Stat 330 Final

Cole Herndon

November 19, 2019

link to our data set <https://www.kaggle.com/jojoker/singapore-airbnb>

library(tidyverse)

## -- Attaching packages --------------------------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggplot2)  
library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

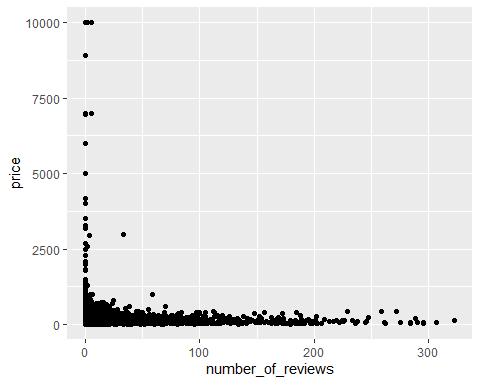
## The following object is masked from 'package:purrr':  
##   
## some

library(ggfortify)  
  
sing\_abb\_all <- read.csv("C:\\Users\\coleh\\OneDrive\\Documents\\BYU Fall 2019\\Stat 330\\Final\\singapore-airbnb\\listings.csv")

sing <- sing\_abb\_all %>%   
 select(id, host\_id, neighbourhood, neighbourhood\_group, room\_type, price, number\_of\_reviews, reviews\_per\_month)  
sing <- sing[-c(3169),]

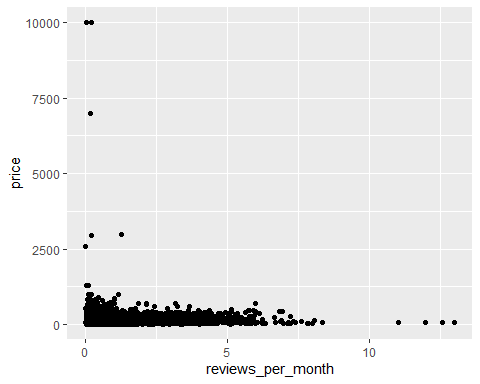
highest\_price <- sing %>%   
 group\_by(neighbourhood) %>%   
 summarise(Avg\_price = mean(price)) %>%   
 arrange(desc(Avg\_price))  
  
price\_group <- sing\_abb\_all %>%   
 group\_by(neighbourhood\_group) %>%   
 summarise(Avg\_price = mean(price), SD = sd(price)) %>%   
 arrange(desc(Avg\_price))  
  
percent\_group <- sing\_abb\_all %>%   
 count(neighbourhood\_group) %>%   
 mutate(Percent = round(n/length(sing\_abb\_all$id), 2))  
  
count\_room\_type <- sing\_abb\_all %>%   
 count(room\_type)

price\_reviews\_plot <- ggplot(data = sing, mapping = aes(y = price, x = number\_of\_reviews)) + geom\_point()   
price\_reviews\_plot

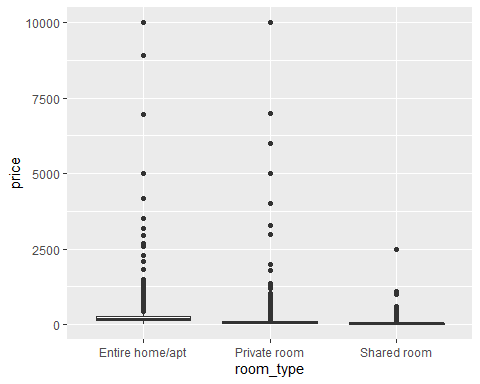


price\_reviews\_per\_month\_plot <- ggplot(data = sing, mapping = aes(y = price, x = reviews\_per\_month)) + geom\_point()   
price\_reviews\_per\_month\_plot

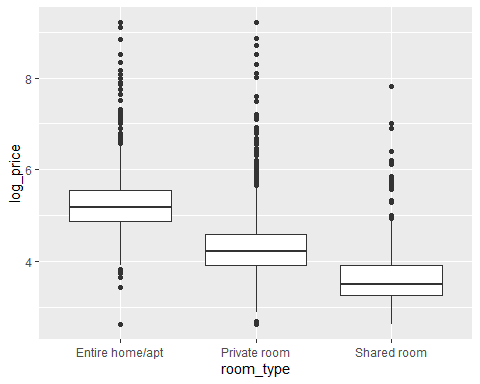
## Warning: Removed 2758 rows containing missing values (geom\_point).



