BA810_Team1_Airbnb_price_prediction

Chunxiaqiu Yang, Kexi Pi, Linh To, Risheng Guo, Qianrong Wen, Ta-Wei Wang

10/12/2021

R Markdown

```
library(data.table)
library(ggplot2)
library(ggthemes)
library(ipred)
library(xgboost)
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

```
library(scales)
library(rpart)
library(rpart.plot)
library(randomForest)
```

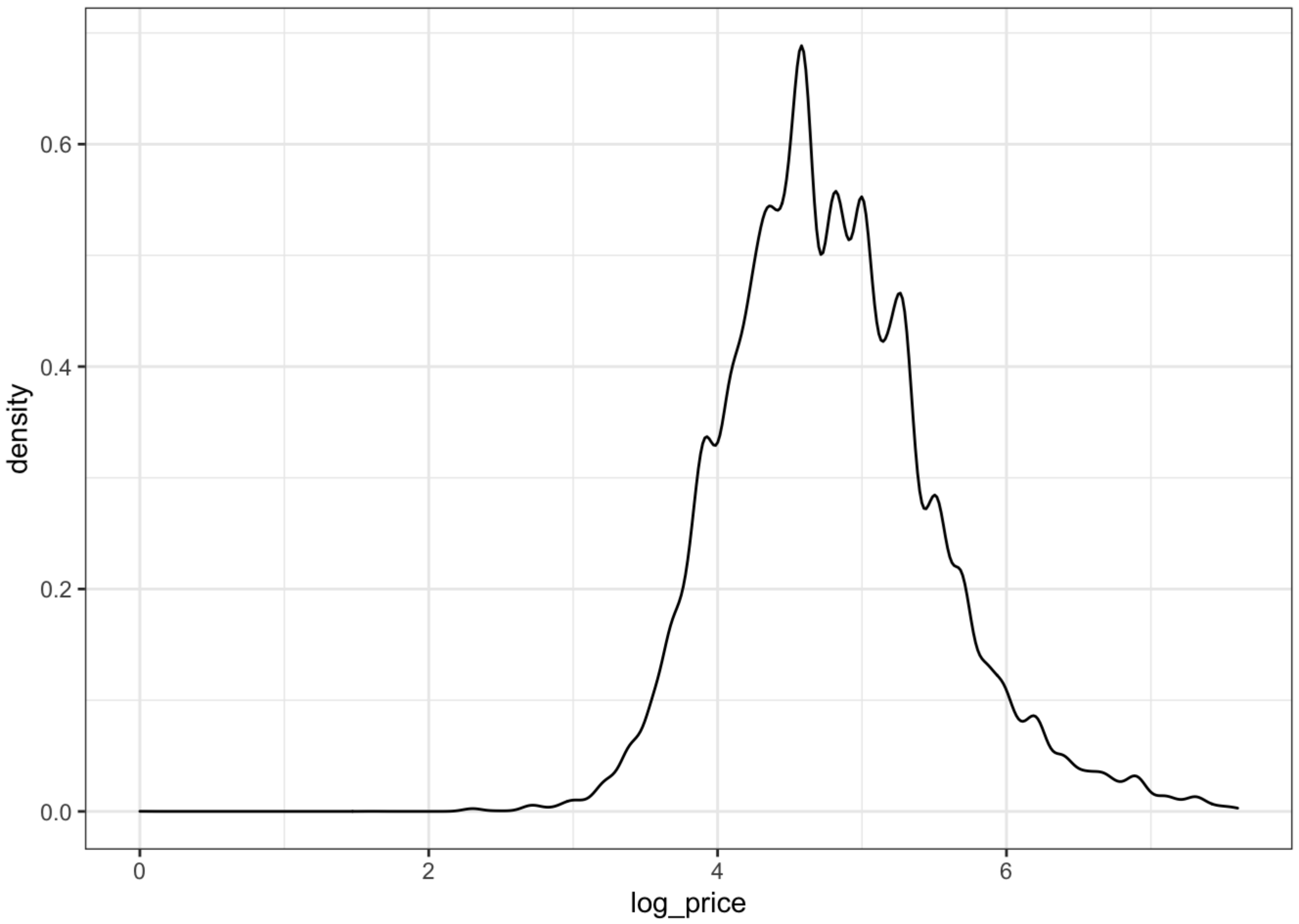
```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

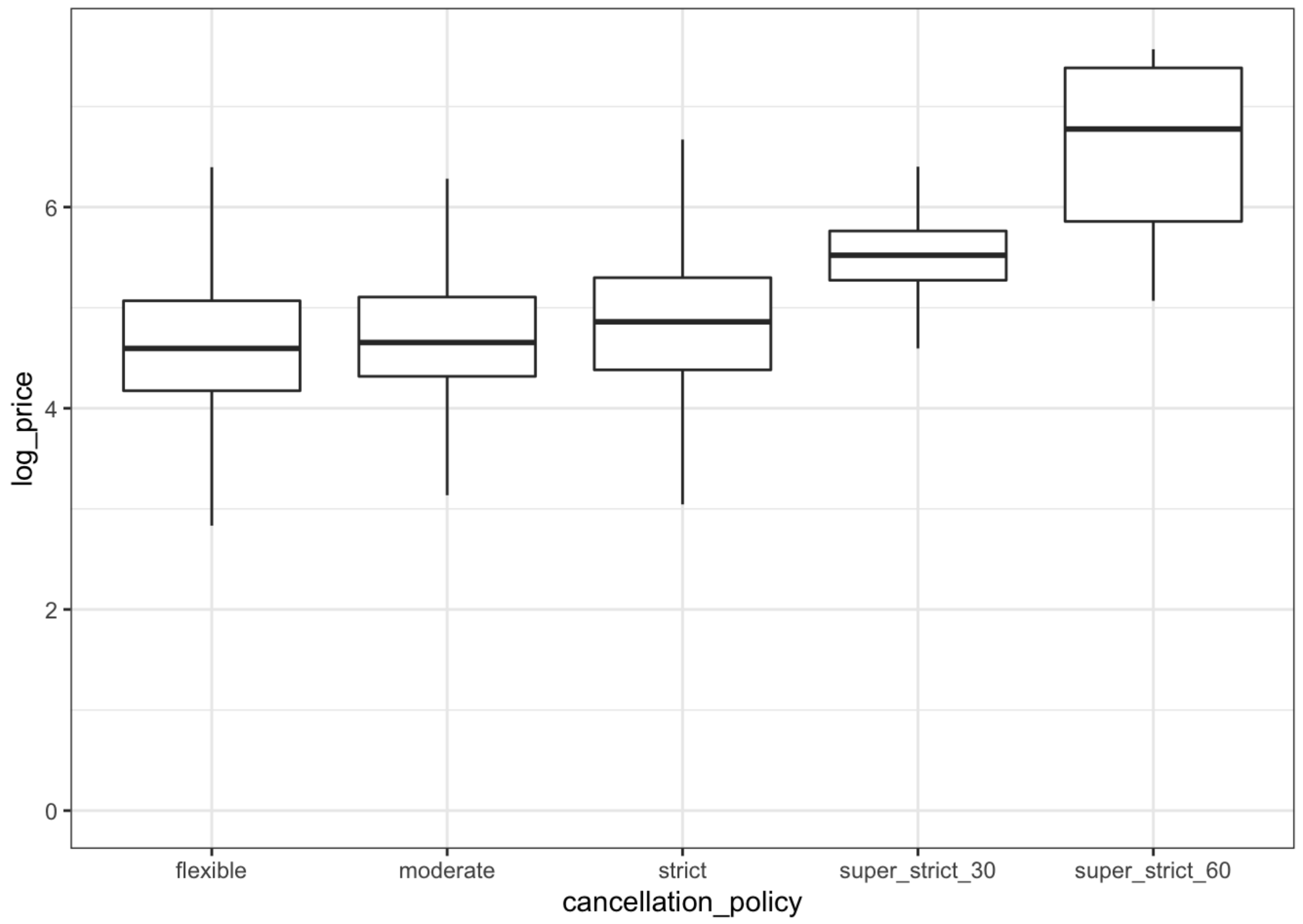
```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

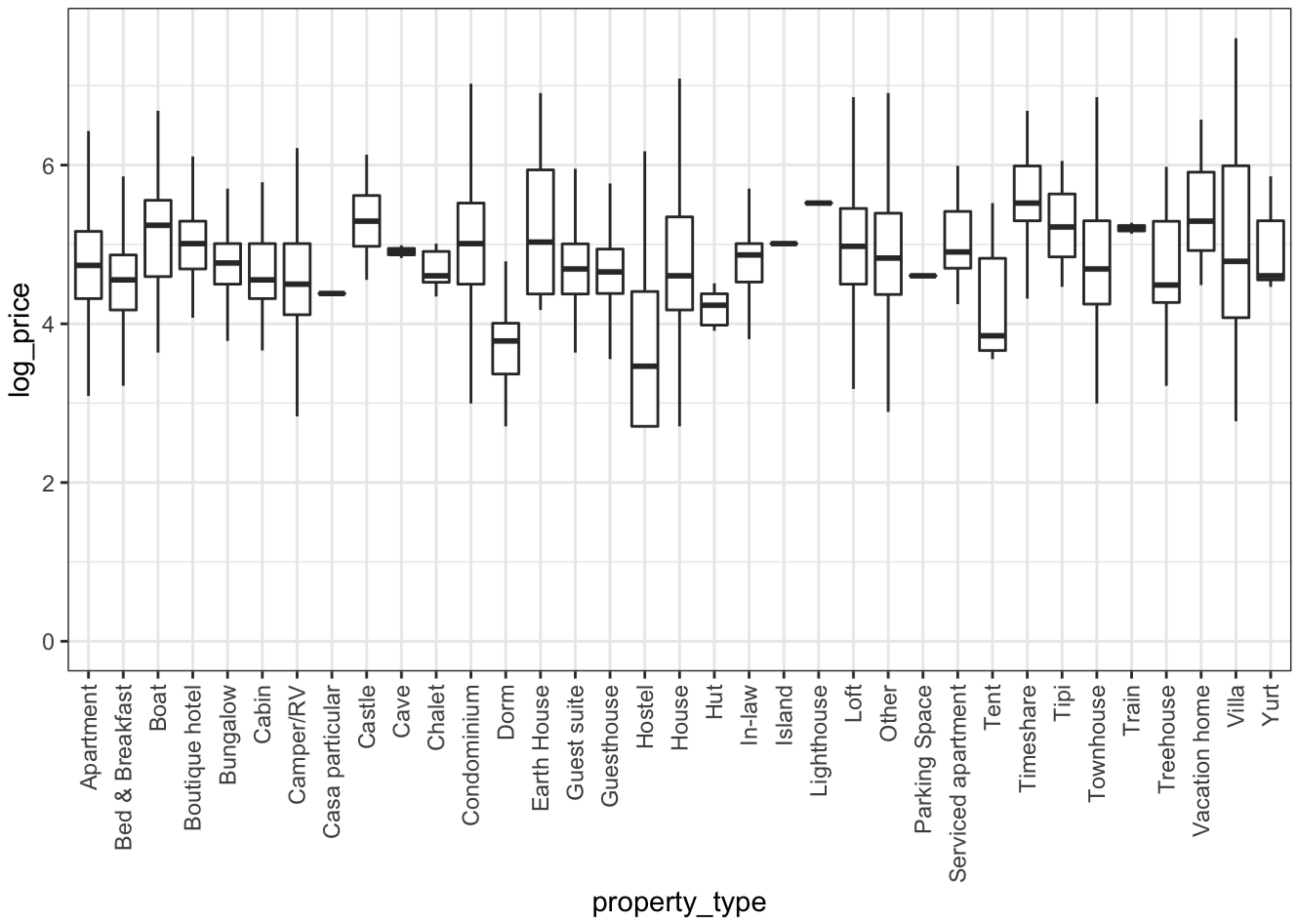
```
theme_set(theme_bw())



###Descriptive Analyse & EDA
train <- fread("/Users/tommy/Downloads/Fall\ 2021/train.csv")
## We will look at the descriptive analyses from three dimenstions: 1. a single dimentio
n; 2. two-variable dimension; 3. three-variable dimension

