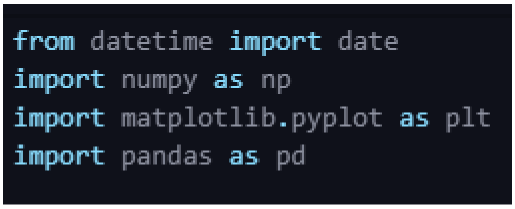
Beer Consumption Analysis using Linear Regression.

Dataset : Consumo_cerveja.csv

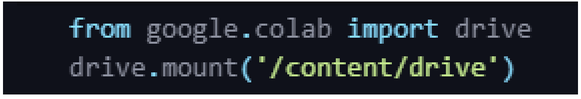
Before performing operations on any dataset, some of the initial steps are always common, the steps followed in this Linear Regression Problem are :

1. Importing Libraries : Importing relevant libs such as datetime, NumPy, matplotlib.pyplot, pandas, sklearn.linearmodel, sklearn.model_selection.



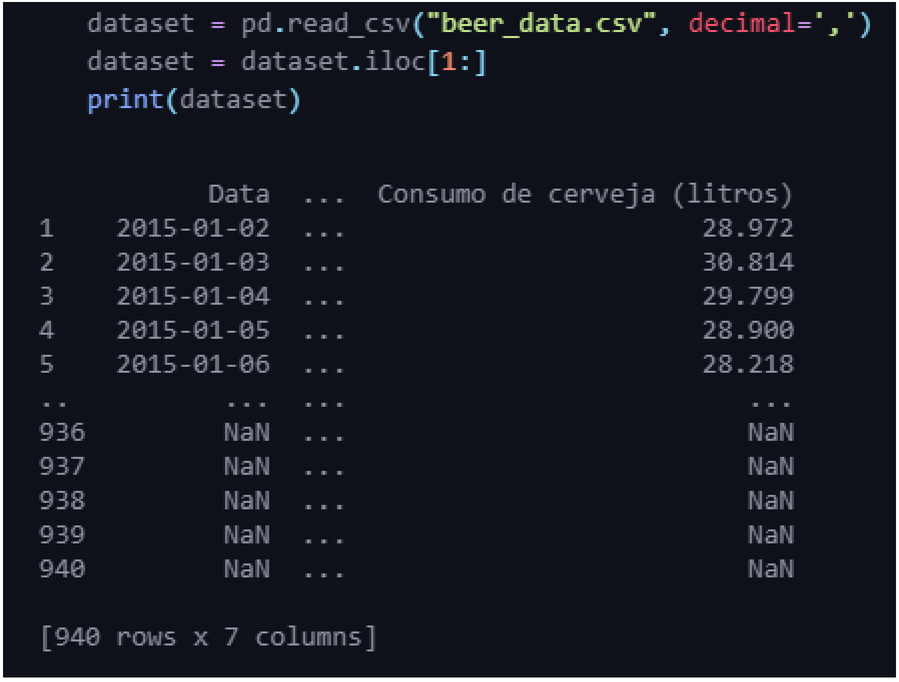
```
from datetime import date
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

2. Linking Google Drive (optional, can directly import csv from project folder) : using google.colab lib and drive attribute, we can access files inside our drive by providing the authorization code and path.



