Shedding Light the Linear Regression

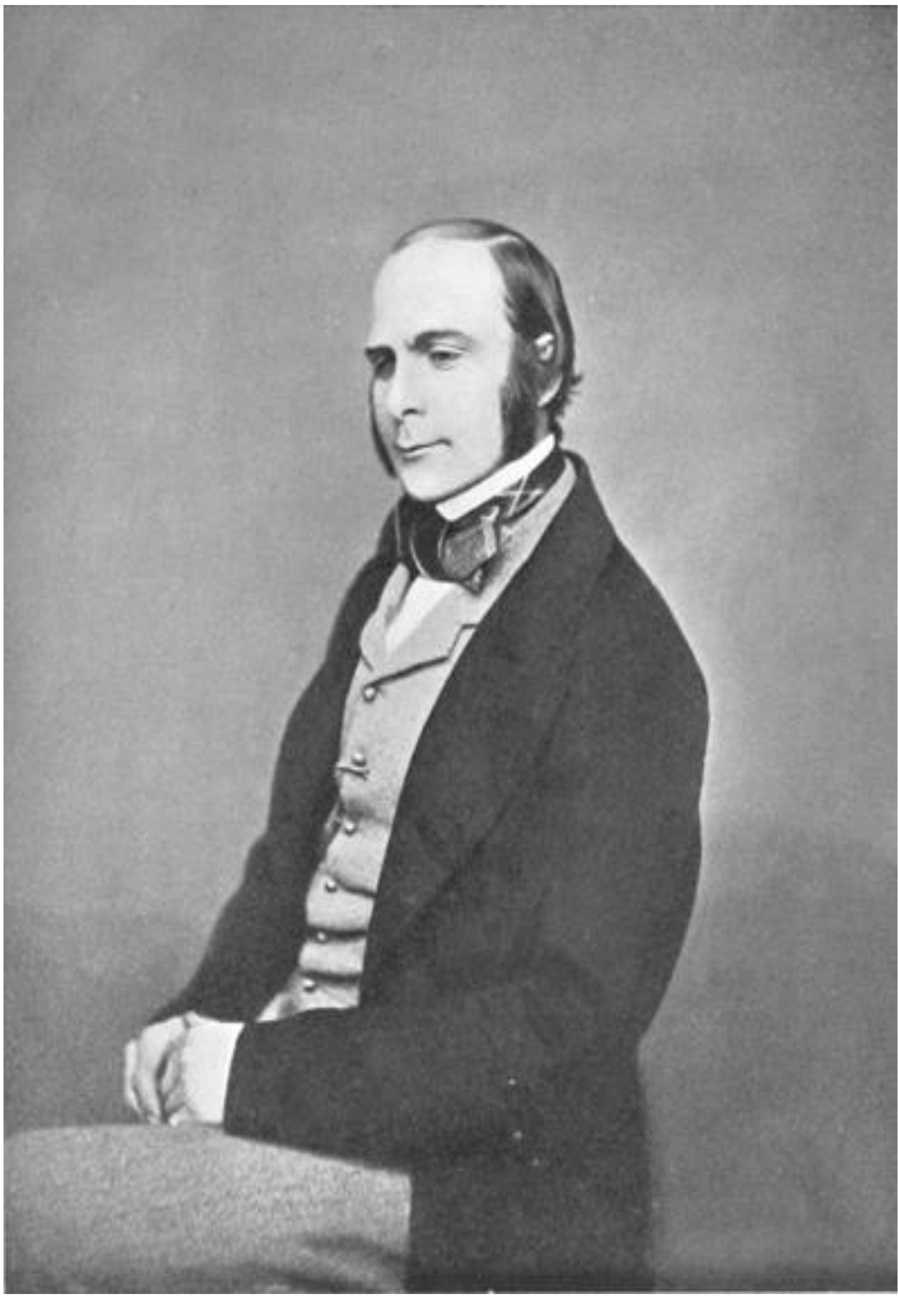
Basic Concepts

An overview about one of the most important algorithms, unfortunately often forgotten for statisticians and data scientists

Linear Regression is particularly important statistical content, it was one of the first ways of trying to identify patterns to make predictions and the basis for developing most other subsequent predict models.

The first notions about linear regression were developed by Francis Galton in 1885, he used this science by applying into Anthropometry (The science of measurement and mathematics of the human body), he observed that children of short parents tend to be taller, and children of tall parents tend to be shorter. In other words, the heights of human beings in general **tend to regress** to the average.

