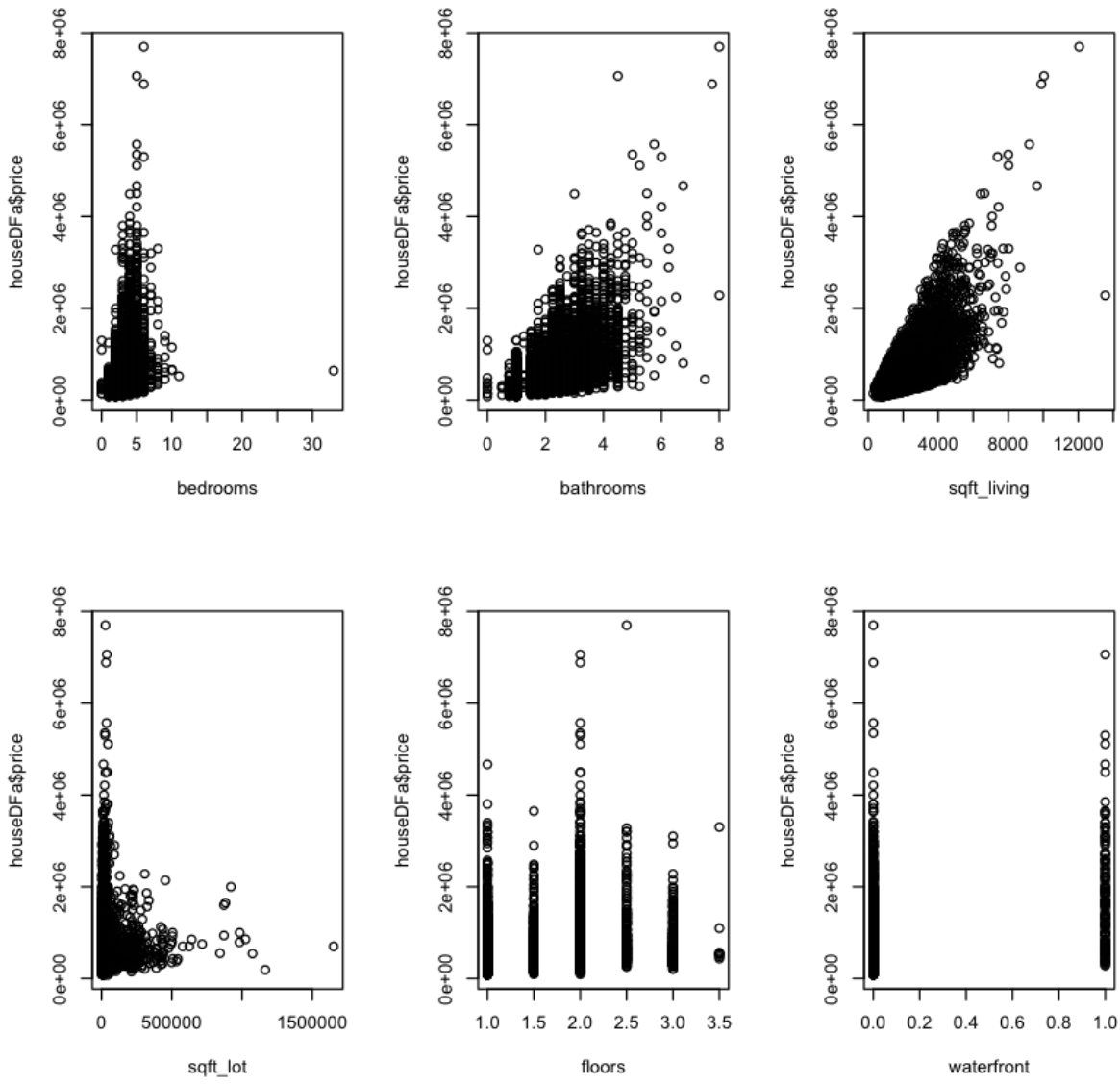
```
Call:
lm(formula = price ~ ., data = houseDFa)

Residuals:
     Min       1Q   Median       3Q      Max 
-1306672   -98900    -8963    77327  4330103 

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    4.618e+06  2.933e+06   1.574  0.11539    
date           1.165e+02  1.213e+01   9.608  < 2e-16 ***
bedrooms      -3.588e+04  1.888e+03 -19.005  < 2e-16 ***
bathrooms      4.137e+04  3.247e+03  12.741  < 2e-16 ***
sqft_living    1.502e+02  4.376e+00  34.327  < 2e-16 ***
sqft_lot       1.257e-01  4.782e-02   2.629  0.00858 ** 
floors         7.158e+03  3.589e+03   1.995  0.04610 *  
waterfront     5.826e+05  1.732e+04  33.628  < 2e-16 ***
view           5.260e+04  2.136e+03  24.629  < 2e-16 ***
condition      2.774e+04  2.351e+03  11.799  < 2e-16 ***
grade          9.624e+04  2.149e+03  44.791  < 2e-16 ***
sqft_above     3.084e+01  4.351e+00   7.088 1.40e-12 ***
sqft_basement         NA         NA      NA       NA    
yr_built      -2.618e+03  7.251e+01 -36.113  < 2e-16 ***
yr_renovated   2.079e+01  3.649e+00   5.698 1.23e-08 ***
zipcode       -5.807e+02  3.292e+01 -17.643  < 2e-16 ***
lat            6.053e+05  1.072e+04  56.487  < 2e-16 ***
long          -2.136e+05  1.311e+04 -16.300  < 2e-16 ***
sqft_living15  2.195e+01  3.441e+00   6.381 1.79e-10 ***
sqft_lot15    -3.825e-01  7.311e-02  -5.232 1.69e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 200800 on 21594 degrees of freedom
Multiple R-squared:  0.701,     Adjusted R-squared:  0.7008 
F-statistic:  2813 on 18 and 21594 DF,  p-value: < 2.2e-16
```

We also run a *general lineary model* on the entire dataset for comparison.