```
from google.colab import drive
drive.mount('/content/drive')
```

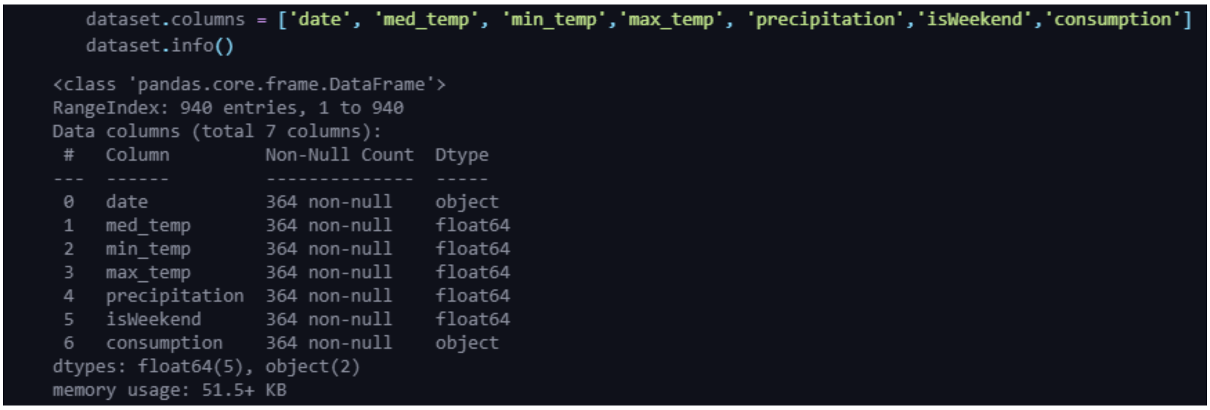
3. Importing the data : Using the read_csv function from the pandas lib, the dataset is stored into a variable, here named, ‘dataset’. (I downloaded and renamed the dataset file so that I don’t have to link drive every time)



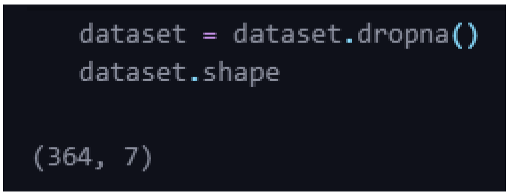
```
dataset = pd.read_csv("beer_data.csv", decimal=',')
dataset = dataset.iloc[1:]
print(dataset)


          Data  ...  Consumo de cerveja (litros)
1   2015-01-02  ...                       28.972
2   2015-01-03  ...                       30.814
3   2015-01-04  ...                       29.799
4   2015-01-05  ...                       28.900
5   2015-01-06  ...                       28.218
..         ...  ...                          ...
936        NaN  ...                          NaN
937        NaN  ...                          NaN
938        NaN  ...                          NaN
939        NaN  ...                          NaN
940        NaN  ...                          NaN

[940 rows x 7 columns]
```

As we see, initially the dataset provided has 940 rows and 7 columns

4. Fine tuning the dataset : Includes adding column names to the values to act as constraints in the late stages, and dropping empty cells (ones showing NaN) from the dataset using the dropna() function.



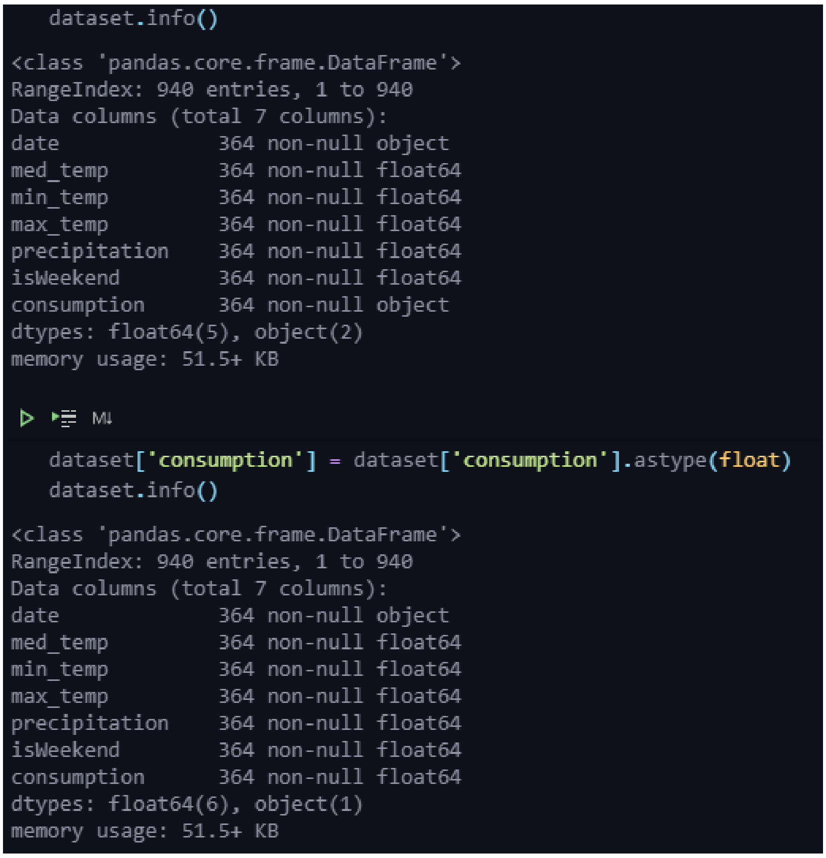
```
    dataset.columns = ['date', 'med_temp', 'min_temp','max_temp', 'precipitation','isWeekend','consumption']
    dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 1 to 940
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   date           364 non-null    object
 1   med_temp       364 non-null    float64
 2   min_temp       364 non-null    float64
 3   max_temp       364 non-null    float64
 4   precipitation  364 non-null    float64
 5   isWeekend      364 non-null    float64
 6   consumption    364 non-null    object
dtypes: float64(5), object(2)
memory usage: 51.5+ KB
```



