**COIMBATORE INSTITUTE OF TECHNOLOGY**

**2. ANALYSIS ON BRAZIL HOUSE RENT DATA TO PREDICT HOUSE RENT**

- BASIL RITHESH

(1832014)

**AIM:**

To predict the house rent of Brazil based on other components in the dataset.

**DESCRIPTION:**

Rent of a house increase or decrease depends on various factors like area, location ,facility, pet, safe and security, etc. But not all factors were responsible to affect the house rent. The project aimed to predict the house rent(Brazil) from the given data. In order to predict the output we have to determine the key factors that affects the house rent. By using such factors , Better results(Rent) can be predicted using it. For this problem Multiple Linear Regression is Best to predict house rent.

**CODE:**

**#IMPORTING PACKAGES**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import statsmodels.api as sm

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score

%matplotlib inline

sns.set\_style('darkgrid')

**#READING DATA FROM LOCAL MACHINE AND STORE IT AS A DATAFRAME**

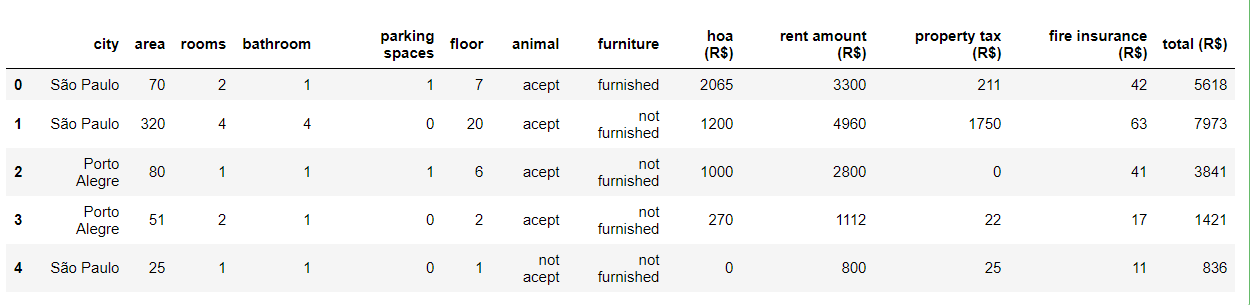
#Loading and Reading the data

filepath='D:\houses\_to\_rent.csv'

houses=pd.read\_csv(filepath,encoding='unicode\_escape')

houses.head()

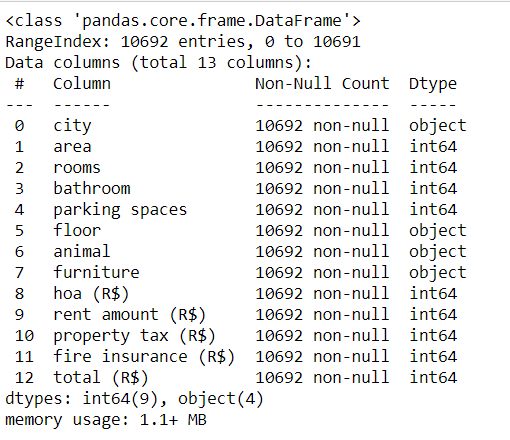
**OUTPUT:**



**#INFORMATION ABOUT THE DATA FRAME**

houses.info()

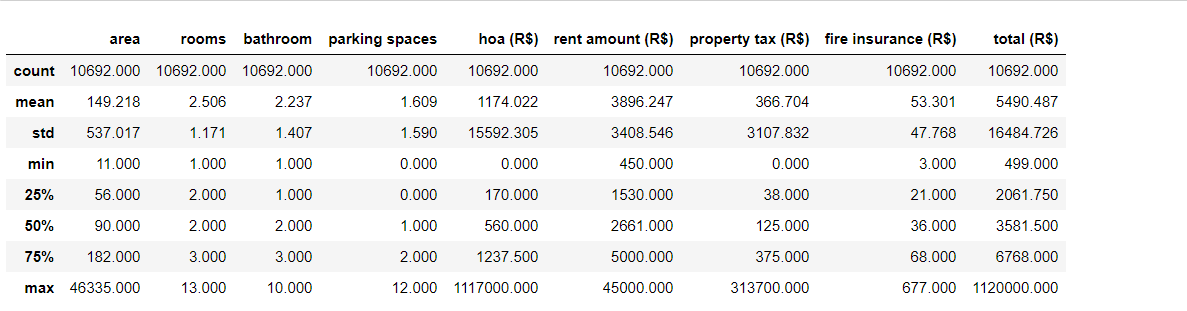
**OUTPUT:**



**#SUMMARY STATISTICS**

houses.describe().round(3)