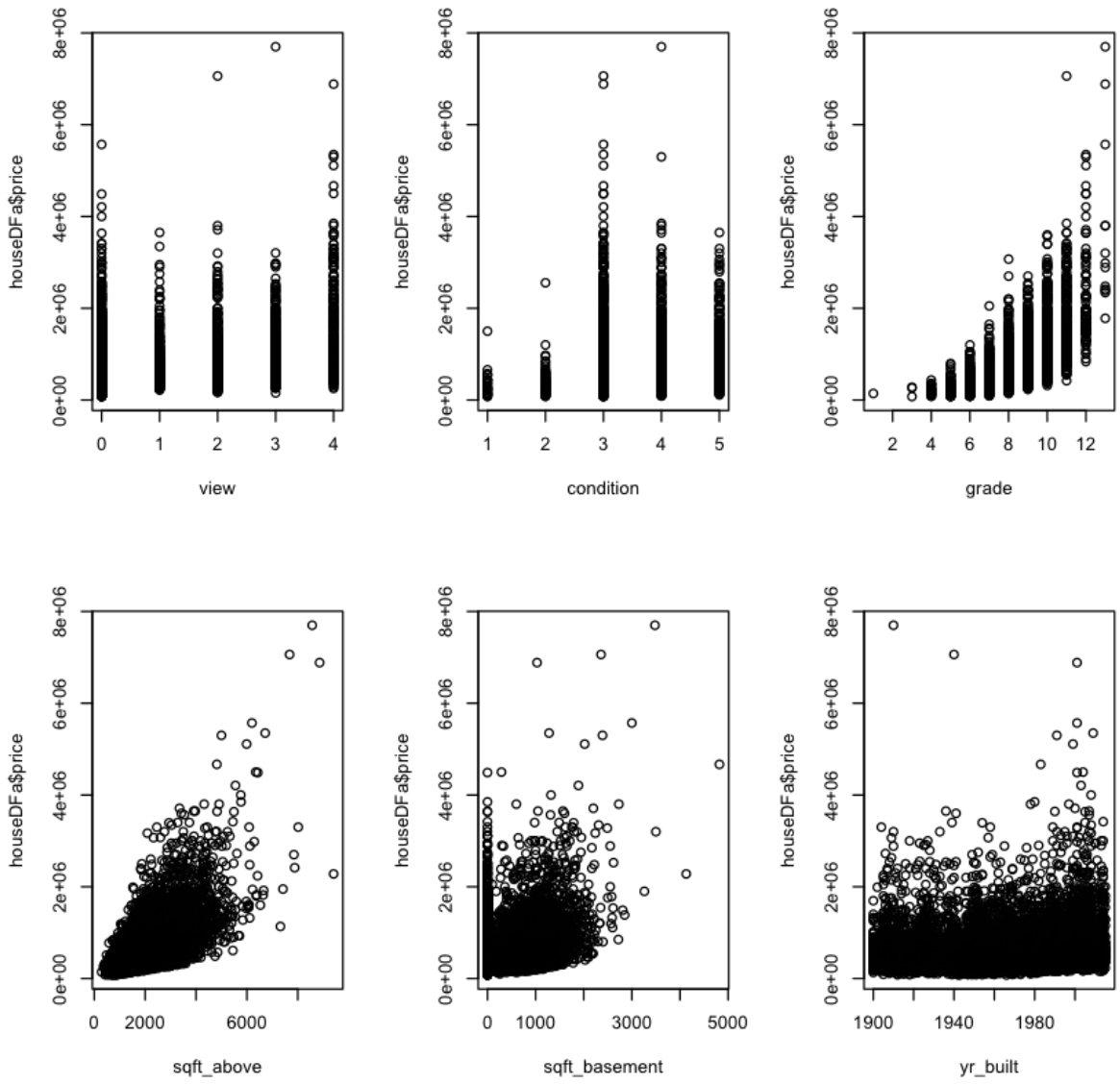
In [19]:
```
house.glm.tot <- glm(price ~., data=houseDFa)
summary(house.glm.tot)
```

```
Call:
glm(formula = price ~ ., data = houseDFa)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1306672    -98900     -8963     77327   4330103

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.618e+06  2.933e+06   1.574  0.11539
date           1.165e+02  1.213e+01   9.608  < 2e-16 ***
bedrooms      -3.588e+04  1.888e+03 -19.005  < 2e-16 ***
bathrooms      4.137e+04  3.247e+03  12.741  < 2e-16 ***
sqft_living    1.502e+02  4.376e+00  34.327  < 2e-16 ***
sqft_lot       1.257e-01  4.782e-02   2.629  0.00858 **
floors         7.158e+03  3.589e+03   1.995  0.04610 *
waterfront     5.826e+05  1.732e+04  33.628  < 2e-16 ***
view           5.260e+04  2.136e+03  24.629  < 2e-16 ***
condition      2.774e+04  2.351e+03  11.799  < 2e-16 ***
grade          9.624e+04  2.149e+03  44.791  < 2e-16 ***
sqft_above     3.084e+01  4.351e+00   7.088 1.40e-12 ***
sqft_basement         NA         NA      NA       NA
yr_built      -2.618e+03  7.251e+01 -36.113  < 2e-16 ***
yr_renovated   2.079e+01  3.649e+00   5.698 1.23e-08 ***
zipcode       -5.807e+02  3.292e+01 -17.643  < 2e-16 ***
lat            6.053e+05  1.072e+04  56.487  < 2e-16 ***
long          -2.136e+05  1.311e+04 -16.300  < 2e-16 ***
sqft_living15  2.195e+01  3.441e+00   6.381 1.79e-10 ***
sqft_lot15    -3.825e-01  7.311e-02  -5.232 1.69e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 40330106193)

    Null deviance: 2.9129e+15  on 21612  degrees of freedom
Residual deviance: 8.7089e+14  on 21594  degrees of freedom
AIC: 589153

Number of Fisher Scoring iterations: 2
```

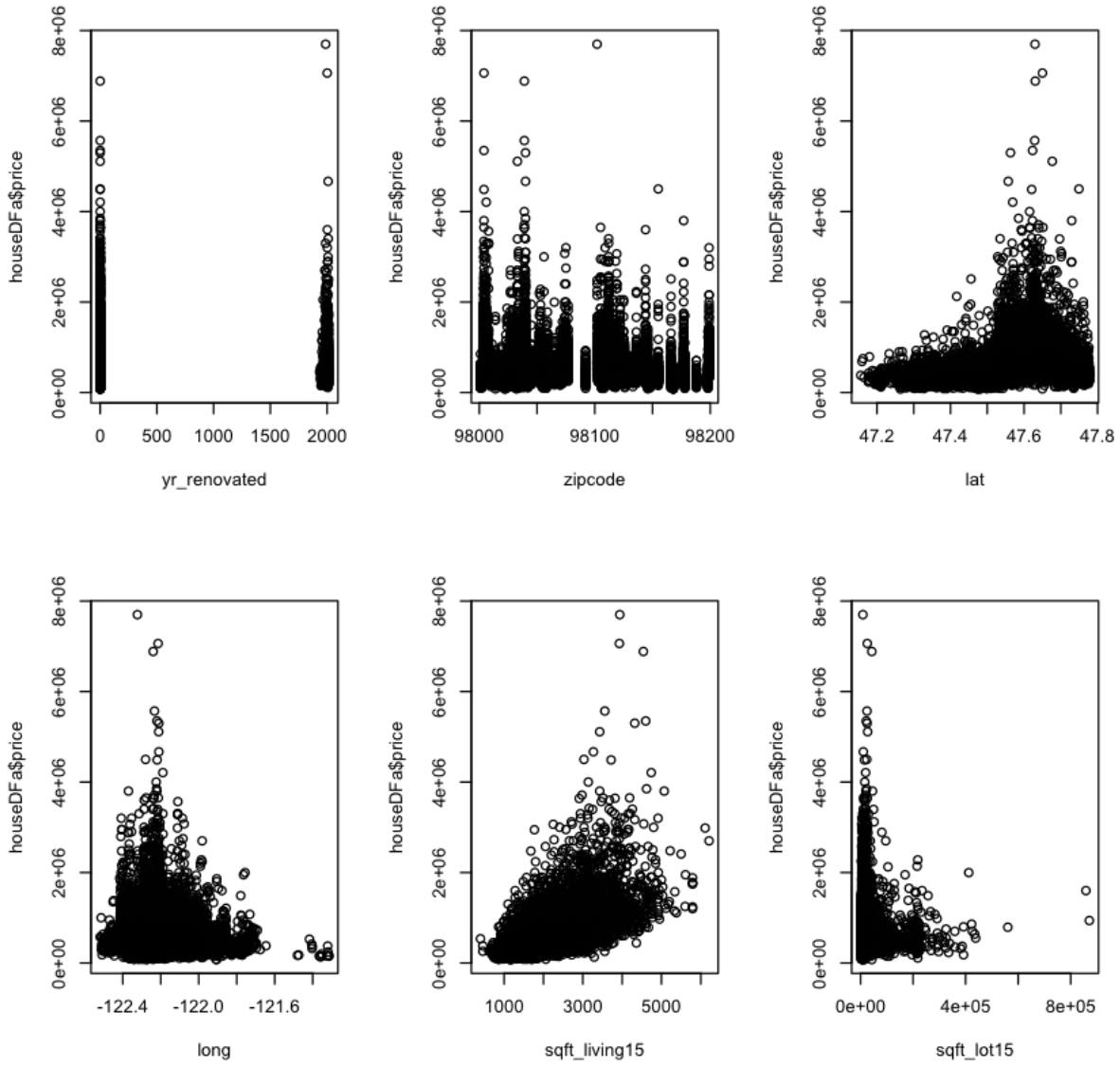
In [20]:
```
par(mfrow=c(2,3))
for(i in 3:8) {plot(houseDFa[,i], houseDFa$price, xlab=names(houseDFa[i
]), ylab=names(houseDFa$price))}
```



In [21]:
```
par(mfrow=c(2,3))
for(i in 9:14) {plot(houseDFa[,i], houseDFa$price, xlab=names(houseDFa[i
]), ylab=names(houseDFa$price))}
```



In [22]:
```
par(mfrow=c(2,3))
for(i in 15:20) {plot(houseDFa[,i], houseDFa$price, xlab=names(houseDFa[
i]), ylab=names(houseDFa$price))}
```




In [23]:
```
#library(party)
```

In [24]:
```
#install.packages("party")
```

In [25]:
```
library(rpart)
```

In [26]:
```
