HedonicModel3:SND

luismor

3/26/2021

VARIABLES SELECTION

Loading the data

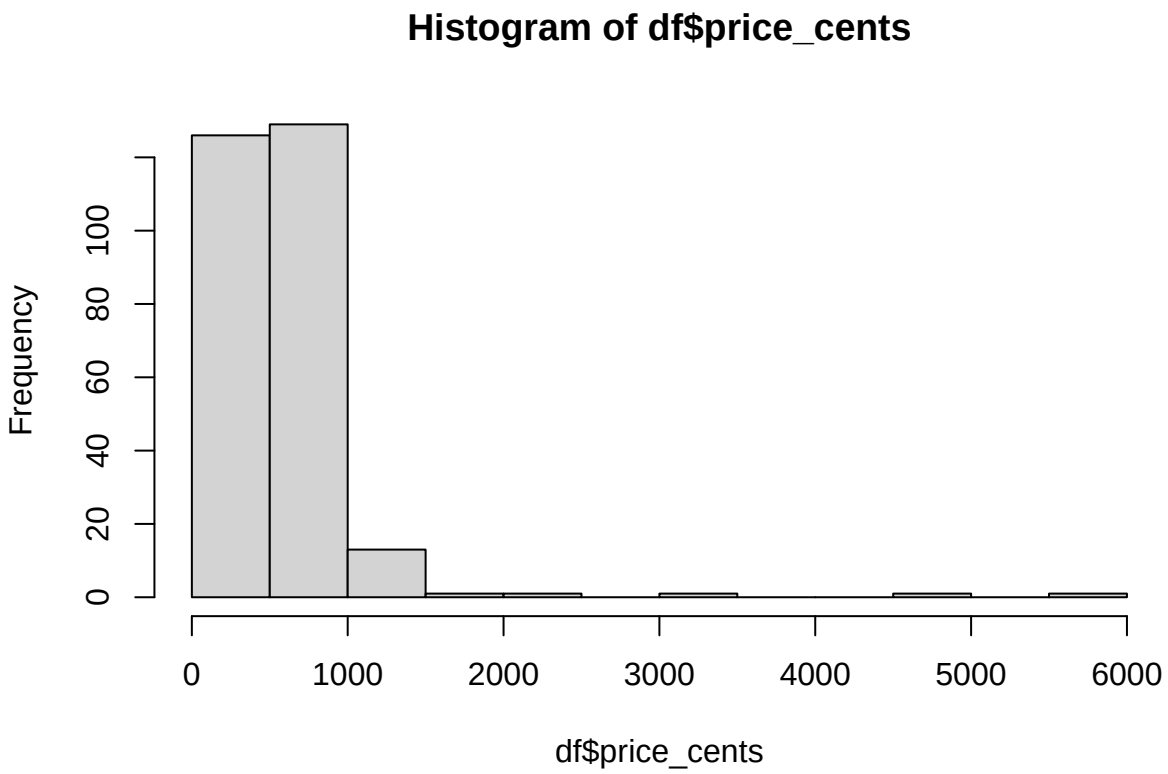
```
library(readxl)
df <- read_excel("/Users/Unimooc/Dropbox/2021/Directorio R/SpaceNextDoor/NextDoor/Optimal pricing/NextD
                 sheet = "Data2")
```

```
head(df)
```

Visualization

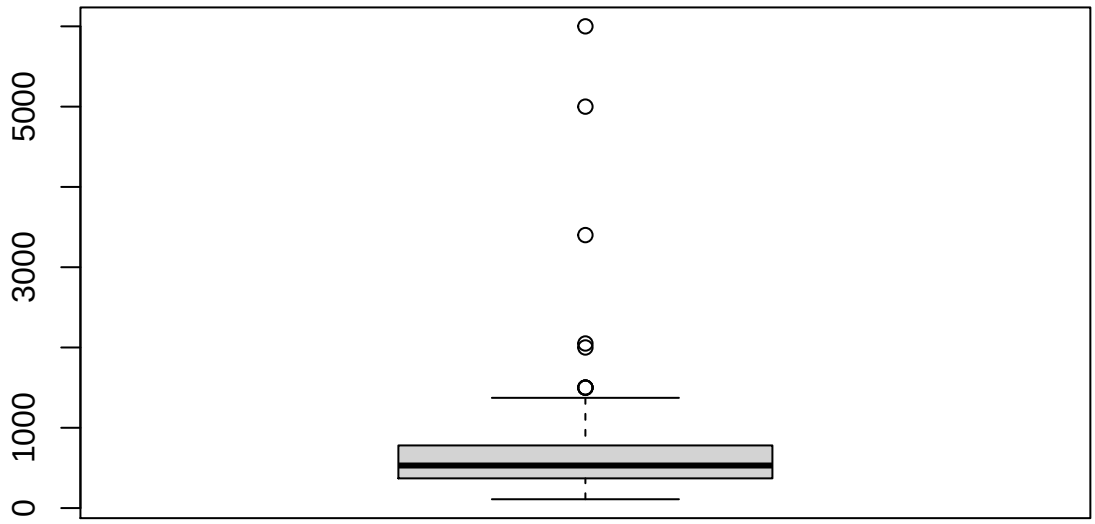
```
library(ggplot2)
library(corrplot)
library(tidyverse)
library(MASS)
```

```
hist(df$price_cents)
```