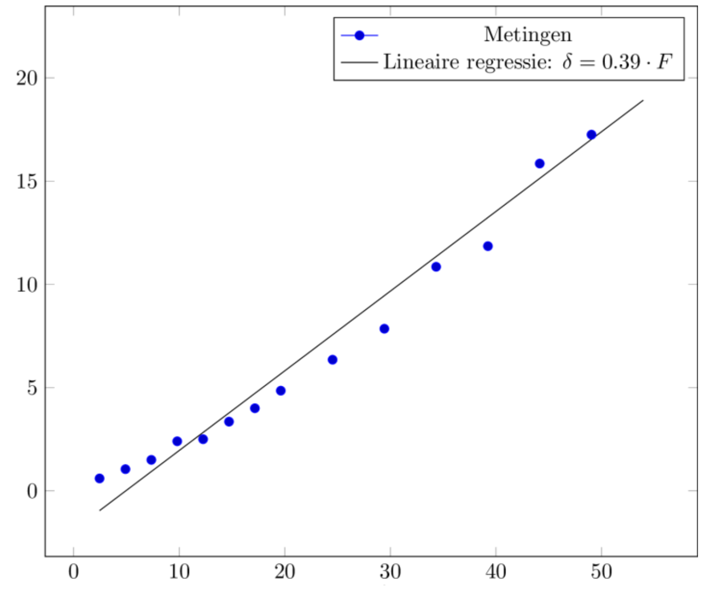
The calculus of linear regression is not simple using only the hands, but with the technological growth, got easier to apply the formula to solve of most diverse “linear problems”.



Written by Leonardo Mota

Initial Concepts

The objective of linear regression is to try find a line that better adjust between the data.



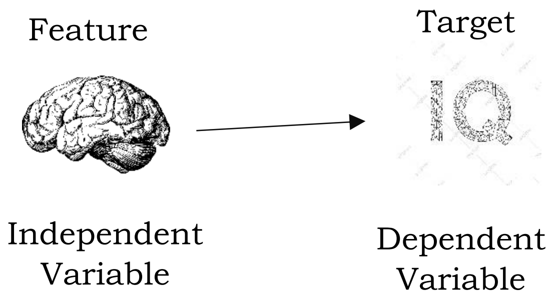
There are two categories of variables, **feature** also called **independent** variable**,** and **target** aka **dependent** variable.

Target is the variable that you want to predict, and feature is the variable that will try to explain how the target variable behaves.

Example

Suppose we want to use Brain Weight to predict how much IQ a person has. IQ is the target variable that will try to be predicted by brain weight.

That is the core objective of a linear regression model, identify correlations to make predictions.



Formula

$$\underset{\text{Dependent Variable}}{Y_i} = \overset{\text{Constant/Intercept}}{\beta_0} + \underset{\text{Slope/Coefficient}}{\beta_1} \overset{\text{Independent Variable}}{X_i}$$

Written by Leonardo Mota

Case Study

Suppose you are a bar owner and want to predict the beer consume, to intend manage better your inventory, decide if you put more tables and chairs or even raises the price, in other words understand better your public to make a better governance. The most of managers, use the famous “felling”. Of course, a knowledgeable person can hit sometimes, but the experience is something that acquirer with a lot of time and the human beings always is liable to silly mistakes, doesn’t matter how experience they are. Being better the use of mathematics to aid decisions.

You have the following Dataset and want to use this to make predictions and answers the followings doubts

