STAT6021_Project_2_Ade_Faparusi

Ade Faparusi

2023-11-06

```
library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## v readr     2.1.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(gapminder)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(datasets)
library(leaps)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggplot2)
library("scales")
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

Read in data

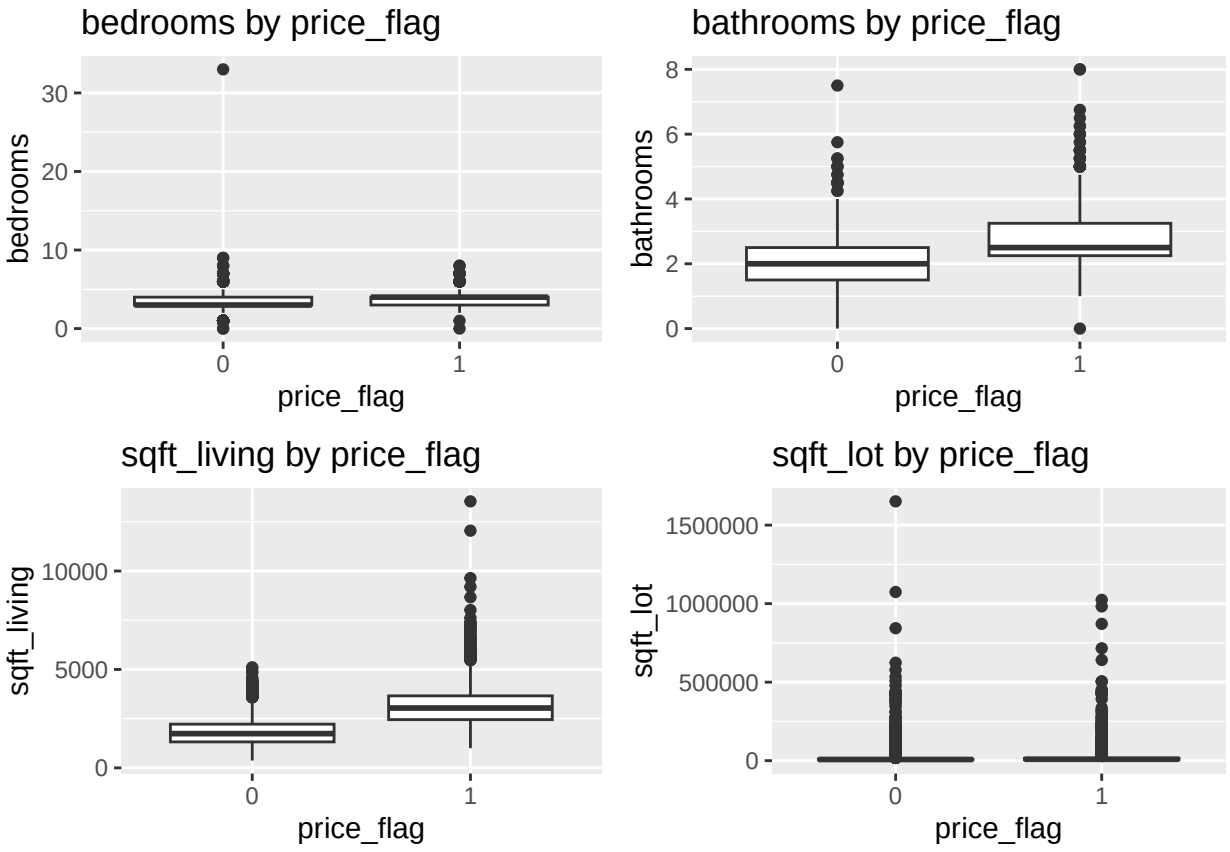
```
#Read in data and split into train vs test data
Data<-read.csv("kc_house_data.csv", sep=",", header=TRUE)

Data$price_flag <- factor(ifelse(Data$price>700000,1,0) ) #flag for price > $700K

Data$basement_flag <- factor(ifelse(Data$sqft_basement>0,1,0) ) #flag for if there is a basement

#Extract year and month from date
Data$year <- substr(Data$date,1,4)
Data$month <- substr(Data$date,5,6)

#change data type of categorical variables
Data$waterfront <-factor(Data$waterfront)
Data$condition <-factor(Data$condition)
#Data$grade <-factor(Data$grade) #???
Data$view <-factor(Data$view)
#Data$zipcode <-factor(Data$zipcode)

#create log of price variable
Data$logprice <-log(Data$price)

#Subset into test and train
set.seed(6021)
sample.data<-sample.int(nrow(Data), floor(.50*nrow(Data)), replace = F)

train<-Data[sample.data, ]
test<-Data[-sample.data, ]