house.dTree <- rpart(houseDFa$price ~., houseDFa[,-c(1,2)], na.action =
 na.rpart)
summary(house.dTree)
```

```
Call:
rpart(formula = houseDFa$price ~ ., data = houseDFa[, -c(1, 2)],
    na.action = na.rpart)
  n= 21613

           CP nsplit rel error    xerror       xstd
1  0.32027044      0 1.0000000 1.0000823 0.04114139
2  0.11478159      1 0.6797296 0.6880212 0.03276257
3  0.06425998      2 0.5649480 0.5749783 0.02466590
4  0.04082237      3 0.5006880 0.5026014 0.02434521
5  0.03003412      4 0.4598656 0.4810306 0.01952726
6  0.02807244      5 0.4298315 0.4500035 0.01506337
7  0.02474470      6 0.4017591 0.4173948 0.01483681
8  0.02077484      7 0.3770144 0.4047731 0.01429445
9  0.01287892      9 0.3354647 0.3690401 0.01318614
10 0.01000000     11 0.3097069 0.3463446 0.01201177

Variable importance
        grade   sqft_living    sqft_above sqft_living15     bathrooms
           23            22            16            12             8
          lat sqft_basement          long       zipcode    sqft_lot15
            7             4             3             3             1
     yr_built      sqft_lot      bedrooms
            1             1             1

Node number 1: 21613 observations,    complexity param=0.3202704
  mean=540088.1, MSE=1.347761e+11
  left son=2 (17362 obs) right son=3 (4251 obs)
  Primary splits:
      grade         < 8.5       to the left,  improve=0.3202704, (0 mis
sing)
      sqft_living   < 3406      to the left,  improve=0.3095934, (0 mis
sing)
      sqft_living15 < 2835      to the left,  improve=0.2480898, (0 mis
sing)
      sqft_above    < 2829      to the left,  improve=0.2245114, (0 mis
sing)
      bathrooms     < 3.125     to the left,  improve=0.2172293, (0 mis
sing)
  Surrogate splits:
      sqft_above    < 2495.5    to the left,  agree=0.885, adj=0.414,
(0 split)
      sqft_living15 < 2644      to the left,  agree=0.884, adj=0.413,
(0 split)
      sqft_living   < 2923.5    to the left,  agree=0.882, adj=0.399,
(0 split)
      bathrooms     < 3.125     to the left,  agree=0.840, adj=0.189,
(0 split)
      sqft_basement < 1515      to the left,  agree=0.810, adj=0.033,
(0 split)

Node number 2: 17362 observations,    complexity param=0.06425998
  mean=437284, MSE=3.834004e+10
  left son=4 (7304 obs) right son=5 (10058 obs)