**OUTPUT:**



**#REPALCING MISSING VALUES**

cols = df.columns

cols = cols.map(lambda x: x.replace(' ','\_') if isinstance(x, (str)) else x)

df.columns = cols

**#CHANGE "$" FOR USE QUERIES**

df.rename(columns={'hoa\_(R$)' : 'hoa',

'rent\_amount\_(R$)' : 'rent\_amount',

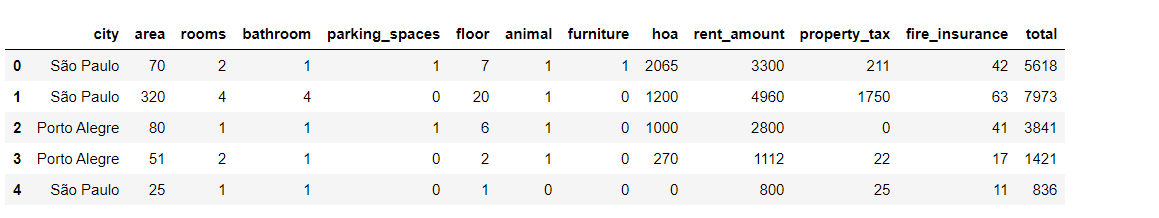
'property\_tax\_(R$)' : 'property\_tax',

'fire\_insurance\_(R$)' : 'fire\_insurance',

'total\_(R$)' : 'total'}, inplace = True)

df.head()

**OUTPUT:**



**#COUNT PLOT FOR FURNITURE**

fc = sns.countplot(df['furniture'], hue = df['city'])

fc.figure.set\_size\_inches(12, 8)

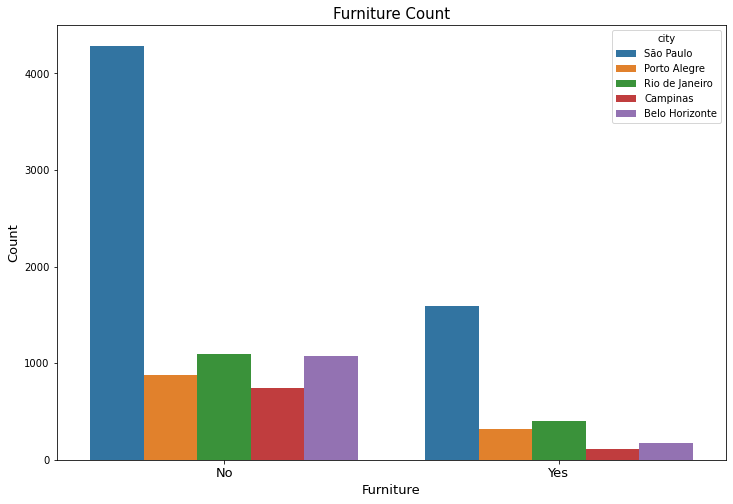
fc.set\_title('Furniture Count',fontsize=15)

fc.set\_xlabel('Furniture',fontsize=13)

fc.set\_ylabel('Count', fontsize=13)

fc.set\_xticklabels(['No','Yes'], fontsize=13)

**OUTPUT:**



**#BARPLOT FOR NUMBER OF ROOMS WITH SIZE OF AREA**

bs = sns.barplot(x='rooms', y='area', data = df, palette = 'dark')

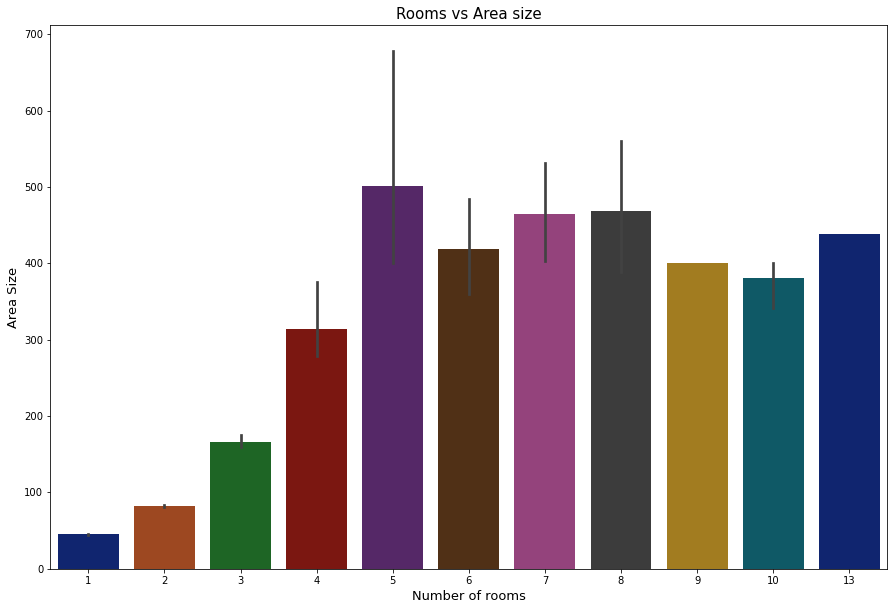
bs.figure.set\_size\_inches(15, 10)

bs.set\_title('Rooms vs Area size',fontsize=15)

bs.set\_xlabel('Number of rooms', fontsize=13)

bs.set\_ylabel('Area Size', fontsize=13)

