A regression Model

This is very quick look at what you can see in Jupyter notebooks. I haven't really included anything of much substance in here, rather just a look at what you can do and how it can be presented to you after I have done an analysis.

Importing all of the necessary modules for writing maths formulas, manipulating data and plotting.

In [1]:
```
import pandas as pd
```

In [2]:
```
import statsmodels.api as sm
```

In [3]:
```
import numpy as np
```

In [4]:
```
from sklearn import linear_model
```

In [5]:
```
from matplotlib import pyplot as plt
```

In [6]:
```
from scipy.stats import ttest_ind
```

In [7]:
```
import matplotlib.cm as cm #latex module
```

Reading the data from the excel sheet (given by Carl)

In [8]:
```
d = pd.read_excel("dataFromCarl.xlsx",sheet_name = 'Sheet2')
```

by calling 'd' which is the variable assigned to the data, you can see the table. 'Sheet2' of the sheet I moved MailingQty, Orders and marketing costs to run some tests.

In [9]:
```
d
```

Out[9]:

	mailingQty	orders	marketingCosts
0	249500	8401	162501.19
1	102887	3514	0.00
2	881	71	0.00
3	110136	5114	0.00
4	67489	2506	0.00
...	...	...	...
85	250000	8923	67850.00
86	60000	1394	21000.00
87	140000	6278	49000.00
88	229672	8280	80385.20
89	175000	3124	61250.00

90 rows × 3 columns

Setting the independant and dependant variables as x and y respectively to split out the table.

In [10]:
```
x = d.drop(["orders","marketingCosts"], axis=1)
```

In [11]:
```
y = d.drop(["mailingQty","marketingCosts"], axis=1)
```

By using the describe function below you can see some of the fundamental properties of the data such as mean, standard deviantion and the quartile values.

In [12]:
```
d.describe()
```

Out[12]:

	mailingQty	orders	marketingCosts
count	90.000000	90.000000	90.000000
mean	80554.811111	3144.600000	25274.678770
std	77325.020775	2992.850889	30107.264507
min	0.000000	1.000000	0.000000
25%	2445.500000	350.500000	1394.325000
50%	57588.000000	2166.500000	12760.127200
75%	143750.000000	5596.500000	48400.458300
max	250000.000000	9816.000000	162501.190000