price\_room\_type\_box <- ggplot(data = sing, mapping = aes(y = price, x = room\_type)) + geom\_boxplot()  
price\_room\_type\_box



sing$log\_price <- log(sing$price)  
  
price\_room\_type\_box <- ggplot(data = sing, mapping = aes(y = log\_price, x = room\_type)) + geom\_boxplot()  
price\_room\_type\_box

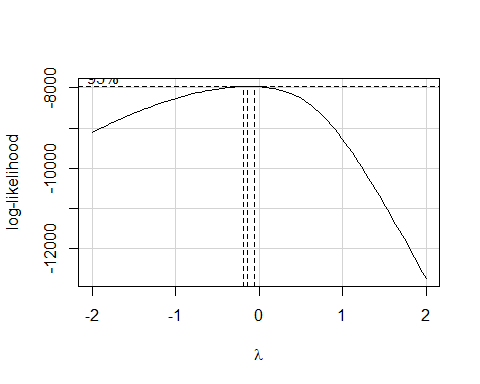


sing$room\_private <- ifelse(sing$room\_type == "Private room", 1, 0)  
sing$room\_entire <- ifelse(sing$room\_type == "Entire home/apt", 1, 0)  
sing$room\_shared <- ifelse(sing$room\_type == "Shared room", 1, 0)

sing.lm <- lm(data = sing, formula = price ~ number\_of\_reviews + room\_entire + room\_shared)  
summary(sing.lm)

##   
## Call:  
## lm(formula = price ~ number\_of\_reviews + room\_entire + room\_shared,   
## data = sing)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -218.0 -76.0 -41.9 9.7 9883.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 116.9195 6.0051 19.47 < 2e-16 \*\*\*  
## number\_of\_reviews -0.4395 0.1267 -3.47 0.000524 \*\*\*  
## room\_entire 115.4884 7.7591 14.88 < 2e-16 \*\*\*  
## room\_shared -45.7678 17.8062 -2.57 0.010178 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 334.5 on 7902 degrees of freedom  
## Multiple R-squared: 0.03374, Adjusted R-squared: 0.03338   
## F-statistic: 91.98 on 3 and 7902 DF, p-value: < 2.2e-16

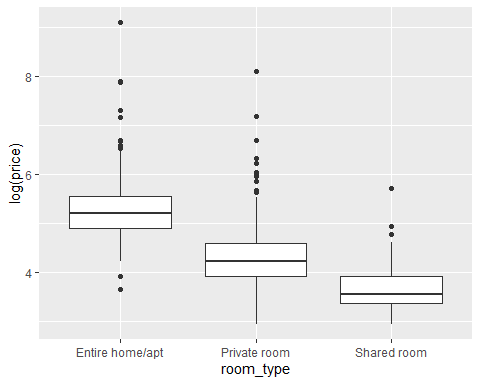
set.seed(42)  
random\_sample\_sz <- 1000  
sing\_no\_na <- sing\_abb\_all  
  
sing\_no\_na$reviews\_per\_month <- ifelse(is.na(sing\_no\_na$reviews\_per\_month) == TRUE, 0, sing\_no\_na$reviews\_per\_month )  
sing\_no\_na$price <- ifelse(is.na(sing\_no\_na$price) == TRUE, 0, sing\_no\_na$price)  
sing\_no\_na <- sing\_no\_na[-c(3169),]  
  
sing\_random <- sing\_no\_na[sample(nrow(sing\_no\_na), random\_sample\_sz),]  
  
bc <-boxCox(sing\_random$price ~ sing\_random$reviews\_per\_month)



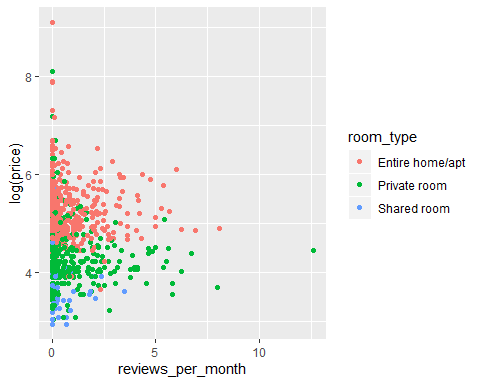
bc$x[which.max(bc$y)]