head(train)
```

```
##              id            date  price bedrooms bathrooms sqft_living sqft_lot
## 8417  1612500090 20150331T000000 225800        4      1.00        1100     7110
## 7001  3401700255 20140729T000000 595000        4      2.00        3090    87120
```

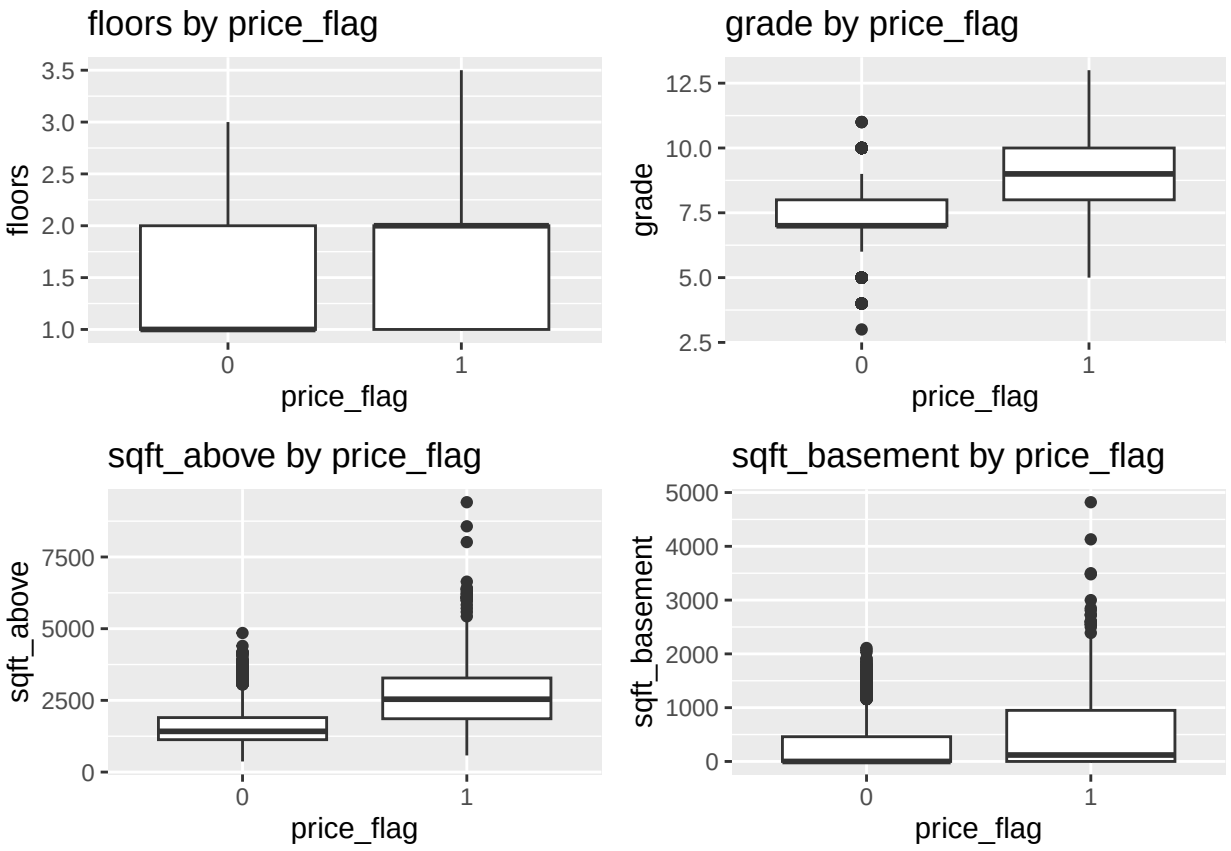
```
## 12812 5425700150 20140804T000000 787500        4      1.75        1580     9382
## 11457 7399000350 20141104T000000 300000        3      2.00        1550     8300
## 5299  7523700305 20141012T000000 243400        4      1.50        1730     7464
## 15174 2927600415 20140821T000000 805000        3      2.25        2860    11250
##       floors waterfront view condition grade sqft_above sqft_basement yr_built
## 8417       1          0    0         4     7        880           220     1907
## 7001       1          0    0         4     7       1590          1500     1974
## 12812      1          0    0         3     7       1080           500     1963
## 11457      1          0    0         4     8       1550             0     1965
## 5299       2          0    0         4     7       1730             0     1959
## 15174      1          0    1         5     8       2290           570     1956
##       yr_renovated zipcode     lat     long sqft_living15 sqft_lot15 price_flag
## 8417             0   98030 47.3858 -122.227          1150       7110          0
## 7001             0   98072 47.7275 -122.122          2560      88426          0
## 12812            0   98039 47.6353 -122.232          2010       9382          1
## 11457            0   98055 47.4654 -122.195          1860       8000          0
## 5299             0   98032 47.3782 -122.304          1370       7860          0
## 15174            0   98166 47.4534 -122.372          2030      11250          1
##       basement_flag year month logprice
## 8417              1 2015    03 12.32740
## 7001              1 2014    07 13.29632
## 12812             1 2014    08 13.57662
## 11457             0 2014    11 12.61154
## 5299              0 2014    10 12.40246
## 15174             1 2014    08 13.59860
```

```r
#The boxplots of the quantitative variables across price flag:

bp1<-ggplot(train, aes(x=price_flag, y=bedrooms))+
geom_boxplot()+
labs(x="price_flag", y="bedrooms", title="bedrooms by price_flag")

bp2<-ggplot(train, aes(x=price_flag, y=bathrooms))+
geom_boxplot()+
labs(x="price_flag", y="bathrooms", title="bathrooms by price_flag")

bp3<-ggplot(train, aes(x=price_flag, y=sqft_living))+
geom_boxplot()+
labs(x="price_flag", y="sqft_living", title="sqft_living by price_flag")

bp4<-ggplot(train, aes(x=price_flag, y=sqft_lot))+
geom_boxplot()+
labs(x="price_flag", y="sqft_lot", title="sqft_lot by price_flag")

bp5<-ggplot(train, aes(x=price_flag, y=floors))+
geom_boxplot()+
labs(x="price_flag", y="floors", title="floors by price_flag")

bp6<-ggplot(train, aes(x=price_flag, y=grade))+
geom_boxplot()+
labs(x="price_flag", y="grade", title="grade by price_flag")