  Primary splits:
      lat           < 47.53435  to the left,  improve=0.28120070, (0 mi
ssing)
```

```
      sqft_living   < 2039      to the left,  improve=0.15821600, (0 mi
ssing)
      grade         < 7.5       to the left,  improve=0.15617670, (0 mi
ssing)
      sqft_living15 < 2009.5    to the left,  improve=0.11275160, (0 mi
ssing)
      sqft_above    < 1416.5    to the left,  improve=0.08694896, (0 mi
ssing)
  Surrogate splits:
      zipcode    < 98071     to the left,  agree=0.666, adj=0.207, (0 s
plit)
      sqft_lot15 < 6442      to the right, agree=0.607, adj=0.066, (0 s
plit)
      sqft_lot   < 7201.5    to the right, agree=0.607, adj=0.065, (0 s
plit)
      grade      < 6.5       to the left,  agree=0.602, adj=0.053, (0 s
plit)
      long       < -122.2255 to the right, agree=0.601, adj=0.051, (0 s
plit)

Node number 3: 4251 observations,    complexity param=0.1147816
  mean=959962.4, MSE=3.091828e+11
  left son=6 (3667 obs) right son=7 (584 obs)
  Primary splits:
      sqft_living < 4185      to the left,  improve=0.2543864, (0 missi
ng)
      grade       < 10.5      to the left,  improve=0.2236842, (0 missi
ng)
      sqft_above  < 4235      to the left,  improve=0.1694131, (0 missi
ng)
      bathrooms   < 3.625     to the left,  improve=0.1654821, (0 missi
ng)
      lat         < 47.5247   to the left,  improve=0.1186550, (0 missi
ng)
  Surrogate splits:
      sqft_above    < 4185      to the left,  agree=0.930, adj=0.491,
(0 split)
      bathrooms     < 3.875     to the left,  agree=0.895, adj=0.235,
(0 split)
      grade         < 10.5      to the left,  agree=0.888, adj=0.185,
(0 split)
      sqft_living15 < 4145      to the left,  agree=0.886, adj=0.168,
(0 split)
      sqft_basement < 1585      to the left,  agree=0.879, adj=0.120,
(0 split)

Node number 4: 7304 observations
  mean=315438.4, MSE=1.373778e+10

Node number 5: 10058 observations,    complexity param=0.02807244
  mean=525766.8, MSE=3.759545e+10
  left son=10 (6761 obs) right son=11 (3297 obs)
  Primary splits:
      sqft_living   < 2035      to the left,  improve=0.2162526, (0 mis
sing)
      sqft_above    < 1448      to the left,  improve=0.1552103, (0 mis
sing)
```

```
      sqft_living15 < 1875      to the left,  improve=0.1535210, (0 mis