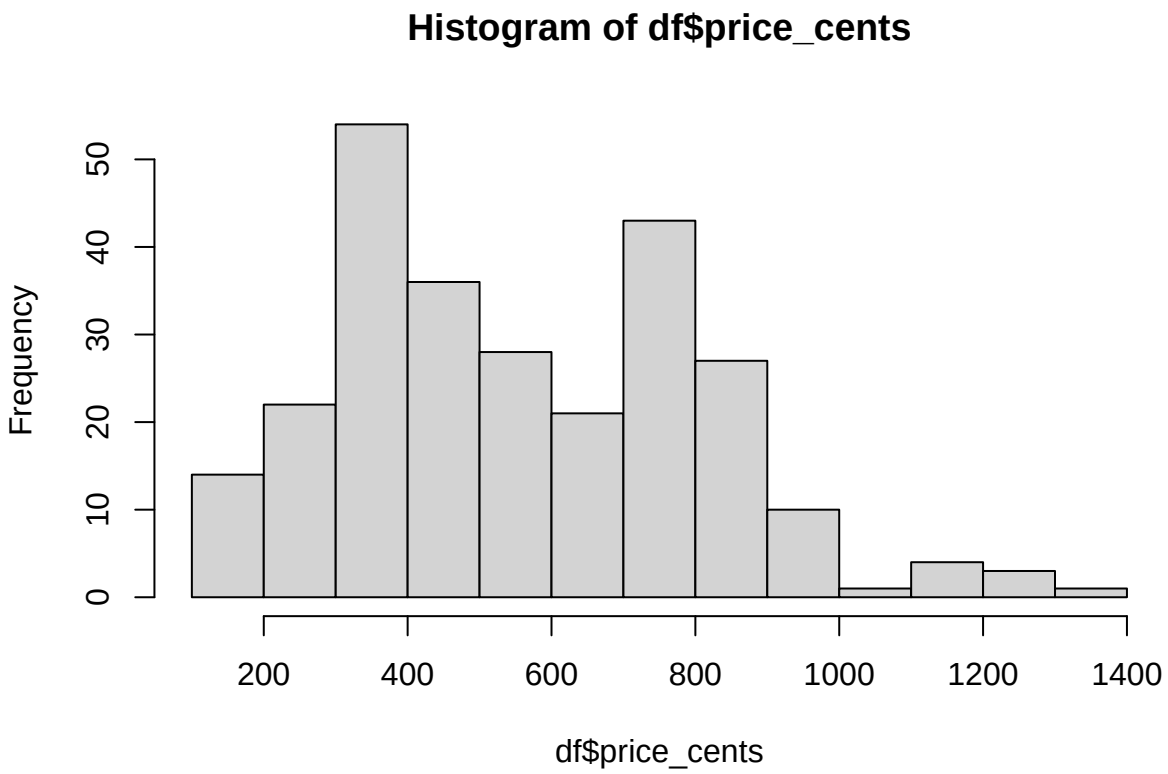



```
#The histogram shows a positive skweness (positioned to the left). There are many outliers that should
```

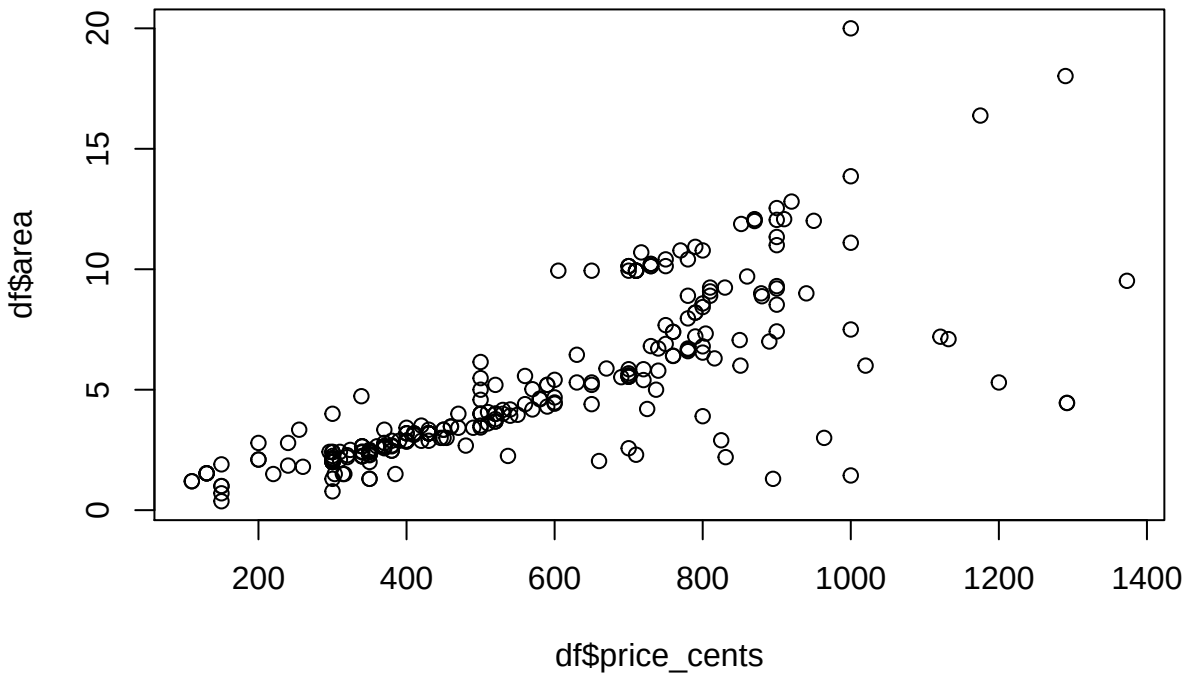
```
gcaja <- boxplot(df$price_cents)
```