**OUTPUT:**



**#SCATTER PLOT FOR TOTAL RENT VS HOA TAX**

df = df.drop(labels=df[(df['hoa'] > 300000)].index)

df = df.drop(labels=df[(df['total'] > 30000)].index)

th = sns.scatterplot(x = 'total', y = 'hoa', data = df)

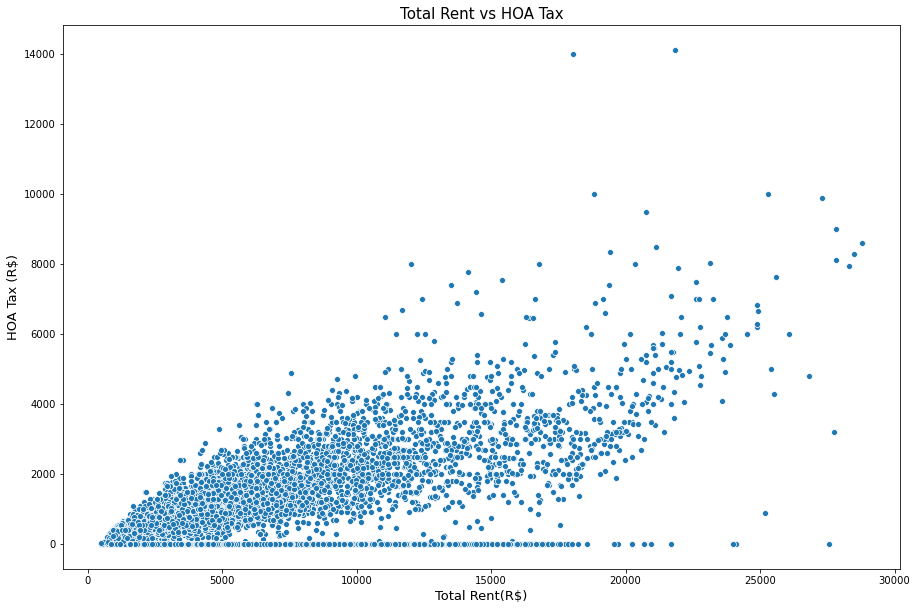
th.figure.set\_size\_inches(15, 10)

th.set\_title('Total Rent vs HOA Tax',fontsize=15)

th.set\_xlabel('Total Rent(R$)', fontsize=13)

th.set\_ylabel('HOA Tax (R$)', fontsize=13)

**OUTPUT:**



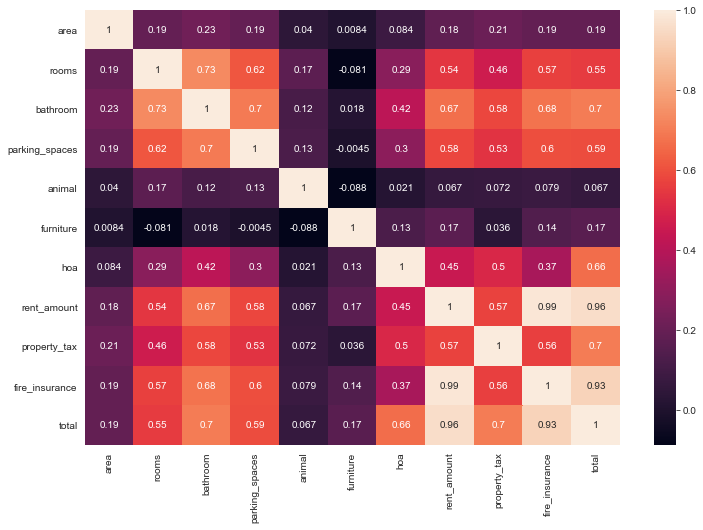
**#HEAT MAP TO FIND BETTER CORRELATED VALUES FOR TOTAL RENT**

cor = df.corr()

plt.figure(figsize=(15,10))

sns.heatmap(df.corr(), annot=True, cmap = 'YlGnBu')