sing)
      grade         < 7.5       to the left,  improve=0.1448457, (0 mis
sing)
      view          < 0.5       to the left,  improve=0.0954927, (0 mis
sing)
  Surrogate splits:
      sqft_above    < 2035      to the left,  agree=0.815, adj=0.436,
(0 split)
      sqft_living15 < 2005      to the left,  agree=0.783, adj=0.339,
(0 split)
      sqft_basement < 725       to the left,  agree=0.783, adj=0.339,
(0 split)
      bedrooms      < 3.5       to the left,  agree=0.778, adj=0.324,
(0 split)
      bathrooms     < 2.375     to the left,  agree=0.747, adj=0.230,
(0 split)

Node number 6: 3667 observations,    complexity param=0.03003412
  mean=848043, MSE=1.345633e+11
  left son=12 (810 obs) right son=13 (2857 obs)
  Primary splits:
      lat         < 47.5231   to the left,  improve=0.1772986, (0 missi
ng)
      yr_built    < 1973.5    to the right, improve=0.1576222, (0 missi
ng)
      sqft_living < 3406      to the left,  improve=0.1260728, (0 missi
ng)
      grade       < 9.5       to the left,  improve=0.1200231, (0 missi
ng)
      long        < -122.1885 to the right, improve=0.1035387, (0 missi
ng)
  Surrogate splits:
      zipcode    < 98003.5   to the left,  agree=0.794, adj=0.067, (0 s
plit)
      long       < -121.849  to the right, agree=0.789, adj=0.044, (0 s
plit)
      sqft_lot   < 92129     to the right, agree=0.788, adj=0.042, (0 s
plit)
      sqft_lot15 < 65809     to the right, agree=0.786, adj=0.031, (0 s
plit)
      sqft_above < 750       to the left,  agree=0.780, adj=0.004, (0 s
plit)

Node number 7: 584 observations,    complexity param=0.04082237
  mean=1662717, MSE=8.331216e+11
  left son=14 (574 obs) right son=15 (10 obs)
  Primary splits:
      sqft_living < 7940      to the left,  improve=0.2444021, (0 missi
ng)
      long        < -122.1875 to the right, improve=0.2019276, (0 missi
ng)
      waterfront  < 0.5       to the left,  improve=0.1626406, (0 missi
ng)
      grade       < 11.5      to the left,  improve=0.1426266, (0 missi
ng)
      sqft_above  < 6115      to the left,  improve=0.1384880, (0 missi
```

```
ng)
  Surrogate splits:
      sqft_above    < 7950      to the left,  agree=0.990, adj=0.4, (0
split)
      bathrooms     < 6.125     to the left,  agree=0.988, adj=0.3, (0
split)
