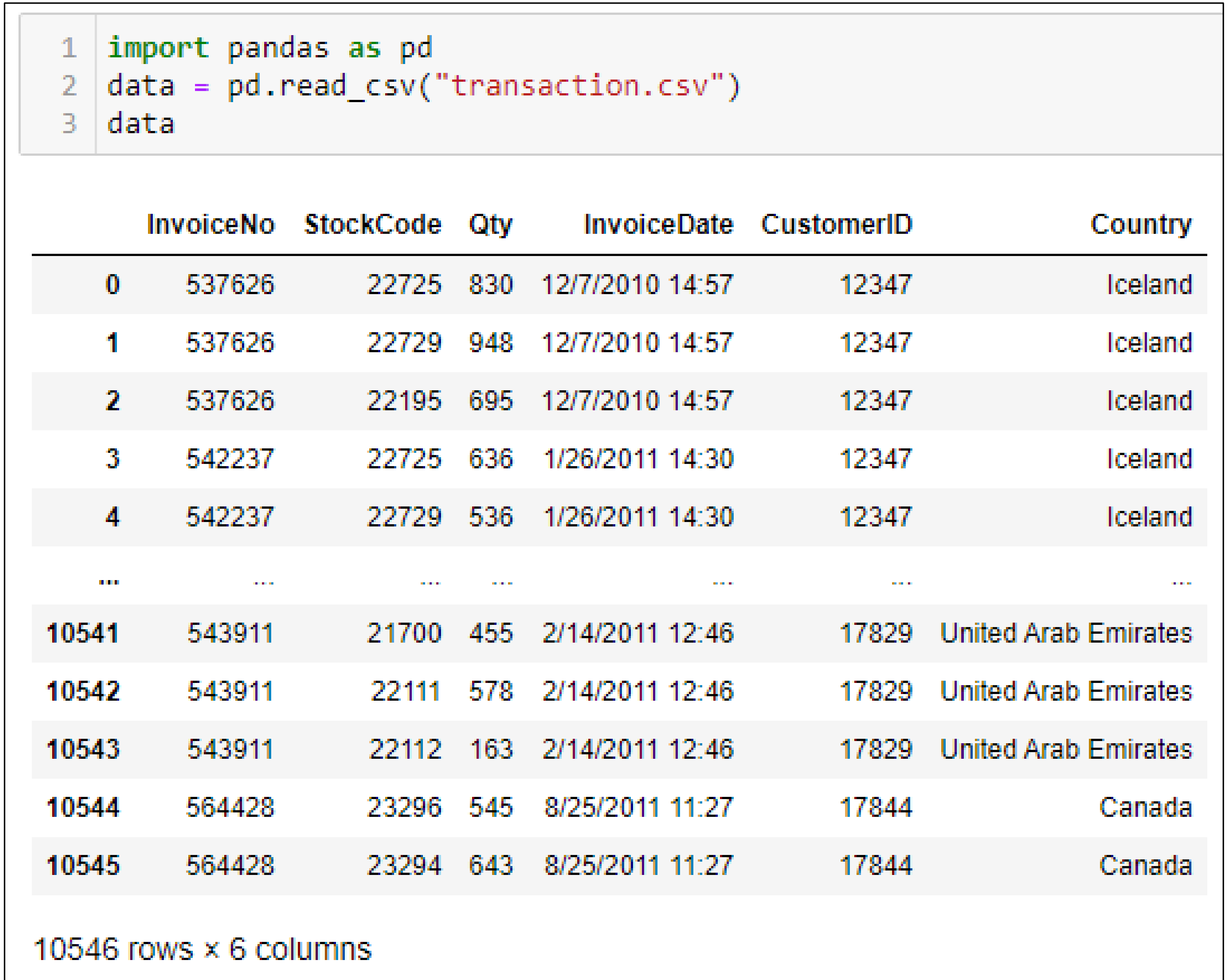
Predictive Mining – Linear Regression

Knowledge Discovery

Hanif Izzudin Rahman

➔1. dataset: transaction.csv, and show it

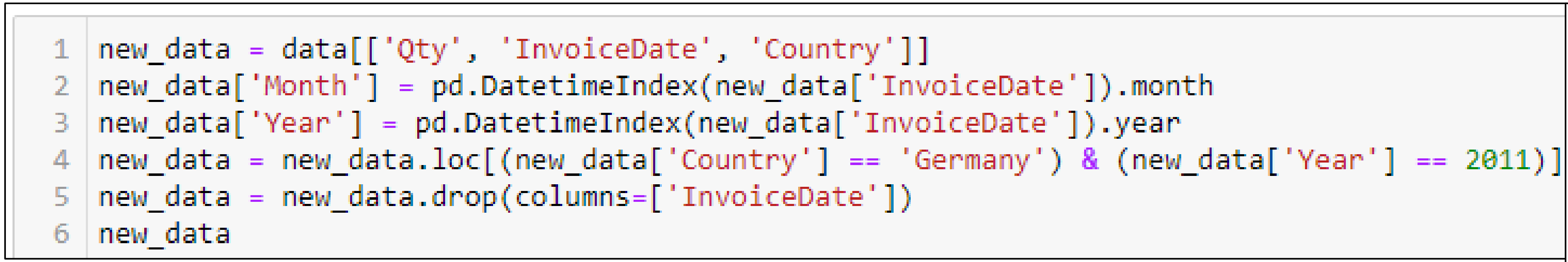