## [1] -0.1414141

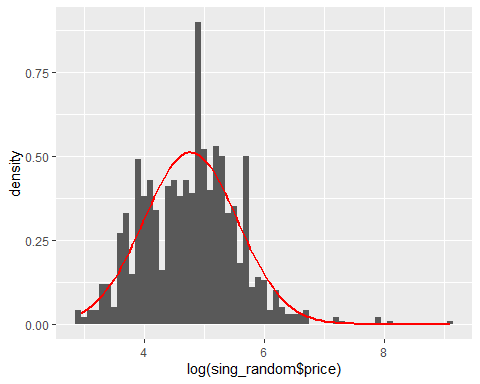
random\_sample\_boxplot <- ggplot(data = sing\_random, mapping = aes(y = log(price), x = room\_type)) + geom\_boxplot()  
random\_sample\_boxplot



r\_s\_plot\_color <- ggplot(data = sing\_random, mapping = aes(y = log(price), x = reviews\_per\_month, color = room\_type)) + geom\_point()  
r\_s\_plot\_color



#frequency plot with log y transformation  
freq\_distribution <- ggplot(data = sing\_random, mapping = aes(x = log(sing\_random$price))) +   
 geom\_histogram(mapping = aes(y = ..density..), binwidth = .1) +  
 stat\_function(fun = dnorm, color = "red", size = 1,  
 args = list(mean = mean(log(sing\_random$price)),   
 sd = sd(log(sing\_random$price))))  
freq\_distribution



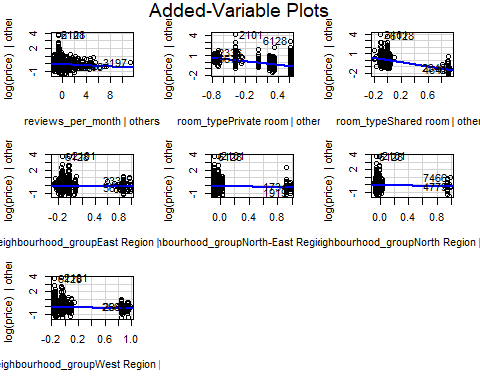
sing\_random\_log\_y.lm <- lm(data = sing\_random, formula = log(price) ~ reviews\_per\_month + room\_type + neighbourhood\_group )  
summary(sing\_random\_log\_y.lm)

##   
## Call:  
## lm(formula = log(price) ~ reviews\_per\_month + room\_type + neighbourhood\_group,   
## data = sing\_random)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.3590 -0.3679 -0.0676 0.2681 3.7974   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.29636 0.02756 192.195 < 2e-16 \*\*\*  
## reviews\_per\_month -0.03375 0.01471 -2.295 0.02194 \*   
## room\_typePrivate room -0.89562 0.03900 -22.962 < 2e-16 \*\*\*  
## room\_typeShared room -1.58075 0.08136 -19.428 < 2e-16 \*\*\*  
## neighbourhood\_groupEast Region -0.09509 0.07236 -1.314 0.18909   
## neighbourhood\_groupNorth-East Region -0.18525 0.09681 -1.914 0.05597 .   
## neighbourhood\_groupNorth Region -0.28828 0.11690 -2.466 0.01383 \*   
## neighbourhood\_groupWest Region -0.18739 0.07185 -2.608 0.00925 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5716 on 992 degrees of freedom  
## Multiple R-squared: 0.4644, Adjusted R-squared: 0.4606   
## F-statistic: 122.9 on 7 and 992 DF, p-value: < 2.2e-16

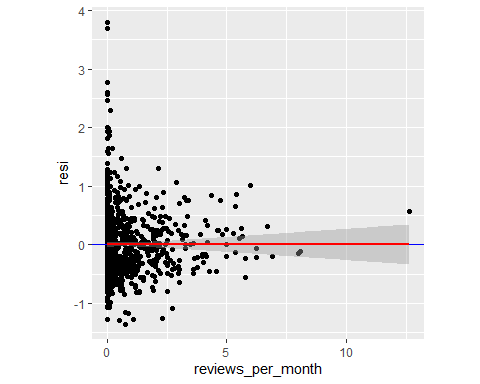
sing\_random.lm <- lm(data = sing\_random, formula = price ~ reviews\_per\_month + room\_type + neighbourhood\_group )  
summary(sing\_random\_log\_y.lm)