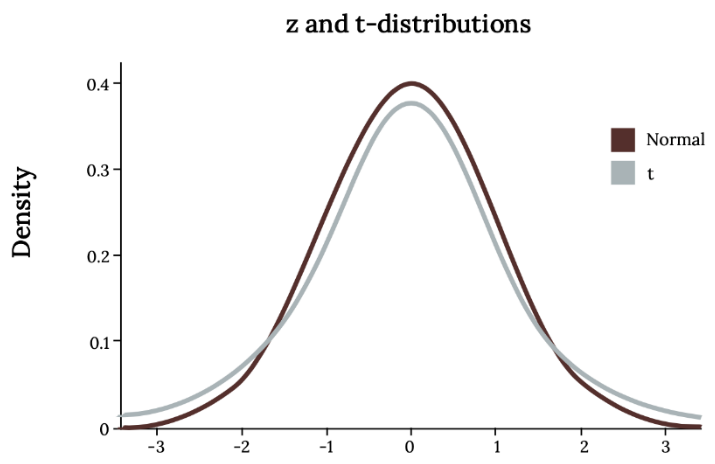
	Date	Average_Temp	Min_Temp	Max_Temp	Precipitation	Weekend	Consume
0	2015-01-01	27.30	23.9	32.5	0.0	0.0	25.461
1	2015-01-02	27.02	24.5	33.5	0.0	0.0	28.972
2	2015-01-03	24.82	22.4	29.9	0.0	1.0	30.814
3	2015-01-04	23.98	21.5	28.6	1.2	1.0	29.799
4	2015-01-05	23.82	21.0	28.3	0.0	0.0	28.900

- Is it possible use all variables?
- Is the Linear Regression the correct algorithm?
- What is the first steps?
- Which variables I would use or discard?
- What prevents to use a variable?
- Is my model valid?
- And more

Written by Leonardo Mota

Should I use Linear Regression in this dataset to make predictions?

It's not in all datasets that you can use Linear Regression, exist some prerequisites to be accomplished for example, it is primal that the output variable follows a normal distribution

