In [2]:

```
%pylab
%matplotlib inline

%config InlineBackend.figure_format = 'retina'
```

```
Using matplotlib backend: Qt5Agg
Populating the interactive namespace from numpy and matplotlib
```

In [50]:

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

import pandas as pd
import numpy as np
from sklearn import datasets, linear_model
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from scipy import stats
```

1- Prediction of the Customer purchase quantity

Database grouped By Customers

In [58]:

```
Database_byCustomers = pd.read_csv('Database_byCustomers.csv', sep = ',',encoding="utf-8")
Database_byCustomers=pd.DataFrame(Database_byCustomers)


Database_byCustomers=Database_byCustomers.rename(columns = {'Total_Time_Taken(min)':'Time',
Database_byCustomers.head(30)
#pd.read_csv('winequality-white.csv', sep = ';')
#wine.head()
```

Out[58]:

stomer ID	Time	Distance_km	Loy_points	Num_employees	Payment Method	Order Status	Latitude	L
457559	14.33	20.29	28	4	CASH	COMPLETED	-1.165012	3
553157	17.75	5.22	158	2	CASH	COMPLETED	-1.293994	3
553157	61.78	5.22	158	2	CASH	COMPLETED	-1.293994	3
554524	1990.72	8.82	58	3	CASH	COMPLETED	-1.260588	3
555286	257.32	13.67	67	1	CASH	COMPLETED	-1.186376	3
...	...	...	...	...	...	...	...	
636098	14.18	6.96	115	5	CASH	COMPLETED	-1.306804	3
636098	14.78	6.96	115	5	CASH	COMPLETED	-1.306804	3
636098	15.52	6.96	115	5	CASH	COMPLETED	-1.306804	3
636098	16.05	6.96	115	5	CASH	COMPLETED	-1.306804	3
636098	19.08	4.54	115	5	CASH	COMPLETED	-1.283330	3

We work only with Kenya orders and customers because there are some inconsistencies in Nigeria data

In [59]:

```
df_mask=((Database_byCustomers['Country']=='Kenya'))
Data_byCustomers_Kenya = Database_byCustomers[df_mask]
Data_byCustomers_Kenya.head(11)
```

Out[59]:

stomer ID	Time	Distance_km	Loy_points	Num_employees	Payment Method	Order Status	Latitude	L
457559	14.33	20.29	28	4	CASH	COMPLETED	-1.165012	3
553157	17.75	5.22	158	2	CASH	COMPLETED	-1.293994	3
553157	61.78	5.22	158	2	CASH	COMPLETED	-1.293994	3
554524	1990.72	8.82	58	3	CASH	COMPLETED	-1.260588	3
555286	257.32	13.67	67	1	CASH	COMPLETED	-1.186376	3
...	...	...	...	...	...	...	...	
636098	0.37	6.96	115	5	CASH	COMPLETED	-1.306804	3
636098	1.43	6.96	115	5	CASH	COMPLETED	-1.306804	3
636098	2.68	6.96	115	5	CASH	COMPLETED	-1.306804	3
636098	4.55	6.96	115	5	CASH	COMPLETED	-1.306804	3
636098	6.20	4.54	115	5	CASH	COMPLETED	-1.283330	3

Lineal Regression with quantitative variables

In [60]:

```
target = 'Purch_quantity'

#predictors:
features=['Time','Distance_km', 'Loy_points']

x = Data_byCustomers_Kenya[features]
y = Data_byCustomers_Kenya[target]


X2 = sm.add_constant(x)
est = sm.OLS(y, X2)
est2 = est.fit()
print(est2.summary())


x_train, x_test, y_train, y_test = train_test_split(x, y)

# Creación de un modelo
model = LinearRegression()
model.fit(x_train, y_train)

predit_train = model.predict(x_train)
predit_test = model.predict(x_test)

# Evaluación de R2
print('R2 en entrenamiento es: ', model.score(x_train, y_train))
print('R2 en validación es: ', model.score(x_test, y_test))
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:         Purch_quantity   R-squared:                       0.0
91
Model:                            OLS   Adj. R-squared:                  0.0
90
Method:                 Least Squares   F-statistic:                     78.
57
Date:                Mon, 21 Feb 2022   Prob (F-statistic):           1.95e-
48
Time:                        22:42:10   Log-Likelihood:                -1047
4.
No. Observations:                2359   AIC:                          2.096e+
04
Df Residuals:                    2355   BIC:                          2.098e+
04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
===
                 coef    std err          t      P>|t|      [0.025      0.9
75]
------------------------------------------------------------------------------
---
const          4.4224      0.569      7.777      0.000       3.307       5.
538
Time           0.0065      0.004      1.836      0.067      -0.000       0.
013
Distance_km    -0.0006      0.001     -0.408      0.683      -0.003       0.
```

```
002
Loy_points      0.0102      0.001     15.262      0.000       0.009       0.
012
==============================================================================
==
Omnibus:                     4704.090   Durbin-Watson:                   1.5
28
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         17917065.4
82
Skew:                          15.542   Prob(JB):                         0.
00
Kurtosis:                     428.815   Cond. No.                     1.07e+
03
==============================================================================
==

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
[2] The condition number is large, 1.07e+03. This might indicate that there
are
strong multicollinearity or other numerical problems.
R2 en entrenamiento es:  0.0832025088921825
R2 en validación es:  0.13776487679665006
```

Lineal Regression with quantitative and dummies variables

In [61]:

```
# Separation among the objective variable and the predictors.