##   
## Call:  
## lm(formula = log(price) ~ reviews\_per\_month + room\_type + neighbourhood\_group,   
## data = sing\_random)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.3590 -0.3679 -0.0676 0.2681 3.7974   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.29636 0.02756 192.195 < 2e-16 \*\*\*  
## reviews\_per\_month -0.03375 0.01471 -2.295 0.02194 \*   
## room\_typePrivate room -0.89562 0.03900 -22.962 < 2e-16 \*\*\*  
## room\_typeShared room -1.58075 0.08136 -19.428 < 2e-16 \*\*\*  
## neighbourhood\_groupEast Region -0.09509 0.07236 -1.314 0.18909   
## neighbourhood\_groupNorth-East Region -0.18525 0.09681 -1.914 0.05597 .   
## neighbourhood\_groupNorth Region -0.28828 0.11690 -2.466 0.01383 \*   
## neighbourhood\_groupWest Region -0.18739 0.07185 -2.608 0.00925 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5716 on 992 degrees of freedom  
## Multiple R-squared: 0.4644, Adjusted R-squared: 0.4606   
## F-statistic: 122.9 on 7 and 992 DF, p-value: < 2.2e-16

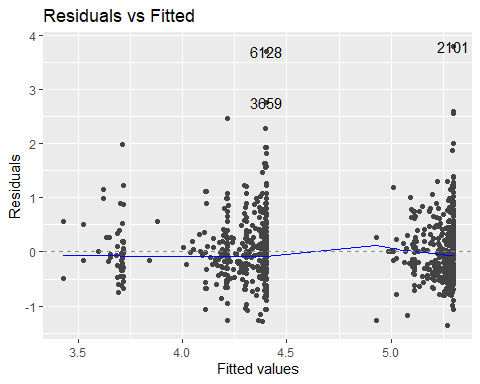
sing\_random$resi <- sing\_random\_log\_y.lm$residuals  
sing\_random$predicted <- predict(sing\_random\_log\_y.lm)  
  
  
#check x vs y is linear  
#partial regression plots  
avPlots(sing\_random\_log\_y.lm)



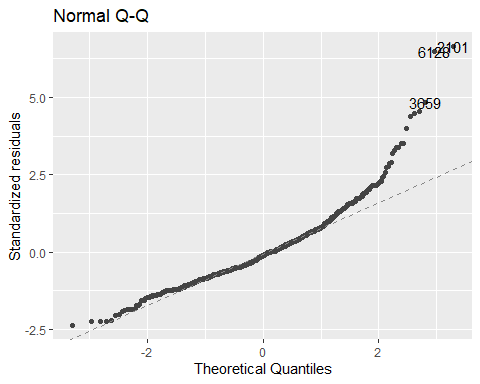
#residuals vs predictor plots  
b\_rpm <- ggplot(sing\_random, aes(x = reviews\_per\_month, y = resi)) +  
 geom\_point() +  
 geom\_hline(yintercept = 0, color = 'blue') +   
 theme(aspect.ratio = 1) +  
 geom\_smooth(method = 'lm', color = 'red')  
  
b\_rpm



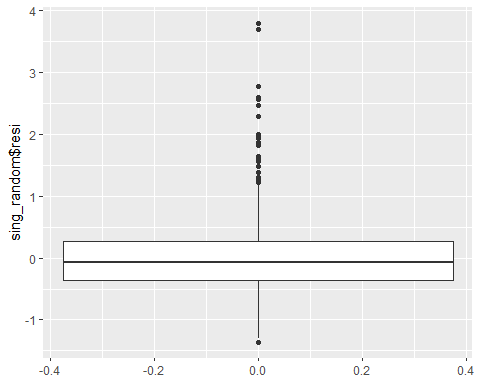
#residuals vs fit values plot  
autoplot(sing\_random\_log\_y.lm, which = 1, ncol = 1, nrow = 1)



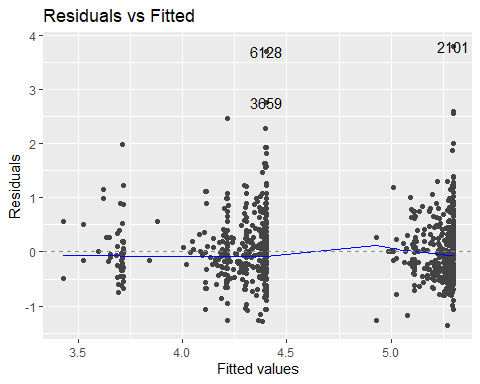
#The residuals are normally distributed and centered at zero  
  