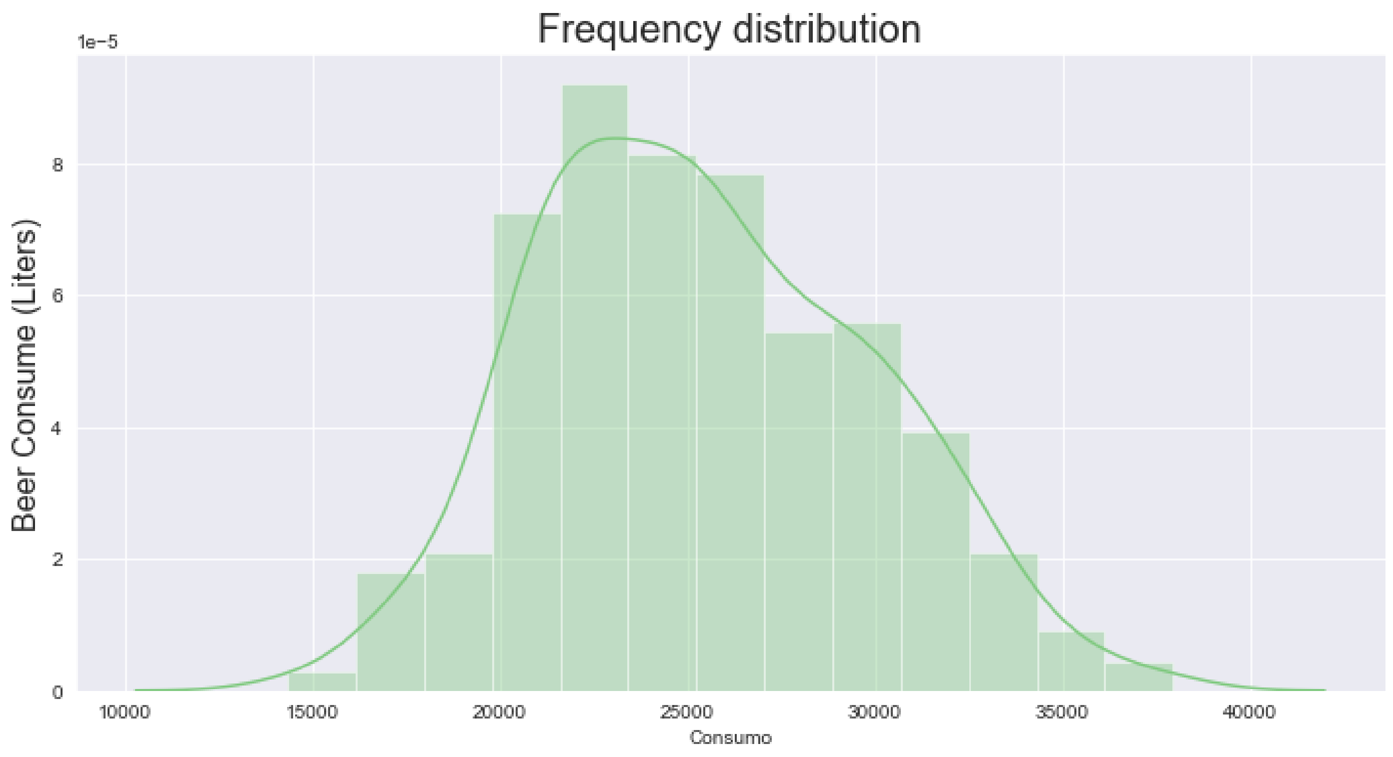


Informal Test

Checking if the output Variable follows a normal distribution just looking at the graphic

```
ax = sns.distplot(data['Consume'])
ax.figure.set_size_inches(12,6)
ax.set_title('Frequency distribution',fontsize = 20)
ax.set_ylabel('Beer Consume(Liters)',fontsize = 16)

plt.show()
```



It seems to be a normal distribution, what is a good signal if you want to use Linear Regression

Written by Leonardo Mota

Formal Test

Checking if the output Variable follows a normal distribution doing a hypothesis test using the P-value

```
import scipy.stats as stats
from scipy.stats import normaltest
from scipy import stats

stat_test,p_value = normaltest(data['Consumo'])

p_value = p_value

if p_value < 0.05:
    print(f'This dataset follows a normal distribution, p-value = {round(p_value,4)}')
else:
    print(f"this dataset doens't follor a noraml distribution, p-value = {round(p_value,4)}")
```

RESULT

H_0: P-value ≥ 0,05

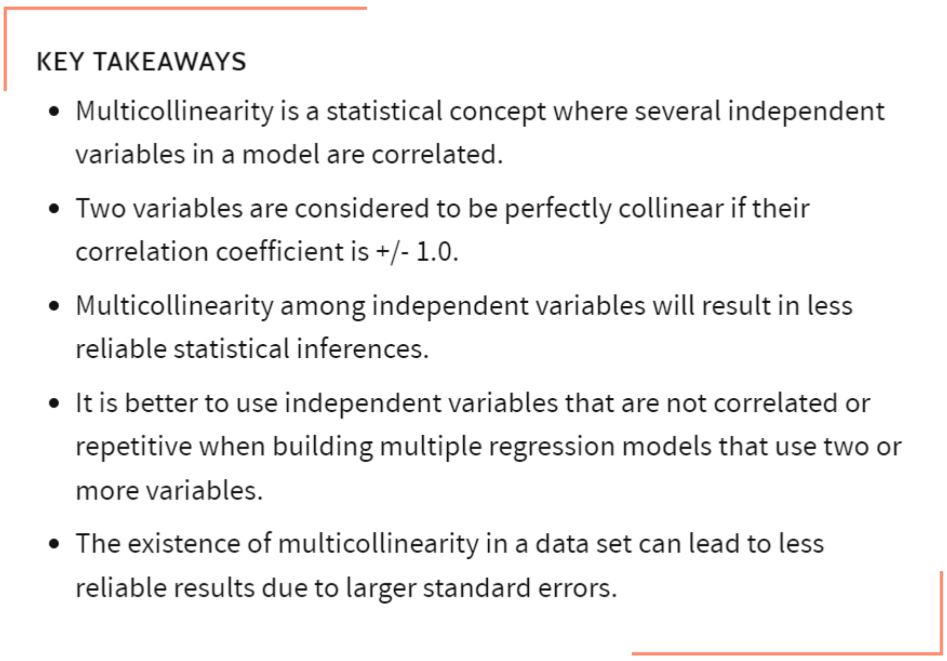
H_1: P-value < 0,05

P-value = **0.0197** Disregard the null hypothesis, the dependent variable follows a normal distribution

Should I use all independent variables to make the prediction?

The problem of Multicollinearity

Multicollinearity is the occurrence of high intercorrelations among two or more independent variables in a multiple regression model. Multicollinearity can lead to skewed or misleading results when a researcher or analyst attempts to determine how well each independent variable can be used most effectively to predict or understand the dependent variable in a statistical model.



KEY TAKEAWAYS

- Multicollinearity is a statistical concept where several independent variables in a model are correlated.
- Two variables are considered to be perfectly collinear if their correlation coefficient is +/- 1.0.
- Multicollinearity among independent variables will result in less reliable statistical inferences.
- It is better to use independent variables that are not correlated or repetitive when building multiple regression models that use two or more variables.
- The existence of multicollinearity in a data set can lead to less reliable results due to larger standard errors.