# the density distribution of our predicted variable: log price
density<-ggplot(train,aes(x=log_price))+geom_density();
density
```



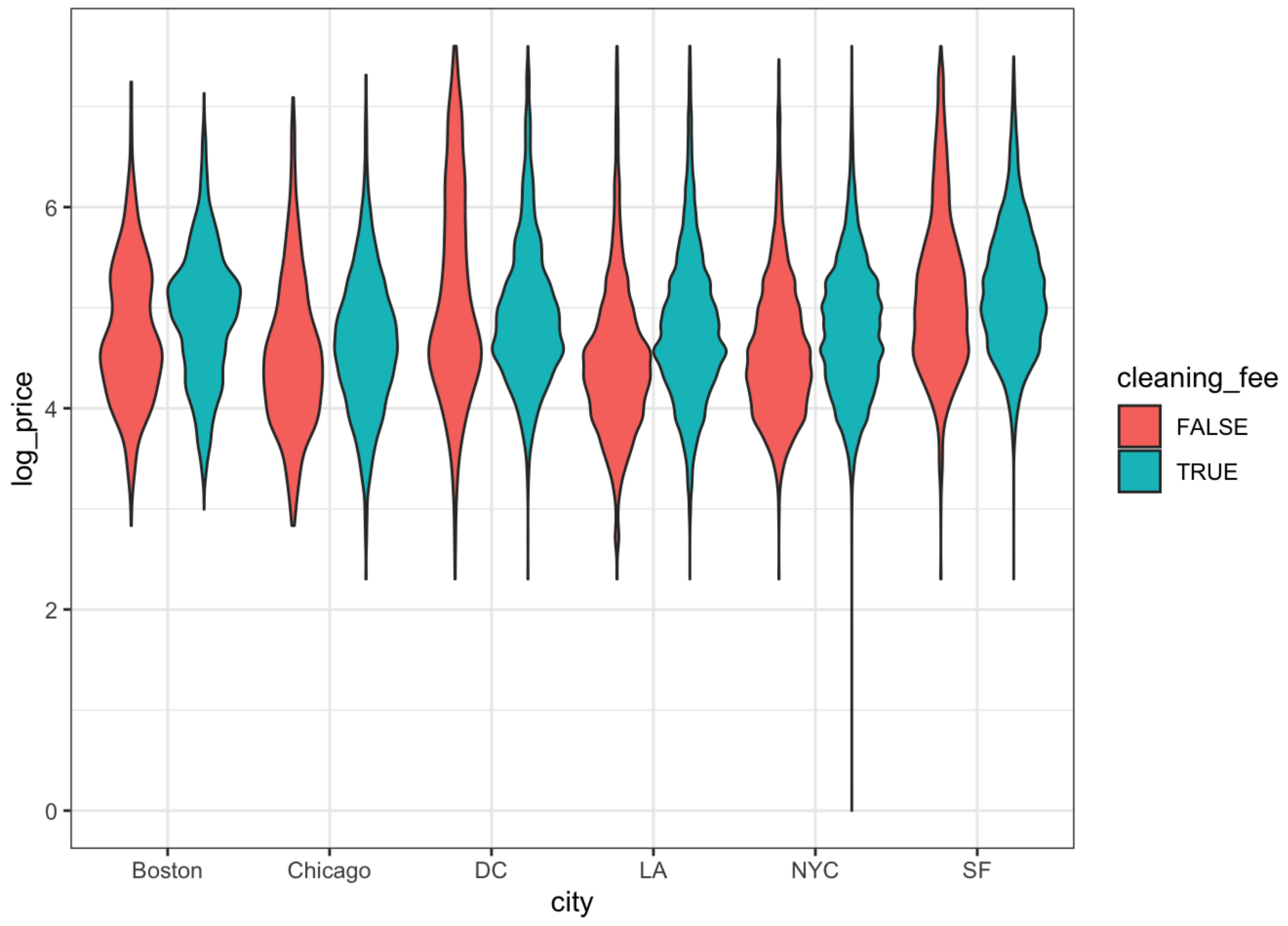
```
# The box plot of log price with the strictness of the cacellation policy
cancellation <- ggplot(train,aes(x=cancellation_policy,y=log_price))+geom_boxplot(outlie
r.shape =
NA);
cancellation
```



```
# box plot of log price with the property types
property<-ggplot(train,aes(x=property_type,y=log_price))+geom_boxplot(outlier.shape =
NA)+theme(axis.text.x = element_text(angle =90, vjust =0.5, hjust=1));
property
```



```
#violin plot of three variables: log price, city location and cleaning fee existance
three<-ggplot(train,aes(x=city,y=log_price,fill=cleaning_fee))+geom_violin()
three
```



```
###Machine Learning
dd <- fread("/Users/tommy/Downloads/Fall\ 2021/Final_num_version.csv")
#Basic summary stats for data column
str(dd)
```

```
## Classes 'data.table' and 'data.frame':   73923 obs. of  118 variables:
##  $ V1                             : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ log_price                      : num  5.01 5.13 4.98 6.62 4.74 ...
##  $ accommodates                   : int  3 7 5 4 2 2 3 2 2 2 ...
##  $ bathrooms                      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ first_review                   : int  618 205 302 0 1021 183 353 437 744 329
...
##  $ last_review                    : int  588 156 165 0 400 174 311 320 155 316
...
##  $ latitude                       : num  40.7 40.8 40.8 37.8 38.9 ...
##  $ longitude                      : num  -74 -74 -73.9 -122.4 -77 ...
##  $ number_of_reviews              : int  2 6 10 0 4 3 15 9 159 2 ...
##  $ review_scores_rating           : num  100 100 100 94.1 40 ...
##  $ host_has_profile_pic_t         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ host_identity_verified_t       : int  1 0 1 1 1 1 0 1 0 0 ...
##  $ instant_bookable_t             : int  0 1 1 0 1 1 1 0 0 1 ...
##  $ thumbnail_url_True             : int  1 1 1 1 0 1 1 1 1 1 ...
##  $ amenities_Air conditioning     : int  1 1 1 0 1 0 1 0 0 1 ...
##  $ amenities_Bath towel           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Bathtub              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Coffee maker         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Cooking basics       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Dishes and silverware    : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Elevator             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Hot water            : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Internet             : int  0 0 1 1 1 0 1 0 0 0 ...
##  $ amenities_Kitchen              : int  1 1 1 1 1 0 1 1 0 1 ...
##  $ amenities_Private bathroom     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Refrigerator         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Self Check-In        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Stove                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Toilet paper         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Apartment        : int  1 1 1 0 1 1 1 0 0 0 ...
##  $ property_type_Bed & Breakfast  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boat             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boutique hotel   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Bungalow         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cabin            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Camper/RV        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Casa particular  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Castle           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cave             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Chalet           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Condominium      : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ property_type_Dorm             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Earth House      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guest suite      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guesthouse       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Hostel           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_House            : int  0 0 0 1 0 0 0 0 1 1 ...
##  $ property_type_Hut              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_In-law           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Island           : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Lighthouse          : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Loft                : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Other               : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Parking Space       : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Serviced apartment  : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Tent                : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Timeshare           : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Tipi                : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Townhouse           : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Train               : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Treehouse           : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Vacation home       : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Villa               : int 0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Yurt                : int 0 0 0 0 0 0 0 0 0 0 ...
## $ room_type_Entire home/apt         : int 1 1 1 1 1 0 1 1 0 0 ...
## $ room_type_Private room            : int 0 0 0 0 0 1 0 0 1 1 ...
## $ room_type_Shared room             : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Airbed                   : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Couch                    : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Futon                    : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Pull-out Sofa            : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Real Bed                 : int 1 1 1 1 1 1 1 1 1 1 ...
## $ cleaning_fee_False                : int 0 0 0 0 0 0 0 0 0 0 ...
## $ cleaning_fee_True                 : int 1 1 1 1 1 1 1 1 1 1 ...
## $ cancellation_policy_flexible      : int 0 0 0 1 0 0 0 0 0 0 ...
## $ cancellation_policy_moderate      : int 0 0 1 0 1 0 1 1 1 1 ...
## $ cancellation_policy_strict        : int 1 1 0 0 0 1 0 0 0 0 ...
## $ cancellation_policy_super_strict_30: int 0 0 0 0 0 0 0 0 0 0 ...
## $ cancellation_policy_super_strict_60: int 0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_0.0                      : int 0 0 0 0 1 0 0 0 0 0 ...
## $ bedrooms_1.0                      : int 1 0 1 0 0 1 1 1 1 1 ...
## $ bedrooms_1.23526268079176         : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_2.0                      : int 0 0 0 1 0 0 0 0 0 0 ...
## $ bedrooms_3.0                      : int 0 1 0 0 0 0 0 0 0 0 ...
## $ bedrooms_4.0                      : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_5.0                      : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_6.0                      : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_7.0                      : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_8.0                      : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_9.0                      : int 0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_10.0                     : int 0 0 0 0 0 0 0 0 0 0 ...
## $ beds_0.0                          : int 0 0 0 0 0 0 0 0 0 0 ...
## $ beds_1.0                          : int 1 0 0 0 1 1 1 1 1 1 ...
## $ beds_1.23526268079176             : int 0 0 0 0 0 0 0 0 0 0 ...
## $ beds_2.0                          : int 0 0 0 1 0 0 0 0 0 0 ...
## $ beds_3.0                          : int 0 1 1 0 0 0 0 0 0 0 ...
## $ beds_4.0                          : int 0 0 0 0 0 0 0 0 0 0 ...
## $ beds_5.0                          : int 0 0 0 0 0 0 0 0 0 0 ...
## $ beds_6.0                          : int 0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
## - attr(*, ".internal.selfref")=<externalptr>
```

```
#Delete the first column in dd
dd = subset(dd, select = -c(V1))
set.seed(810)
#take 30% random rows and stick them in the test set
test_index <- sample(nrow(dd), nrow(dd) * 0.3)
dd_test <- dd[test_index,]
dd_train <- dd[-test_index,]
x_test <- dd_test[,-1]
x_train <- dd_train[,-1]
y_test <- dd_test$log_price
y_train <- dd_train$log_price

##linear Regression
model <- lm(log_price ~ ., data = dd_train)
y_hat_train <- predict(model, dd_train)
```

```
## Warning in predict.lm(model, dd_train): prediction from a rank-deficient fit may
## be misleading
```

```
mse_train <- mean((y_train - y_hat_train)^2)

print(model)
```

```
##
## Call:
## lm(formula = log_price ~ ., data = dd_train)
##
## Coefficients:
##                     (Intercept)                             accommodates
##                      -1.189e+02                                8.340e-02
##                       bathrooms                             first_review
##                       1.299e-01                                7.854e-06
##                     last_review                                 latitude
##                      -1.964e-04                               -3.409e-02
##                       longitude                        number_of_reviews
##                      -1.020e+00                               -6.583e-04
##            review_scores_rating                     host_has_profile_pic_t
##                       3.856e-03                               -9.551e-02
##         host_identity_verified_t                       instant_bookable_t
##                      -2.204e-02                               -1.968e-02
##                 thumbnail_url_True          `amenities_Air conditioning`
##                      -6.797e-02                                8.097e-02
##           `amenities_Bath towel`                         amenities_Bathtub
##                      -3.401e-02                               -4.866e-03
##         `amenities_Coffee maker`              `amenities_Cooking basics`
##                       4.195e-02                                2.490e-02
##   `amenities_Dishes and silverware`                       amenities_Elevator
##                       7.460e-02                                1.742e-01
##            `amenities_Hot water`                        amenities_Internet
##                      -5.013e-05                                3.097e-02
##               amenities_Kitchen            `amenities_Private bathroom`
##                      -5.702e-02                                2.055e-01
##           amenities_Refrigerator                `amenities_Self Check-In`
##                      -1.297e-01                               -5.584e-02
##                 amenities_Stove                 `amenities_Toilet paper`
##                       8.266e-03                                       NA
##          property_type_Apartment         `property_type_Bed & Breakfast`
##                       5.619e-02                                2.350e-01
##               property_type_Boat          `property_type_Boutique hotel`
##                       4.252e-01                                4.126e-01
##           property_type_Bungalow                      property_type_Cabin
##                       6.385e-02                               -8.813e-02
##         `property_type_Camper/RV`         `property_type_Casa particular`
##                      -1.822e-01                                3.057e-01
##             property_type_Castle                       property_type_Cave
##                       6.787e-01                                3.653e-01
##             property_type_Chalet                property_type_Condominium
##                       1.224e-01                                1.864e-01
##               property_type_Dorm             `property_type_Earth House`
##                      -3.971e-01                                3.225e-01
##        `property_type_Guest suite`                 property_type_Guesthouse
##                       9.331e-04                               -1.122e-02
##             property_type_Hostel                      property_type_House
##                      -3.970e-01                                6.309e-02
##                property_type_Hut                     `property_type_In-law`
##                      -4.729e-01                               -1.373e-01
##                property_type_Island              property_type_Lighthouse 
##                           7.721e-01                                    NA 
##                  property_type_Loft                   property_type_Other 
##                           2.134e-01                             2.052e-01 
##         `property_type_Parking Space`   `property_type_Serviced apartment` 
##                           9.842e-01                             2.704e-01 
##                  property_type_Tent               property_type_Timeshare 
##                          -3.006e-01                             5.153e-01 
##                  property_type_Tipi               property_type_Townhouse 
##                                  NA                             8.818e-02 
##                 property_type_Train               property_type_Treehouse 
##                           5.781e-01                            -2.213e-02 
##         `property_type_Vacation home`                   property_type_Villa 
##                           3.253e-01                             2.976e-01 
##                  property_type_Yurt         `room_type_Entire home/apt` 
##                                  NA                             9.874e-01 
##            `room_type_Private room`               `room_type_Shared room` 
##                           4.255e-01                                    NA 
##                     bed_type_Airbed                        bed_type_Couch 
##                          -1.528e-02                             8.194e-02 
##                      bed_type_Futon              `bed_type_Pull-out Sofa` 
##                          -6.455e-02                            -1.593e-02 
##                 `bed_type_Real Bed`                    cleaning_fee_False 
##                                  NA                             3.915e-02 
##                   cleaning_fee_True         cancellation_policy_flexible 
##                                  NA                            -5.531e-01 
##        cancellation_policy_moderate           cancellation_policy_strict 
##                          -5.756e-01                            -5.444e-01 
## cancellation_policy_super_strict_30  cancellation_policy_super_strict_60 
##                          -3.085e-01                                    NA 
##                        bedrooms_0.0                          bedrooms_1.0 
##                          -7.882e-01                            -7.172e-01 
##          bedrooms_1.23526268079176                          bedrooms_2.0 
##                          -7.430e-01                            -5.571e-01 
##                        bedrooms_3.0                          bedrooms_4.0 
##                          -3.741e-01                            -2.234e-01 
##                        bedrooms_5.0                          bedrooms_6.0 
##                          -1.745e-01                            -4.698e-02 
##                        bedrooms_7.0                          bedrooms_8.0 
##                          -4.147e-02                            -2.038e-01 
##                        bedrooms_9.0                         bedrooms_10.0 
##                           5.778e-02                                    NA 
##                            beds_0.0                              beds_1.0 
##                           5.792e-01                             3.149e-01 
##              beds_1.23526268079176                              beds_2.0 
##                           4.112e-01                             3.086e-01 
##                            beds_3.0                              beds_4.0 
##                           2.922e-01                             2.267e-01 
##                            beds_5.0                              beds_6.0 
##                           2.004e-01                             1.138e-01 
##                            beds_7.0                              beds_8.0 
##                           4.403e-02                            -1.634e-01 
##                            beds_9.0                             beds_10.0 
##                          -2.520e-01                            -3.641e-01 
##                        beds_11.0                               beds_12.0
##                       -2.362e-01                              -2.411e-01
##                        beds_13.0                               beds_14.0
##                       -7.828e-01                               1.691e-01
##                        beds_15.0                               beds_16.0
##                       -4.010e-01                              -7.645e-02
##                        beds_18.0                             city_Boston
##                               NA                               5.221e+01
##                     city_Chicago                                 city_DC
##                        3.502e+01                               4.609e+01
##                          city_LA                                city_NYC
##                        3.614e+00                               4.917e+01
##                          city_SF                       host_since_days_n
##                               NA                               3.690e-05
##           host_response_rate_num
##                       -1.143e-03
```