      sqft_basement < 2925      to the left,  agree=0.986, adj=0.2, (0
split)

Node number 10: 6761 observations
  mean=462801.4, MSE=2.013469e+10

Node number 11: 3297 observations
  mean=654887.1, MSE=4.859925e+10

Node number 12: 810 observations
  mean=557955.7, MSE=4.060606e+10

Node number 13: 2857 observations,    complexity param=0.02077484
  mean=930286.8, MSE=1.305796e+11
  left son=26 (1583 obs) right son=27 (1274 obs)
  Primary splits:
      long        < -122.1865 to the right, improve=0.1478167, (0 missi
ng)
      yr_built    < 1973.5    to the right, improve=0.1464955, (0 missi
ng)
      sqft_living < 3067.5    to the left,  improve=0.1445104, (0 missi
ng)
      view        < 2.5       to the left,  improve=0.1211965, (0 missi
ng)
      waterfront  < 0.5       to the left,  improve=0.1110150, (0 missi
ng)
  Surrogate splits:
      zipcode       < 98089.5   to the left,  agree=0.831, adj=0.622,
(0 split)
      sqft_basement < 30        to the left,  agree=0.709, adj=0.347,
(0 split)
      yr_built      < 1975.5    to the right, agree=0.704, adj=0.337,
(0 split)
      sqft_living15 < 2405      to the right, agree=0.699, adj=0.324,
(0 split)
      sqft_lot15    < 6438.5    to the right, agree=0.674, adj=0.269,
(0 split)

Node number 14: 574 observations,    complexity param=0.0247447
  mean=1603157, MSE=5.875979e+11
  left son=28 (315 obs) right son=29 (259 obs)
  Primary splits:
      long       < -122.1875 to the right, improve=0.2137067, (0 missin
g)
      waterfront < 0.5       to the left,  improve=0.1632810, (0 missin
g)
      view       < 3.5       to the left,  improve=0.1201055, (0 missin
g)
      lat        < 47.55455  to the left,  improve=0.1121319, (0 missin
g)
      grade      < 10.5      to the left,  improve=0.1083140, (0 missin
```

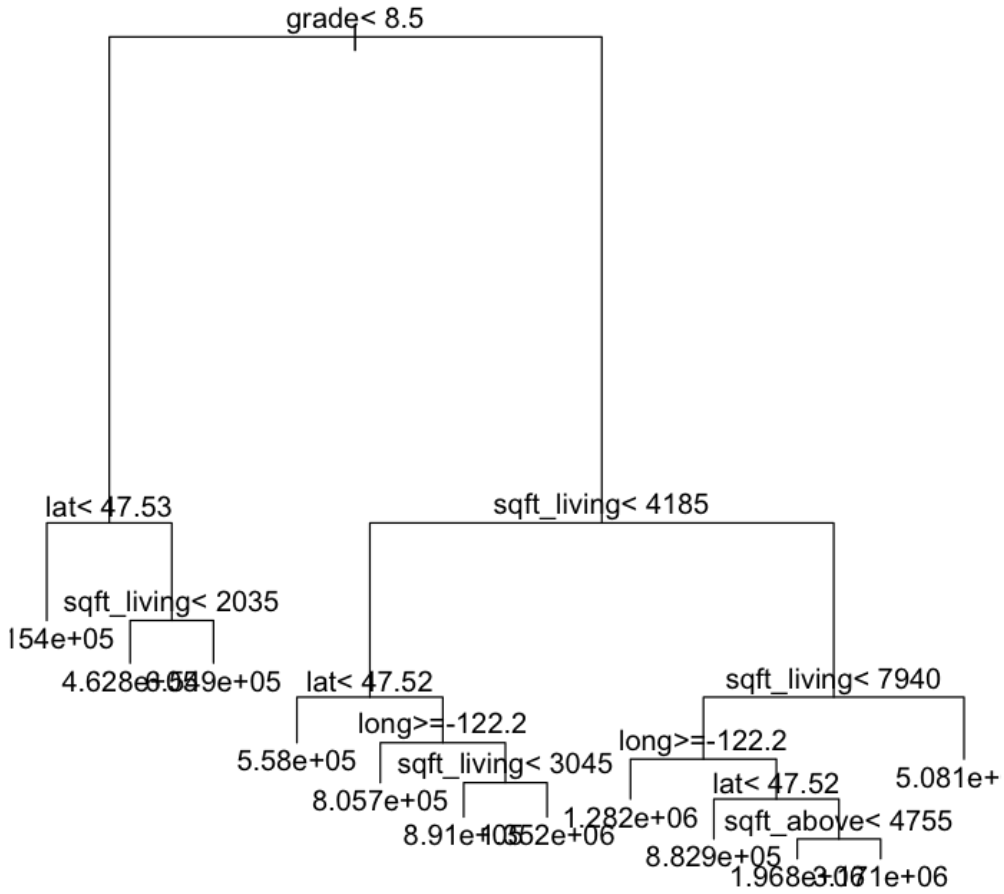
```
g)
  Surrogate splits:
      zipcode       < 98097     to the left,  agree=0.725, adj=0.390,
(0 split)
      yr_built      < 1980.5    to the right, agree=0.706, adj=0.347,
(0 split)
      sqft_lot      < 24426.5   to the right, agree=0.672, adj=0.274,
(0 split)
      sqft_lot15    < 22994.5   to the right, agree=0.671, adj=0.270,
(0 split)
      sqft_living15 < 3715      to the right, agree=0.652, adj=0.228,
(0 split)

Node number 15: 10 observations