Causes of Multicollinearity?

Multicollinearity can exist when two independent variables are highly correlated. It can also happen if an independent variable is computed from other variables in the data set or if two independent variables provide similar and repetitive results.

Effects of Multicollinearity?

Although multicollinearity does not affect the regression estimates, it makes them vague, imprecise, and unreliable. Thus, it can be hard to determine how the independent variables influence the dependent variable individually. This inflates the standard errors of some or all of the regression coefficients.

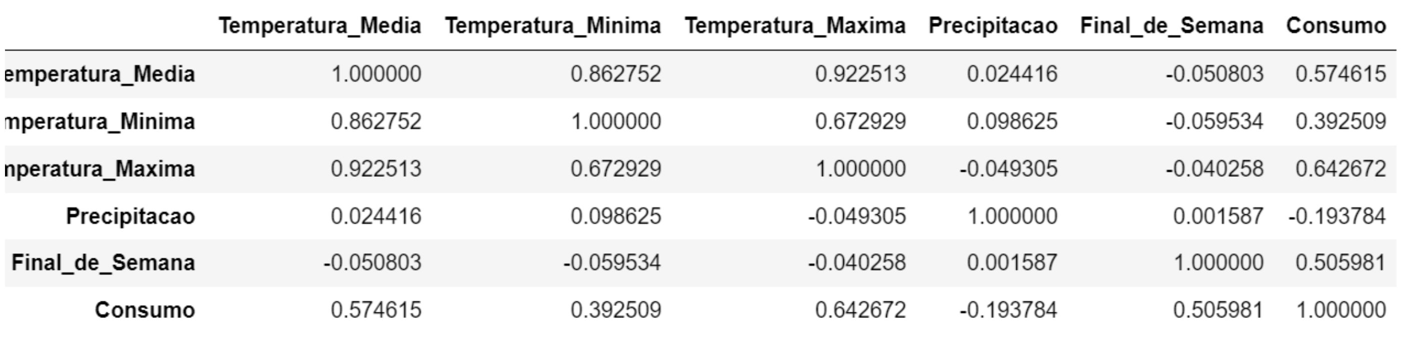
This page is a resume of an article than can be found in the link below

https://www.investopedia.com/terms/m/multicollinearity.asp#:~:text=Multicollinearity%20is%20a%20statistical%20concept,in%20less%20reliable%20statistical%20inferences.

What everyone does

A lot of statisticians and data scientists try to solve this problem using the Pearson Correlation, but it is not enough and exist cases that **correlation is not multicollinearity**.

```
data.corr(method='pearson')
```



	Temperatura_Media	Temperatura_Minima	Temperatura_Maxima	Precipitacao	Final_de_Semana	Consumo
emperatura_Media	1.000000	0.862752	0.922513	0.024416	-0.050803	0.574615
mperatura_Minima	0.862752	1.000000	0.672929	0.098625	-0.059534	0.392509
nperatura_Maxima	0.922513	0.672929	1.000000	-0.049305	-0.040258	0.642672
Precipitacao	0.024416	0.098625	-0.049305	1.000000	0.001587	-0.193784
Final_de_Semana	-0.050803	-0.059534	-0.040258	0.001587	1.000000	0.505981
Consumo	0.574615	0.392509	0.642672	-0.193784	0.505981	1.000000

what everyone should do

One of the best ways to analyze if there is multicollinearity is using the VIF (Variance Inflator Indicator)

VIF definition second by Wikipedia

In statistics, the **variance inflation factor** (**VIF**) is the ratio (quotient) of the variance of estimating some parameter in a model that includes multiple other terms (parameters) by the variance of a model constructed using only one term.[1] It quantifies the severity of multicollinearity in an ordinary least squares regression analysis. It provides an index that measures how much the variance (the square of the estimate's standard deviation) of an estimated regression coefficient is increased because of collinearity.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor


# the independent variables set

X = data[['Temperatura_Media', 'Temperatura_Minima', 'Temperatura_Maxima',
          'Precipitacao','Final_de_Semana']]

# VIF dataframe

vif_data = pd.DataFrame()
vif_data["feature"] = X.columns

# Calculating VIF for each feature

vif_data["VIF"] = [variance_inflation_factor(X.values, i)
                          for i in range(len(X.columns))]

print(vif_data)
```

```
     feature          VIF

0    Average_Temp    1087.797835
1    Min_Temp         249.112601
2    Max_Temp         436.529190
3    Precipitation      1.220276
4    Weekend            1.374283
```

How to interpreter this?

VIF greater than 5 is considered too large and must be put away.