```
y_hat_test <- predict(model, dd_test)
```

```
## Warning in predict.lm(model, dd_test): prediction from a rank-deficient fit may
## be misleading
```

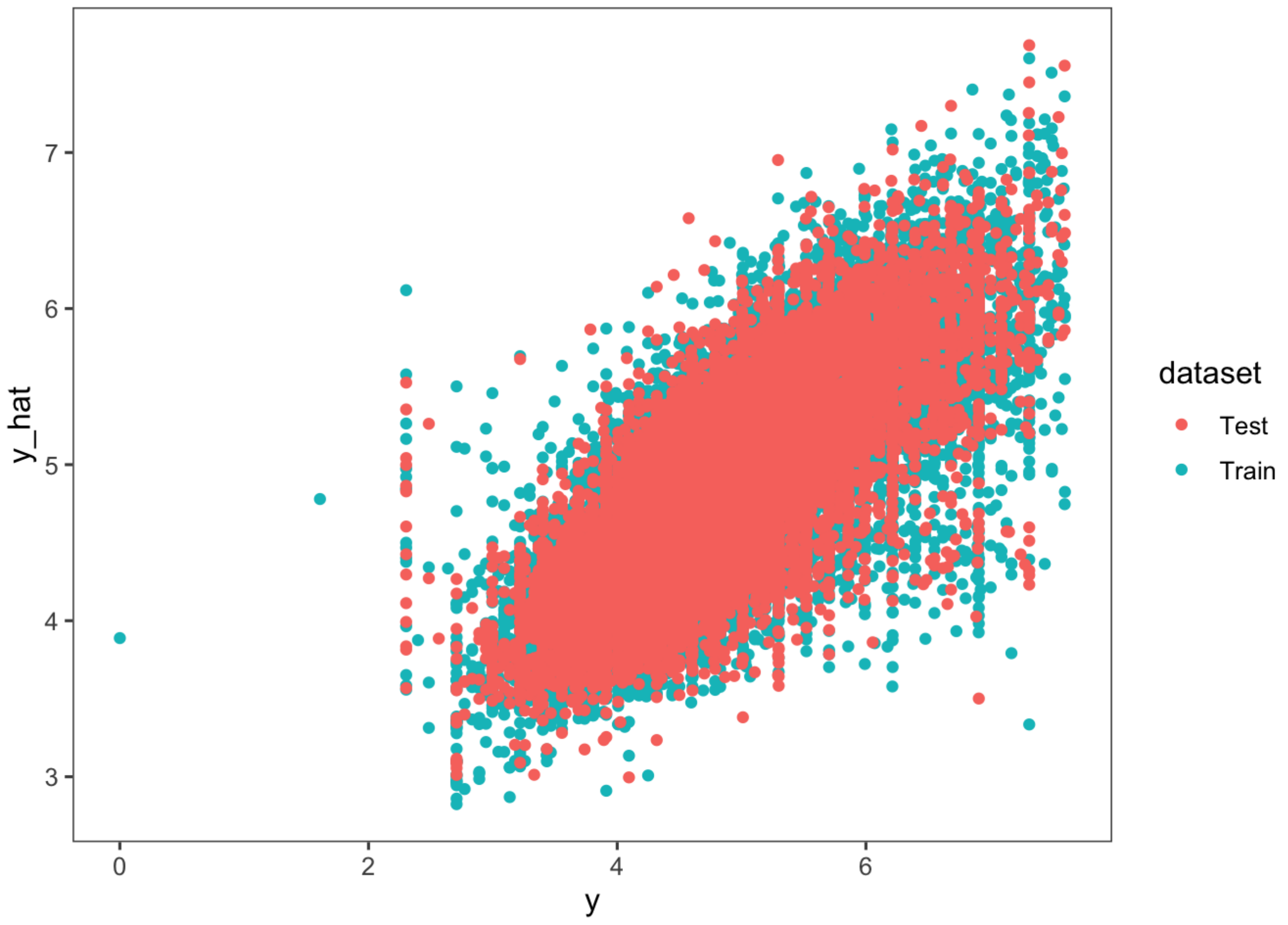
```
mse_test <- mean((y_test - y_hat_test)^2)

#PRINT MSE(training)
print(mse_train)
```

```
## [1] 0.2072443
```

```
#PRINT MSE(testing)
print(mse_test)
```

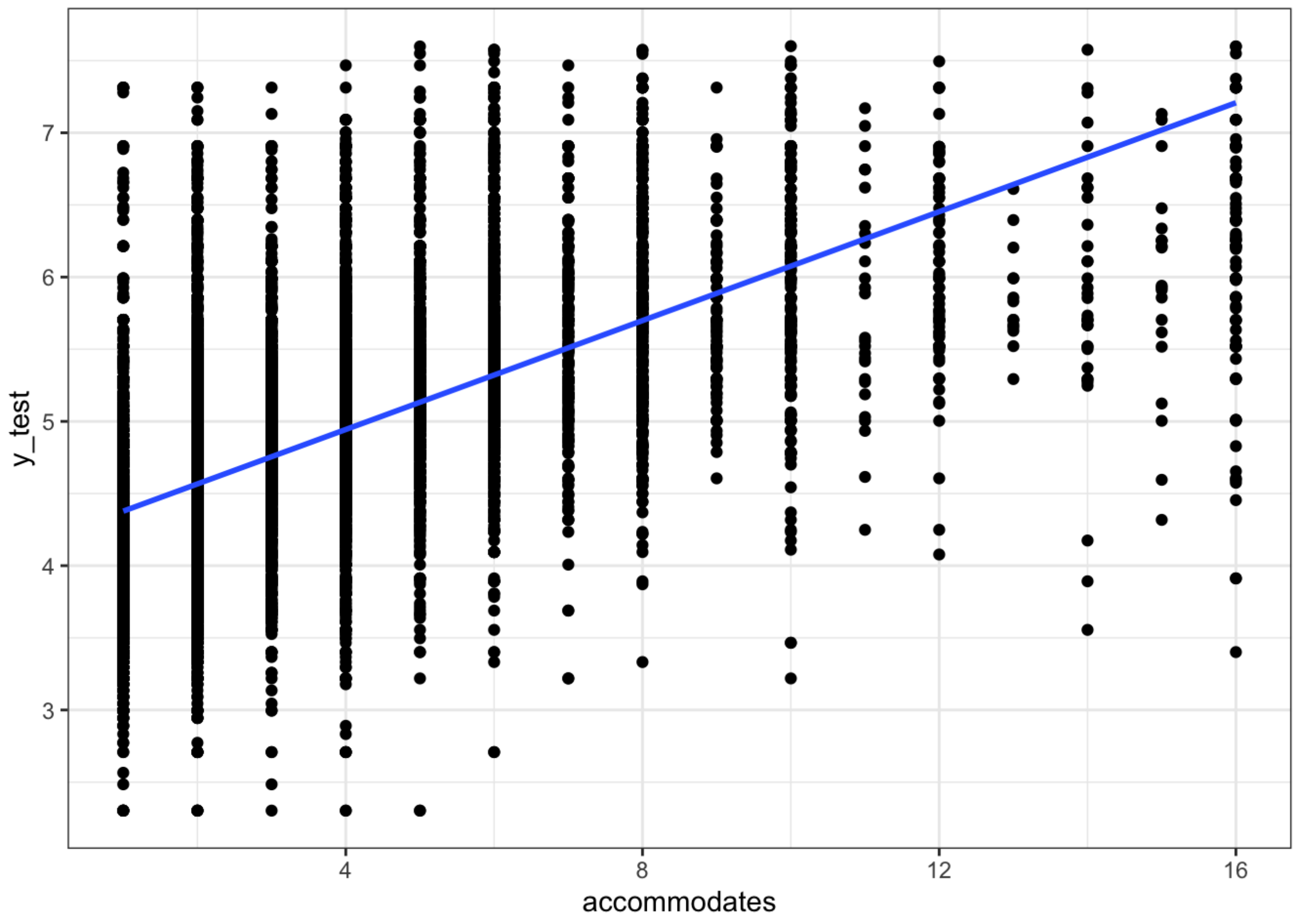
```
## [1] 0.2108787
```

```
#create tables
dd_y <- data.table(
y = y_train,
y_hat = y_hat_train,
dataset = "Train"
)

dd_y <- rbind(dd_y, data.table(
y = y_test,
y_hat = y_hat_test,
dataset = "Test"
))
#plot
ggplot(dd_y, aes(y, y_hat, color = dataset)) + geom_point() + theme_few()
```



```
ggplot(dd_test, aes(y=y_test, x = accommodates)) + geom_point() + stat_smooth(method =
'lm', se=FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
##Ridge Regression

airbnb <- fread("/Users/tommy/Downloads/Fall\ 2021/Final_num_version.csv",stringsAsFacto
rs = T)
airbnb = subset(airbnb, select = -c(V1) )
set.seed(810)
sample <- sample.int(n=nrow(airbnb),size=floor(.7*nrow(airbnb)),replace=F)
train <- airbnb[sample,]
test <- airbnb[-sample,]
y_train <- train$log_price
x_train <- data.matrix(subset(train,select = -c(log_price)))
y_test <- test$log_price
x_test <- data.matrix(subset(test,select = -c(log_price)))

#Cross validation
cv_model <- cv.glmnet(x_train,y_train,alpha = 0)
#Getting the minimum lambda
best_lambda <- cv_model$lambda.min
#Fitting the best model
best_model <- glmnet(x_train,y_train,alpha = 0, lambda = best_lambda)
coef(best_model)
```

