COIMBATORE INSTITUTE OF TECHNOLOGY

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

- THANGAVEL V (1832056)

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 numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt
```

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

```
df = pd.read_csv(r'C:/Users/THANGAVEL/Desktop/houses_to_rent.csv')

df.head()
```

OUTPUT:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
0	São Paulo	70	2	1	1	7	acept	furnished	2065	3300	211	42	5618
1	São Paulo	320	4	4	0	20	acept	not furnished	1200	4960	1750	63	7973
2	Porto Alegre	80	1	1	1	6	acept	not furnished	1000	2800	0	41	3841
3	Porto Alegre	51	2	1	0	2	acept	not furnished	270	1112	22	17	1421
4	São Paulo	25	1	1	0	1	not acept	not furnished	0	800	25	11	836

#EXPLORATORY DATA ANALYSIS

#ALL COLUMNS IN DATASET

all_cols = df.columns

all_cols

OUTPUT:

```
Index(['city', 'area', 'rooms', 'bathroom', 'parking spaces', 'floor',
       'animal', 'furniture', 'hoa (R$)', 'rent amount (R$)',
       'property tax (R$)', 'fire insurance (R$)', 'total (R$)'],
      dtype='object')
```

#INFORMATION ABOUT THE DATA FRAME

df.info()

OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10692 entries, 0 to 10691
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   city                 10692 non-null  object
 1   area                 10692 non-null  int64
 2   rooms                10692 non-null  int64
 3   bathroom             10692 non-null  int64
 4   parking spaces       10692 non-null  int64
 5   floor                10692 non-null  object
 6   animal               10692 non-null  object
 7   furniture            10692 non-null  object
 8   hoa (R$)             10692 non-null  int64
 9   rent amount (R$)     10692 non-null  int64
 10  property tax (R$)    10692 non-null  int64
 11  fire insurance (R$)  10692 non-null  int64
 12  total (R$)           10692 non-null  int64
dtypes: int64(9), object(4)
memory usage: 1.1+ MB
```

#SUMMARY STATISTICS

df.describe().round(3)

OUTPUT:

	area	rooms	bathroom	parking spaces	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
count	10692.000	10692.000	10692.000	10692.000	10692.000	10692.000	10692.000	10692.000	10692.000
mean	149.218	2.506	2.237	1.609	1174.022	3896.247	366.704	53.301	5490.487
std	537.017	1.171	1.407	1.590	15592.305	3408.546	3107.832	47.768	16484.726
min	11.000	1.000	1.000	0.000	0.000	450.000	0.000	3.000	499.000
25%	56.000	2.000	1.000	0.000	170.000	1530.000	38.000	21.000	2061.750
50%	90.000	2.000	2.000	1.000	560.000	2661.000	125.000	36.000	3581.500
75%	182.000	3.000	3.000	2.000	1237.500	5000.000	375.000	68.000	6768.000
max	46335.000	13.000	10.000	12.000	1117000.000	45000.000	313700.000	677.000	1120000.000

#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()

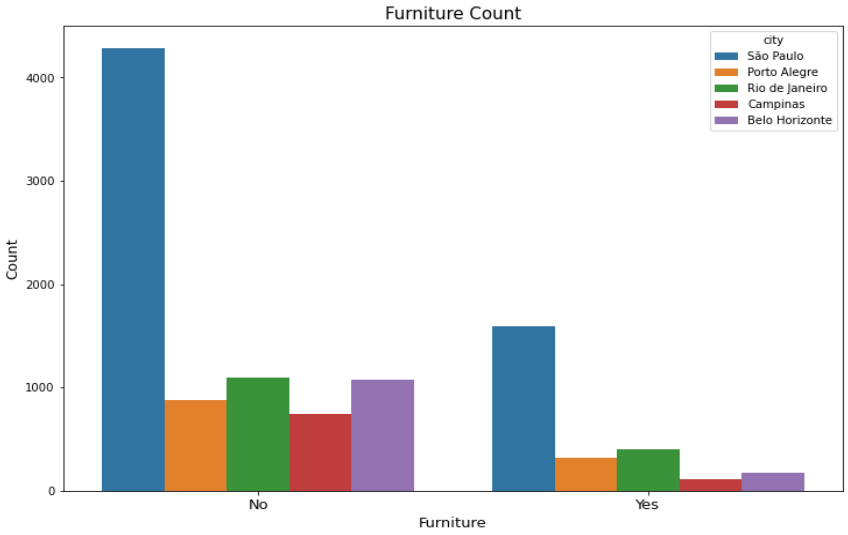
OUTPUT:

	city	area	rooms	bathroom	parking_spaces	floor	animal	furniture	hoa	rent_amount	property_tax	fire_insurance	total
0	São Paulo	70	2	1	1	7	1	1	2065	3300	211	42	5618
1	São Paulo	320	4	4	0	20	1	0	1200	4960	1750	63	7973
2	Porto Alegre	80	1	1	1	6	1	0	1000	2800	0	41	3841
3	Porto Alegre	51	2	1	0	2	1	0	270	1112	22	17	1421
4	São Paulo	25	1	1	0	1	0	0	0	800	25	11	836