bp7<-ggplot(train, aes(x=price_flag, y=sqft_above))+
geom_boxplot()+
```

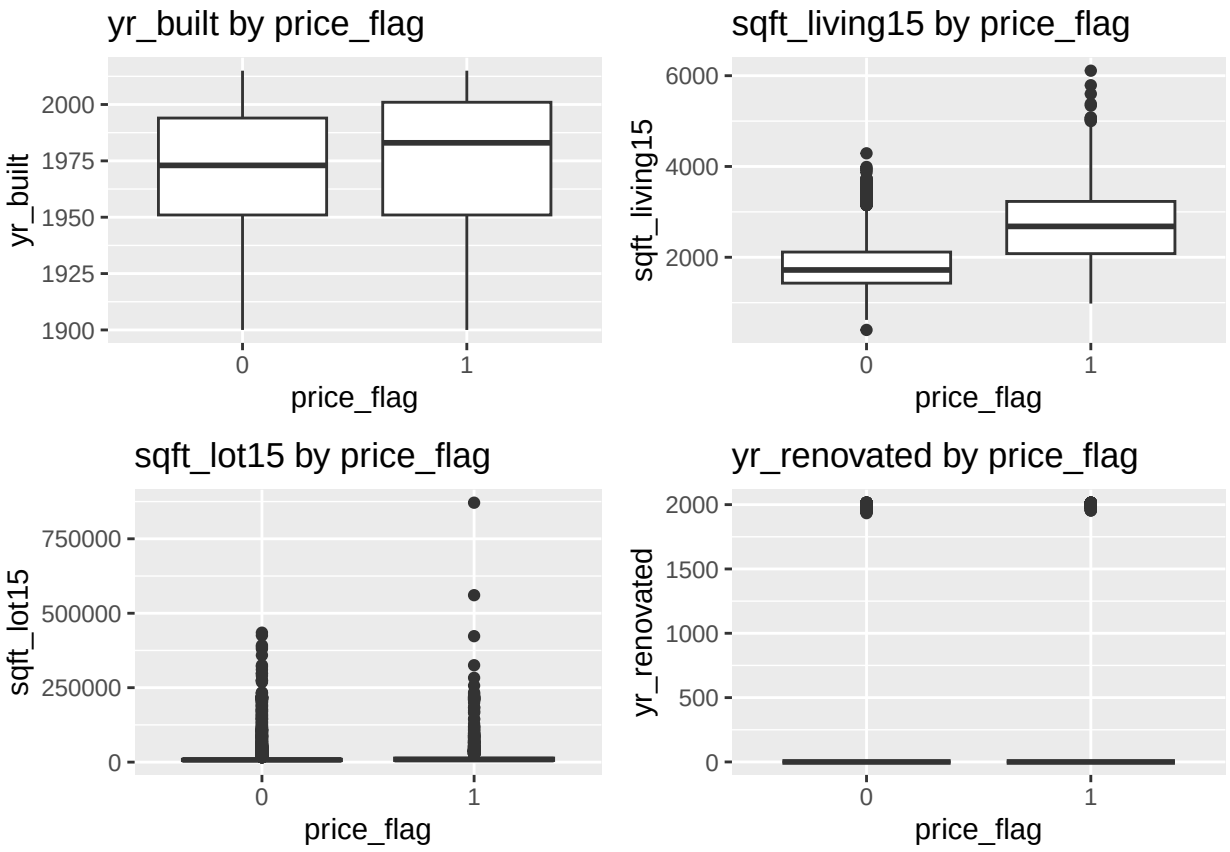
```
labs(x="price_flag", y="sqft_above", title="sqft_above by price_flag")

bp8<-ggplot(train, aes(x=price_flag, y=sqft_basement))+
geom_boxplot()+
labs(x="price_flag", y="sqft_basement", title="sqft_basement by price_flag")

bp9<-ggplot(train, aes(x=price_flag, y=yr_built))+
geom_boxplot()+
labs(x="price_flag", y="yr_built", title="yr_built by price_flag")

bp10<-ggplot(train, aes(x=price_flag, y=sqft_living15))+
geom_boxplot()+
labs(x="price_flag", y="sqft_living15", title="sqft_living15 by price_flag")

bp11<-ggplot(train, aes(x=price_flag, y=sqft_lot15))+
geom_boxplot()+
labs(x="price_flag", y="sqft_lot15", title="sqft_lot15 by price_flag")

bp12<-ggplot(train, aes(x=price_flag, y=yr_renovated))+
geom_boxplot()+
labs(x="price_flag", y="yr_renovated", title="yr_renovated by price_flag")


##produce the boxplots in a 2 by 2 matrix
grid.arrange(bp1, bp2, bp3, bp4, ncol = 2, nrow = 2)
```