```
## 117 x 1 sparse Matrix of class "dgCMatrix"
##                                              s0
## (Intercept)                        3.952396e+00
## accommodates                       6.902291e-02
## bathrooms                          1.358707e-01
## first_review                      -6.635765e-06
## last_review                       -1.566547e-04
## latitude                           3.051688e-03
## longitude                         -9.153105e-04
## number_of_reviews                 -5.213289e-04
## review_scores_rating               4.163939e-03
## host_has_profile_pic_t            -5.996529e-02
## host_identity_verified_t          -2.564810e-02
## instant_bookable_t                -3.410331e-02
## thumbnail_url_True                -6.597457e-02
## amenities_Air conditioning         6.298421e-02
## amenities_Bath towel               1.282988e-02
## amenities_Bathtub                 -5.438652e-03
## amenities_Coffee maker             3.610599e-02
## amenities_Cooking basics           4.225581e-02
## amenities_Dishes and silverware    3.310074e-02
## amenities_Elevator                 1.698072e-01
## amenities_Hot water               -8.905121e-03
## amenities_Internet                 2.836596e-02
## amenities_Kitchen                 -3.281136e-02
## amenities_Private bathroom         1.947149e-01
## amenities_Refrigerator            -7.038374e-02
## amenities_Self Check-In           -5.223783e-02
## amenities_Stove                   -1.871192e-02
## amenities_Toilet paper             9.471573e-03
## property_type_Apartment           -1.211101e-02
## property_type_Bed & Breakfast      1.479256e-01
## property_type_Boat                 2.964345e-01
## property_type_Boutique hotel       2.719308e-01
## property_type_Bungalow            -9.299951e-03
## property_type_Cabin               -3.203758e-02
## property_type_Camper/RV           -9.876544e-02
## property_type_Casa particular      3.275791e-01
## property_type_Castle               4.446168e-01
## property_type_Cave                 1.950466e-01
## property_type_Chalet               2.152503e-02
## property_type_Condominium          1.148726e-01
## property_type_Dorm                -4.511059e-01
## property_type_Earth House          7.378739e-01
## property_type_Guest suite         -6.971181e-02
## property_type_Guesthouse          -4.581692e-02
## property_type_Hostel              -5.083008e-01
## property_type_House               -2.760003e-02
## property_type_Hut                 -5.677500e-01
## property_type_In-law              -1.916653e-01
## property_type_Island               .
## property_type_Lighthouse           1.286268e-01
## property_type_Loft                 1.465496e-01
## property_type_Other                   1.073972e-01
## property_type_Parking Space           .
## property_type_Serviced apartment      2.754744e-01
## property_type_Tent                   -2.273363e-01
## property_type_Timeshare               4.582811e-01
## property_type_Tipi                    5.925272e-01
## property_type_Townhouse              -5.876573e-04
## property_type_Train                   5.636192e-01
## property_type_Treehouse               2.072202e-02
## property_type_Vacation home           2.233082e-01
## property_type_Villa                   1.265950e-01
## property_type_Yurt                    2.285728e-01
## room_type_Entire home/apt             3.092108e-01
## room_type_Private room               -2.348936e-01
## room_type_Shared room                -6.201617e-01
## bed_type_Airbed                      -7.440672e-02
## bed_type_Couch                        3.517345e-02
## bed_type_Futon                       -4.697792e-02
## bed_type_Pull-out Sofa                2.821083e-02
## bed_type_Real Bed                     2.038974e-02
## cleaning_fee_False                    1.464916e-02
## cleaning_fee_True                    -1.450925e-02
## cancellation_policy_flexible         -7.368696e-04
## cancellation_policy_moderate         -2.377365e-02
## cancellation_policy_strict            1.680882e-02
## cancellation_policy_super_strict_30  2.349397e-01
## cancellation_policy_super_strict_60  7.198056e-01
## bedrooms_0.0                         -1.569351e-01
## bedrooms_1.0                         -9.910440e-02
## bedrooms_1.23526268079176            -1.154131e-01
## bedrooms_2.0                          7.565915e-02
## bedrooms_3.0                          2.537887e-01
## bedrooms_4.0                          3.932980e-01
## bedrooms_5.0                          5.099700e-01
## bedrooms_6.0                          5.495670e-01
## bedrooms_7.0                          5.472334e-01
## bedrooms_8.0                          4.821198e-01
## bedrooms_9.0                          6.925693e-02
## bedrooms_10.0                         2.649338e-01
## beds_0.0                              4.843186e-01
## beds_1.0                             -3.985492e-03
## beds_1.23526268079176                 6.851256e-02
## beds_2.0                              1.723517e-02
## beds_3.0                              2.579128e-02
## beds_4.0                             -1.774342e-02
## beds_5.0                             -2.430871e-02
## beds_6.0                             -1.042354e-01
## beds_7.0                             -2.271246e-01
## beds_8.0                             -2.830252e-01
## beds_9.0                             -3.649829e-01
## beds_10.0                            -4.868212e-01
## beds_11.0                            -3.462168e-01
## beds_12.0                            -3.823974e-01
## beds_13.0                            -4.415403e-01
## beds_14.0                           -2.218990e-01
## beds_15.0                           -6.109262e-01
## beds_16.0                           -8.302594e-02
## beds_18.0                             .
## city_Boston                           3.901053e-02
## city_Chicago                         -2.274987e-01
## city_DC                               8.762466e-02
## city_LA                              -8.284772e-02
## city_NYC                             -1.948668e-02
## city_SF                               3.075430e-01
## host_since_days_n                     4.673323e-05
## host_response_rate_num               -1.100087e-03
```

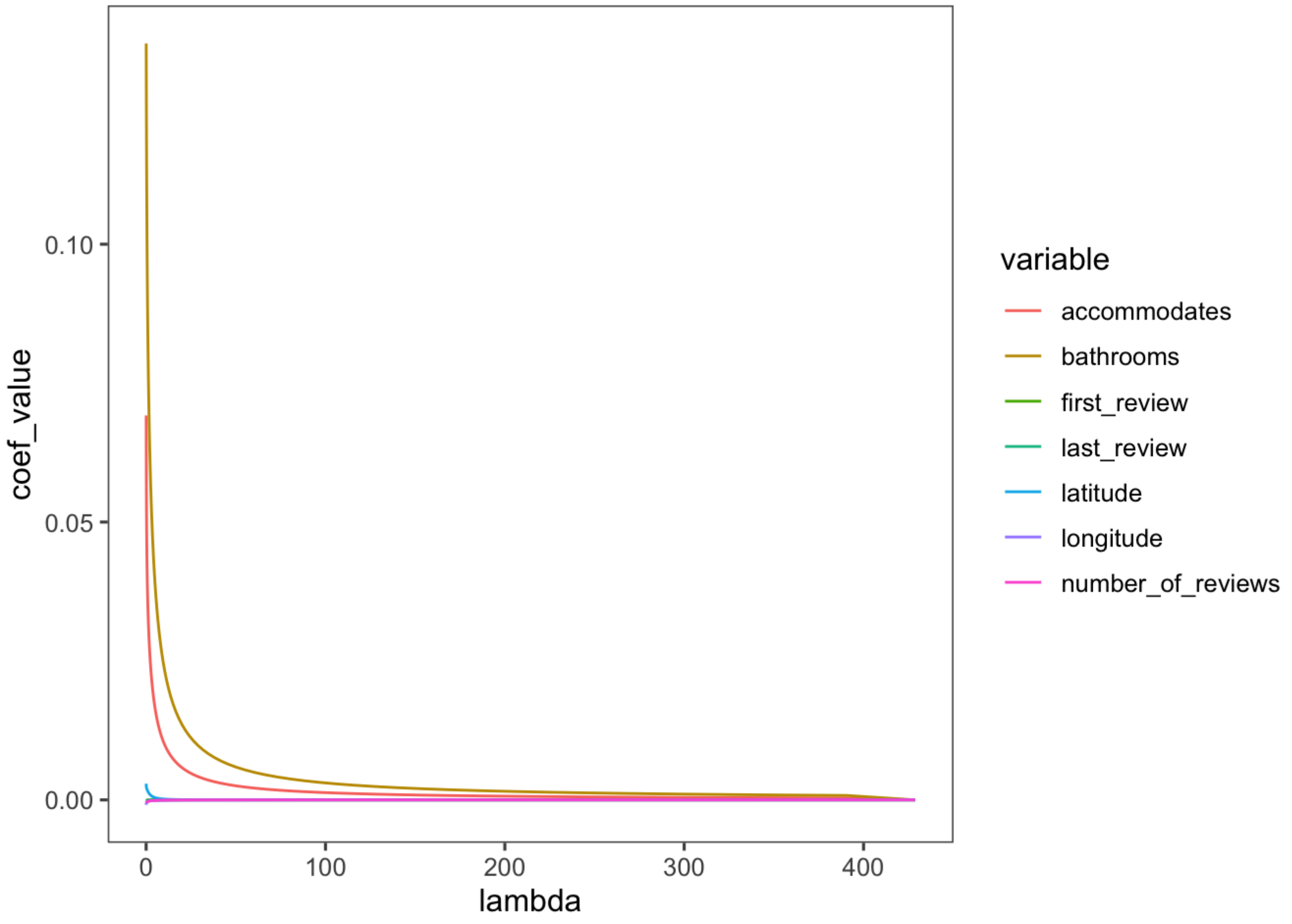
```
#mse train : 0.2274799
yhat_train_ridge <- predict(best_model, x_train)
print(mse_train_ridge <- mean((y_train - yhat_train_ridge)^2))
```

```
## [1] 0.2172679
```

```
#mse test : 0.2272783
yhat_test_ridge <- predict(best_model, x_test)
print(mse_test_ridge <- mean((y_test - yhat_test_ridge)^2))
```

```
## [1] 0.2169934
```

```
#coef for ridge regression
ridge.coef <- predict(cv_model, type = "coefficients", s = cv_model$lambda)
#Creat table to plot
to_plot <- data.table(lambda = cv_model$lambda, coef_value = ridge.coef[2,], variable =
"accommodates")
to_plot <- rbind(to_plot, data.table(lambda = cv_model$lambda, coef_value = ridge.coef[3
,], variable = "bathrooms"))
to_plot <- rbind(to_plot, data.table(lambda = cv_model$lambda, coef_value = ridge.coef[4
,], variable = "first_review"))
to_plot <- rbind(to_plot, data.table(lambda = cv_model$lambda, coef_value = ridge.coef[5
,], variable = "last_review"))
to_plot <- rbind(to_plot, data.table(lambda = cv_model$lambda, coef_value = ridge.coef[6
,], variable = "latitude"))
to_plot <- rbind(to_plot, data.table(lambda = cv_model$lambda, coef_value = ridge.coef[7
,], variable = "longitude"))
to_plot <- rbind(to_plot, data.table(lambda = cv_model$lambda, coef_value = ridge.coef[8
,], variable = "number_of_reviews"))

#Plot the coef_values with different values of lambda
#ridge regression will penalize the coefficients and shrink them towards zero
ggplot(to_plot, aes(lambda, coef_value, color=variable)) + geom_line() + theme_few()
```



```
##Lasso Regression
#load the data set
dd <- fread("/Users/tommy/Downloads/Fall\ 2021/Final_num_version.csv")
```

```
#have a look at the data
str(dd)
```

```
## Classes 'data.table' and 'data.frame':   73923 obs. of  118 variables:
##  $ V1                             : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ log_price                      : num  5.01 5.13 4.98 6.62 4.74 ...
##  $ accommodates                   : int  3 7 5 4 2 2 3 2 2 2 ...
##  $ bathrooms                      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ first_review                   : int  618 205 302 0 1021 183 353 437 744 329
...
##  $ last_review                    : int  588 156 165 0 400 174 311 320 155 316
...
##  $ latitude                       : num  40.7 40.8 40.8 37.8 38.9 ...
##  $ longitude                      : num  -74 -74 -73.9 -122.4 -77 ...
##  $ number_of_reviews              : int  2 6 10 0 4 3 15 9 159 2 ...
##  $ review_scores_rating           : num  100 100 100 94.1 40 ...
##  $ host_has_profile_pic_t         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ host_identity_verified_t       : int  1 0 1 1 1 1 0 1 0 0 ...
##  $ instant_bookable_t             : int  0 1 1 0 1 1 1 0 0 1 ...
##  $ thumbnail_url_True             : int  1 1 1 1 0 1 1 1 1 1 ...
##  $ amenities_Air conditioning     : int  1 1 1 0 1 0 1 0 0 1 ...
##  $ amenities_Bath towel           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Bathtub              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Coffee maker         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Cooking basics       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Dishes and silverware    : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Elevator             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Hot water            : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Internet             : int  0 0 0 1 1 0 1 0 0 0 ...
##  $ amenities_Kitchen              : int  1 1 1 1 1 0 1 1 0 1 ...
##  $ amenities_Private bathroom     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Refrigerator         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Self Check-In        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Stove                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Toilet paper         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Apartment        : int  1 1 1 0 1 1 1 0 0 0 ...
##  $ property_type_Bed & Breakfast      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boat             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boutique hotel   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Bungalow         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cabin            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Camper/RV        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Casa particular      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Castle           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cave             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Chalet           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Condominium      : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ property_type_Dorm             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Earth House      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guest suite      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guesthouse       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Hostel           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_House            : int  0 0 0 1 0 0 0 0 1 1 ...
##  $ property_type_Hut              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_In-law           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Island           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Lighthouse          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Loft                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Other               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Parking Space       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Serviced apartment  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Tent                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Timeshare           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Tipi                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Townhouse           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Train               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Treehouse           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Vacation home       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Villa               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Yurt                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ room_type_Entire home/apt         : int  1 1 1 1 1 0 1 1 0 0 ...
##  $ room_type_Private room            : int  0 0 0 0 0 1 0 0 1 1 ...
##  $ room_type_Shared room             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Airbed                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Couch                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Futon                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Pull-out Sofa            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Real Bed                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cleaning_fee_False                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cleaning_fee_True                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cancellation_policy_flexible      : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ cancellation_policy_moderate      : int  0 0 1 0 1 0 1 1 1 1 ...
##  $ cancellation_policy_strict        : int  1 1 0 0 0 1 0 0 0 0 ...
##  $ cancellation_policy_super_strict_30: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cancellation_policy_super_strict_60: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_0.0                      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ bedrooms_1.0                      : int  1 0 1 0 0 1 1 1 1 1 ...
##  $ bedrooms_1.23526268079176         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_2.0                      : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ bedrooms_3.0                      : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_4.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_5.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_6.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_7.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_8.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_9.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_10.0                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_0.0                          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_1.0                          : int  1 0 0 0 1 1 1 1 1 1 ...
##  $ beds_1.23526268079176             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_2.0                          : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ beds_3.0                          : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ beds_4.0                          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_5.0                          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_6.0                          : int  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
# create our train and test data sets
set.seed(810)

