Model Ideal

Load Libraries

```
library(tidyverse) # common libraries
library(caret)
```

Load Data

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

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

Split Data

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

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

train_data<-df[training.ids,]
test_data<- df[-training.ids,]
```

Model Creation

```
lm<-lm(rent.amount~.,data=train_data)
```

MSE

Measures the average error between the predicted value and the original.

R^2

It measures the degree of fit of the model. The higher the value, the better the model fit.

```
summary(lm)
```

```
##
## Call:
## lm(formula = rent.amount ~ ., data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51030 -0.10254 -0.01088  0.08536  0.82536
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.112122   0.021883 233.612  < 2e-16 ***
## furniture            0.051093   0.005609   9.110  < 2e-16 ***
## area                 0.018235   0.006474   2.817  0.00487 **
## fire.insurance       0.459329   0.008097  56.728  < 2e-16 ***
## rooms               -0.009335   0.003868  -2.413  0.01584 *
## bathrooms            0.013907   0.003262   4.263 2.06e-05 ***
## parking.spaces       0.006737   0.002548   2.644  0.00823 **
## quality_departament  0.341469   0.005858  58.290  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1496 on 4248 degrees of freedom
## Multiple R-squared:  0.963,  Adjusted R-squared:  0.963
## F-statistic: 1.581e+04 on 7 and 4248 DF,  p-value: < 2.2e-16
```

The model has a good fit to the data. It can explain 96% of the cases, which can generalize very well.

```
y_pred_test<-predict(lm,newdata = test_data)
```

```
test_data<-test_data %>% mutate(y_pred=y_pred_test)
```

```
test_data %>% summarise(rmse=RMSE(y_pred,rent.amount))
```

rmse <dbl>
0.1603487

1 row

```
test_data %>% summarise(mse=R2(y_pred,rent.amount))
```

mse <dbl>
0.9563073

1 row

It has similar metrics with the training data, indicating that it's not just good for the data it was trained on. If not, it is also suitable, for values that I have never seen.

Exponential transformation

```
test_data<-test_data %>%

  mutate(rent.amount=exp(rent.amount)) %>%

  mutate(predictions=exp(y_pred))
```

Let us remember that the rent.amount variable is in logarithmic scale and we perform the exponential transformation because it is its opposite operation.

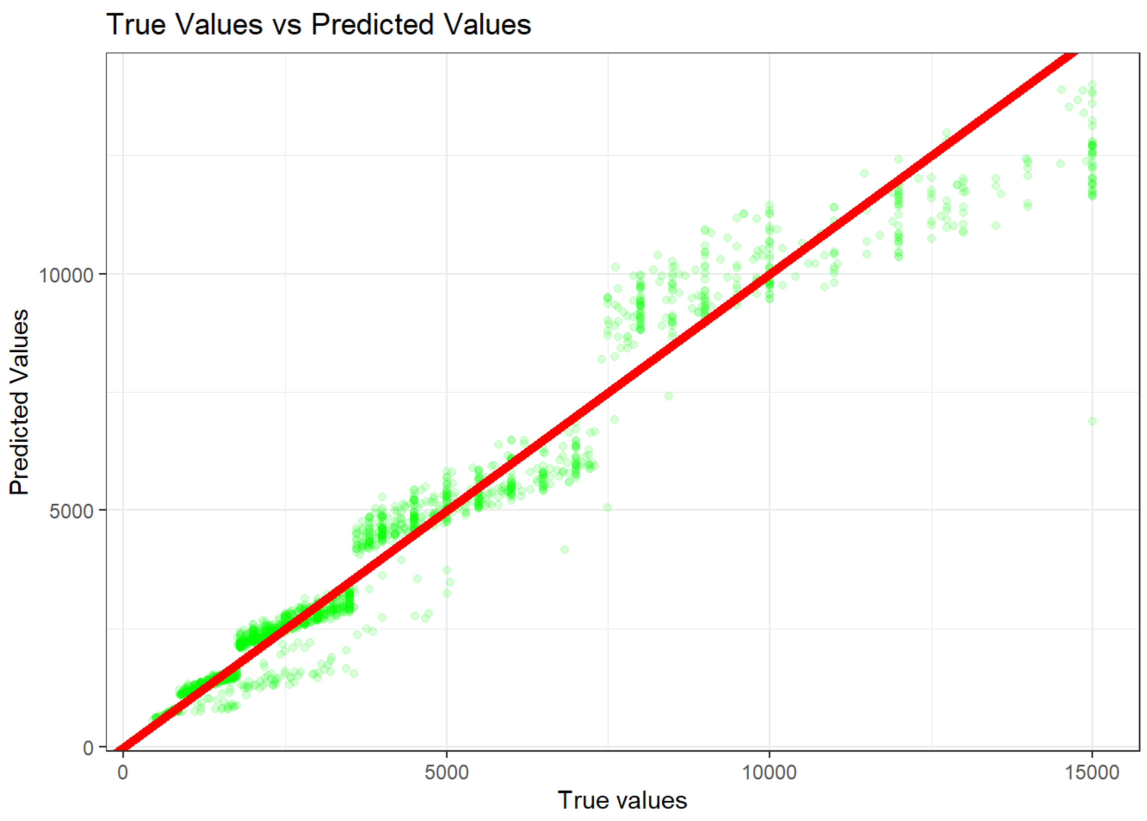
Variable coefficients

```
coef(lm)
```

```
##         (Intercept)           furniture                area      fire.insurance
##         5.112121968         0.051093103         0.018234716         0.459329059
##               rooms           bathrooms      parking.spaces quality_departament
##        -0.009334583         0.013907228         0.006736941         0.341469148
```

Coefficients determined by the Linear Regression model.

```
test_data %>%
  ggplot(aes(x=rent.amount,y=predictions))+
  geom_point(color="green",alpha=0.15) +
  geom_abline(intercept = 0,slope = 1,color="red",lwd=2) +
  theme_bw() +
  labs(title = "True Values vs Predicted Values",
       x="True values",y="Predicted Values")
```




```
test_data %>% select(rent.amount,predictions) %>% head(20)
```

	rent.amount <dbl>	predictions <dbl>
7	5000	5032.991
10	2900	2616.611
11	720	728.121
15	3950	4534.735
17	3010	3125.526
18	3500	3136.057
19	7800	8545.846
23	3500	2932.991
27	6500	5412.758
29	1800	2167.663

1-10 of 20 rows Previous 1 2 Next

In most predictions, it gives predictions very close to the original value.

Save Model

```
saveRDS(lm,"lm_rent_amount.RDS")
```