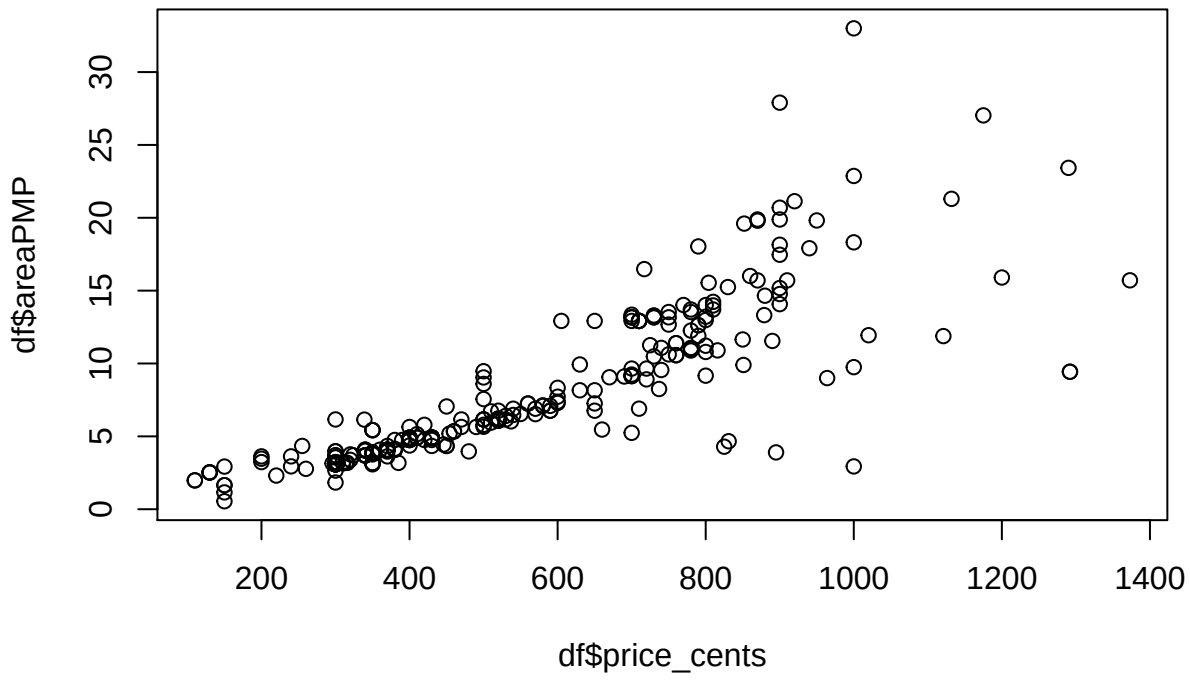

```
df<-df[!(df$price_cents %in% gcaja$out),]
hist(df$price_cents)
```




```
plot(df$price_cents,df$area)
```