#Delete the first column in dd
dd <- subset(dd, select = -c(V1))
set.seed(810)
#take 30% random rows and stick them in the test set
test_index <- sample(nrow(dd), nrow(dd) * 0.3)
dd.test <- dd[test_index,]
dd.train <- dd[-test_index,]
x.test <- dd.test[,-1]
x.train <- dd.train[,-1]
y.test <- dd.test$log_price
y.train <- dd.train$log_price
```

```
# fit our lasso model with cross validation, nlambda=100
fit.lasso <- glmnet(x.train, y.train, alpha = 1, nlambda = 100)
#view the values of lambdas
fit.lasso$lambda
```

```
##  [1] 0.4333510697 0.3948533551 0.3597756713 0.3278141923 0.2986920830
##  [6] 0.2721571016 0.2479794148 0.2259496071 0.2058768667 0.1875873333
## [11] 0.1709225917 0.1557382997 0.1419029384 0.1292966725 0.1178103125
## [16] 0.1073443690 0.0978081911 0.0891191810 0.0812020786 0.0739883097
## [21] 0.0674153922 0.0614263946 0.0559694430 0.0509972719 0.0464668149
## [26] 0.0423388311 0.0385775660 0.0351504414 0.0320277731 0.0291825140
## [31] 0.0265900199 0.0242278358 0.0220755016 0.0201143749 0.0183274693
## [36] 0.0166993074 0.0152157870 0.0138640584 0.0126324137 0.0115101849
## [41] 0.0104876518 0.0095559577 0.0087070327 0.0079335240 0.0072287315
## [46] 0.0065865510 0.0060014199 0.0054682703 0.0049824843 0.0045398541
## [51] 0.0041365460 0.0037690666 0.0034342331 0.0031291453 0.0028511607
## [56] 0.0025978714 0.0023670837 0.0021567985 0.0019651945 0.0017906120
## [61] 0.0016315390 0.0014865975 0.0013545323 0.0012341994 0.0011245565
## [66] 0.0010246540 0.0009336266 0.0008506858 0.0007751132 0.0007062543
## [71] 0.0006435126 0.0005863447 0.0005342555 0.0004867937 0.0004435483
## [76] 0.0004041447 0.0003682416 0.0003355280 0.0003057206 0.0002785612
## [81] 0.0002538146 0.0002312664
```

```
#make predictions
chosenlam<-fit.lasso$lambda.min
newXtest <- model.matrix(~.-y.test,data=x.test)
```

```
## Warning in terms.formula(object, data = data): 'varlist' has changed (from
## nvar=116) to new 117 after EncodeVars() -- should no longer happen!
```

```
## Warning in terms.formula(formula, data = data): 'varlist' has changed (from
## nvar=116) to new 117 after EncodeVars() -- should no longer happen!
```

```
newXtest<-newXtest[,-1]
newXtrain<- model.matrix(~.-y.train,data=x.train)
```

```
## Warning in terms.formula(object, data = data): 'varlist' has changed (from
## nvar=116) to new 117 after EncodeVars() -- should no longer happen!

## Warning in terms.formula(object, data = data): 'varlist' has changed (from
## nvar=116) to new 117 after EncodeVars() -- should no longer happen!
```

```
newXtrain<-newXtrain[,-1]
```

```
str(newXtest)
```

```
##  num [1:22176, 1:116] 4 2 1 1 2 2 3 4 6 5 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:22176] "1" "2" "3" "4" ...
##   ..$ : chr [1:116] "accommodates" "bathrooms" "first_review" "last_review" ...
```

```
yhat.train.lasso <- predict(fit.lasso,s=chosenlam,newXtest)
yhat.test.lasso <- predict(fit.lasso,s=chosenlam,newXtrain)
print(mse.test.lasso <- mean((y.test - yhat.test.lasso)^2))
```

```
## Warning in y.test - yhat.test.lasso: longer object length is not a multiple of
## shorter object length
```

```
## [1] 0.7402901
```

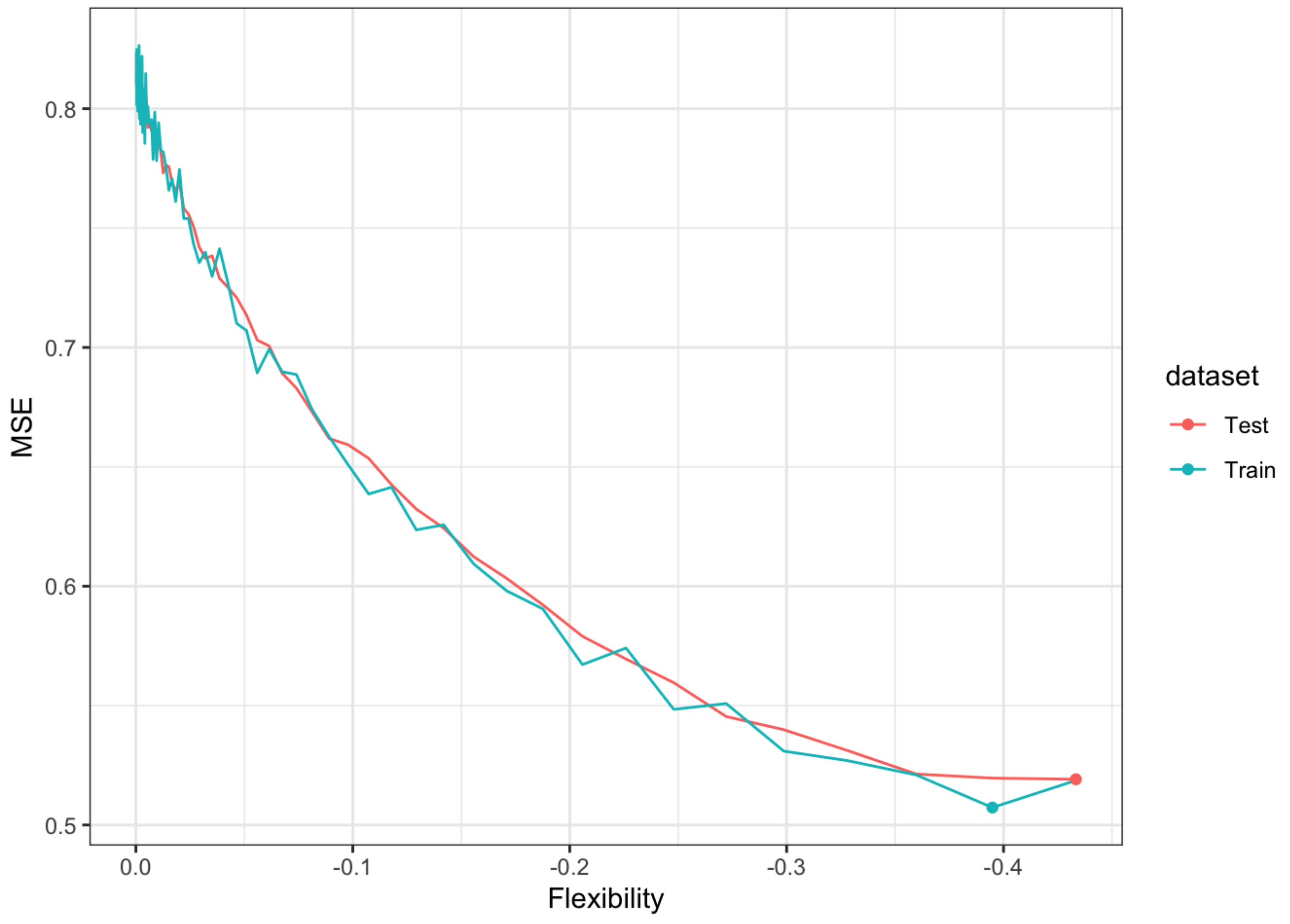
```
#plot
mse_train <- colMeans((y.train - yhat.train.lasso)^2)
```

```
## Warning in y.train - yhat.train.lasso: longer object length is not a multiple of
## shorter object length
```

```
mse_test <- colMeans((y.test - yhat.test.lasso)^2)
```

```
## Warning in y.test - yhat.test.lasso: longer object length is not a multiple of
## shorter object length
```

```
lambda_min_mse_train <- fit.lasso$lambda[which.min(mse_train)]
lambda_min_mse_test <- fit.lasso$lambda[which.min(mse_test)]

dd_mse <- data.table(
 lambda = fit.lasso$lambda,
 mse = mse_train,
 dataset = "Train",
 is_min = mse_train == min(mse_train)
)
dd_mse <- rbind(dd_mse, data.table(
 lambda = fit.lasso$lambda,
 mse = mse_test,
 dataset = "Test",
 is_min = mse_test == min(mse_test)
))

ggplot(dd_mse, aes(-lambda, mse, color=dataset)) +
geom_line() +
geom_point(data=dd_mse[is_min==TRUE]) +
scale_y_continuous("MSE") +
scale_x_reverse("Flexibility")
```



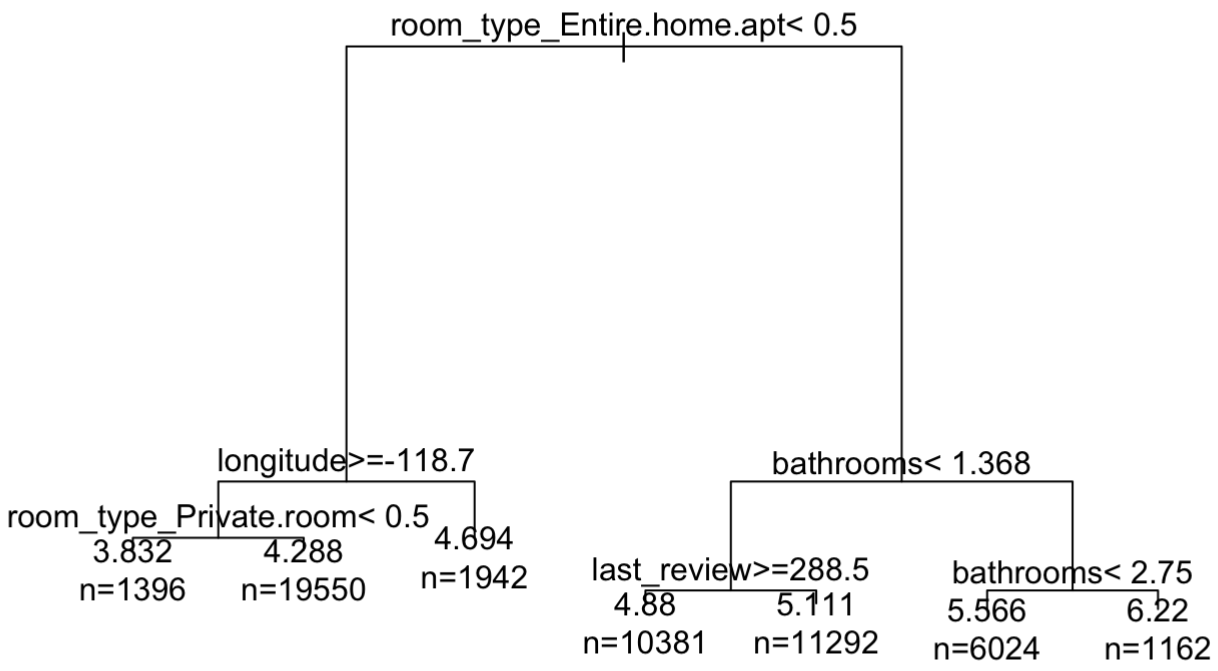
```r
##Trees
#Download the dataset Final_num_version.csv and load it into R
dd <- read.csv("/Users/tommy/Downloads/Fall\ 2021/Final_num_version.csv")