# normal probability plot  
autoplot(sing\_random\_log\_y.lm, which = 2, ncol = 1, nrow = 1)



# shapiro wilk test  
#shapiro.test(sing\_random\_log\_y.lm)  
  
#boxplot  
sing\_log\_y\_boxplot <- ggplot(sing\_random\_log\_y.lm, mapping = aes(y = sing\_random$resi)) + geom\_boxplot()  
sing\_log\_y\_boxplot



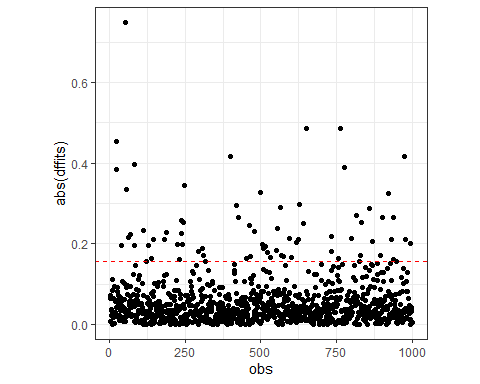
#The residuals are homoscedastic  
  
#residuals vs fit values plot  
autoplot(sing\_random\_log\_y.lm, which = 1, ncol = 1, nrow = 1)



#brown-forsythe test  
grp <- as.factor(c(rep("lower", floor(dim(sing\_random)[1] / 2)),   
 rep("upper", ceiling(dim(sing\_random)[1] / 2))))  
leveneTest(sing\_random$resi ~ grp, center = median)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 1 1.1398 0.286  
## 998

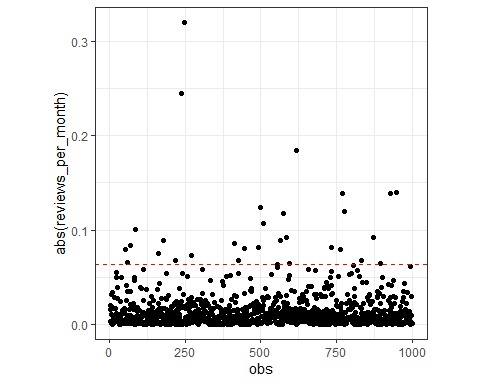
#The model describes all observations - there are no influential points (use the DFBETAS, DFFITS, partial regression plots  
  
# DFFITS  
sing\_random.dffits <- data.frame("dffits" = dffits(sing\_random\_log\_y.lm))  
sing\_random.dffits$obs <- 1:length(sing\_random$price)  
  
ggplot(data = sing\_random.dffits) +  
 geom\_point(mapping = aes(x = obs, y = abs(dffits))) +  
 # geom\_hline(mapping = aes(yintercept = 1),  
 # color = "red", linetype = "dashed") + # for n <= 30  
 geom\_hline(mapping = aes(yintercept = 2 \* sqrt(6 / length(obs))),  
 color = "red", linetype = "dashed") + # for n > 30  
 theme\_bw() +  
 theme(aspect.ratio = 1)



sing\_random.dffits[abs(sing\_random.dffits$dffits) > 1, ]

## [1] dffits obs   
## <0 rows> (or 0-length row.names)

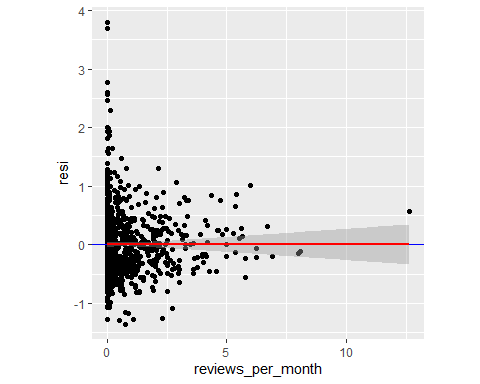
#DFFBETAS  
  
sing\_random.dfbetas <- as.data.frame(dfbetas(sing\_random\_log\_y.lm))  
sing\_random.dfbetas$obs <- 1:length(sing\_random$price)  
# age  
ggplot(data = sing\_random.dfbetas) +  
 geom\_point(mapping = aes(x = obs, y = abs(reviews\_per\_month))) +  
 geom\_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),  
 color = "red", linetype = "dashed") + # for n > 30  
 theme\_bw() +  
 theme(aspect.ratio = 1)



reviews.extreme.dfbetas <- sing\_random.dfbetas[abs(sing\_random.dfbetas$reviews\_per\_month) > .25, ]  
reviews.extreme.dfbetas[order(reviews.extreme.dfbetas$reviews\_per\_month), ]