```
    dataset = dataset.dropna()
    dataset.shape

(364, 7)
```

After dropping the empty cells, the number of rows and columns lay 364x7

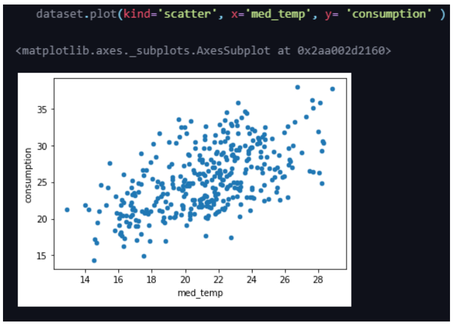
5. Typecasting : After checking all the constraints we find out that the consumption field is saved as an object class, so we convert it into float for being able to plot data.



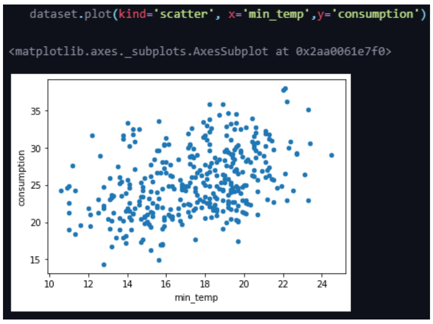
```
   dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 1 to 940
Data columns (total 7 columns):
date             364 non-null object
med_temp         364 non-null float64
min_temp         364 non-null float64
max_temp         364 non-null float64
precipitation    364 non-null float64
isWeekend        364 non-null float64
consumption      364 non-null object
dtypes: float64(5), object(2)
memory usage: 51.5+ KB
```