#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)
```

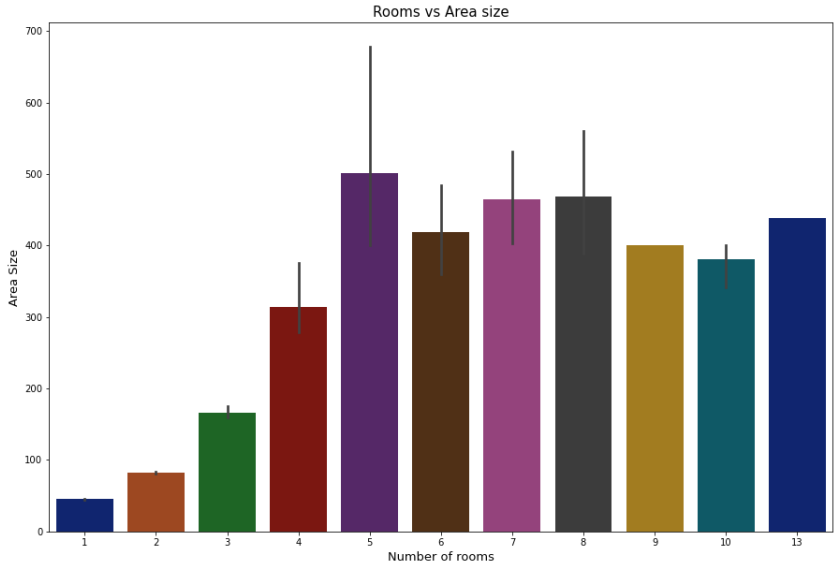
OUTPUT:



#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)
```

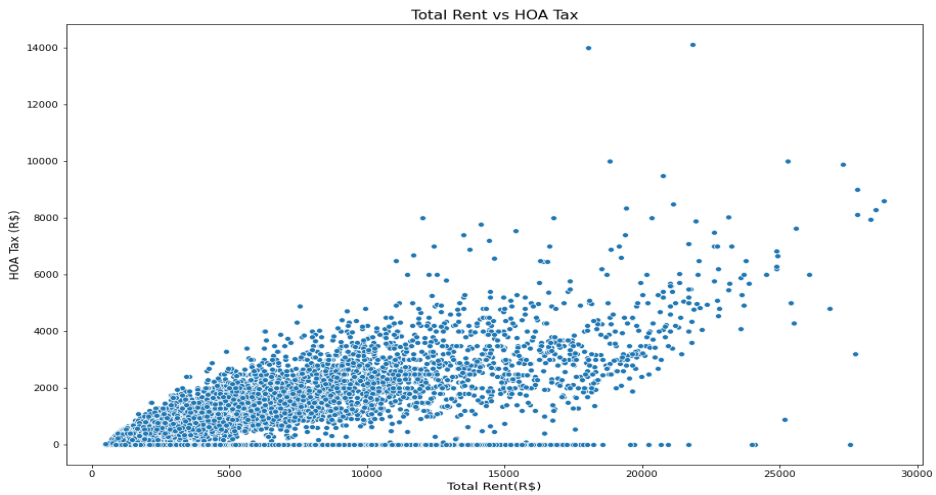
OUTPUT:



#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)
```

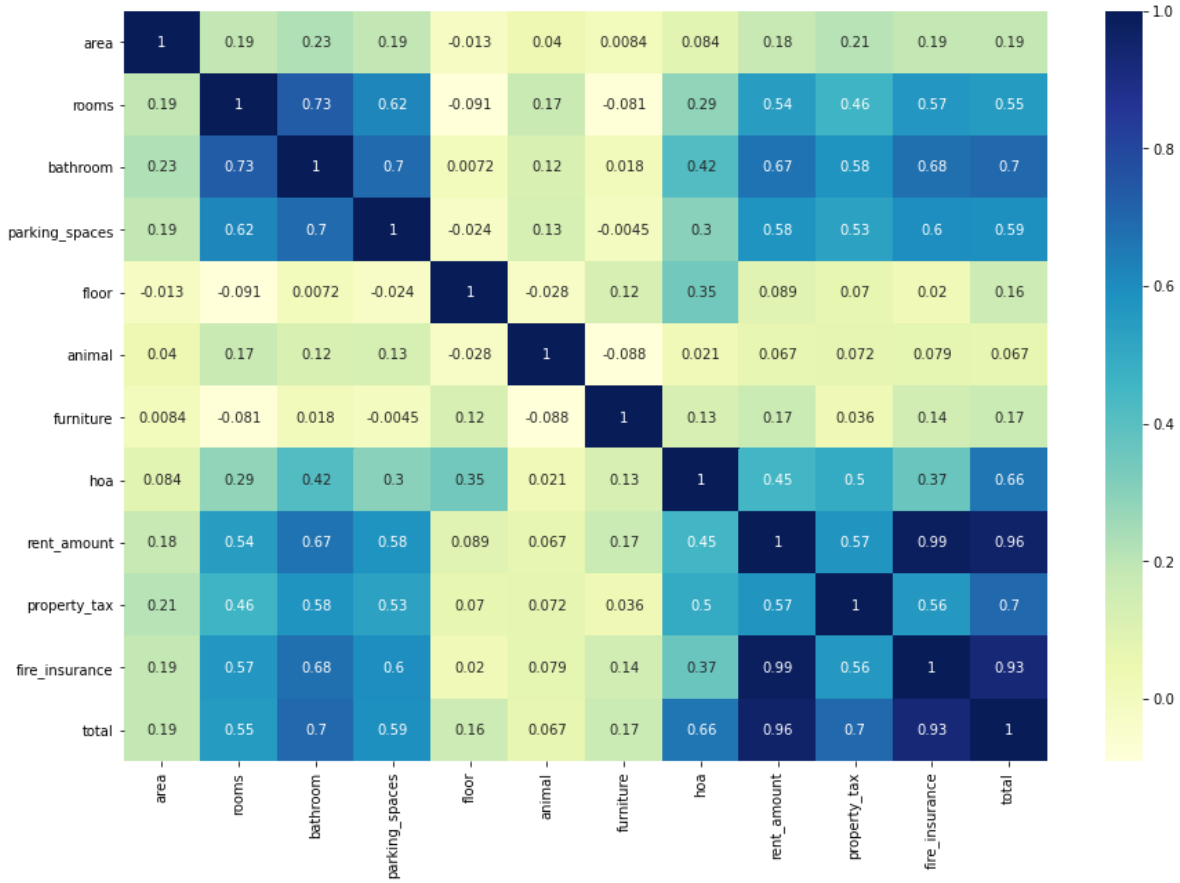
OUTPUT:



#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
```

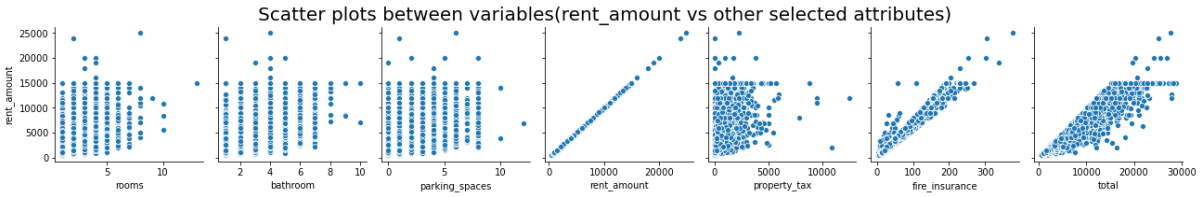
OUTPUT:

```
Index(['rooms', 'bathroom', 'parking_spaces', 'rent_amount', 'property_tax',
       'fire_insurance', 'total'],
      dtype='object')
```

#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:



#SELECTING X AND Y VALUES

```
metrics = []

y = df['rent_amount']

x = df[req_cols]
```

#IMPORTING PACKAGES FOR MODELS, SPLIT AND ACCURACY SCORES

```
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
```

#SPLITTING INTO TRAINING AND TEST DATA USIG TRAIN_TEST_SPLIT

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state = 8)
```

#FITTING AND TRAINING MODEL

```
lr = LinearRegression()

lr.fit(x_train, y_train)

predict = lr.predict(x_test)
```

#ADD A CONSTANT AND LOOKING THE SUMMARY

```
import statsmodels.api as sm

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:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 4.984e+31
Date:                Sun, 07 Mar 2021   Prob (F-statistic):               0.00
Time:                        10:03:03   Log-Likelihood:             1.7544e+05
No. Observations:                7474   AIC:                        -3.509e+05
Df Residuals:                    7466   BIC:                        -3.508e+05
Df Model:                           7
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const          -1.421e-13   4.48e-13     -0.317      0.751   -1.02e-12    7.36e-13
rooms           2.402e-12   2.35e-13     10.209      0.000    1.94e-12    2.86e-12
bathroom       -1.592e-12   2.37e-13     -6.728      0.000   -2.06e-12   -1.13e-12
parking_spaces  8.171e-13   1.68e-13      4.860      0.000    4.88e-13    1.15e-12
rent_amount        1.0000   6.23e-16   1.61e+15      0.000       1.000       1.000
property_tax    7.494e-16    5.4e-16      1.388      0.165   -3.09e-16    1.81e-15
fire_insurance -5.329e-14   3.15e-14     -1.690      0.091   -1.15e-13    8.54e-15
total          -2.151e-16   2.41e-16     -0.894      0.371   -6.87e-16    2.57e-16
==============================================================================
Omnibus:                     1862.698   Durbin-Watson:                   0.672
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3897.781
Skew:                          -1.471   Prob(JB):                         0.00
Kurtosis:                       4.965   Cond. No.                     2.24e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.24e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
MAE:  0.0
RMSE:  0.0
R2: 1.0
```

INFERENCE:

The test results shows that the r-square values(R2) is 1.00 with 0 RMSE and)MAE, 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.