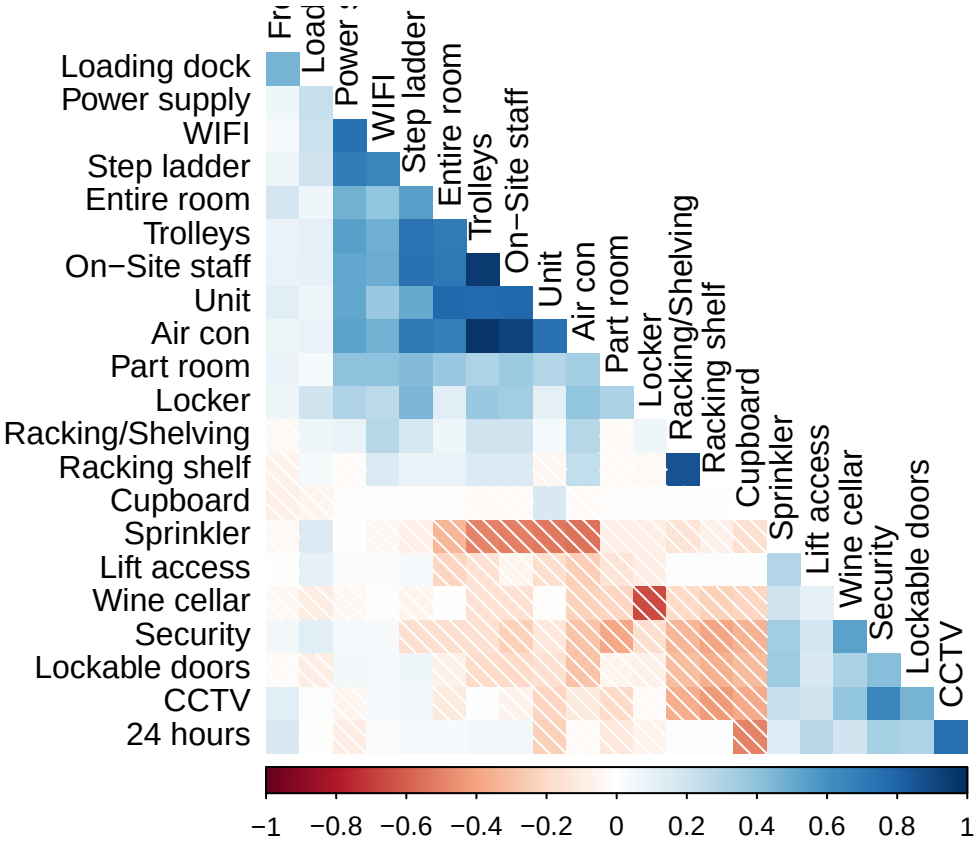

```
plot(df$price_cents,df$areaPMP)
```




```
#Deleting outlyers
```

```
df.cor <- cor(df[,c(8:29)], method = "kendall")
round(df.cor, digits = 1)
```

```
corrplot(df.cor, method ="shade",
         tl.col ="black",
         order = "AOE", type = "lower", diag = F)
```




```
#The correlation matrix indicates the presence of strong autocorrelation between some variables. We sho
```

```
library(PanJen)
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
formBase <- formula(price_cents~ area + areaPMP + PMP, data=df)
summary(gam(formBase, method="GCV.Cp",data=df))
```

```
##
## Family: gaussian
## Link function: identity
```

```
##
## Formula:
## price_cents ~ area + areaPMP + PMP
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -59.593     72.420  -0.823  0.41133
## area          35.856     13.452   2.665  0.00817 **
## areaPMP       16.905      8.384   2.016  0.04479 *
## PMP          186.786     42.456   4.399 1.58e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.708   Deviance explained = 71.2%
## GCV =  18893  Scale est. = 18606     n = 264
```

```
PanJenArea<-fform(data=df,"area",formBase)
```

```
##               AIC     BIC ranking (BIC)
## smoothing 3258.29 3294.28           1.0
## x^2       3275.67 3297.12           2.0
## log(x)    3284.31 3305.77           3.0
## sqr(x)    3294.63 3316.09           4.5
## x+x^2     3294.63 3316.09           4.5
## 1/x       3345.91 3367.37           6.0
## base      3350.62 3368.50           7.5
## x         3350.62 3368.50           7.5
## [1] "Smoothing is a semi-parametric and data-driven transformation, please see Wood (2006) for an ela
## [1] "please note that you included area in the base-formula and it is also the variable you test"
```

```
PanJenArea<-fform(data=df,"areaPMP",formBase)
```

```
##               AIC     BIC ranking (BIC)
## x^2       3267.93 3289.39           1.0
## sqr(x)    3271.05 3292.50           2.5
## x+x^2     3271.05 3292.50           2.5
## smoothing 3258.79 3299.90           4.0
## log(x)    3281.79 3303.25           5.0
## base      3350.62 3368.50           6.5
## x         3350.62 3368.50           6.5
## 1/x       3348.06 3369.51           8.0
## [1] "Smoothing is a semi-parametric and data-driven transformation, please see Wood (2006) for an ela
## [1] "please note that you included areaPMP in the base-formula and it is also the variable you test"
```

```
PanJenArea<-fform(data=df,"PMP",formBase)
```

```
##               AIC     BIC ranking (BIC)
## base      3350.62 3368.50           1.5
## x         3350.62 3368.50           1.5
## smoothing 3334.21 3371.24           3.0
## sqr(x)    3351.12 3372.58           4.5
```

```
## x+x^2     3351.12 3372.58           4.5
## x^2       3351.32 3372.77           6.0
## log(x)    3351.40 3372.86           7.0
## 1/x       3351.76 3373.21           8.0
## [1] "Smoothing is a semi-parametric and data-driven transformation, please see Wood (2006) for an ela
## [1] "please note that you included PMP in the base-formula and it is also the variable you test"
```

```
df$areaPMP2 <- df$areaPMP^2
df$area2 <- df$area^2
```

Training and test sample division

```
library(dplyr)
library(caret)
```

```
  df.sel <- df[,-c(1,3,4)]

  set.seed(2021)
  dfPartition <- createDataPartition(y = df.sel$price_cents,
                                     p = 0.7, list = F)

  Training <- df.sel[dfPartition,]
```

```
## Warning: The ‘i‘ argument of ‘‘[‘()‘ can’t be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
```

```
  Test <- df.sel[-dfPartition,]
```