**OUTPUT:**

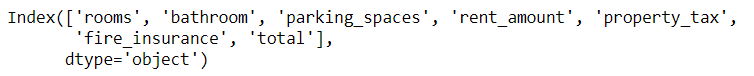


**#BATROOMS, HOA PROPERTY\_TAX, FIRE\_INSURANCE WERE MORE CORRELATED WITH RENT\_AMOUNT**

req\_cols = cor[cor.loc['rent\_amount']>0.5].T.columns

req\_cols

**OUTPUT:**



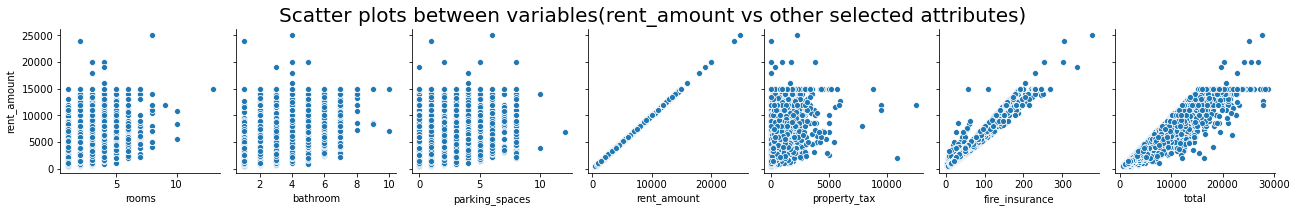
**#PAIRPLOT FOR RENT\_AMOUNT WITH REQ\_COLS**

ax = sns.pairplot(df, y\_vars='rent\_amount', x\_vars=req\_cols)

ax.fig.suptitle('Scatter plots between variables(rent\_amount vs other selected attributes)', fontsize=20, y=1.1)

ax

**OUTPUT:**



**#LINEAR REGRESSION MODEL**

metrics = [] #list with RMSE values

y = houses['rent\_amount'] #Creating a Series for target variable

#Creating a DF for explanatory variables

x = houses.drop(['rent\_amount','city','parking\_spaces','area',

'bathroom','furniture','animal','floor','rooms', 'total'], axis = 1)

#Splitting data arrays into two subset

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state = 8)

reg = LinearRegression() #Instantiating the model

reg.fit(x\_train, y\_train) #Training data

predict = reg.predict(x\_test)

#add a constant and looking the summary

x\_train\_constant = sm.add\_constant(x\_train)

model\_sm = sm.OLS(y\_train, x\_train\_constant, hascont = True).fit()

print(model\_sm.summary())

#looking the metrics

print('MAE: ', mean\_absolute\_error(y\_test, predict).round(3))

print('RMSE: ', np.sqrt(mean\_squared\_error(y\_test, predict)).round(3))

print('R2:', r2\_score(y\_test, predict).round(3))

metrics.append(np.sqrt(mean\_squared\_error(y\_test, predict)))

**OUTPUT:**

OLS Regression Results

==============================================================================

Dep. Variable: rent\_amount R-squared: 0.985

Model: OLS Adj. R-squared: 0.985

Method: Least Squares F-statistic: 1.615e+05

Date: Sun, 07 Mar 2021 Prob (F-statistic): 0.00

Time: 17:22:52 Log-Likelihood: -55698.

No. Observations: 7475 AIC: 1.114e+05

Df Residuals: 7471 BIC: 1.114e+05

Df Model: 3

Covariance Type: nonrobust

==================================================================================

coef std err t P>|t| [0.025 0.975]

----------------------------------------------------------------------------------

const -23.9498 7.662 -3.126 0.002 -38.969 -8.931

hoa 0.3272 0.005 65.262 0.000 0.317 0.337

property\_tax -0.1223 0.011 -11.555 0.000 -0.143 -0.102

fire\_insurance 68.7181 0.125 551.712 0.000 68.474 68.962

==============================================================================

Omnibus: 2524.562 Durbin-Watson: 2.013

Prob(Omnibus): 0.000 Jarque-Bera (JB): 232868.171

Skew: 0.665 Prob(JB): 0.00

Kurtosis: 30.311 Cond. No. 2.42e+03

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 2.42e+03. This might indicate that there are

strong multicollinearity or other numerical problems.

MAE: 239.729

RMSE: 459.726

R2: 0.981

new\_houses = houses.copy()

features = ['floor','bathroom','rooms','area','parking\_spaces',

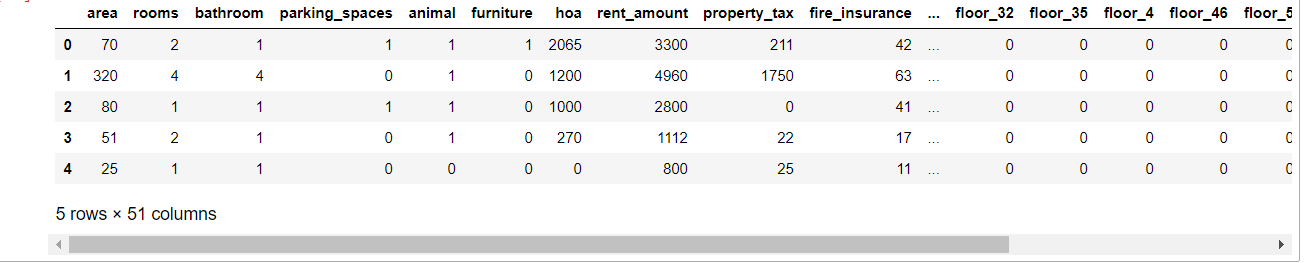
'hoa', 'rent\_amount', 'property\_tax', 'fire\_insurance']