## (Intercept) reviews\_per\_month room\_typePrivate room room\_typeShared room  
## 3197 -0.1292251 0.320289 0.01790715 0.01603595  
## neighbourhood\_groupEast Region neighbourhood\_groupNorth-East Region  
## 3197 0.08304192 -0.001366735  
## neighbourhood\_groupNorth Region neighbourhood\_groupWest Region obs  
## 3197 -0.01273612 0.01881182 247

#partial pregression  
b\_rpm



#no mulitcollinlearnity  
  
vif(sing\_random\_log\_y.lm)

## GVIF Df GVIF^(1/(2\*Df))  
## reviews\_per\_month 1.021333 1 1.010610  
## room\_type 1.100557 2 1.024243  
## neighbourhood\_group 1.118002 4 1.014041

confint(sing\_random\_log\_y.lm, level = .95)

## 2.5 % 97.5 %  
## (Intercept) 5.24228532 5.350439681  
## reviews\_per\_month -0.06260957 -0.004892606  
## room\_typePrivate room -0.97215909 -0.819078095  
## room\_typeShared room -1.74041693 -1.421084324  
## neighbourhood\_groupEast Region -0.23707729 0.046898853  
## neighbourhood\_groupNorth-East Region -0.37522887 0.004725880  
## neighbourhood\_groupNorth Region -0.51768837 -0.058873721  
## neighbourhood\_groupWest Region -0.32839762 -0.046389428

#adjusted r squared  
summary(sing\_random\_log\_y.lm)

##   
## Call:  
## lm(formula = log(price) ~ reviews\_per\_month + room\_type + neighbourhood\_group,   
## data = sing\_random)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.3590 -0.3679 -0.0676 0.2681 3.7974   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.29636 0.02756 192.195 < 2e-16 \*\*\*  
## reviews\_per\_month -0.03375 0.01471 -2.295 0.02194 \*   
## room\_typePrivate room -0.89562 0.03900 -22.962 < 2e-16 \*\*\*  
## room\_typeShared room -1.58075 0.08136 -19.428 < 2e-16 \*\*\*  
## neighbourhood\_groupEast Region -0.09509 0.07236 -1.314 0.18909   
## neighbourhood\_groupNorth-East Region -0.18525 0.09681 -1.914 0.05597 .   
## neighbourhood\_groupNorth Region -0.28828 0.11690 -2.466 0.01383 \*   
## neighbourhood\_groupWest Region -0.18739 0.07185 -2.608 0.00925 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5716 on 992 degrees of freedom  
## Multiple R-squared: 0.4644, Adjusted R-squared: 0.4606   
## F-statistic: 122.9 on 7 and 992 DF, p-value: < 2.2e-16

no\_group\_analysis <- lm(formula = log(price) ~ reviews\_per\_month + neighbourhood\_group, data = sing\_random)  
summary(no\_group\_analysis)

##   
## Call:  
## lm(formula = log(price) ~ reviews\_per\_month + neighbourhood\_group,   
## data = sing\_random)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.9179 -0.4824 0.0288 0.4769 4.2170   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.87681 0.03004 162.339 < 2e-16 \*\*\*  
## reviews\_per\_month -0.02187 0.01947 -1.123 0.261576   
## neighbourhood\_groupEast Region -0.40545 0.09382 -4.321 1.71e-05 \*\*\*  
## neighbourhood\_groupNorth-East Region -0.44008 0.12747 -3.452 0.000579 \*\*\*  
## neighbourhood\_groupNorth Region -0.64282 0.15403 -4.173 3.27e-05 \*\*\*  
## neighbourhood\_groupWest Region -0.40656 0.09350 -4.348 1.51e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7579 on 994 degrees of freedom  
## Multiple R-squared: 0.05629, Adjusted R-squared: 0.05155   
## F-statistic: 11.86 on 5 and 994 DF, p-value: 3.661e-11