This line just calls the function to initialize it from the module

In [13]:
```
regr = linear_model.LinearRegression()
```

In [14]:
```
regr.fit(x,y)
```

Out[14]:
```
▾ LinearRegression
LinearRegression()
```

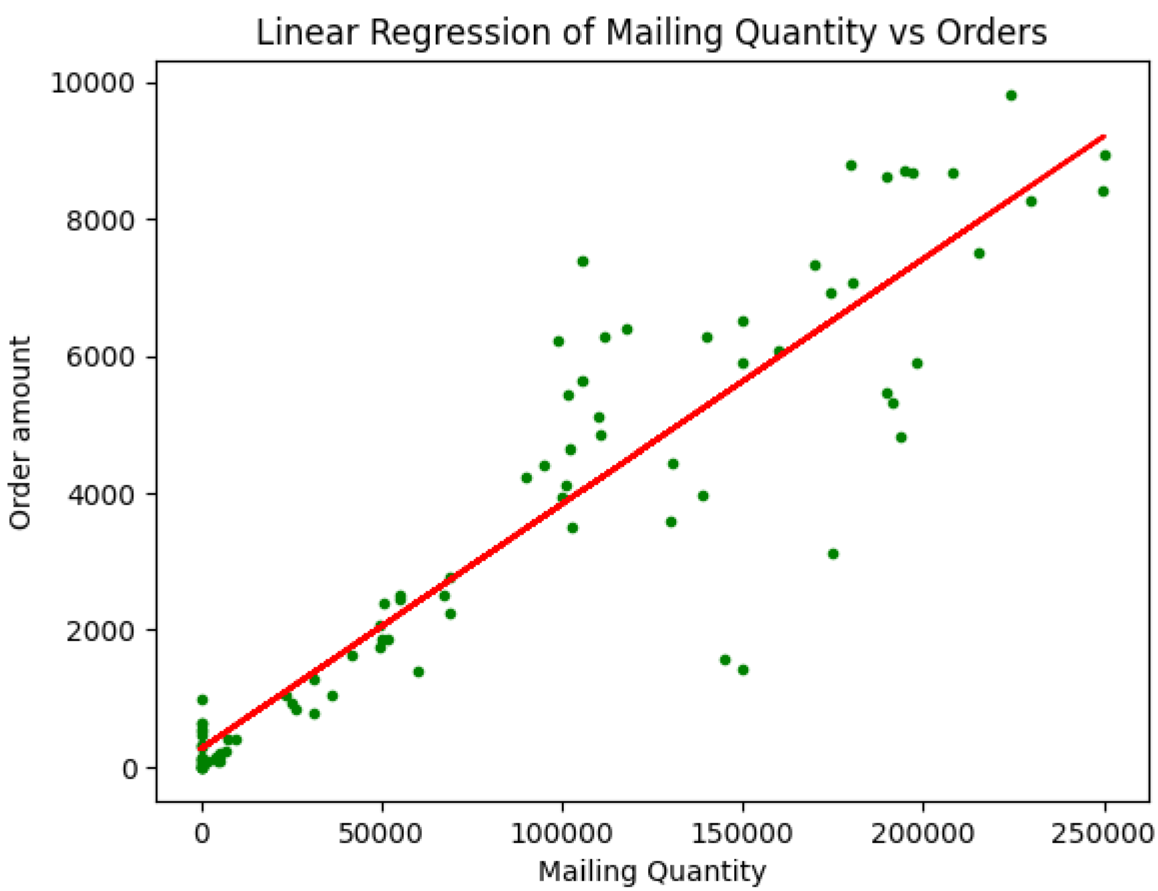
The predict function creates the regression line (best fit) by creating a simle y = mx +c function

In [15]:
```
Y_pred = regr.predict(x)
```

Using the matplotlib library to plot the regression model

In [16]:
```
plt.scatter(x, y, color = 'green', marker = '.')
plt.plot(x, Y_pred, color='red')
plt.xlabel("Mailing Quantity")
plt.ylabel("Order amount")
plt.title("Linear Regression of Mailing Quantity vs Orders")
```

Out[16]:
```
Text(0.5, 1.0, 'Linear Regression of Mailing Quantity vs Orders')
```



In [17]:
```
d.mean()
```

Out[17]:
```
mailingQty        80554.811111
orders             3144.600000
marketingCosts    25274.678770
dtype: float64
```

In [18]:
```
#coefficient of determination
r_sq = regr.score(x,y)
r_sq
```

Out[18]:
```
0.85318447280849
```

the R^2 value determines how much of the dependant varibale is explained by the shape of the independant. The value given of 0.853 means that 85.3% of the data can be explained by this regression model which is very good. 0.7 < means a high level of correlation and 0.4 > shows little correlation between the variables.

In [19]:
```
res = ttest_ind(x,y).pvalue
res
```

Out[19]:
```
array([1.52245875e-17])
```

T-tests determine how much of the data represents the null hypothesis, i.e. The percentage of data that demonstrate that the variables aren't related. This value is much smaller than 0.05 (5%) meaning that the null hypothesis is not true.

Looking into Multivariable Linear Regression

For this test, there will be two independant variables and one dependant. This study will determine how the mailing quantity and the marketing costs will affect the total orders.

First the x varialbe must be overwritten to become a n x 2 array

In [20]:
```
x = d[["mailingQty", "marketingCosts"]]
```

In [21]:
```
y = d[["orders"]]
```

In [22]:
```
regr = linear_model.LinearRegression()
```

In [23]:
```
regr.fit(x,y)
```

Out[23]:
```
▾ LinearRegression
LinearRegression()
```

In [24]:
```
print("Intecept: \n", regr.intercept_)
```

```
Intecept:
 [267.30953125]
```

In [25]:
```
print('Coefficients: \n', regr.coef_)
```

```
Coefficients:
 [[0.03532648 0.00124918]]
```

In [26]:
```
x = sm.add_constant(x) # adding a constant
```

In [27]:
```
model = sm.OLS(y, x).fit()
```

In [28]:
```
predictions = model.predict(x)
```

In [29]:
```
print_model = model.summary()
```

In [30]:
```
print(model.params)
```

```
const             267.309531
mailingQty          0.035326
marketingCosts      0.001249
dtype: float64
```

These values are the coefficients of the regression model; The amount that orders increases with every unit increase in mailing qunatity or marketing price.