```
grid.arrange(bp5, bp6, bp7, bp8, ncol = 2, nrow = 2)
```

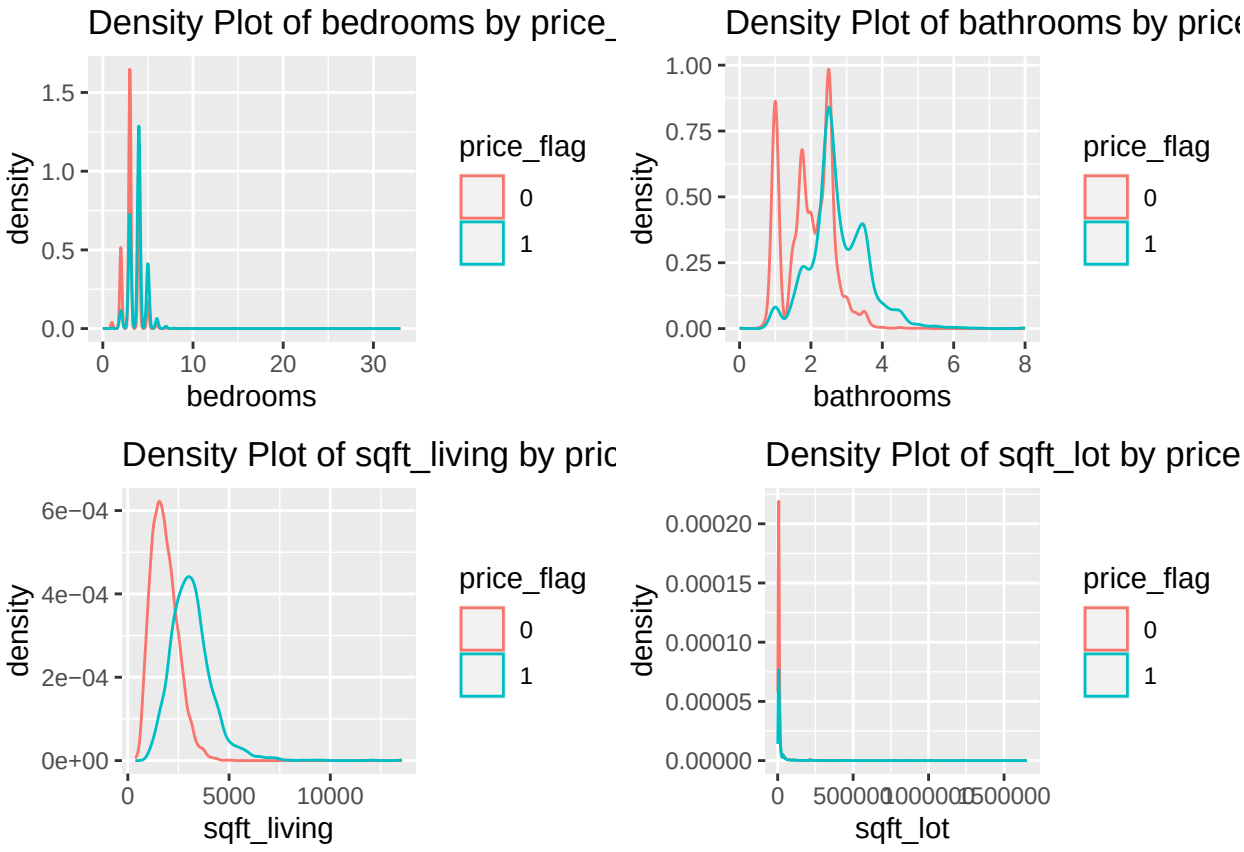



```
grid.arrange(bp9, bp10, bp11, bp12, ncol = 2, nrow = 2)
```

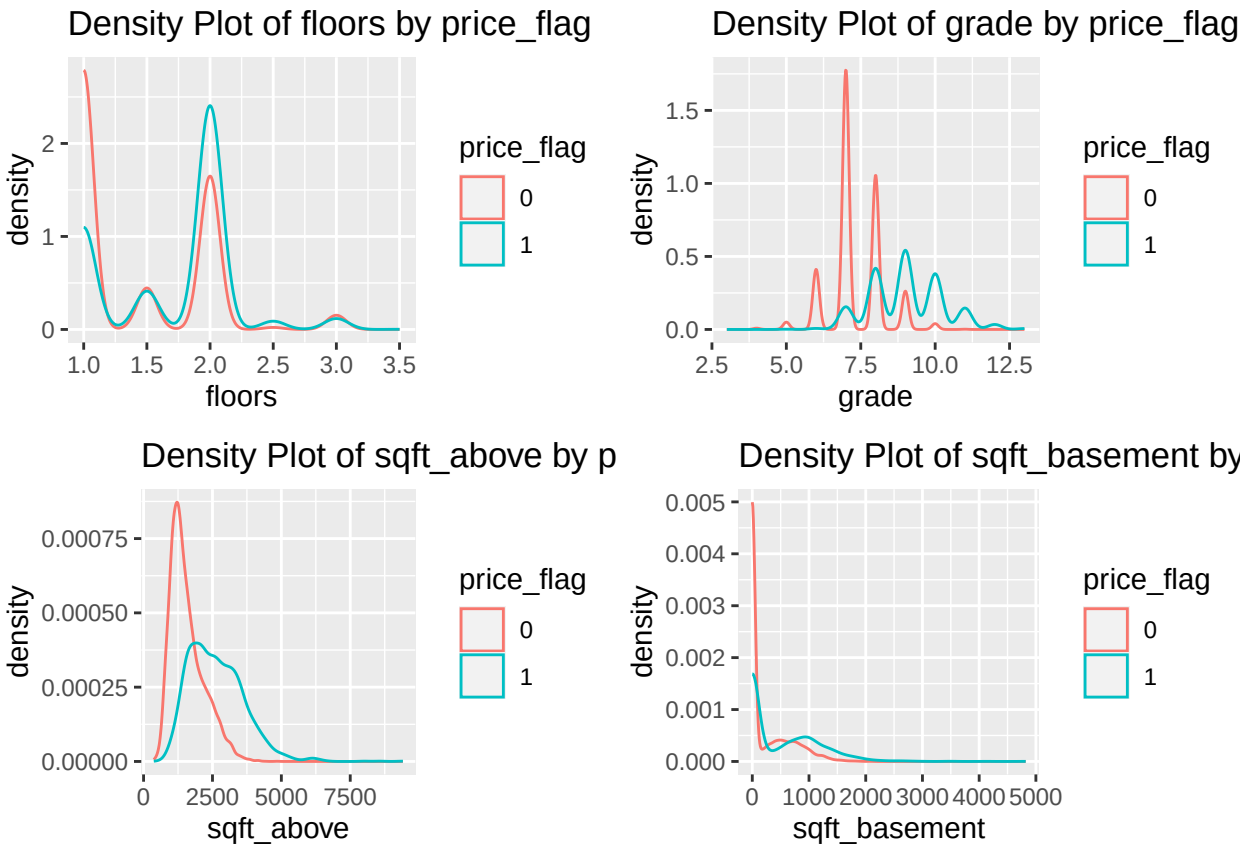