Variables selection

AIC forward selection

```
Modelzero <- lm(price_cents~1,data=Training)
  summary(Modelzero)

  FitAll = lm(price_cents ~ ., data=Training)
  formula(FitAll)

  model.forward <- step(Modelzero,direction="forward",scope=formula(FitAll))
```

```
summary(model.forward)
```

```
##
## Call:
## lm(formula = price_cents ~ areaPMP + areaPMP2 + Trolleys + Locker +
```

```
##     `Lift access` + `Part room` + `Step ladder` + `24 hours` +
##     `Power supply` + `Wine cellar` + CCTV + `Racking shelf` +
##     `Lockable doors` + `Entire room` + `On-Site staff` + PMP,
##     data = Training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -299.85  -31.73    2.00   40.91  351.25
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        720.7162   116.6113   6.181 4.54e-09 ***
## areaPMP             71.6360     3.7315  19.198  < 2e-16 ***
## areaPMP2            -1.3958     0.1442  -9.682  < 2e-16 ***
## Trolleys           283.4565   101.7978   2.785 0.005965 **
## Locker            -276.1819    47.6418  -5.797 3.18e-08 ***
## `Lift access`     -180.0903    74.2699  -2.425 0.016357 *
## `Part room`       -304.9518    49.7279  -6.132 5.82e-09 ***
## `Step ladder`      259.4567    53.5774   4.843 2.85e-06 ***
## `24 hours`       -1072.1018   146.1035  -7.338 8.37e-12 ***
## `Power supply`    -143.2721    43.3152  -3.308 0.001147 **
## `Wine cellar`     -159.1852    52.1847  -3.050 0.002650 **
## CCTV               915.6844   145.9520   6.274 2.79e-09 ***
## `Racking shelf`    412.2588    95.1061   4.335 2.49e-05 ***
## `Lockable doors`  -220.1758    59.1372  -3.723 0.000267 ***
## `Entire room`      138.1340    41.2172   3.351 0.000990 ***
## `On-Site staff`   -284.5849   105.1520  -2.706 0.007491 **
## PMP                 61.8493    33.3704   1.853 0.065545 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 82.69 on 171 degrees of freedom
## Multiple R-squared:  0.8992, Adjusted R-squared:  0.8898
## F-statistic: 95.36 on 16 and 171 DF,  p-value: < 2.2e-16
```

```
predict.for.tr <- predict(model.forward, newdata = Training)

training.for.mse <- mean((predict.for.tr - Training$price_cents)^2)
paste("Training MSE error:", training.for.mse)
```

```
## [1] "Training MSE error: 6218.84749850434"
```

```
predict.for.tst <- predict(model.forward, newdata = Test)

test.for.mse <- mean((predict.for.tst - Test$price_cents)^2)
paste("Test MSE error:", test.for.mse)
```

```
## [1] "Test MSE error: 33491.9618712716"
```

AIC backward selection

```r
model.backward <- stepAIC(FitAll, trace=TRUE, direction="backward")
```

```r
summary(model.backward)
```

```
##
## Call:
## lm(formula = price_cents ~ area + PMP + areaPMP + Locker + `Racking shelf` +
##     `Part room` + `Entire room` + `Wine cellar` + `Air con` +
##     `24 hours` + `Lift access` + Security + CCTV + `Lockable doors` +
##     `On-Site staff` + Trolleys + `Step ladder` + `Power supply` +
##     areaPMP2 + area2, data = Training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -257.51  -30.02    0.81   34.30  316.43
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1390.3759   197.4060   7.043 4.68e-11 ***
## area              -206.9883    43.6297  -4.744 4.47e-06 ***
## PMP               -193.0689    72.7750  -2.653 0.008749 **
## areaPMP            199.3332    27.3882   7.278 1.26e-11 ***
## Locker            -241.9684    48.4049  -4.999 1.45e-06 ***
## `Racking shelf`    428.3879    92.3749   4.637 7.08e-06 ***
## `Part room`       -322.5346    57.5388  -5.606 8.42e-08 ***
## `Entire room`      123.9874    41.3909   2.996 0.003157 **
## `Wine cellar`     -118.8612    54.6932  -2.173 0.031171 *
## `Air con`         -264.9318   182.8798  -1.449 0.149306
## `24 hours`       -1036.5202   187.1490  -5.538 1.17e-07 ***
## `Lift access`     -234.0499    90.2713  -2.593 0.010366 *
## Security          -221.1852   121.1279  -1.826 0.069629 .
## CCTV               873.3448   200.5938   4.354 2.33e-05 ***
## `Lockable doors`  -226.9956    68.0640  -3.335 0.001051 **
## `On-Site staff`   -422.2022   105.1282  -4.016 8.93e-05 ***
## Trolleys           680.5225   198.6777   3.425 0.000773 ***
## `Step ladder`      280.5002    55.6191   5.043 1.18e-06 ***
## `Power supply`    -158.6295    46.5463  -3.408 0.000820 ***
## areaPMP2            -4.4883     0.6519  -6.885 1.12e-10 ***
## area2                8.0578     1.6664   4.836 3.00e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 77.89 on 167 degrees of freedom
## Multiple R-squared:  0.9127, Adjusted R-squared:  0.9022
## F-statistic: 87.27 on 20 and 167 DF,  p-value: < 2.2e-16
```

```r
predict.bck.tr <- predict(model.backward, newdata = Training)

training.bck.mse <- mean((predict.bck.tr - Training$price_cents)^2)
paste("Training MSE error:", training.bck.mse)
```

Summary

```
## [1] "Training MSE error: 5388.60327239268"
```

```
predict.bck.tst <- predict(model.backward, newdata = Test)

test.bck.mse <- mean((predict.bck.tst - Test$price_cents)^2)
paste("Test MSE error:", test.bck.mse)
```

```
## [1] "Test MSE error: 47216.582975243"
```