  mean=5081430, MSE=3.034967e+12

Node number 26: 1583 observations
  mean=805650.8, MSE=4.091497e+10

Node number 27: 1274 observations,    complexity param=0.02077484
  mean=1085152, MSE=1.987065e+11
  left son=54 (737 obs) right son=55 (537 obs)
  Primary splits:
      sqft_living   < 3045      to the left,  improve=0.2602601, (0 mis
sing)
      sqft_living15 < 2975      to the left,  improve=0.2089785, (0 mis
sing)
      grade         < 9.5       to the left,  improve=0.1847518, (0 mis
sing)
      sqft_above    < 1815      to the left,  improve=0.1238427, (0 mis
sing)
      sqft_lot      < 3306.5    to the left,  improve=0.1010572, (0 mis
sing)
  Surrogate splits:
      sqft_above    < 3045      to the left,  agree=0.745, adj=0.395,
(0 split)
      bathrooms     < 2.875     to the left,  agree=0.730, adj=0.359,
(0 split)
      sqft_basement < 825       to the left,  agree=0.705, adj=0.300,
(0 split)
      sqft_living15 < 2985      to the left,  agree=0.692, adj=0.270,
(0 split)
      grade         < 9.5       to the left,  agree=0.688, adj=0.261,
(0 split)

Node number 28: 315 observations
  mean=1281833, MSE=2.380825e+11

Node number 29: 259 observations,    complexity param=0.01287892
  mean=1993957, MSE=7.34386e+11
  left son=58 (27 obs) right son=59 (232 obs)
  Primary splits:
      lat         < 47.52195  to the left,  improve=0.1956192, (0 missi
ng)
      grade       < 11.5      to the left,  improve=0.1930182, (0 missi
ng)
      sqft_above  < 4735      to the left,  improve=0.1772601, (0 missi
```

```
ng)
      sqft_living < 5005      to the left,  improve=0.1704191, (0 missi
ng)
      bathrooms   < 5.375     to the left,  improve=0.1049654, (0 missi
ng)
  Surrogate splits:
      zipcode < 98003.5   to the left,  agree=0.919, adj=0.222, (0 spli
t)

Node number 54: 737 observations
  mean=891035.6, MSE=9.798018e+10

Node number 55: 537 observations
  mean=1351566, MSE=2.142557e+11

Node number 58: 27 observations
  mean=882916.7, MSE=2.453867e+11

Node number 59: 232 observations,    complexity param=0.01287892
  mean=2123260, MSE=6.309163e+11
  left son=118 (202 obs) right son=119 (30 obs)
  Primary splits:
      sqft_above  < 4755      to the left,  improve=0.2583987, (0 missi
ng)
      sqft_living < 5005      to the left,  improve=0.2289437, (0 missi