In [31]:
```
print(print_model)
```

OLS Regression Results

```
==============================================================================
===
Dep. Variable:                 orders   R-squared:                       0.
853
Model:                            OLS   Adj. R-squared:                  0.
850
Method:                 Least Squares   F-statistic:                     25
2.9
Date:                Wed, 14 Dec 2022   Prob (F-statistic):           5.62e
-37
Time:                        13:49:35   Log-Likelihood:                -76
1.21
No. Observations:                  90   AIC:                             15
28.
Df Residuals:                      87   BIC:                             15
36.
Df Model:                           2

Covariance Type:            nonrobust

==============================================================================
=======
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const            267.3095    177.904      1.503      0.137     -86.294
620.913
mailingQty         0.0353      0.003     10.856      0.000       0.029
  0.042
marketingCosts     0.0012      0.008      0.149      0.882      -0.015
  0.018
==============================================================================
===
Omnibus:                       19.772   Durbin-Watson:                   2.
039
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               49.
248
Skew:                          -0.701   Prob(JB):                     2.02e
-11
Kurtosis:                       6.342   Cond. No.                     1.71e
+05
==============================================================================
===

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The condition number is large, 1.71e+05. This might indicate that there
are
strong multicollinearity or other numerical problems.
```

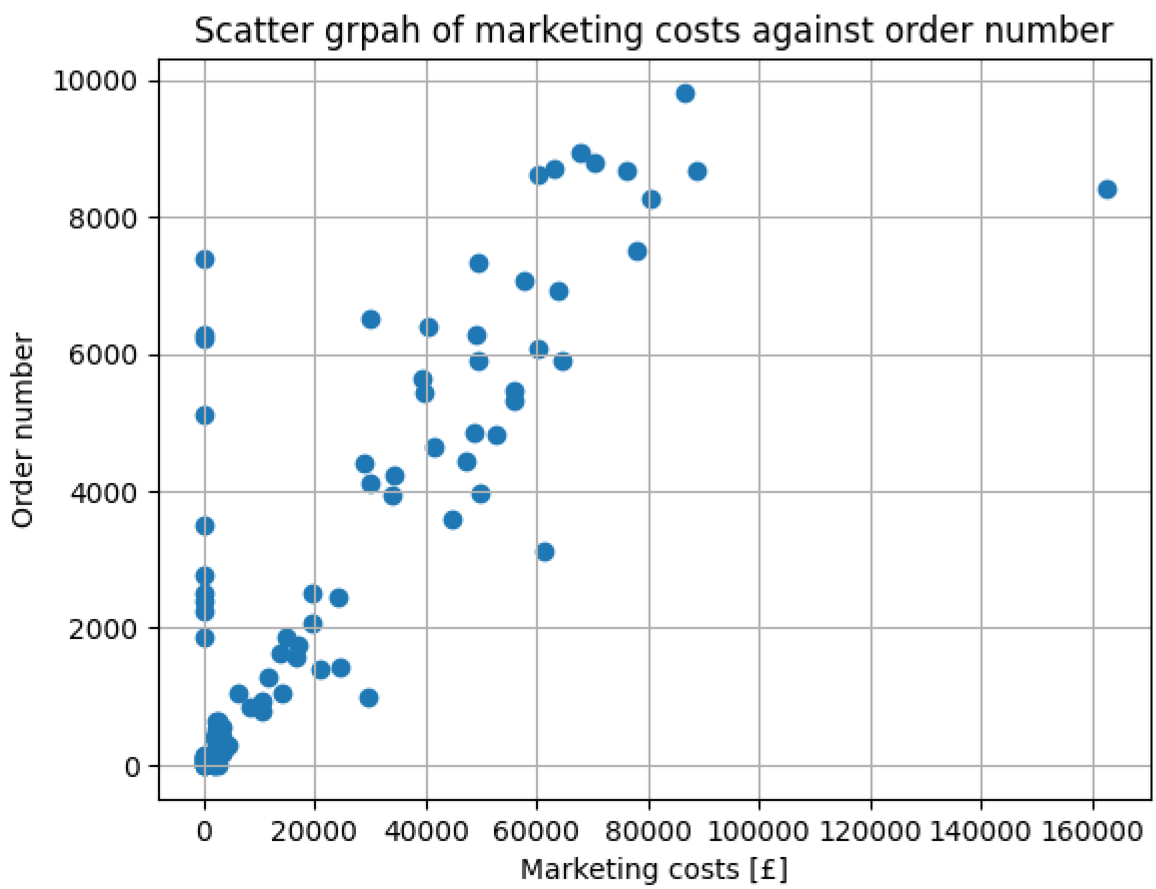
$\underline{The Summary Table Explained}$