```
#Density plots of quantitative variables
dp1<-ggplot(train,aes(x=bedrooms, color=price_flag))+
geom_density()+
labs(title="Density Plot of bedrooms by price_flag")

dp2<-ggplot(train,aes(x=bathrooms, color=price_flag))+
geom_density()+
labs(title="Density Plot of bathrooms by price_flag")

dp3<-ggplot(train,aes(x=sqft_living, color=price_flag))+
geom_density()+
labs(title="Density Plot of sqft_living by price_flag")

dp4<-ggplot(train,aes(x=sqft_lot, color=price_flag))+
geom_density()+
labs(title="Density Plot of sqft_lot by price_flag")

dp5<-ggplot(train,aes(x=floors, color=price_flag))+
geom_density()+
labs(title="Density Plot of floors by price_flag")

dp6<-ggplot(train,aes(x=grade, color=price_flag))+
geom_density()+
labs(title="Density Plot of grade by price_flag")

dp7<-ggplot(train,aes(x=sqft_above, color=price_flag))+
```

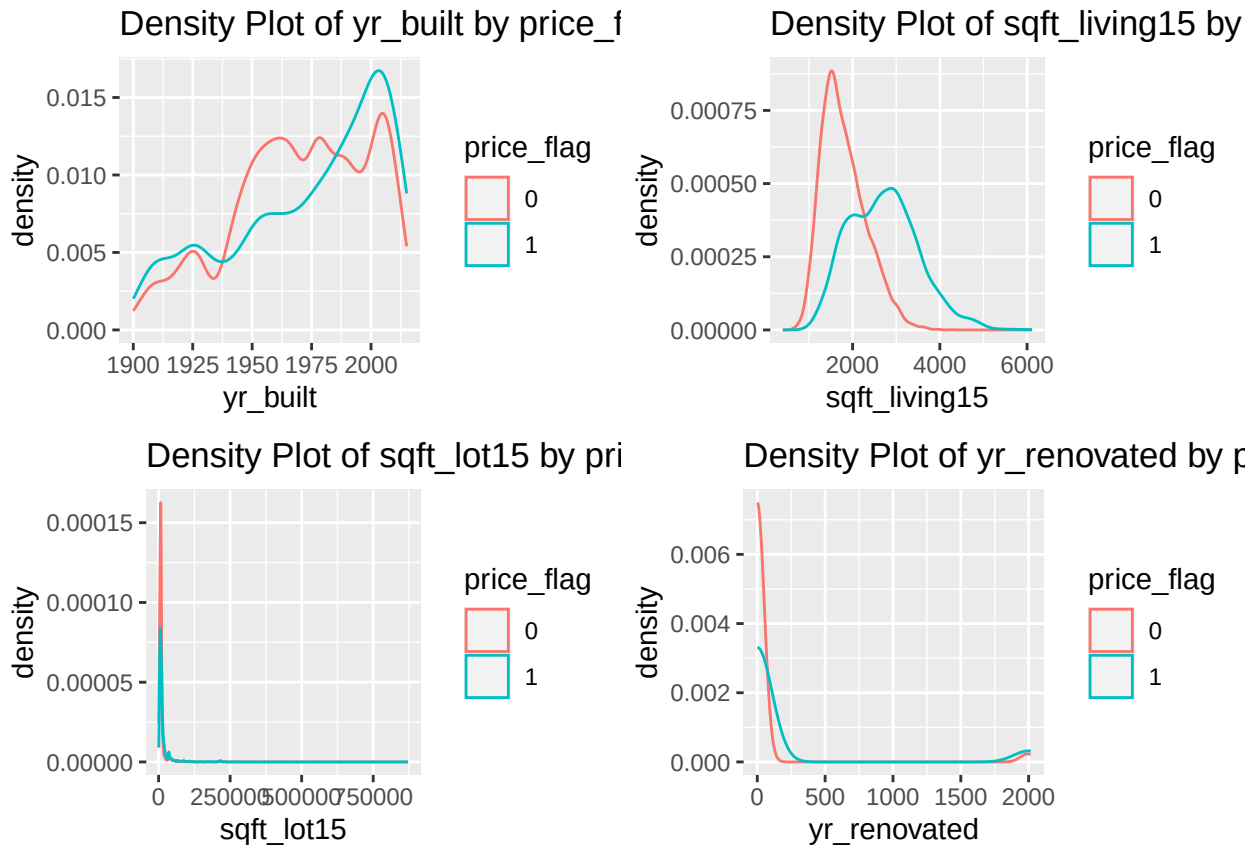
```
geom_density()+
labs(title="Density Plot of sqft_above by price_flag")

dp8<-ggplot(train,aes(x=sqft_basement, color=price_flag))+
geom_density()+
labs(title="Density Plot of sqft_basement by price_flag")

dp9<-ggplot(train,aes(x=yr_built, color=price_flag))+
geom_density()+
labs(title="Density Plot of yr_built by price_flag")

dp10<-ggplot(train,aes(x=sqft_living15, color=price_flag))+
geom_density()+
labs(title="Density Plot of sqft_living15 by price_flag")

dp11<-ggplot(train,aes(x=sqft_lot15, color=price_flag))+
geom_density()+
labs(title="Density Plot of sqft_lot15 by price_flag")

dp12<-ggplot(train,aes(x=yr_renovated, color=price_flag))+
geom_density()+
labs(title="Density Plot of yr_renovated by price_flag")