#Basic summary stats for data column
str(dd)
```

```
## 'data.frame':    73923 obs. of  118 variables:
##  $ X                                  : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ log_price                          : num  5.01 5.13 4.98 6.62 4.74 ...
##  $ accommodates                       : int  3 7 5 4 2 2 3 2 2 2 ...
##  $ bathrooms                          : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ first_review                       : int  618 205 302 0 1021 183 353 437 744 329
...
##  $ last_review                        : int  588 156 165 0 400 174 311 320 155 316
...
##  $ latitude                           : num  40.7 40.8 40.8 37.8 38.9 ...
##  $ longitude                          : num  -74 -74 -73.9 -122.4 -77 ...
##  $ number_of_reviews                  : int  2 6 10 0 4 3 15 9 159 2 ...
##  $ review_scores_rating               : num  100 100 100 94.1 40 ...
##  $ host_has_profile_pic_t             : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ host_identity_verified_t           : int  1 0 1 1 1 1 0 1 0 0 ...
##  $ instant_bookable_t                 : int  0 1 1 0 1 1 1 0 0 1 ...
##  $ thumbnail_url_True                 : int  1 1 1 1 0 1 1 1 1 1 ...
##  $ amenities_Air.conditioning         : int  1 1 1 0 1 0 1 0 0 1 ...
##  $ amenities_Bath.towel               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Bathtub                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Coffee.maker             : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Cooking.basics           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Dishes.and.silverware    : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Elevator                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Hot.water                : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Internet                 : int  0 0 0 1 1 0 1 0 0 0 ...
##  $ amenities_Kitchen                  : int  1 1 1 1 1 0 1 1 0 1 ...
##  $ amenities_Private.bathroom         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Refrigerator             : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Self.Check.In            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Stove                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Toilet.paper             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Apartment            : int  1 1 1 0 1 1 1 0 0 0 ...
##  $ property_type_Bed...Breakfast      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boat                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boutique.hotel       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Bungalow             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cabin                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Camper.RV            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Casa.particular      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Castle               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cave                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Chalet               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Condominium          : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ property_type_Dorm                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Earth.House          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guest.suite          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guesthouse           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Hostel               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_House                : int  0 0 0 1 0 0 0 0 1 1 ...
##  $ property_type_Hut                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_In.law               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Island               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Lighthouse           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Loft                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Other                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Parking.Space        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Serviced.apartment   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Tent                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Timeshare            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Tipi                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Townhouse            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Train                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Treehouse            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Vacation.home        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Villa                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Yurt                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ room_type_Entire.home.apt          : int  1 1 1 1 1 0 1 1 0 0 ...
##  $ room_type_Private.room             : int  0 0 0 0 0 1 0 0 1 1 ...
##  $ room_type_Shared.room              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Airbed                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Couch                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Futon                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Pull.out.Sofa             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Real.Bed                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cleaning_fee_False                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cleaning_fee_True                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cancellation_policy_flexible       : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ cancellation_policy_moderate       : int  0 0 1 0 1 0 1 1 1 1 ...
##  $ cancellation_policy_strict         : int  1 1 0 0 0 1 0 0 0 0 ...
##  $ cancellation_policy_super_strict_30: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cancellation_policy_super_strict_60: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_0.0                       : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ bedrooms_1.0                       : int  1 0 1 0 0 1 1 1 1 1 ...
##  $ bedrooms_1.23526268079176          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_2.0                       : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ bedrooms_3.0                       : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_4.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_5.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_6.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_7.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_8.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_9.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_10.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_0.0                           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_1.0                           : int  1 0 0 0 1 1 1 1 1 1 ...
##  $ beds_1.23526268079176              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_2.0                           : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ beds_3.0                           : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ beds_4.0                           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_5.0                           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_6.0                           : int  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
```

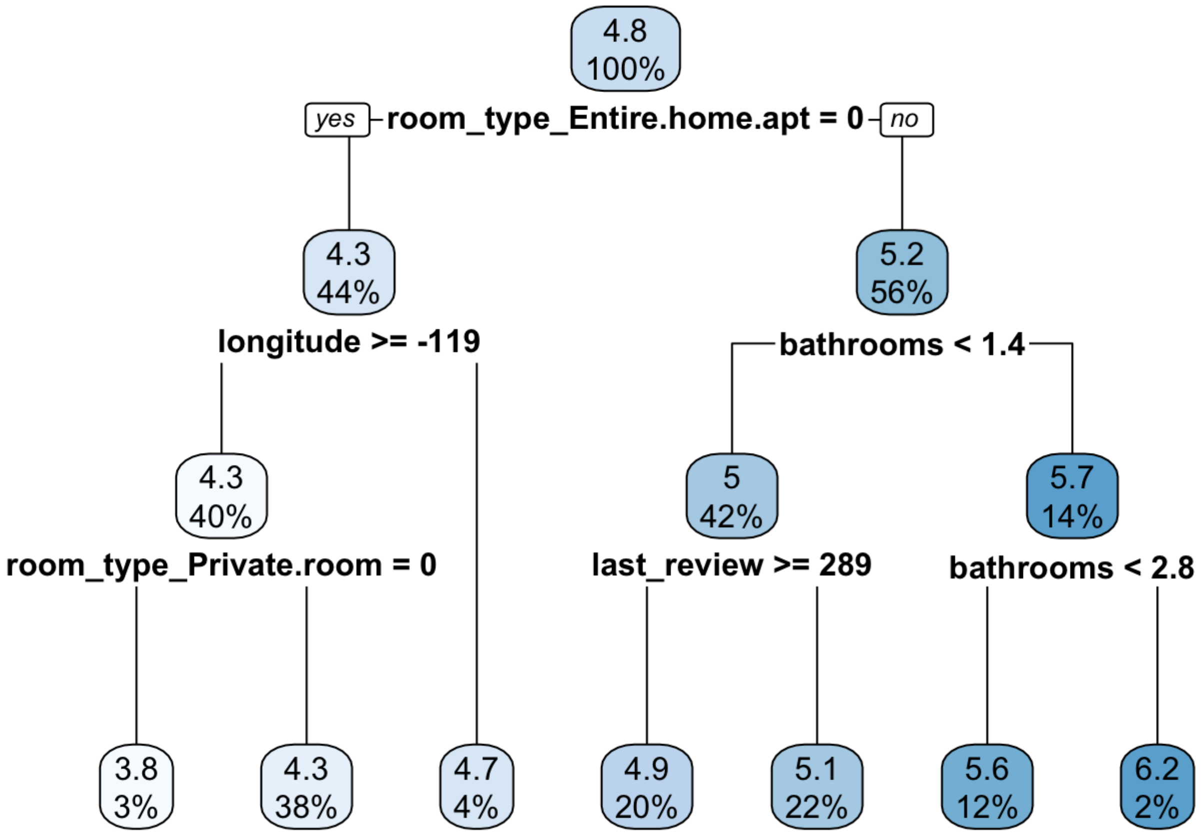
```
#Delete the first column in dd
dd = subset(dd, select = -c(X))
set.seed(810)
#take 30% random rows and stick them in the test set
test_index <- sample(nrow(dd), nrow(dd) * 0.3)
dd_test <- dd[test_index,]
dd_train <- dd[-test_index,]
x_test <- dd_test[,-1]
x_train <- dd_train[,-1]
y_test <- dd_test$log_price
y_train <- dd_train$log_price

#start with a straightforward regression tree.
fit.tree <- rpart(log_price~.,
dd_train,
control = rpart.control(cp = 0.01))

print(fit.tree)
```

```
## n= 51747
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 51747 26584.4600 4.781565
##    2) room_type_Entire.home.apt< 0.5 22888  6217.9900 4.294960
##      4) longitude>=-118.6632 20946  5386.4250 4.257984
##        8) room_type_Private.room< 0.5 1396   595.5682 3.832191 *
##        9) room_type_Private.room>=0.5 19550  4519.6900 4.288389 *
##      5) longitude< -118.6632 1942   494.0590 4.693766 *
##    3) room_type_Entire.home.apt>=0.5 28859 10648.7400 5.167490
##      6) bathrooms< 1.367631 21673  5292.5290 5.000316
##       12) last_review>=288.5 10381  2028.7410 4.879885 *
##       13) last_review< 288.5 11292  2974.8120 5.111031 *
##      7) bathrooms>=1.367631 7186  2923.7330 5.671686
##       14) bathrooms< 2.75 6024  2075.5920 5.565959 *
##       15) bathrooms>=2.75 1162   431.7096 6.219796 *
```

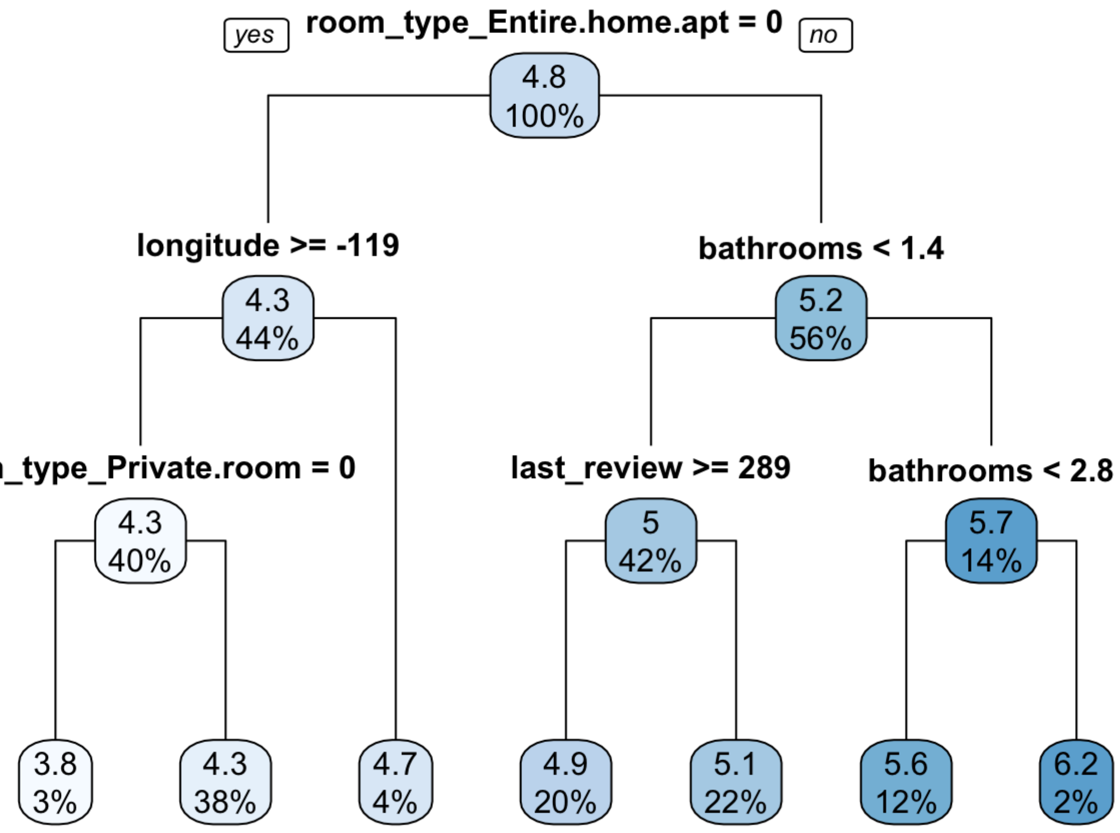
```
##You can control the complexity of the tree using the cp parameter. Smaller values will
give you more complex trees. One of the advantages of trees is that they are simple enou
gh to plot.

par(xpd = TRUE)
plot(fit.tree, compress=TRUE)
text(fit.tree, use.n=TRUE)
```



```
rpart.plot(fit.tree)
```



```
#The rpart.plot function accepts a numeric type argument that creates different styles o
f plots.
rpart.plot(fit.tree, type = 1)
```



```
#make some predictions and compute a train MSE
y_hat.tree <- predict(fit.tree, dd_train)
mse.tree <- mean((y_hat.tree - y_train) ^ 2)
print(mse.tree)
```

```
## [1] 0.2535446
```

```
y_hat_test <-predict(fit.tree, dd_test)
mse_test_tree <- mean((y_hat_test - y_test) ^ 2)
print(mse_test_tree)
```

```
## [1] 0.2565106
```

```
## Bagging
#Download the dataset Final_num_version.csv and load it into R
dd <- read.csv("/Users/tommy/Downloads/Fall\ 2021/Final_num_version.csv")
#Basic summary stats for data column
str(dd)
```