```
   dataset['consumption'] = dataset['consumption'].astype(float)
   dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 1 to 940
Data columns (total 7 columns):
date             364 non-null object
med_temp         364 non-null float64
min_temp         364 non-null float64
max_temp         364 non-null float64
precipitation    364 non-null float64
isWeekend        364 non-null float64
consumption      364 non-null float64
dtypes: float64(6), object(1)
memory usage: 51.5+ KB
```

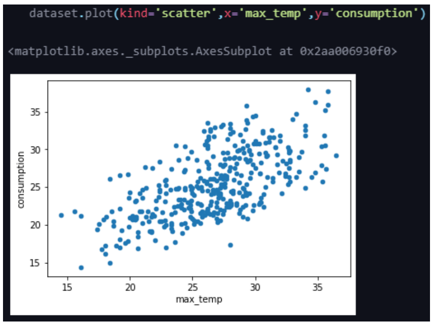
6. The most exciting part : Hence, the plotting –



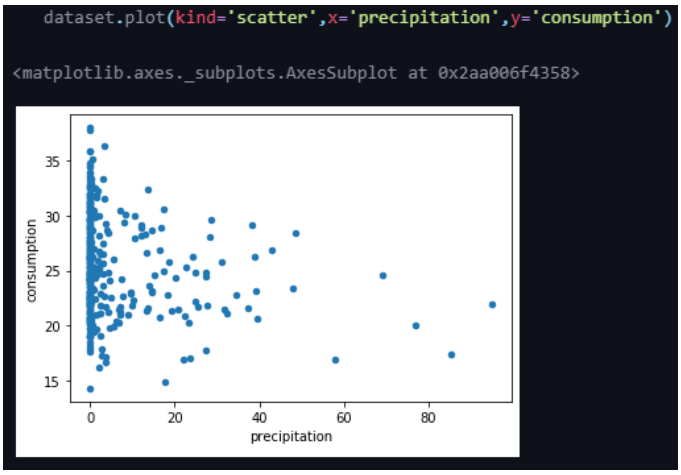
```
dataset.plot(kind='scatter', x='med_temp', y= 'consumption' )

<matplotlib.axes._subplots.AxesSubplot at 0x2aa002d2160>
```



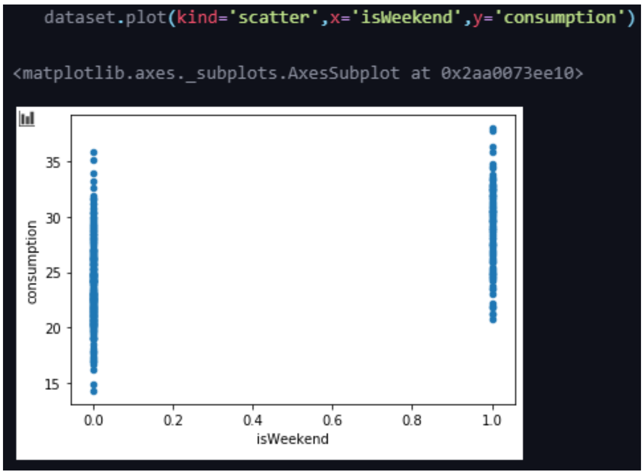
```
dataset.plot(kind='scatter', x='min_temp',y='consumption')

<matplotlib.axes._subplots.AxesSubplot at 0x2aa0061e7f0>
```



```
dataset.plot(kind='scatter',x='max_temp',y='consumption')

<matplotlib.axes._subplots.AxesSubplot at 0x2aa006930f0>
```



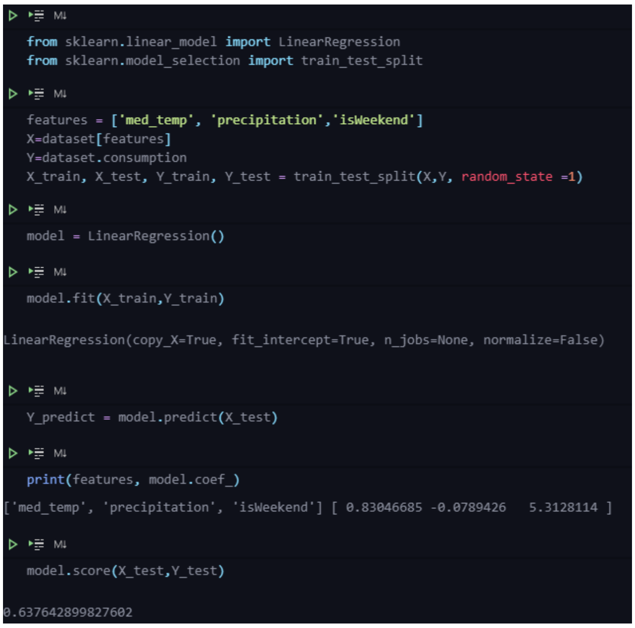
```
dataset.plot(kind='scatter',x='precipitation',y='consumption')