##produce the density plots in a 2 by 2 matrix
grid.arrange(dp1, dp2, dp3, dp4, ncol = 2, nrow = 2)
```



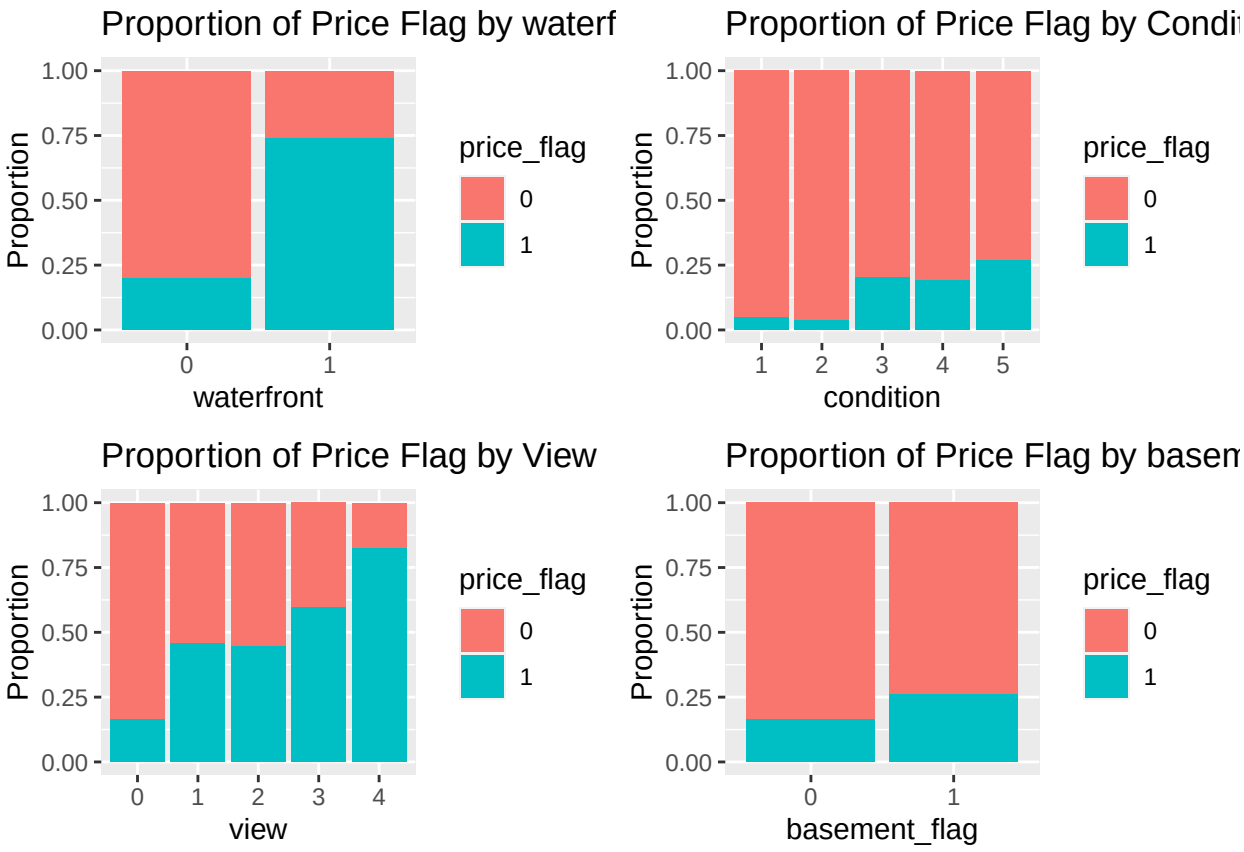
```
grid.arrange(dp5, dp6, dp7, dp8, ncol = 2, nrow = 2)
```



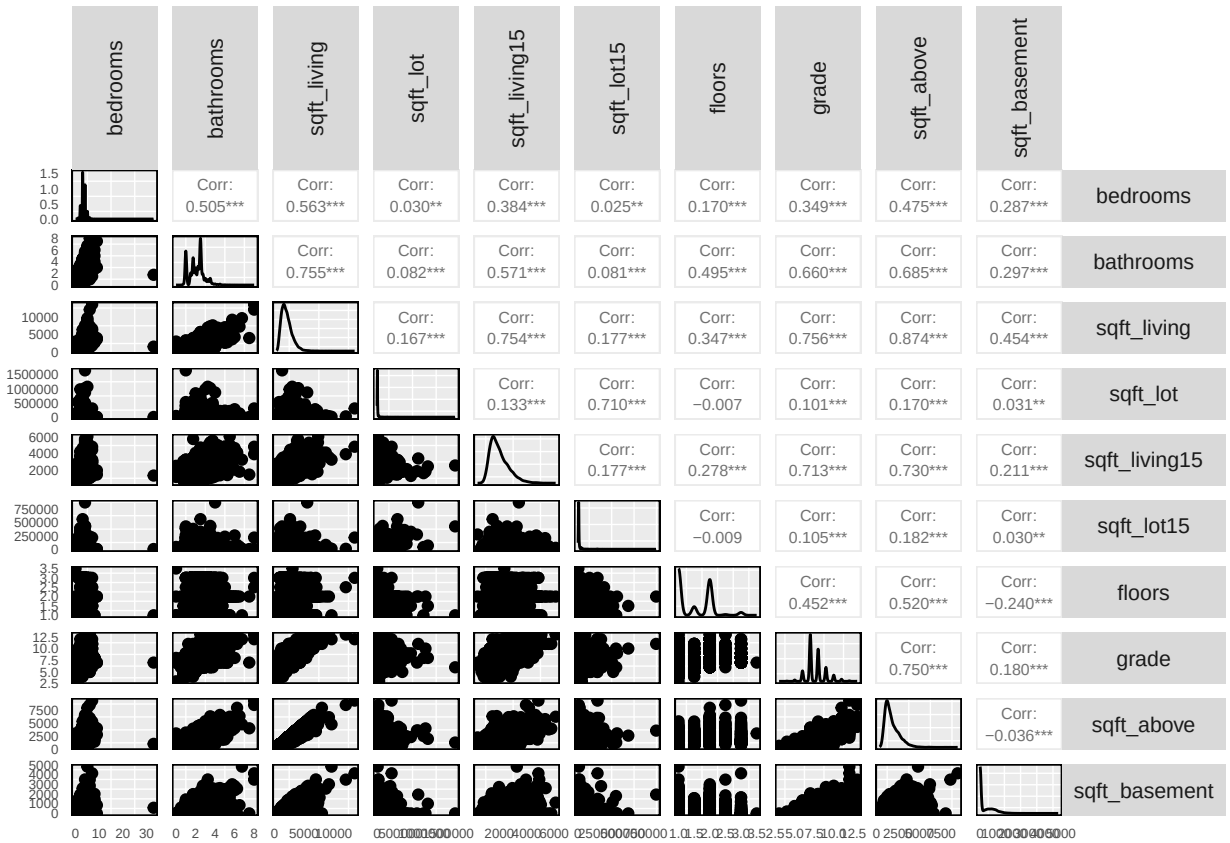