```
## 'data.frame':    73923 obs. of  118 variables:
##  $ X                                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ log_price                        : num  5.01 5.13 4.98 6.62 4.74 ...
##  $ accommodates                     : int  3 7 5 4 2 2 3 2 2 2 ...
##  $ bathrooms                        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ first_review                     : int  618 205 302 0 1021 183 353 437 744 329
...
##  $ last_review                      : int  588 156 165 0 400 174 311 320 155 316
...
##  $ latitude                         : num  40.7 40.8 40.8 37.8 38.9 ...
##  $ longitude                        : num  -74 -74 -73.9 -122.4 -77 ...
##  $ number_of_reviews                : int  2 6 10 0 4 3 15 9 159 2 ...
##  $ review_scores_rating             : num  100 100 100 94.1 40 ...
##  $ host_has_profile_pic_t           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ host_identity_verified_t         : int  1 0 1 1 1 1 0 1 0 0 ...
##  $ instant_bookable_t               : int  0 1 1 0 1 1 1 0 0 1 ...
##  $ thumbnail_url_True               : int  1 1 1 1 0 1 1 1 1 1 ...
##  $ amenities_Air.conditioning       : int  1 1 1 0 1 0 1 0 0 1 ...
##  $ amenities_Bath.towel             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Bathtub                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Coffee.maker           : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Cooking.basics         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Dishes.and.silverware  : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Elevator               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Hot.water              : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Internet               : int  0 0 1 1 1 0 1 0 0 0 ...
##  $ amenities_Kitchen                : int  1 1 1 1 1 0 1 1 0 1 ...
##  $ amenities_Private.bathroom       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Refrigerator           : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Self.Check.In          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Stove                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Toilet.paper           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Apartment          : int  1 1 1 0 1 1 1 0 0 0 ...
##  $ property_type_Bed...Breakfast    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boat               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boutique.hotel     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Bungalow           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cabin              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Camper.RV          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Casa.particular    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Castle             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cave               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Chalet             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Condominium        : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ property_type_Dorm               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Earth.House        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guest.suite        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guesthouse         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Hostel             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_House              : int  0 0 0 1 0 0 0 0 1 1 ...
##  $ property_type_Hut                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_In.law             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Island             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Lighthouse          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Loft                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Other               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Parking.Space       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Serviced.apartment  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Tent                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Timeshare           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Tipi                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Townhouse           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Train               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Treehouse           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Vacation.home       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Villa               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Yurt                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ room_type_Entire.home.apt         : int  1 1 1 1 1 0 1 1 0 0 ...
##  $ room_type_Private.room            : int  0 0 0 0 0 1 0 0 1 1 ...
##  $ room_type_Shared.room             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Airbed                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Couch                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Futon                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Pull.out.Sofa            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bed_type_Real.Bed                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cleaning_fee_False                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cleaning_fee_True                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cancellation_policy_flexible      : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ cancellation_policy_moderate      : int  0 0 1 0 1 0 1 1 1 1 ...
##  $ cancellation_policy_strict        : int  1 1 0 0 0 1 0 0 0 0 ...
##  $ cancellation_policy_super_strict_30: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cancellation_policy_super_strict_60: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_0.0                      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ bedrooms_1.0                      : int  1 0 1 0 0 1 1 1 1 1 ...
##  $ bedrooms_1.23526268079176         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_2.0                      : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ bedrooms_3.0                      : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_4.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_5.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_6.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_7.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_8.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_9.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedrooms_10.0                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_0.0                          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_1.0                          : int  1 0 0 0 1 1 1 1 1 1 ...
##  $ beds_1.23526268079176             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_2.0                          : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ beds_3.0                          : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ beds_4.0                          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_5.0                          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ beds_6.0                          : int  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
```

```
#Delete the first column in dd
dd = subset(dd, select = -c(X))
set.seed(810)
#take 30% random rows and stick them in the test set
test_index <- sample(nrow(dd), nrow(dd) * 0.3)
dd_test <- dd[test_index,]
dd_train <- dd[-test_index,]
x_test <- dd_test[,-1]
x_train <- dd_train[,-1]
y_test <- dd_test$log_price
y_train <- dd_train$log_price

#Bagging model
fit.bagging <- bagging(
formula = y_train ~ .,
data = x_train,
nbagg =100,
coob =TRUE,
control = rpart.control(minsplit =2, cp =0.01)
)
print(fit.bagging)
```

```
##
## Bagging regression trees with 100 bootstrap replications
##
## Call: bagging.data.frame(formula = y_train ~ ., data = x_train, nbagg = 100,
##     coob = TRUE, control = rpart.control(minsplit = 2, cp = 0.01))
##
## Out-of-bag estimate of root mean squared error:  0.5017
```

```
# now let's make predictions
y_hat.bagging <- predict(fit.bagging, x_train)
# calculate the mse
mse_train.bagging <- mean((y_hat.bagging - y_train) ^2)
print(mse_train.bagging)
```

```
## [1] 0.2510397
```

```
y_hat_test.bagging <-predict(fit.bagging, x_test)
mse_test.bagging <- mean((y_hat_test.bagging - y_test) ^2)
print(mse_test.bagging)
```

```
## [1] 0.2535688
```

```
#Boosting
dd <- fread("/Users/tommy/Downloads/Fall\ 2021/Final_num_version.csv", stringsAsFactors
 = T)

set.seed(810)
nrow(dd)
```

```
## [1] 73923
```

```
dd[, test:=0]
dd[sample(nrow(dd), 22176), test:=1]
# take 100K random rows and stick them in the test set
# now split
dd.test <- dd[test==1]
dd.train <- dd[test==0]

dd.train.sample.size <- 51747
dd.train.sample <- dd.train[sample(nrow(dd.train), dd.train.sample.size)]

f1 <- as.formula(log_price ~ .-V1)

# the [, -1] means take all columns of the matrix except the first column,
# which is an intercept added by default
x1.train.sample <- model.matrix(f1, dd.train.sample)[, -1]
# and this the response
y.train <- dd.train$log_price
y.train.sample <- dd.train.sample$log_price

x1.test <- model.matrix(f1, dd.test)[, -1]
y.test <- dd.test$log_price

fit.btree <- gbm(
  f1,
  data = dd.train.sample,
  distribution = "gaussian",
  n.trees = 40,
  interaction.depth = 20,
  shrinkage = 0.001,
  cv.folds = 10)
```

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 49: property_type_Lighthouse has no variation.
```

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 56: property_type_Tipi has no variation.
```

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 117: test has no variation.
```

```
relative.influence(fit.btree)
```

```
## n.trees not given. Using 40 trees.
##                        accommodates                           bathrooms
##                          7278.11804                         56229.86169
##                        first_review                         last_review
##                           437.64203                          8759.61406
##                            latitude                           longitude
##                          4551.16571                         16548.91059
##                   number_of_reviews                review_scores_rating
##                             0.00000                             0.00000
##              host_has_profile_pic_t            host_identity_verified_t
##                             0.00000                             0.00000
##                  instant_bookable_t                  thumbnail_url_True
##                             0.00000                             0.00000
##        `amenities_Air conditioning`              `amenities_Bath towel`
##                             0.00000                             0.00000
##                   amenities_Bathtub            `amenities_Coffee maker`
##                             0.00000                             0.00000
##          `amenities_Cooking basics`   `amenities_Dishes and silverware`
##                             0.00000                             0.00000
##                  amenities_Elevator               `amenities_Hot water`
##                           131.16900                             0.00000
##                  amenities_Internet                   amenities_Kitchen
##                             0.00000                             0.00000
##        `amenities_Private bathroom`              amenities_Refrigerator
##                             0.00000                             0.00000
##           `amenities_Self Check-In`                     amenities_Stove
##                             0.00000                             0.00000
##            `amenities_Toilet paper`             property_type_Apartment
##                             0.00000                             0.00000
##     `property_type_Bed & Breakfast`                  property_type_Boat
##                             0.00000                             0.00000
##      `property_type_Boutique hotel`              property_type_Bungalow
##                             0.00000                             0.00000
##                 property_type_Cabin           `property_type_Camper/RV`
##                             0.00000                             0.00000
##     `property_type_Casa particular`                property_type_Castle
##                             0.00000                             0.00000
##                  property_type_Cave                property_type_Chalet
##                             0.00000                             0.00000
##           property_type_Condominium                  property_type_Dorm
##                             0.00000                             0.00000
##         `property_type_Earth House`         `property_type_Guest suite`
##                             0.00000                             0.00000
##            property_type_Guesthouse                property_type_Hostel
##                             0.00000                             0.00000
##                 property_type_House                   property_type_Hut
##                            61.95589                             0.00000
##              `property_type_In-law`                property_type_Island
##                             0.00000                             0.00000
##            property_type_Lighthouse                  property_type_Loft
##                             0.00000                             0.00000
##                 property_type_Other       `property_type_Parking Space`
##                             0.00000                             0.00000
##  `property_type_Serviced apartment`                  property_type_Tent
##                          0.00000                          0.00000
##          property_type_Timeshare               property_type_Tipi
##                          0.00000                          0.00000
##          property_type_Townhouse              property_type_Train
##                          0.00000                          0.00000
##          property_type_Treehouse      `property_type_Vacation home`
##                          0.00000                          0.00000
##              property_type_Villa               property_type_Yurt
##                          0.00000                          0.00000
##        `room_type_Entire home/apt`          `room_type_Private room`
##                     186640.76353                       3473.58709
##           `room_type_Shared room`                  bed_type_Airbed
##                       1893.50080                          0.00000
##                   bed_type_Couch                   bed_type_Futon
##                          0.00000                          0.00000
##           `bed_type_Pull-out Sofa`               `bed_type_Real Bed`
##                          0.00000                          0.00000
##               cleaning_fee_False                cleaning_fee_True
##                          0.00000                          0.00000
##     cancellation_policy_flexible     cancellation_policy_moderate
##                          0.00000                          0.00000
##       cancellation_policy_strict cancellation_policy_super_strict_30
##                          0.00000                          0.00000
## cancellation_policy_super_strict_60                    bedrooms_0.0
##                          0.00000                          0.00000
##                     bedrooms_1.0       bedrooms_1.23526268079176
##                          0.00000                          0.00000
##                     bedrooms_2.0                     bedrooms_3.0
##                        257.83251                          0.00000
##                     bedrooms_4.0                     bedrooms_5.0
##                          0.00000                          0.00000
##                     bedrooms_6.0                     bedrooms_7.0
##                          0.00000                          0.00000
##                     bedrooms_8.0                     bedrooms_9.0
##                          0.00000                          0.00000
##                    bedrooms_10.0                         beds_0.0
##                          0.00000                          0.00000
##                         beds_1.0           beds_1.23526268079176
##                          0.00000                          0.00000
##                         beds_2.0                         beds_3.0
##                          0.00000                          0.00000
##                         beds_4.0                         beds_5.0
##                          0.00000                          0.00000
##                         beds_6.0                         beds_7.0
##                          0.00000                          0.00000
##                         beds_8.0                         beds_9.0
##                          0.00000                          0.00000
##                        beds_10.0                        beds_11.0
##                          0.00000                          0.00000
##                        beds_12.0                        beds_13.0
##                          0.00000                          0.00000
##                        beds_14.0                        beds_15.0
##                          0.00000                          0.00000
##                        beds_16.0                        beds_18.0
##                          0.00000                        0.00000
##                      city_Boston                   city_Chicago
##                          0.00000                       41.78949
##                          city_DC                        city_LA
##                       1049.00174                        0.00000
##                         city_NYC                        city_SF
##                          0.00000                        0.00000
##                 host_since_days_n          host_response_rate_num
##                          0.00000                      644.46148
##                             test
##                          0.00000
```

```
# mse_train: 0.471164
yhat.btree <- predict(fit.btree, dd.train)
```

```
## Using 40 trees...
```

```
mse.btree <- mean((yhat.btree - y.train) ^ 2)
print(mse.btree)
```

```
## [1] 0.4915582
```

```
#mse_test: 0.4751472
yhat.test.btree <- predict(fit.btree, dd.test[,-1])
```

```
## Using 40 trees...
```

```
mse.test.btree <- mean((yhat.btree - dd.test$log_price) ^ 2)
```

```
## Warning in yhat.btree - dd.test$log_price: longer object length is not a
## multiple of shorter object length
```

```
print(mse.test.btree)
```

```
## [1] 0.5187581
```