Ridge and Lasso regularizations

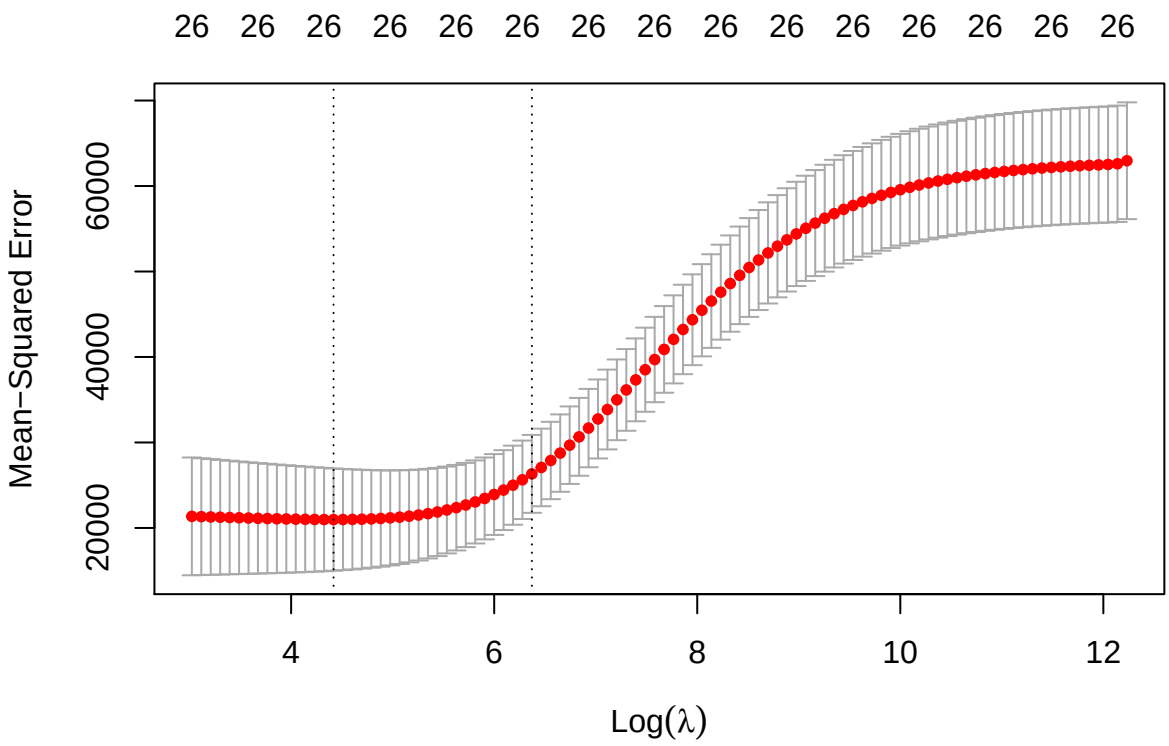
```
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.0-2
```

```
# Convert into a matrix train and test data
train.mat <- model.matrix(price_cents ~ ., data = Training)
test.mat <- model.matrix(price_cents ~ ., data = Test)
```

Ridge

```
# Cross validation to obtain the best value of lambda. Error evolution.
cv.ridge <- cv.glmnet(x = train.mat, y = Training$price_cents, alpha = 0,
                      lambda = NULL, type.measure="mse")

plot(cv.ridge)
```




```
paste("Best lambda:", cv.ridge$lambda.min)
```

```
## [1] "Best lambda: 82.8962040386565"
```

```
paste("Best lambda + y sd:", cv.ridge$lambda.1se)
```

```
## [1] "Best lambda + y sd: 584.816330737625"
```

```
# Training the model
mod.ridge.train <- glmnet(x = train.mat, y = Training$price_cents, alpha = 0,
                          lambda = cv.ridge$lambda.1se)

dim(coef(mod.ridge.train))
```

```
## [1] 29  1
```

```
coef(mod.ridge.train, s = "lambda.1se")
```

```
## 29 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)        519.9546855
## (Intercept)          .
## area                 9.5555486
```

```
## PMP                 16.1704369
## areaPMP              6.7590258
## Locker             -61.7338679
## Cupboard             .
## `Racking shelf`     50.3237893
## `Part room`        -32.9606682
## `Entire room`       32.8389610
## Unit                35.1664430
## `Wine cellar`       62.8528510
## `Air con`           14.7398559
## `24 hours`         -68.5651762
## `Lift access`      -93.5450013
## Security             3.9605439
## `Loading dock`      11.4918662
## CCTV               -37.0390277
## `Lockable doors`     3.2126782
## `On-Site staff`     17.2551282
## `Free Parking`       4.0930558
## Trolleys            15.4875343
## `Step ladder`       17.0748527
## `Racking/Shelving`  50.3044594
## Sprinkler           -3.3464127
## `Power supply`       3.2118782
## WIFI                -2.5807604
## areaPMP2             0.1792083
## area2                0.4311332
```

```
# Training predictions
pred.ridge <- predict(mod.ridge.train, newx = train.mat)

# Training error (MSE)
tr.ridge.mse <- mean((pred.ridge - Training$price_cents)^2)
paste("Training MSE error:", tr.ridge.mse)
```