```
grid.arrange(dp9, dp10, dp11, dp12, ncol = 2, nrow = 2)
```




```
#Bar charts of categorical variables

dp13<- ggplot(train, aes(x=waterfront, fill=price_flag))+
geom_bar(position = "fill")+
labs(x="waterfront", y="Proportion",
title="Proportion of Price Flag by waterfront")

dp14<- ggplot(train, aes(x=condition, fill=price_flag))+
geom_bar(position = "fill")+
labs(x="condition", y="Proportion",
title="Proportion of Price Flag by Condition")

dp15<- ggplot(train, aes(x=view, fill=price_flag))+
geom_bar(position = "fill")+
labs(x="view", y="Proportion",
title="Proportion of Price Flag by View")

dp16<- ggplot(train, aes(x=basement_flag, fill=price_flag))+
geom_bar(position = "fill")+
labs(x="basement_flag", y="Proportion",
title="Proportion of Price Flag by basement_flag")

grid.arrange(dp13, dp14, dp15,dp16, ncol = 2, nrow = 2)
```




```
#ggpairs plot of numerical variables
my_Data_numer <- train[,c("bedrooms", "bathrooms","sqft_living", "sqft_lot", "sqft_living15", "sqft_lot

GGally::ggpairs(my_Data_numer, progress = FALSE,upper=list(continuous = wrap("cor", size=2))) +
  theme(axis.text = element_text(size = 5),
        strip.text.y = element_text(size = 8,angle = 0),
        strip.text.x = element_text(size = 8,angle = 90),
        legend.position = "none",
        panel.grid.major = element_blank(),
        axis.ticks = element_blank(),
        axis.title.y = element_text(angle = 90, vjust = 1, color = "black"),
        panel.border = element_rect(fill = NA))
```




```
# Create model
train_sub <- subset(train, select = -c(date, price,id,logprice, sqft_basement))
result<-glm(price_flag ~ ., family="binomial", data=train_sub)
result
```

```
##
## Call:  glm(formula = price_flag ~ ., family = "binomial", data = train_sub)
##
## Coefficients:
##    (Intercept)        bedrooms       bathrooms     sqft_living        sqft_lot
##     -1.228e+02      -1.066e-01       2.515e-01       9.083e-04       4.654e-06
##         floors     waterfront1           view1           view2           view3
##      2.356e-01       2.163e-01       5.317e-01       6.232e-01       8.365e-01
##           view4      condition2      condition3      condition4      condition5
##      2.292e+00      -4.707e-02       3.184e-02       5.996e-01       1.072e+00
##           grade      sqft_above       yr_built    yr_renovated         zipcode
##      1.312e+00       4.755e-04      -3.076e-02       2.307e-04      -6.715e-03
##             lat            long   sqft_living15      sqft_lot15  basement_flag1
##      7.324e+00      -3.885e+00       7.956e-04      -6.630e-06       1.400e-01
##       year2015         month02         month03         month04         month05
##      6.841e-01      -1.898e-01       3.198e-01       4.251e-01       4.675e-01
##         month06         month07         month08         month09         month10
##      5.433e-01       5.213e-01       8.147e-01       5.190e-01       5.324e-01
##         month11         month12
##      6.590e-01       3.793e-01
##
## Degrees of Freedom: 10805 Total (i.e. Null);  10769 Residual
```

```
## Null Deviance:       10870
## Residual Deviance: 4741  AIC: 4815
```

```r
#Wald test
summary(result)
```

```
##
## Call:
## glm(formula = price_flag ~ ., family = "binomial", data = train_sub)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.228e+02  8.915e+01  -1.377  0.16847
## bedrooms       -1.066e-01  5.280e-02  -2.019  0.04348 *
## bathrooms       2.515e-01  8.714e-02   2.887  0.00390 **
## sqft_living     9.083e-04  1.632e-04   5.567 2.59e-08 ***
## sqft_lot        4.654e-06  1.063e-06   4.377 1.20e-05 ***
## floors          2.356e-01  1.023e-01   2.303  0.02130 *
## waterfront1     2.163e-01  5.810e-01   0.372  0.70971
## view1           5.317e-01  2.302e-01   2.310  0.02089 *
## view2           6.232e-01  1.542e-01   4.042 5.29e-05 ***
## view3           8.365e-01  2.101e-01   3.982 6.83e-05 ***
## view4           2.292e+00  4.230e-01   5.418 6.02e-08 ***
## condition2     -4.707e-02  1.434e+00  -0.033  0.97381
## condition3      3.184e-02  1.134e+00   0.028  0.97761
## condition4      5.996e-01  1.133e+00   0.529  0.59673
## condition5      1.072e+00  1.137e+00   0.942  0.34612
## grade           1.312e+00  6.099e-02  21.515  < 2e-16 ***
## sqft_above      4.755e-04  1.800e-04   2.642  0.00824 **
## yr_built       -3.076e-02  2.005e-03 -15.340  < 2e-16 ***
## yr_renovated    2.307e-04  8.860e-05   2.604  0.00921 **
## zipcode        -6.715e-03  9.880e-04  -6.796 1.08e-11 ***
## lat             7.324e+00  3.664e-01  19.988  < 2e-16 ***
## long           -3.885e+00  4.307e-01  -9.019  < 2e-16 ***
## sqft_living15   7.956e-04  9.092e-05   8.750  < 2e-16 ***
## sqft_lot15     -6.630e-06  2.122e-06  -3.125  0.00178 **
## basement_flag1  1.400e-01  1.542e-01   0.908  0.36371
## year2015        6.841e-01  2.551e-01   2.682  0.00733 **
## month02        -1.898e-01  2.367e-01  -0.802  0.42269
## month03         3.198e-01  2.102e-01   1.521  0.12821
## month04         4.251e-01  2.031e-01   2.093  0.03635 *
## month05         4.675e-01  2.743e-01   1.704  0.08835 .
## month06         5.433e-01  3.292e-01   1.650  0.09889 .
## month07         5.213e-01  3.288e-01   1.586  0.11283
## month08         8.147e-01  3.299e-01   2.470  0.01353 *
## month09         5.190e-01  3.332e-01   1.558  0.11934
## month10         5.324e-01  3.319e-01   1.604  0.10868
## month11         6.590e-01  3.425e-01   1.924  0.05438 .
## month12         3.793e-01  3.410e-01   1.112  0.26609
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10874.9  on 10805  degrees of freedom
```

