Selection of model ideal

Load Libraries

```
library(tidymodels) # basic libraries
```

Load Data

```
df<-read.csv("rent-amount-brazil-clear.csv")
```

Data Preprocessing

One Hot Encoding

t is used for qualitative variables, for example if the house is furnished or not. Dummy variables are created according to the number of categories where a 1 is assigned when the condition is met. While in the opposite case it is filled with 0, in this case it is a binary variable. It doesn't make sense to create a new column, just assign it a 1 where the condition is met. Since it is perfectly understood.

```
unique(df$furniture)
```

```
## [1] "furnished"     "not furnished"
```

One Hot Binary Transform

```
df<- df %>%
  mutate(furniture=ifelse(furniture=="furnished",1,0))
```

With the mutate function we perform the transformation of variables.

Scaler data

We do it so that the variables can be comparable to each other. Where for each observation the average will be subtracted and divided by the standard deviation.

```
df_sc<- df %>% mutate(rooms=scale(rooms),
                      bathrooms=scale(bathrooms),
                      parking.spaces=scale(parking.spaces),
                      quality_departament=scale(quality_departament))
```

```
library(caret) # machine learning library
```

Split Data

We split the data, to see if our model is capable of solving new predictions, with data that it has never seen.

```
set.seed(2018) # random state

training.ids<-createDataPartition(df_sc$rent.amount,p=0.7,list = F)

train_data<-df[training.ids,] # select train data index
test_data<-df[-training.ids,] # select test data index
```

```
dim(train_data)
```

```
## [1] 4256    8
```

```
dim(test_data)
```

```
## [1] 1824    8
```

We have 4256 observations to train the model. And the rest to perform a validation, to see if the model is capable of generalizing the problem.

Cross Validation

It is a technique that consists of making sub samples in the data, where each sample would be evaluated. In order to see the average generalization of the model.

```
train_control<-trainControl(method = "cv",number = 10)
```

Performance metrics

- **Mean Square Error** Measures the average error between the predicted and original values. It is very sensitive to outliers, but it is good for making sure that our model does not have predictions that are too far out.
- **Mean Absolute Error** It is similar to the MSE with the difference that the MAE is robust with the predictions with outliers, so it is more complicated to determine if our model generates predictions.
- R^2 It measures the degree of fit between the predictions and the original value. It is measured from 0 to 1, the closer it is to 1, the greater the accumulation of the predictions with respect to the original value.

```
cv<-train(rent.amount~.,data=train_data,
              trControl=train_control,
              method="lm")
```

```
cv
```

```
## Linear Regression
##
## 4256 samples
##    7 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3831, 3830, 3829, 3830, 3831, 3830, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.1495644  0.9628778  0.1114936
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Not Scaler Data

```
set.seed(2018) # random state

training.ids<-createDataPartition(df$rent.amount,p=0.7,list = F)

train_data<-df[training.ids,] # select train data index
test_data<-df[-training.ids,] # select test data index
```

```
cv<-train(rent.amount~.,data=train_data,
              trControl=train_control,
              method="lm")
```

```
cv
```

```
## Linear Regression
##
## 4256 samples
##    7 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3831, 3830, 3829, 3830, 3831, 3830, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.1495644  0.9628778  0.1114936
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Conclusion

Using a Linear Regression for this case is enough to solve the problem. On this occasion the difference is zero between processing the data or not. So we can use the variables without using scaling.

We will only apply one hot transformation to the furniture variable, since it is a qualitative variable. We will apply a logarithmic transformation to the new data that refers to the area, the price of fire insurance.