```
X = data[['Temperatura_Maxima',
         'Precipitacao','Final_de_Semana']]

# VIF dataframe

vif_data = pd.DataFrame()
vif_data["feature"] = X.columns

# Calculating VIF for each feature

vif_data["VIF"] = [variance_inflation_factor(X.values, i)
                          for i in range(len(X.columns))]

print(vif_data)
```

```
     feature          VIF

0    Max_Temp         1.528790
1    Precipitation    1.163160
2    Weekend          1.373842
```

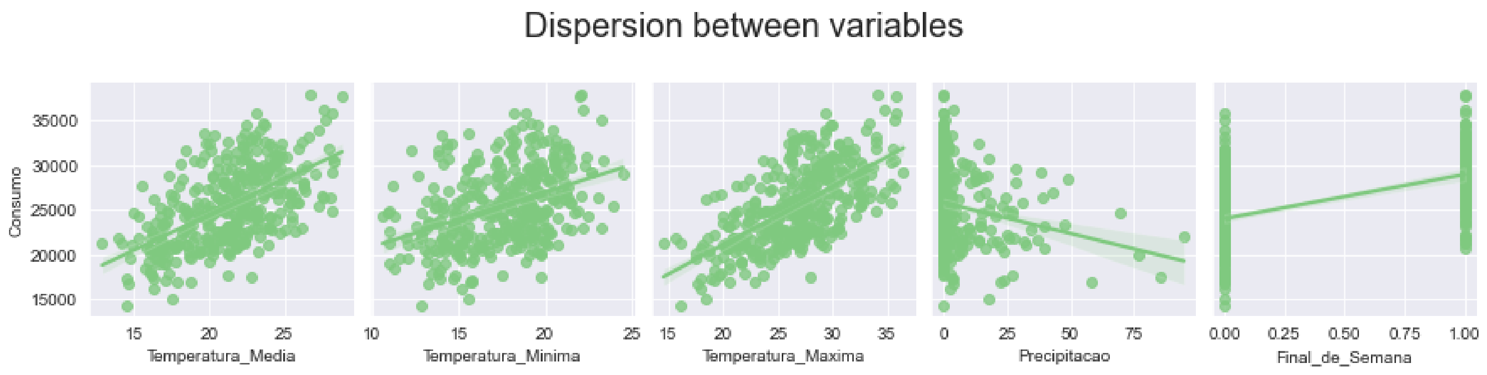
Now we have a better dataset to make the prediction and acquire insights

Analyzing the dispersion of variables

Is important to understand how the independents variables (X) relate with the dependent variable (Y)

```
ax = sns.pairplot(data,y_vars = 'Consumo', x_vars = ['Temperatura_Media','Temperatura_Minima',
                                                      'Temperatura_Maxima','Precipitacao',
                                                      'Final_de_Semana'],kind = 'reg')

ax.fig.suptitle('Dispersion between variables',fontsize = 20,y = 1.20)
plt.show()
```



At first there doesn't seem to be anything wrong with the variables and just looking to this graphics we can obtain the next insights

- People Consume more with the higher temperature
- The rain drives away consumers
- People consume more during the weekend

Creating and training the model

```
from sklearn.model_selection import train_test_split

y = data['Consumo']
x = data[['Temperatura_Maxima','Precipitacao',
         'Final_de_Semana']]

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.30,random_state = 2811)

from sklearn.linear_model import LinearRegression
from sklearn import metrics

model = LinearRegression()
model.fit(x_train,y_train)
print(f'The R² is de {round(model.score(x_test,y_test),2)}')
```

Obtaining the coefficients