for item in features:

new\_houses[item] = np.log1p(new\_houses[item])

new\_houses = pd.get\_dummies(new\_houses)

new\_houses.head()

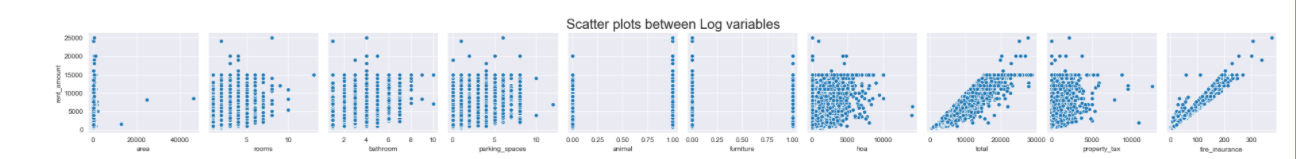


ax = sns.pairplot(new\_houses, y\_vars='rent\_amount',

x\_vars = ['area', 'rooms', 'bathroom', 'parking\_spaces', 'animal', 'furniture', 'hoa', 'total', 'property\_tax', 'fire\_insurance'])

ax.fig.suptitle('Scatter plots between Log variables',

fontsize=20, y=1.1)



**#LINEAR REGRESSION MODEL**

y = new\_houses['rent\_amount']

x = new\_houses.drop(['rent\_amount','animal','rooms','total','area'], axis = 1)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state = 8)

reg = LinearRegression()

reg.fit(x\_train, y\_train)

predict = reg.predict(x\_test)

x\_train\_constant = sm.add\_constant(x\_train)

model\_sm = sm.OLS(y\_train, x\_train\_constant, hascont = True).fit()

print(model\_sm.summary())

print('MAE: ', mean\_absolute\_error(y\_test, predict).round(3))

print('RMSE: ', np.sqrt(mean\_squared\_error(y\_test, predict)).round(3))

print('R2:', r2\_score(y\_test, predict).round(3))

metrics.append(np.sqrt(mean\_squared\_error(y\_test, predict)))

OLS Regression Results

==============================================================================

Dep. Variable: rent\_amount R-squared: 0.990

Model: OLS Adj. R-squared: 0.990

Method: Least Squares F-statistic: 1.724e+04

Date: Sun, 07 Mar 2021 Prob (F-statistic): 0.00

Time: 17:28:25 Log-Likelihood: -54286.

No. Observations: 7475 AIC: 1.087e+05

Df Residuals: 7433 BIC: 1.089e+05

Df Model: 41

Covariance Type: nonrobust

=======================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------