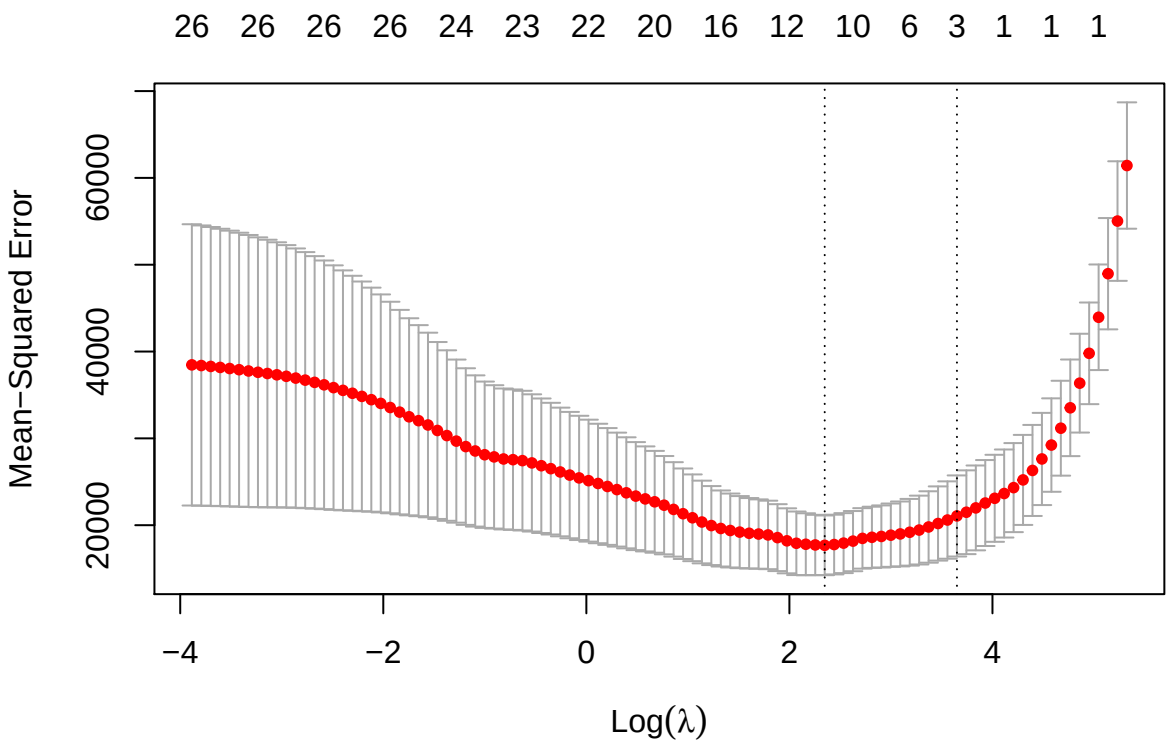
```
## [1] "Training MSE error: 22768.5893704842"
```

```
#Test predictions: using training model
pred.test.ridge <- predict(mod.ridge.train,newx = test.mat)

test.ridge.mse <- mean((pred.test.ridge - Test$price_cents)^2)
paste("Test MSE error:",test.ridge.mse)
```

```
## [1] "Test MSE error: 29051.6852001325"
```

Lasso

```
cv.lasso <- cv.glmnet(x = train.mat, y = Training$price_cents, alpha = 1,
                      lambda = NULL, type.measure="mse")

plot(cv.lasso)
```




```
paste("Best lambda:", cv.lasso$lambda.min)
```

```
## [1] "Best lambda: 10.4603145174933"
```

```
paste("Best lambda + y sd:", cv.lasso$lambda.1se)
```

```
## [1] "Best lambda + y sd: 38.4770092814394"
```

```
# Training the model
mod.lasso.train <- glmnet(x = train.mat, y = Training$price_cents, alpha = 1,
                          lambda = cv.lasso$lambda.1se)

dim(coef(mod.lasso.train))
```

```
## [1] 29  1
```

```
coef(mod.lasso.train, s = "lambda.1se")
```

```
## 29 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)        311.40725
## (Intercept)          .
## area                 .
```

```
## PMP                    .
## areaPMP             31.04453
## Locker             -17.87422
## Cupboard               .
## `Racking shelf`        .
## `Part room`            .
## `Entire room`          .
## Unit                56.28222
## `Wine cellar`          .
## `Air con`              .
## `24 hours`             .
## `Lift access`          .
## Security               .
## `Loading dock`         .
## CCTV                   .
## `Lockable doors`       .
## `On-Site staff`        .
## `Free Parking`         .
## Trolleys               .
## `Step ladder`          .
## `Racking/Shelving`     .
## Sprinkler              .
## `Power supply`         .
## WIFI                   .
## areaPMP2               .
## area2                  .
```

```
# Training predictions
pred.lasso <- predict(mod.lasso.train, newx = train.mat)

# Training error (MSE)
tr.lasso.mse <- mean((pred.lasso - Training$price_cents)^2)
paste("Training MSE error:", tr.lasso.mse)
```

```
## [1] "Training MSE error: 19127.0633609672"
```