```
index = ['Intercept','Max Temperature','Precipitation (mm)','Weekend']

pd.DataFrame(data = np.append(model.intercept_,model.coef_),
             index = index,columns = ['Parameters'])
```

	Parameters
Intercept	5951.976339
Max Temperature	684.736759
Precipitation (mm)	-60.782435
Weekend	5401.083339

Using these parameters, we can find the formula that will be used to make ***predictions*** and obtain **deeper *insights***

X_1 – Max Temperature
X_2 – Precipitation
X_3 - Weekend

$$\hat{Y} = \beta_0 + \beta_1 X_1 - \beta_2 X_2 + \beta_3 X_3$$

$$\hat{Y} = 5951.98 + 684.74\, X_1 - 60.78\, X_2 + 5401\, X_3$$

$$\hat{Y} = 5951.98 + 684.74 \times 23.78 - 60.78 \times 12.2 + 5401 \times 0$$

$$\hat{Y} \cong \boldsymbol{21{,}493.58}$$

Insights

Max Temperature - For every 1 degree more temperature, consumption tends to grow by 684.74 liters

Precipitation – The greater the precipitation, the less consumption tends to be

Weekend – During the weekend the beer consumption grow about 5401 liters

Observe that looking the graphics we have answers about the consume behavior, but we don’t have numbers

Analyzing the accuracy

Definitions

Mean absolute error: This is the average of absolute errors of all the data points in the given dataset.

Mean squared error: This is the average of the squares of the errors of all the data points in the given dataset. It is one of the most popular metrics out there!

Median absolute error: This is the median of all the errors in the given dataset. The main advantage of this metric is that it's robust to outliers. A single bad point in the test dataset wouldn't skew the entire error metric, as opposed to a mean error metric.

Explained variance score: This score measures how well our model can account for the variation in our dataset. A score of 1.0 indicates that our model is perfect.

R2 score: This is pronounced as R-squared, and this score refers to the coefficient of determination. This tells us how well the unknown samples will be predicted by our model. The best possible score is 1.0, but the score can be negative as well.

```
from sklearn.metrics import mean_absolute_error,r2_score,median_absolute_error
from sklearn.metrics import explained_variance_score
from sklearn.metrics import mean_squared_error,mean_absolute_error
import numpy as np

y_pred = model.predict(x_test)

print("Mean absolute error :", round(mean_absolute_error(y_test,y_pred),2))
print("Mean squared error :", round(mean_squared_error(y_test,y_pred),2))
print("Median absolute error :", round(median_absolute_error(y_test,y_pred),2))
print("Explained variance score :", round(explained_variance_score(y_test,y_pred),5))
print("R² : ", round(r2_score(y_test, y_pred),5))
```