const 115.1887 17.611 6.541 0.000 80.667 149.711

bathroom -19.7188 4.705 -4.191 0.000 -28.942 -10.496

parking\_spaces -29.3756 3.793 -7.744 0.000 -36.811 -21.940

furniture 48.1505 9.715 4.956 0.000 29.106 67.195

hoa 0.1869 0.005 35.214 0.000 0.177 0.197

property\_tax -0.0313 0.009 -3.351 0.001 -0.050 -0.013

fire\_insurance 70.6802 0.129 548.576 0.000 70.428 70.933

city\_Belo Horizonte -30.4581 11.097 -2.745 0.006 -52.211 -8.705

city\_Campinas 104.3999 12.982 8.042 0.000 78.952 129.848

city\_Porto Alegre -243.2764 11.365 -21.407 0.000 -265.554 -220.999

city\_Rio de Janeiro 71.3885 10.664 6.694 0.000 50.483 92.294

city\_SÃ£o Paulo 213.1349 7.305 29.178 0.000 198.816 227.454

floor\_- -563.0260 22.228 -25.329 0.000 -606.600 -519.452

floor\_1 -91.3654 22.882 -3.993 0.000 -136.220 -46.511

floor\_10 -71.0168 28.827 -2.464 0.014 -127.526 -14.507

floor\_11 -51.3271 29.284 -1.753 0.080 -108.732 6.078

floor\_12 -13.9460 32.127 -0.434 0.664 -76.924 49.032

floor\_13 -2.1211 34.706 -0.061 0.951 -70.154 65.912

floor\_14 -8.6343 35.784 -0.241 0.809 -78.781 61.512

floor\_15 16.5658 37.472 0.442 0.658 -56.889 90.021

floor\_16 20.3057 44.499 0.456 0.648 -66.926 107.537

floor\_17 101.2522 45.371 2.232 0.026 12.313 190.192

floor\_18 83.8837 47.744 1.757 0.079 -9.707 177.475

floor\_19 112.3665 59.906 1.876 0.061 -5.067 229.800

floor\_2 -70.7433 23.435 -3.019 0.003 -116.682 -24.805

floor\_20 199.9562 61.500 3.251 0.001 79.399 320.513

floor\_21 322.0862 68.545 4.699 0.000 187.718 456.455

floor\_22 10.3473 103.001 0.100 0.920 -191.563 212.258

floor\_23 26.2610 88.722 0.296 0.767 -147.658 200.180

floor\_24 -44.3479 107.823 -0.411 0.681 -255.712 167.016

floor\_25 -37.5753 77.499 -0.485 0.628 -189.495 114.344

floor\_26 -146.1179 98.758 -1.480 0.139 -339.712 47.476

floor\_27 210.3425 168.799 1.246 0.213 -120.550 541.235

floor\_28 67.3274 168.769 0.399 0.690 -263.507 398.162

floor\_29 220.8629 168.815 1.308 0.191 -110.061 551.787

floor\_3 -58.9196 23.573 -2.499 0.012 -105.129 -12.710

floor\_301 1.085e-13 7.6e-14 1.427 0.154 -4.05e-14 2.58e-13

floor\_32 -106.2029 335.861 -0.316 0.752 -764.585 552.179

floor\_35 7.349e-15 3.29e-14 0.223 0.823 -5.72e-14 7.19e-14

floor\_4 -44.5221 24.435 -1.822 0.068 -92.422 3.378

floor\_46 380.0206 336.142 1.131 0.258 -278.913 1038.954

floor\_5 -93.1974 25.589 -3.642 0.000 -143.359 -43.036

floor\_51 0 0 nan nan 0 0

floor\_6 -77.8223 26.153 -2.976 0.003 -129.090 -26.554

floor\_7 -77.7381 26.295 -2.956 0.003 -129.283 -26.193

floor\_8 -78.1553 26.567 -2.942 0.003 -130.234 -26.077

floor\_9 -19.6105 28.613 -0.685 0.493 -75.701 36.480

==============================================================================

Omnibus: 4116.684 Durbin-Watson: 1.993

Prob(Omnibus): 0.000 Jarque-Bera (JB): 465944.389

Skew: 1.684 Prob(JB): 0.00

Kurtosis: 41.531 Cond. No. 2.53e+16

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 2.7e-23. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

MAE: 214.158

RMSE: 399.011

R2: 0.986

**#PLOTTING THE GRAPH**

group = houses.groupby('city')['rent\_amount']

Q1 = group.quantile(.25)

Q3 = group.quantile(.75)

IIQ = Q3 - Q1

down\_limit = Q1 - 1.5\* IIQ

up\_limit = Q3 + 1.5\* IIQ

new\_df = pd.DataFrame() #Creating a empty df

for city in group.groups.keys():

cities = houses['city'] == city #selecting boolean values

limits = (houses['rent\_amount'] >= down\_limit[city]) & (houses['rent\_amount'] <= up\_limit[city]) #put the boxplot limits

queries = cities & limits #query cities and limits

selected = houses[queries] #put the query above in houses

new\_df = pd.concat([new\_df, selected]) #concatenating the empty df in the df who I processed

plt.figure(figsize=(17,12))

color = {'Belo Horizonte':'g', 'Campinas':'b', 'Porto Alegre':'m',

'Rio de Janeiro':'r', 'SÃ£o Paulo':'c'}

plt.subplot(2, 2, 1)

ax = sns.boxplot(x = 'city', y = 'rent\_amount', data = new\_df, palette = color)

ax.set\_xlabel('City', fontsize=13)

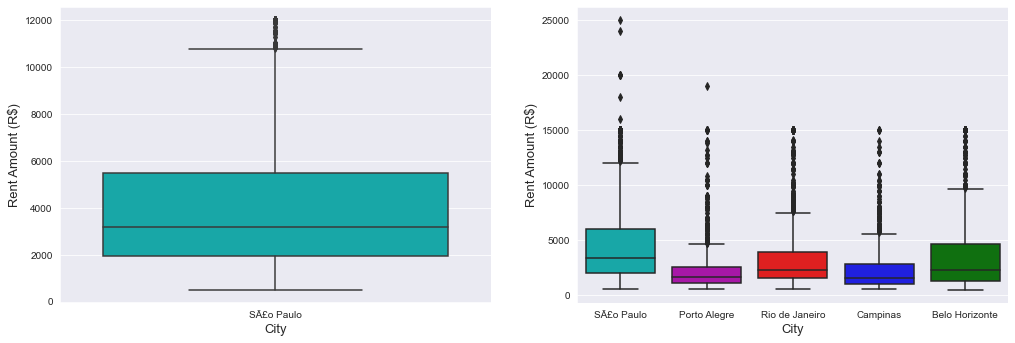
ax.set\_ylabel('Rent Amount (R$)', fontsize=13)