ng)
      grade       < 11.5      to the left,  improve=0.2099801, (0 missi
ng)
      bathrooms   < 5.375     to the left,  improve=0.1630861, (0 missi
ng)
      waterfront  < 0.5       to the left,  improve=0.1411986, (0 missi
ng)
  Surrogate splits:
      sqft_living < 6405      to the left,  agree=0.914, adj=0.333, (0
split)
      bathrooms   < 5.125     to the left,  agree=0.897, adj=0.200, (0
split)
      grade       < 12.5      to the left,  agree=0.888, adj=0.133, (0
split)

Node number 118: 202 observations
  mean=1967657, MSE=4.447485e+11

Node number 119: 30 observations
  mean=3170982, MSE=6.236966e+11
```

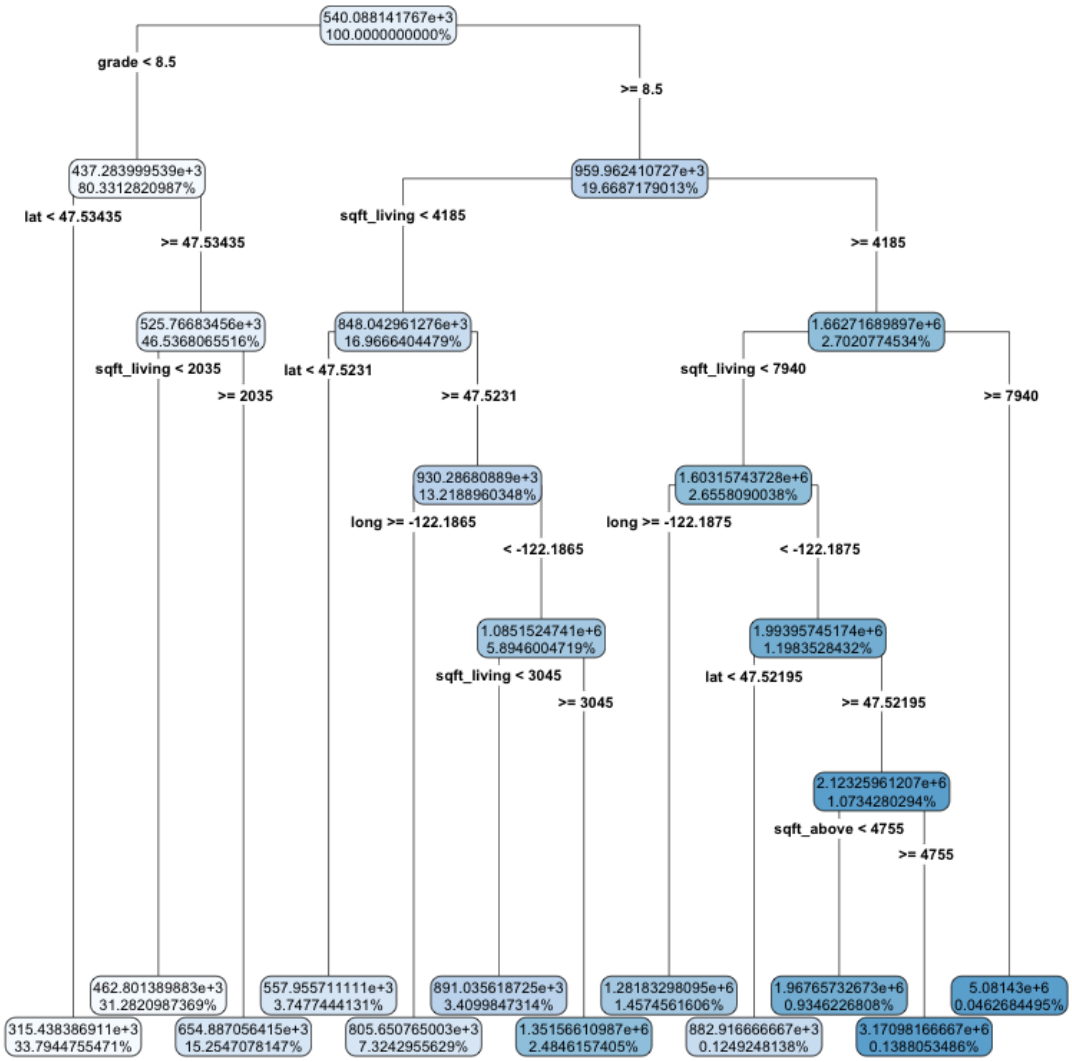
In [27]:
```
house.dTree
```

```
n= 21613

node), split, n, deviance, yval
      * denotes terminal node

  1) root 21613 2.912917e+15  540088.1
    2) grade< 8.5 17362 6.656597e+14  437284.0
      4) lat< 47.53435 7304 1.003407e+14  315438.4 *
      5) lat>=47.53435 10058 3.781350e+14  525766.8
       10) sqft_living< 2035 6761 1.361306e+14  462801.4 *
       11) sqft_living>=2035 3297 1.602317e+14  654887.1 *
    3) grade>=8.5 4251 1.314336e+15  959962.4
      6) sqft_living< 4185 3667 4.934437e+14  848043.0
       12) lat< 47.5231 810 3.289091e+13  557955.7 *
       13) lat>=47.5231 2857 3.730659e+14  930286.8
         26) long>=-122.1865 1583 6.476840e+13  805650.8 *
         27) long< -122.1865 1274 2.531521e+14 1085152.0
           54) sqft_living< 3045 737 7.221140e+13  891035.6 *
           55) sqft_living>=3045 537 1.150553e+14 1351566.0 *
      7) sqft_living>=4185 584 4.865430e+14 1662717.0
       14) sqft_living< 7940 574 3.372812e+14 1603157.0
         28) long>=-122.1875 315 7.499599e+13 1281833.0 *
         29) long< -122.1875 259 1.902060e+14 1993957.0
           58) lat< 47.52195 27 6.625440e+12  882916.7 *
           59) lat>=47.52195 232 1.463726e+14 2123260.0
            118) sqft_above< 4755 202 8.983921e+13 1967657.0 *
            119) sqft_above>=4755 30 1.871090e+13 3170982.0 *
       15) sqft_living>=7940 10 3.034967e+13 5081430.0 *
```

In [28]:
```
plot(house.dTree, margin = 0.00001)
text(house.dTree)
```



In [29]:
```
library(rpart.plot)
rpart.plot(house.dTree,digits=12,fallen.leaves=TRUE,type=4)
```



In [30]:
```
plotcp(house.dTree)
```