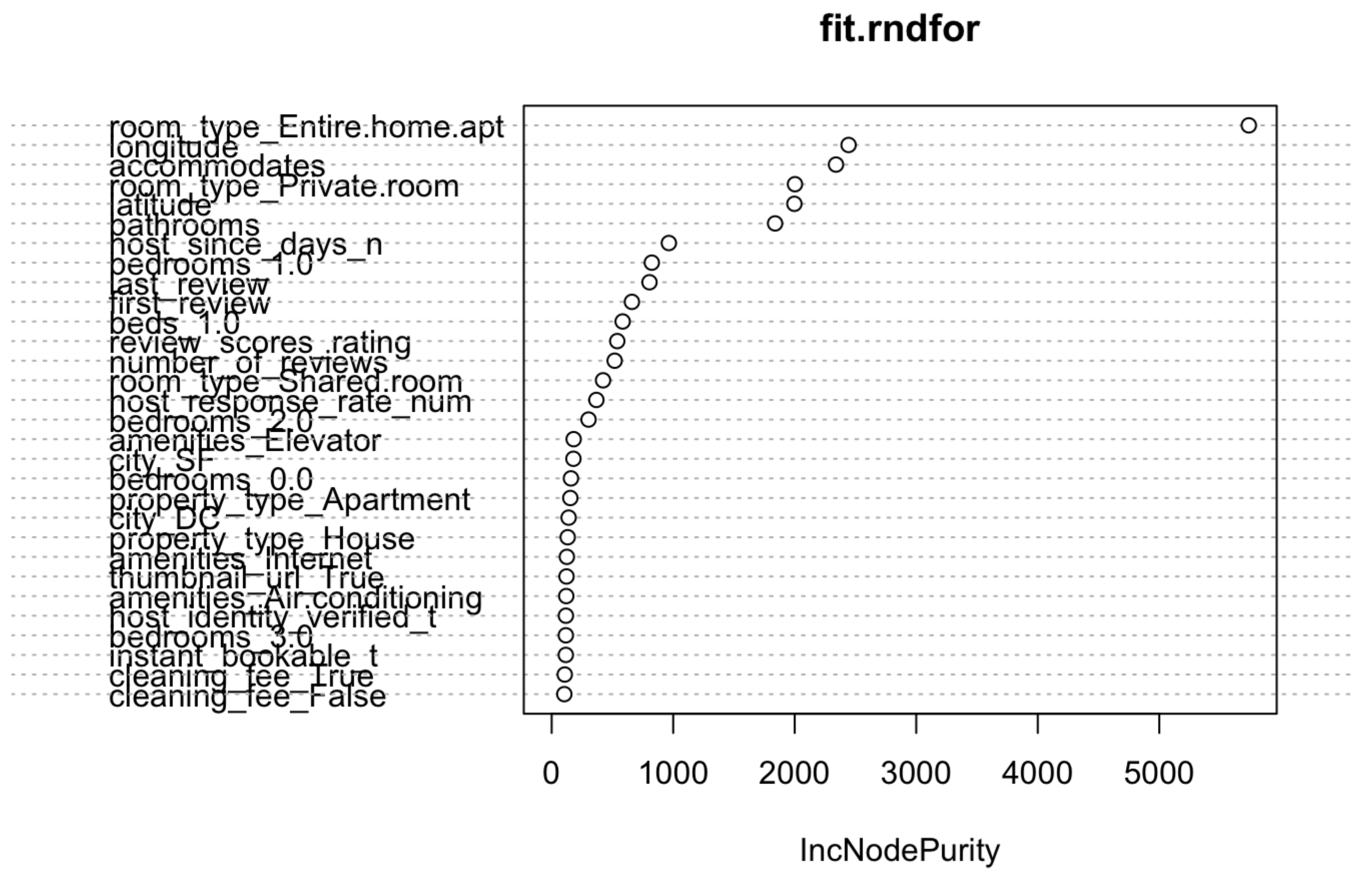
```
##Random Forest
#Download the dataset Final_num_version.csv and load it into R
dd <- read.csv("/Users/tommy/Downloads/Fall\ 2021/Final_num_version.csv")

#Basic summary stats for data column
str(dd)
```

```
## 'data.frame':    73923 obs. of  118 variables:
##  $ X                                 : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ log_price                         : num  5.01 5.13 4.98 6.62 4.74 ...
##  $ accommodates                      : int  3 7 5 4 2 2 3 2 2 2 ...
##  $ bathrooms                         : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ first_review                      : int  618 205 302 0 1021 183 353 437 744 329
...
##  $ last_review                       : int  588 156 165 0 400 174 311 320 155 316
...
##  $ latitude                          : num  40.7 40.8 40.8 37.8 38.9 ...
##  $ longitude                         : num  -74 -74 -73.9 -122.4 -77 ...
##  $ number_of_reviews                 : int  2 6 10 0 4 3 15 9 159 2 ...
##  $ review_scores_rating              : num  100 100 100 94.1 40 ...
##  $ host_has_profile_pic_t            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ host_identity_verified_t          : int  1 0 1 1 1 1 0 1 0 0 ...
##  $ instant_bookable_t                : int  0 1 1 0 1 1 1 0 0 1 ...
##  $ thumbnail_url_True                : int  1 1 1 1 0 1 1 1 1 1 ...
##  $ amenities_Air.conditioning        : int  1 1 1 0 1 0 1 0 0 1 ...
##  $ amenities_Bath.towel              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Bathtub                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Coffee.maker            : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Cooking.basics          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Dishes.and.silverware   : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Elevator                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Hot.water               : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Internet                : int  0 0 0 1 1 0 1 0 0 0 ...
##  $ amenities_Kitchen                 : int  1 1 1 1 1 0 1 1 0 1 ...
##  $ amenities_Private.bathroom        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Refrigerator            : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ amenities_Self.Check.In           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Stove                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities_Toilet.paper            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Apartment           : int  1 1 1 0 1 1 1 0 0 0 ...
##  $ property_type_Bed...Breakfast     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boat                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Boutique.hotel      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Bungalow            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cabin               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Camper.RV           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Casa.particular     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Castle              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Cave                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Chalet              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Condominium         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ property_type_Dorm                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Earth.House         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guest.suite         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Guesthouse          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Hostel              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_House               : int  0 0 0 1 0 0 0 0 1 1 ...
##  $ property_type_Hut                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_In.law              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ property_type_Island              : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Lighthouse           : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Loft                 : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Other                : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Parking.Space        : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Serviced.apartment   : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Tent                 : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Timeshare            : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Tipi                 : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Townhouse            : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Train                : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Treehouse            : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Vacation.home        : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Villa                : int  0 0 0 0 0 0 0 0 0 0 ...
## $ property_type_Yurt                 : int  0 0 0 0 0 0 0 0 0 0 ...
## $ room_type_Entire.home.apt          : int  1 1 1 1 1 0 1 1 0 0 ...
## $ room_type_Private.room             : int  0 0 0 0 0 1 0 0 1 1 ...
## $ room_type_Shared.room              : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Airbed                    : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Couch                     : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Futon                     : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Pull.out.Sofa             : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bed_type_Real.Bed                  : int  1 1 1 1 1 1 1 1 1 1 ...
## $ cleaning_fee_False                 : int  0 0 0 0 0 0 0 0 0 0 ...
## $ cleaning_fee_True                  : int  1 1 1 1 1 1 1 1 1 1 ...
## $ cancellation_policy_flexible       : int  0 0 0 1 0 0 0 0 0 0 ...
## $ cancellation_policy_moderate       : int  0 0 1 0 1 0 1 1 1 1 ...
## $ cancellation_policy_strict         : int  1 1 0 0 0 1 0 0 0 0 ...
## $ cancellation_policy_super_strict_30: int  0 0 0 0 0 0 0 0 0 0 ...
## $ cancellation_policy_super_strict_60: int  0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_0.0                       : int  0 0 0 0 1 0 0 0 0 0 ...
## $ bedrooms_1.0                       : int  1 0 1 0 0 1 1 1 1 1 ...
## $ bedrooms_1.23526268079176          : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_2.0                       : int  0 0 0 1 0 0 0 0 0 0 ...
## $ bedrooms_3.0                       : int  0 1 0 0 0 0 0 0 0 0 ...
## $ bedrooms_4.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_5.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_6.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_7.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_8.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_9.0                       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ bedrooms_10.0                      : int  0 0 0 0 0 0 0 0 0 0 ...
## $ beds_0.0                           : int  0 0 0 0 0 0 0 0 0 0 ...
## $ beds_1.0                           : int  1 0 0 0 1 1 1 1 1 1 ...
## $ beds_1.23526268079176              : int  0 0 0 0 0 0 0 0 0 0 ...
## $ beds_2.0                           : int  0 0 0 1 0 0 0 0 0 0 ...
## $ beds_3.0                           : int  0 1 1 0 0 0 0 0 0 0 ...
## $ beds_4.0                           : int  0 0 0 0 0 0 0 0 0 0 ...
## $ beds_5.0                           : int  0 0 0 0 0 0 0 0 0 0 ...
## $ beds_6.0                           : int  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
```

```
#Delete the first column in dd
dd = subset(dd, select = -c(X))
set.seed(810)
#take 30% random rows and stick them in the test set
test_index <- sample(nrow(dd), nrow(dd) * 0.3)
dd_test <- dd[test_index,]
dd_train <- dd[-test_index,]
x_test <- dd_test[,-1]
x_train <- dd_train[,-1]
y_test <- dd_test$log_price
y_train <- dd_train$log_price

##Random Forest model
fit.rndfor <- randomForest(y_train~., x_train, ntree=100,do.trace=T)
```

```
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##    1 |   0.2995    58.30 |
##    2 |   0.2859    55.65 |
##    3 |   0.2711    52.77 |
##    4 |   0.2572    50.07 |
##    5 |    0.245    47.70 |
##    6 |   0.2321    45.18 |
##    7 |   0.2228    43.37 |
##    8 |   0.2153    41.91 |
##    9 |   0.2077    40.44 |
##   10 |   0.2013    39.18 |
##   11 |   0.1962    38.19 |
##   12 |    0.192    37.38 |
##   13 |   0.1878    36.56 |
##   14 |   0.1842    35.85 |
##   15 |   0.1816    35.35 |
##   16 |   0.1796    34.95 |
##   17 |   0.1772    34.50 |
##   18 |   0.1756    34.18 |
##   19 |   0.1736    33.79 |
##   20 |   0.1719    33.47 |
##   21 |    0.171    33.28 |
##   22 |   0.1698    33.06 |
##   23 |   0.1687    32.83 |
##   24 |   0.1676    32.63 |
##   25 |   0.1667    32.46 |
##   26 |   0.1659    32.29 |
##   27 |   0.1651    32.14 |
##   28 |   0.1644    32.00 |
##   29 |   0.1639    31.89 |
##   30 |   0.1632    31.76 |
##   31 |   0.1626    31.65 |
##   32 |   0.1622    31.58 |
##   33 |   0.1619    31.52 |
##   34 |   0.1616    31.46 |
##   35 |   0.1613    31.39 |
##   36 |   0.1609    31.33 |
##   37 |   0.1608    31.31 |
##   38 |   0.1604    31.22 |
##   39 |   0.1601    31.16 |
##   40 |   0.1598    31.11 |
##   41 |   0.1597    31.08 |
##   42 |   0.1595    31.04 |
##   43 |   0.1591    30.97 |
##   44 |   0.1589    30.93 |
##   45 |   0.1586    30.87 |
##   46 |   0.1584    30.83 |
##   47 |   0.1583    30.81 |
##   48 |   0.1581    30.77 |
##   49 |    0.158    30.75 |
##   50 |   0.1577    30.70 |
##   51 |   0.1575    30.67 |
##   52 |   0.1574    30.63 |
##   53 |   0.1572    30.61 |
##   54 |   0.1571    30.58 |
##   55 |   0.1569    30.54 |
##   56 |   0.1568    30.51 |
##   57 |   0.1567    30.50 |
##   58 |   0.1566    30.49 |
##   59 |   0.1565    30.46 |
##   60 |   0.1564    30.45 |
##   61 |   0.1563    30.42 |
##   62 |   0.1561    30.39 |
##   63 |    0.156    30.36 |
##   64 |   0.1558    30.33 |
##   65 |   0.1557    30.32 |
##   66 |   0.1557    30.30 |
##   67 |   0.1556    30.28 |
##   68 |   0.1554    30.25 |
##   69 |   0.1553    30.24 |
##   70 |   0.1553    30.23 |
##   71 |   0.1552    30.21 |
##   72 |   0.1551    30.18 |
##   73 |    0.155    30.18 |
##   74 |    0.155    30.16 |
##   75 |   0.1548    30.14 |
##   76 |   0.1548    30.12 |
##   77 |   0.1546    30.10 |
##   78 |   0.1546    30.10 |
##   79 |   0.1546    30.09 |
##   80 |   0.1545    30.07 |
##   81 |   0.1545    30.07 |
##   82 |   0.1543    30.03 |
##   83 |   0.1542    30.02 |
##   84 |   0.1542    30.01 |
##   85 |   0.1541    30.00 |
##   86 |   0.1541    29.99 |
##   87 |    0.154    29.97 |
##   88 |   0.1539    29.95 |
##   89 |   0.1539    29.95 |
##   90 |   0.1538    29.94 |
##   91 |   0.1538    29.93 |
##   92 |   0.1537    29.92 |
##   93 |   0.1537    29.91 |
##   94 |   0.1536    29.90 |
##   95 |   0.1535    29.89 |
##   96 |   0.1535    29.88 |
##   97 |   0.1535    29.87 |
##   98 |   0.1534    29.86 |
##   99 |   0.1534    29.86 |
##  100 |   0.1533    29.85 |
```

```
#We can check which variables are most predictice using a variable importtance plot
varImpPlot(fit.rndfor)
```



```
#Make predictions
y_hat.rndfor <- predict(fit.rndfor, x_train)
mse_train.tree <- mean((y_hat.rndfor - y_train) ^2)
print(mse_train.tree)
```

```
## [1] 0.03051325
```

```
y_hat_test.rndfor <-predict(fit.rndfor, x_test)
mse_test.tree <- mean((y_hat_test.rndfor - y_test) ^2)
print(mse_test.tree)
```

```
## [1] 0.1541398
```