plt.subplot(2, 2, 2)

ax = sns.boxplot(x = 'city', y = 'rent\_amount', data = houses, palette = color)

ax.set\_xlabel('City', fontsize=13)

ax.set\_ylabel('Rent Amount (R$)', fontsize=13)



reg = LinearRegression()

reg.fit(x\_train, y\_train)

predict = reg.predict(x\_test)

x\_train\_constant = sm.add\_constant(x\_train)

model\_sm = sm.OLS(y\_train, x\_train\_constant, hascont = True).fit()

print(model\_sm.summary())

print('MAE: ', mean\_absolute\_error(y\_test, predict).round(3))

print('RMSE: ', np.sqrt(mean\_squared\_error(y\_test, predict)).round(3))

print('R2:', r2\_score(y\_test, predict).round(3))

metrics.append(np.sqrt(mean\_squared\_error(y\_test, predict)))

OLS Regression Results

==============================================================================

Dep. Variable: rent\_amount R-squared: 0.990

Model: OLS Adj. R-squared: 0.990

Method: Least Squares F-statistic: 1.724e+04

Date: Sun, 07 Mar 2021 Prob (F-statistic): 0.00

Time: 17:34:54 Log-Likelihood: -54286.

No. Observations: 7475 AIC: 1.087e+05

Df Residuals: 7433 BIC: 1.089e+05

Df Model: 41

Covariance Type: nonrobust

=======================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------