```
## Residual deviance:  4740.6  on 10769  degrees of freedom
## AIC: 4814.6
##
## Number of Fisher Scoring iterations: 7
```

```
##predicted class for test data based on training data
test_sub <- subset(test, select = -c(date, price,id,logprice, sqft_basement))

preds<-predict(result,newdata=test_sub, type="response")
##add predicted probabilities and classification based on threshold
test.new<-data.frame(test_sub,preds,preds>0.5)
```

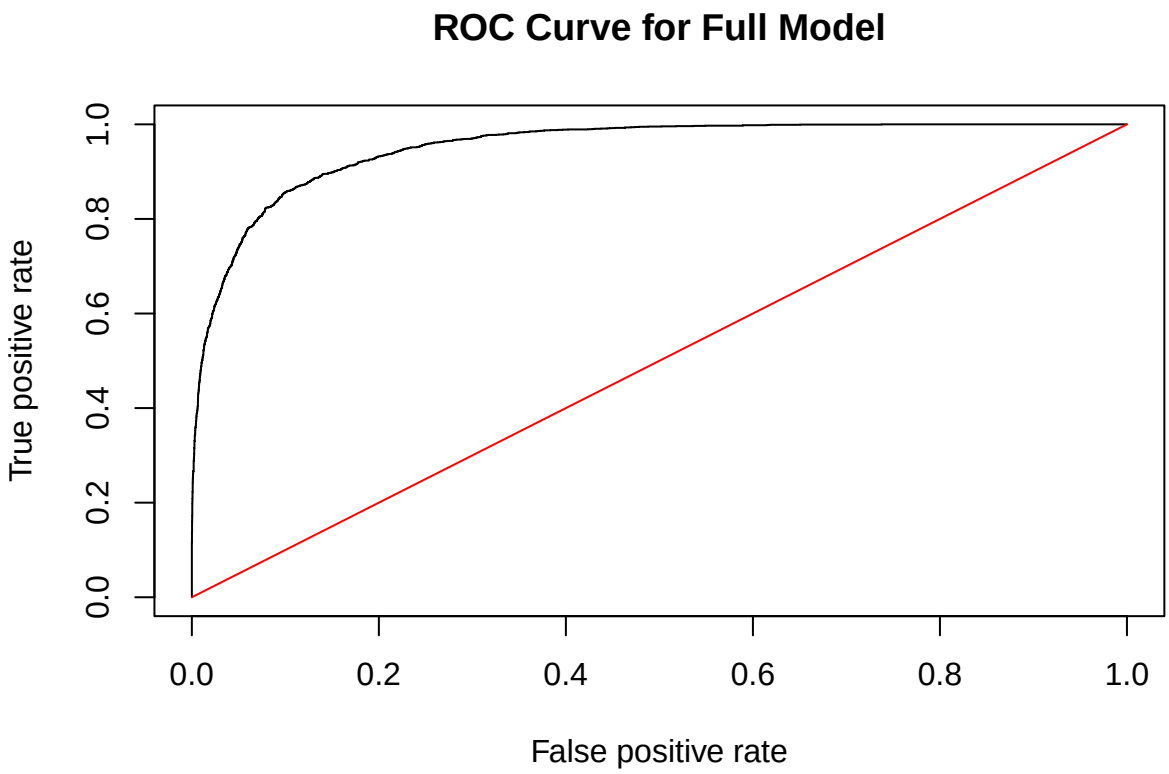
```
##confusion matrix with 0.5 threshold
table(test_sub$price_flag, preds>0.5)
```

```
##
##     FALSE TRUE
##   0  8333  334
##   1   666 1474
```

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.3.2
```

```
##produce the numbers associated with classification table
rates<-ROCR::prediction(preds, test_sub$price_flag)

##store the true positive and false positive rates
roc_result<-ROCR::performance(rates,measure="tpr", x.measure="fpr")
##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Full Model")
lines(x = c(0,1), y = c(0,1), col="red")
```




```
##compute the AUC
auc<-performance(rates, measure = "auc")
auc@y.values
```

```
## [[1]]
## [1] 0.951602
```

```
#Likelihood ratio tests

deltaG2<-result$null.deviance-result$deviance
deltaG2
```

```
## [1] 6134.217
```

```
#1-pchisq(deltaG2,x)  ????
```

‘‘‘