#Objective variable
target = 'Purch_quantity'
y = pd.DataFrame(Data_byCustomers_Kenya[target])

#predictors:
#features=['Time', 'Distance_km', 'Loy_points', 'Payment Method', 'Order Status']
x_0 = pd.DataFrame(Data_byCustomers_Kenya, columns = ['Time', 'Distance_km', 'Loy_points',
x = pd.concat([x_0['Time'], x_0['Distance_km'], x_0['Loy_points'], pd.get_dummies(x_0['Paym

#x_0 = pd.DataFrame(Database_byCustomers, columns = ['Time', 'Distance_km', 'Loy_points', '
#x = pd.concat([x_0['Time'], x_0['Distance_km'], x_0['Loy_points'], pd.get_dummies(x_0['Ord

# Polinomio de grado 5
model_dummies = LinearRegression(fit_intercept = False)
model_dummies.fit(x, y)

print("Modelo dummies - R^2:", model_dummies.score(x, y))
```

```
Modelo dummies - R^2: 0.13224229805481458
```

2- Prediction of the Revenue by Store

In [62]:

```
Database_byStore = pd.read_csv('Database_byStore.csv', sep = ',',encoding="utf-8")
Database_byStore=pd.DataFrame(Database_byStore)


Database_byStore=Database_byStore.rename(columns = {'Total_Time_Taken(min)':'Time','Distanc
Database_byStore.head(30)
#pd.read_csv('winequality-white.csv', sep = ';')
#wine.head()
```

Out[62]:

	Lat	Long	Nro_Employees	Payment Method	Order Status	Rating	Country	N_Orders
0	-8.901959	13.197016	1	CASH	CANCELLED	0.0	Kenya	1
1	-1.475960	36.959040	2	CASH	COMPLETED	0.0	Kenya	2
2	-1.396854	36.758234	3	CASH	CANCELLED	0.0	Kenya	1
3	-1.396816	36.749753	4	CASH	COMPLETED	0.0	Kenya	1
4	-1.396806	36.749739	4	CASH	CANCELLED	0.0	Kenya	1
...	...	...	...	...	...	...	...	...
25	-1.383046	36.676590	6	Pay Later	COMPLETED	0.0	Kenya	4
26	-1.369013	36.940145	4	CASH	COMPLETED	0.0	Kenya	3
27	-1.364193	36.911635	3	CASH	CANCELLED	0.0	Kenya	1
28	-1.360832	36.655943	8	CASH	COMPLETED	0.0	Kenya	1
29	-1.360832	36.655943	8	Pay Later	COMPLETED	0.0	Kenya	3

We work only with Kenya orders and customers because there are some inconsistencies in Nigeria data

In [63]:

```
df_mask=((Database_byStore['Country']=='Kenya'))
Data_byStore_Kenya = Database_byStore[df_mask]
Data_byStore_Kenya.head(11)
```

Out[63]:

	Lat	Long	Nro_Employees	Payment Method	Order Status	Rating	Country	N_Orders
0	-8.901959	13.197016	1	CASH	CANCELLED	0.0	Kenya	1
1	-1.475960	36.959040	2	CASH	COMPLETED	0.0	Kenya	2
2	-1.396854	36.758234	3	CASH	CANCELLED	0.0	Kenya	1
3	-1.396816	36.749753	4	CASH	COMPLETED	0.0	Kenya	1
4	-1.396806	36.749739	4	CASH	CANCELLED	0.0	Kenya	1
...	...	...	...	...	...	...	...	...
6	-1.396545	36.765326	2	CASH	COMPLETED	0.0	Kenya	11
7	-1.396390	36.940280	1	CASH	COMPLETED	0.0	Kenya	2
8	-1.396276	36.762251	3	CASH	COMPLETED	0.0	Kenya	2
9	-1.395300	36.764000	3	CASH	COMPLETED	0.0	Kenya	3
10	-1.395300	36.764000	4	CASH	COMPLETED	0.0	Kenya	5

In [65]:

```
## Lineal Regression with quantitative variables

target = 'Revenue_byStore'

#predictors:
features=['Nro_Employees', 'N_Orders', 'Rating']

x = Data_byStore_Kenya[features]
y = Data_byStore_Kenya[target]



X2 = sm.add_constant(x)
est = sm.OLS(y, X2)
est2 = est.fit()
print(est2.summary())


x_train, x_test, y_train, y_test = train_test_split(x, y)

# Creación de un modelo
model = LinearRegression()
model.fit(x_train, y_train)

predit_train = model.predict(x_train)
predit_test = model.predict(x_test)

# Evaluación de R2
print('R2 en entrenamiento es: ', model.score(x_train, y_train))
print('R2 en validación es: ', model.score(x_test, y_test))
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:        Revenue_byStore   R-squared:                       0.4
64
Model:                            OLS   Adj. R-squared:                  0.4
61
Method:                 Least Squares   F-statistic:                     13
7.0
Date:                Mon, 21 Feb 2022   Prob (F-statistic):           6.25e-
64
Time:                        23:05:54   Log-Likelihood:                -618
0.8
No. Observations:                 478   AIC:                          1.237e+
04
Df Residuals:                     474   BIC:                          1.239e+
04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
=====
                    coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
```

```
-----
const         -2.252e+04   6588.336     -3.418      0.001   -3.55e+04   -957
6.067
Nro_Employees  3890.1496   1037.554      3.749      0.000    1851.375    592
8.924
N_Orders       1.151e+04    591.168     19.476      0.000    1.04e+04    1.2
7e+04
Rating         5365.6012   9070.187      0.592      0.554   -1.25e+04    2.3
2e+04
==============================================================================
==
Omnibus:                      570.169   Durbin-Watson:                   1.8
44
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            49356.2
84
Skew:                           5.612   Prob(JB):                         0.
00
Kurtosis:                      51.499   Cond. No.                         1
9.3
==============================================================================
==

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
R2 en entrenamiento es:  0.4866995919031063
R2 en validación es:  0.41747895257749645
```