<matplotlib.axes._subplots.AxesSubplot at 0x2aa006f4358>
```



```
dataset.plot(kind='scatter',x='isWeekend',y='consumption')

<matplotlib.axes._subplots.AxesSubplot at 0x2aa0073ee10>
```

After plotting the constraints, we notice a few traits as to which one to use for best fitting, and for training, out of the temp columns, I pick the median temp, because it is not much scattered like min temp, nor secluded like max temp.

7. Sklearn : calling LinearRegression() function, fitting the data, predicting the variable and then calculating the score for the model.



```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

```
features = ['med_temp', 'precipitation','isWeekend']
X=dataset[features]
Y=dataset.consumption
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, random_state =1)
```

```
model = LinearRegression()
```

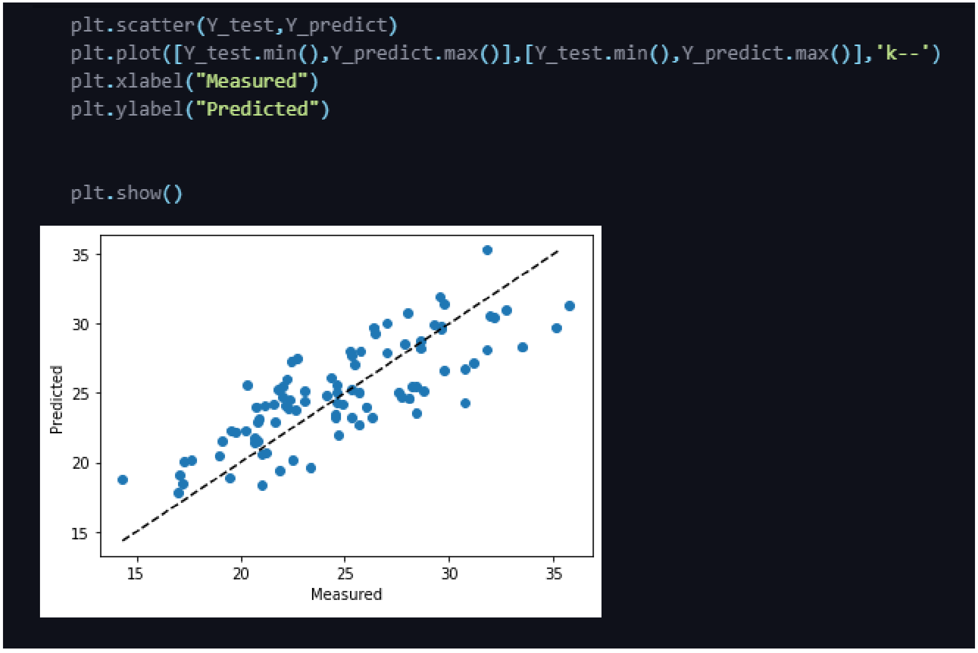
```
model.fit(X_train,Y_train)

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
Y_predict = model.predict(X_test)
```

```
print(features, model.coef_)

['med_temp', 'precipitation', 'isWeekend'] [ 0.83046685 -0.0789426   5.3128114 ]
```

```
model.score(X_test,Y_test)

0.637642899827602
```

8. Plotting the observations :



```
plt.scatter(Y_test,Y_predict)
plt.plot([Y_test.min(),Y_predict.max()],[Y_test.min(),Y_predict.max()],'k--')
plt.xlabel("Measured")
plt.ylabel("Predicted")


plt.show()
```

Performed by : Josh Trivedi

Available link for submission : https://github.com/joshtrivedi/Beer-Consumption