```
Mean absolute error: 1966.56
Mean squared error: 5471976.38
Median absolute error: 1852.72
Explained variance score: 0.69075
R²: 0.69074
```

Analyzing if the model is valid

Definition

To have a valid model we need a dataset that meets the next topics

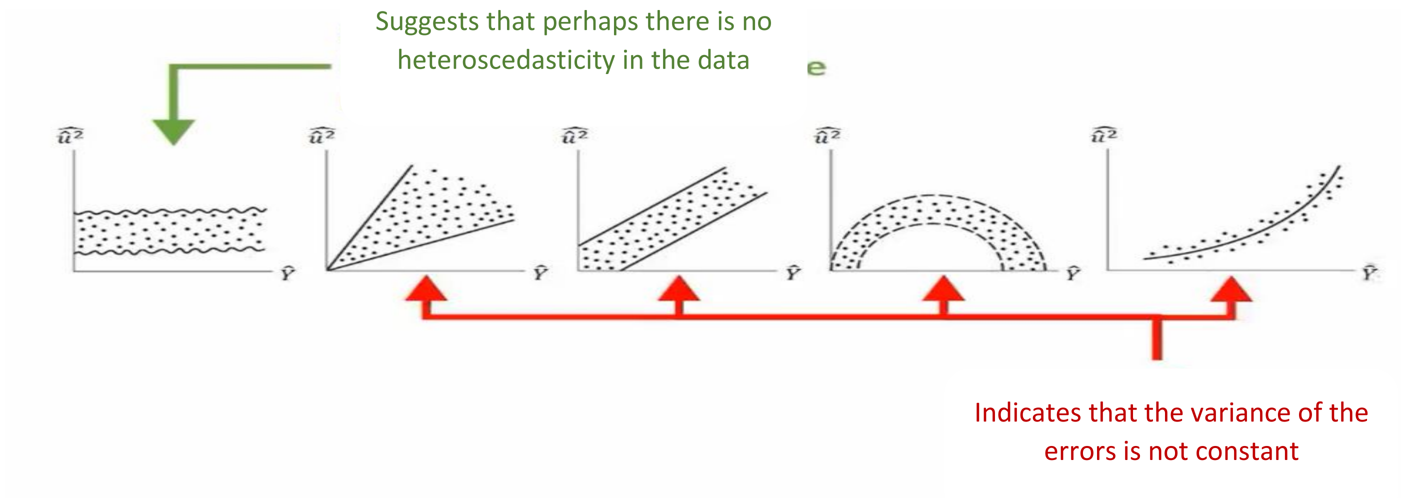
- **Weak Exogeneity**: We assume that the predictor variables are error free, for example that there are no field measurement errors.
- **Linearity**: The mean of the response variable is a linear combination of the parameters and the predictor variables.
- **Homoscedasticity**: Different values of the response variable have the same variance in their errors (homogeneity of variance).
- **Independence of Errors**: Errors of the response variables are uncorrelated with each other.
- **Lack of Perfect Multicollinearity**: For least square estimation methods, the design matrix X must have full rank, otherwise it could mean that one predictor variable is given twice.

We've already dealt in the last few pages about exogeneity using a reliable dataset, Linearity looking at the graph and doing the hypothesis test and solving the multicollinearity problem by dropping some columns.

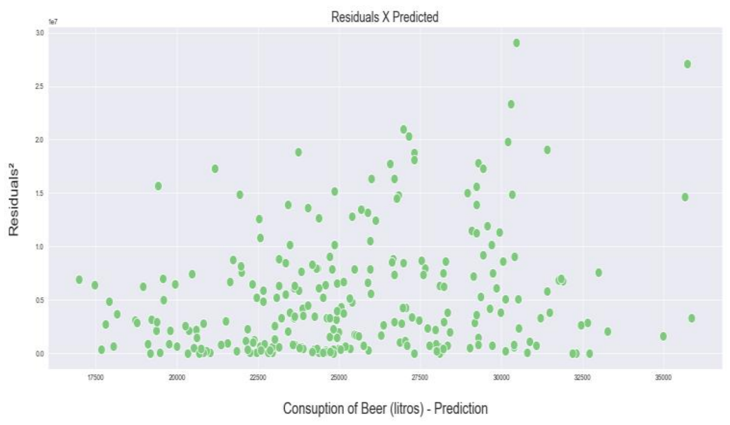
But after training the model we need to see the residuals of this Linear Regression.

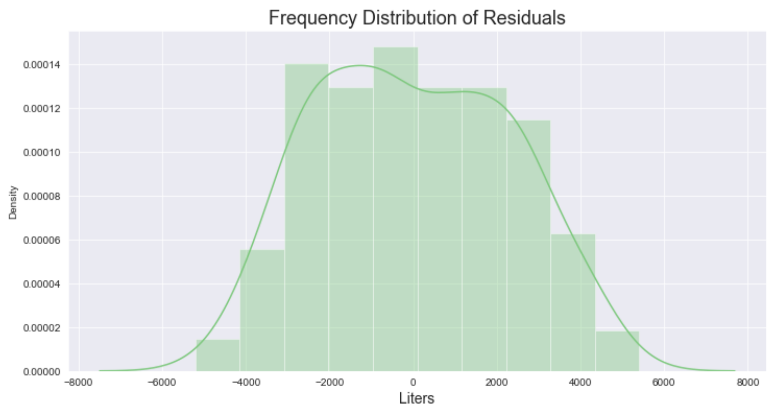
Residuals - The difference between an observed value of the response variable and the value of the response variable predicted from the regression line.

How should residuals behave?


Suggests that perhaps there is no heteroscedasticity in the data
e
$\widehat{u^2}$
$\hat{Y}$
Indicates that the variance of the errors is not constant

We need to avoid the Homoscedasticity, and our graphic of residuals its similar to the second graphic in the image that is a good thing. Another form to analyze if the residuals its 0k is looking for the distribution of the residuals, they must follow a normal distribution for the model to be valid





Our model is valid and can be used to make prediction and the insights can be considered true

Written by Leonardo Mota

FINAL CONSIDERATIONS

This article was a short summary on basic Linear Regression topics, this subject is very broad and of course there is much more that can be done using this concept, I will try to write more about it soon. Feel free to comment or criticize the article, I think this is very healthy for the data science community and thank you very much for reading and I hope I have added some value to you reader.

Best Regards – Leonardo Mota