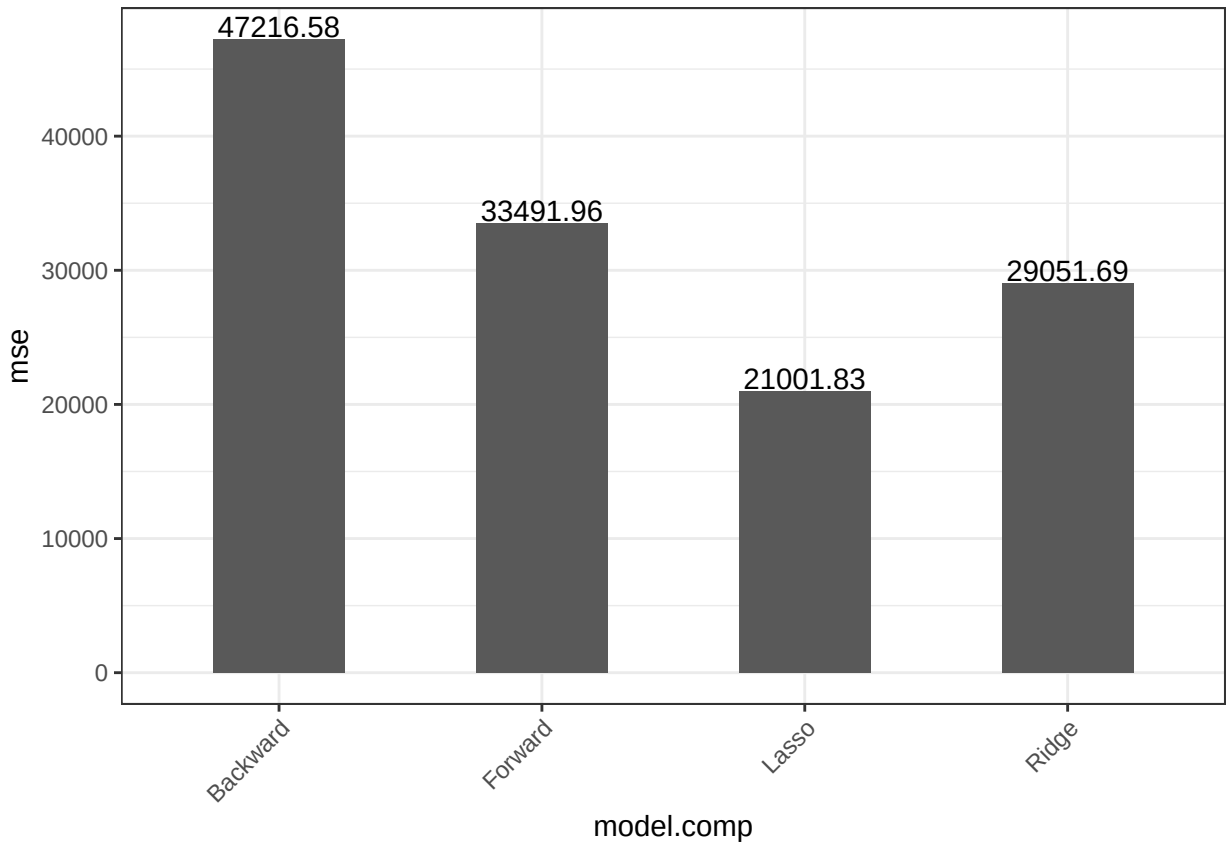
```
#Test predictions: using training model
pred.test.lasso <- predict(mod.lasso.train, newx = test.mat)

test.lasso.mse <- mean((pred.test.lasso - Test$price_cents)^2)
paste("Test MSE error:",test.lasso.mse)
```

```
## [1] "Test MSE error: 21001.8332374298"
```

Comparing results

```
df_compar <- data.frame(model.comp = c("Forward", "Backward", "Ridge", "Lasso"),

mse = c(test.for.mse, test.bck.mse, test.ridge.mse, test.lasso.mse))

ggplot(data = df_compar, aes(x = model.comp, y = mse)) + geom_col(width = 0.5) +
```

```
    geom_text(aes(label = round(mse, 2)), vjust = -0.1) + theme_bw() + theme(axis.text.x = element_text
    hjust = 1))
```



Linear Model

```
lmodel.Train <- lm (price_cents ~ areaPMP + Locker + Unit, data = Training)

summary(lmodel.Train)
```

```
##
## Call:
## lm(formula = price_cents ~ areaPMP + Locker + Unit, data = Training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -467.36  -69.53  -13.66   66.63  628.32
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  264.617     18.179  14.556  < 2e-16 ***
## areaPMP       36.447      1.791  20.350  < 2e-16 ***
## Locker      -126.104     33.151  -3.804 0.000194 ***
## Unit         179.120     29.668   6.038 8.45e-09 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 125.3 on 184 degrees of freedom
## Multiple R-squared:  0.7509, Adjusted R-squared:  0.7468
## F-statistic: 184.9 on 3 and 184 DF,  p-value: < 2.2e-16
```

```
lmodel.Test <- lm (price_cents ~ areaPMP + Locker + Unit, data = Test)

summary(lmodel.Test)
```

```
##
## Call:
## lm(formula = price_cents ~ areaPMP + Locker + Unit, data = Test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -517.05  -57.83  -10.30   51.64  517.17
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  236.428     30.289   7.806 3.53e-11 ***
## areaPMP       37.576      3.121  12.038  < 2e-16 ***
## Locker        -5.139     59.003  -0.087  0.93083
## Unit         132.255     45.755   2.890  0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 138.7 on 72 degrees of freedom
## Multiple R-squared:  0.7319, Adjusted R-squared:  0.7207
## F-statistic: 65.52 on 3 and 72 DF,  p-value: < 2.2e-16
```