P>|t| is the p-value. A p-value below 0.05 means that the variable is significant. This means that the mailingQty is significant but the marketing costs may not be.

The *Prob (F-statistic)* shows how the F-Statistic compares to the significance level. In other words, how true is the null hypothesis. In this case it is extremely low and therefore the null hypothesis can probably be ignored.

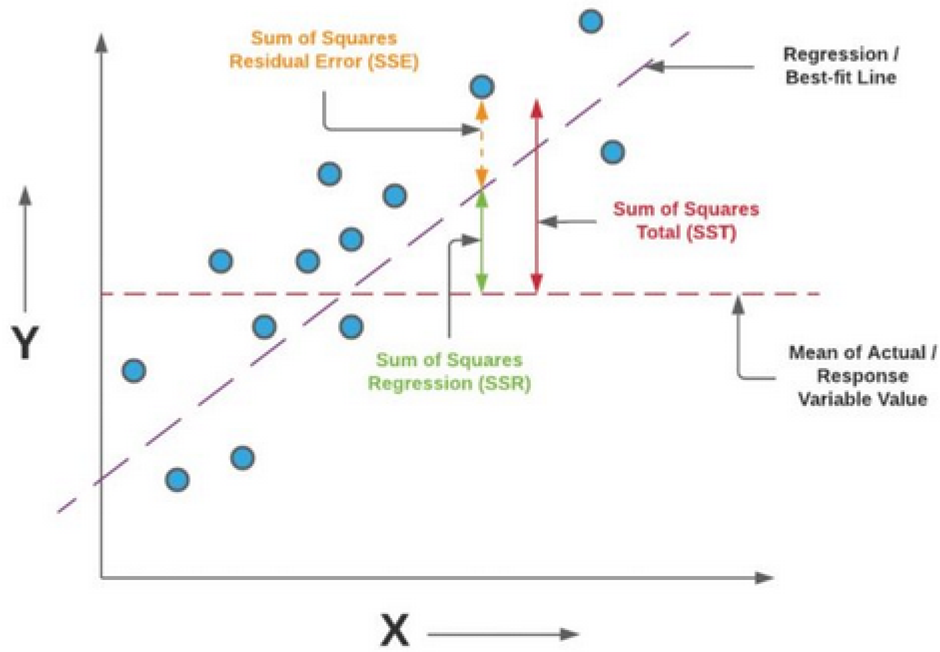
The R^2 Value shows that 85.3% of the dependant variable's variation is explained by the independant variables.

In [32]:
```
plt.scatter(d["marketingCosts"], d["orders"])
plt.title("Scatter grpah of marketing costs against order number")
plt.xlabel("Marketing costs [£]")
plt.ylabel("Order number")
plt.grid()
```



Expalination of the t-test

The best way to explain the f-test is with the below diagram



The F-Statistic

The f test of a regression model determines the significance of the trend. It tests the null hpothesis, which states that the model with no independant variabes fits the data as well as the model

The F-statistic is required in conjunction with the p-value

f-tests test determines the significance of all the coefficients wheras the t-test determines the significance of the coefficients individually. It is useful to use a t-test as well to see the effect of the independant variables seperately on the dependant.

In [ ]:

```

```