const 115.1887 17.611 6.541 0.000 80.667 149.711

bathroom -19.7188 4.705 -4.191 0.000 -28.942 -10.496

parking\_spaces -29.3756 3.793 -7.744 0.000 -36.811 -21.940

furniture 48.1505 9.715 4.956 0.000 29.106 67.195

hoa 0.1869 0.005 35.214 0.000 0.177 0.197

property\_tax -0.0313 0.009 -3.351 0.001 -0.050 -0.013

fire\_insurance 70.6802 0.129 548.576 0.000 70.428 70.933

city\_Belo Horizonte -30.4581 11.097 -2.745 0.006 -52.211 -8.705

city\_Campinas 104.3999 12.982 8.042 0.000 78.952 129.848

city\_Porto Alegre -243.2764 11.365 -21.407 0.000 -265.554 -220.999

city\_Rio de Janeiro 71.3885 10.664 6.694 0.000 50.483 92.294

city\_SÃ£o Paulo 213.1349 7.305 29.178 0.000 198.816 227.454

floor\_- -563.0260 22.228 -25.329 0.000 -606.600 -519.452

floor\_1 -91.3654 22.882 -3.993 0.000 -136.220 -46.511

floor\_10 -71.0168 28.827 -2.464 0.014 -127.526 -14.507

floor\_11 -51.3271 29.284 -1.753 0.080 -108.732 6.078

floor\_12 -13.9460 32.127 -0.434 0.664 -76.924 49.032

floor\_13 -2.1211 34.706 -0.061 0.951 -70.154 65.912

floor\_14 -8.6343 35.784 -0.241 0.809 -78.781 61.512

floor\_15 16.5658 37.472 0.442 0.658 -56.889 90.021

floor\_16 20.3057 44.499 0.456 0.648 -66.926 107.537

floor\_17 101.2522 45.371 2.232 0.026 12.313 190.192

floor\_18 83.8837 47.744 1.757 0.079 -9.707 177.475

floor\_19 112.3665 59.906 1.876 0.061 -5.067 229.800

floor\_2 -70.7433 23.435 -3.019 0.003 -116.682 -24.805

floor\_20 199.9562 61.500 3.251 0.001 79.399 320.513

floor\_21 322.0862 68.545 4.699 0.000 187.718 456.455

floor\_22 10.3473 103.001 0.100 0.920 -191.563 212.258

floor\_23 26.2610 88.722 0.296 0.767 -147.658 200.180

floor\_24 -44.3479 107.823 -0.411 0.681 -255.712 167.016

floor\_25 -37.5753 77.499 -0.485 0.628 -189.495 114.344

floor\_26 -146.1179 98.758 -1.480 0.139 -339.712 47.476

floor\_27 210.3425 168.799 1.246 0.213 -120.550 541.235

floor\_28 67.3274 168.769 0.399 0.690 -263.507 398.162

floor\_29 220.8629 168.815 1.308 0.191 -110.061 551.787

floor\_3 -58.9196 23.573 -2.499 0.012 -105.129 -12.710

floor\_301 1.085e-13 7.6e-14 1.427 0.154 -4.05e-14 2.58e-13

floor\_32 -106.2029 335.861 -0.316 0.752 -764.585 552.179

floor\_35 7.349e-15 3.29e-14 0.223 0.823 -5.72e-14 7.19e-14

floor\_4 -44.5221 24.435 -1.822 0.068 -92.422 3.378

floor\_46 380.0206 336.142 1.131 0.258 -278.913 1038.954

floor\_5 -93.1974 25.589 -3.642 0.000 -143.359 -43.036

floor\_51 0 0 nan nan 0 0

floor\_6 -77.8223 26.153 -2.976 0.003 -129.090 -26.554

floor\_7 -77.7381 26.295 -2.956 0.003 -129.283 -26.193

floor\_8 -78.1553 26.567 -2.942 0.003 -130.234 -26.077

floor\_9 -19.6105 28.613 -0.685 0.493 -75.701 36.480

==============================================================================

Omnibus: 4116.684 Durbin-Watson: 1.993

Prob(Omnibus): 0.000 Jarque-Bera (JB): 465944.389

Skew: 1.684 Prob(JB): 0.00

Kurtosis: 41.531 Cond. No. 2.53e+16

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 2.7e-23. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

MAE: 214.158

RMSE: 399.011

R2: 0.986

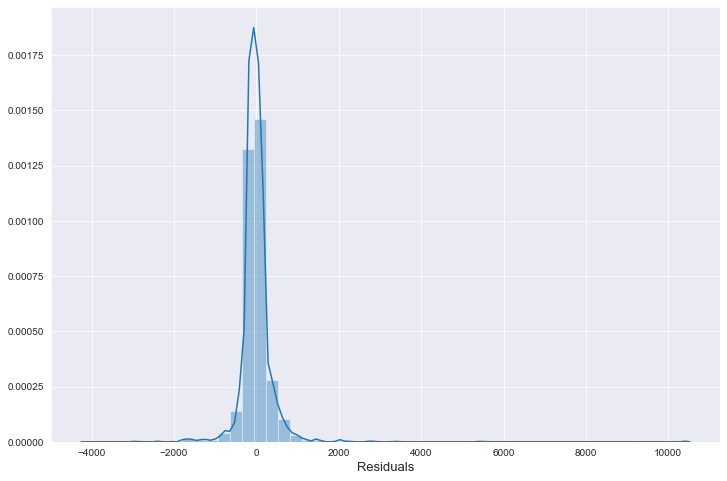
**#PREDICTION**

resid = y\_test - predict

ax = sns.distplot(resid)

ax.figure.set\_size\_inches(12,8)

ax.set\_xlabel('Residuals', fontsize=13)



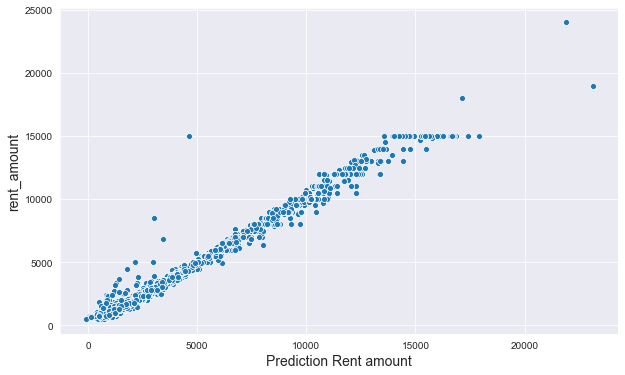
**#PLOTTING**

plt.figure(figsize=(10,6))

plt.xlabel('Prediction Rent amount', fontsize = 14)

plt.ylabel('rent\_amount', fontsize = 14)

sns.scatterplot(predict, y\_test)



**#COMPARING THE RESULTS**

attempts = ['first','second','third','fourth']

for attempt, metric in zip(attempts, metrics):

print(attempt, ':', metric)

**OUTPUT:**

first : 459.7260506804912

second : 399.0113689750958

third : 399.0113689750958

**INFERENCE:**

The test results shows that the r-square values(R2) is 1.00 with 0 RMSE which means the model is good enough to predict the house rent.

**RESULT:**

The value of r2(r-square) resembles 100% of accuracy score, which means the model is perfectly ready to predict the output(House rent). From my observation bathrooms, HOA property tax, fire insurance are the major factors that affects the house rent in Brazil. If this factors are high the rent of the house will be high, Otherwise the rent will decrease depends on the factors variation.