```python
import pandas as pd
data = pd.read_csv("transaction.csv")
data
```

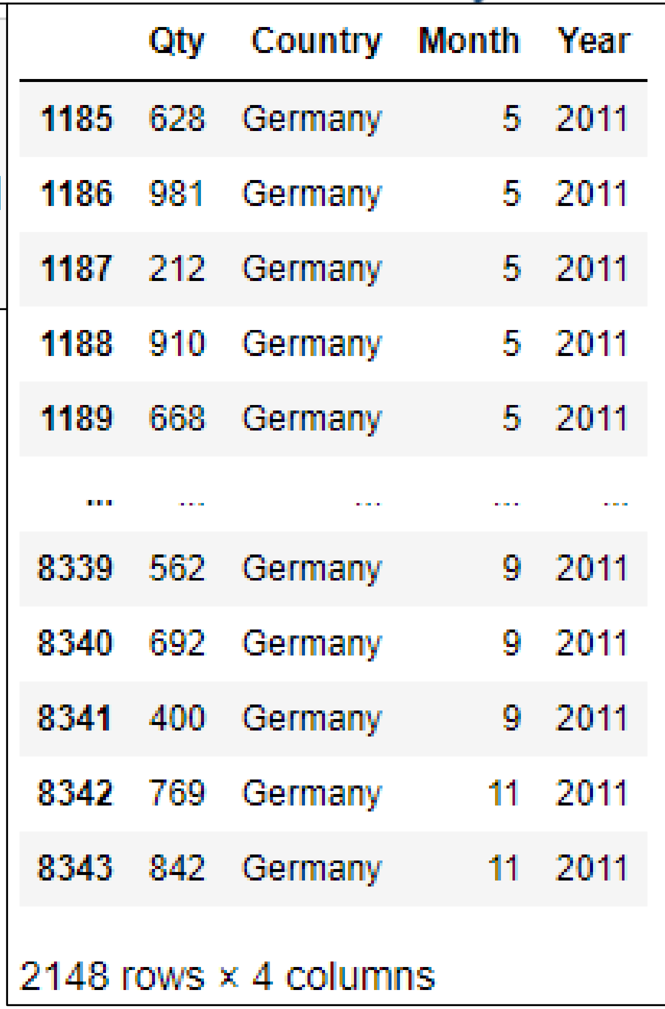
	InvoiceNo	StockCode	Qty	InvoiceDate	CustomerID	Country
0	537626	22725	830	12/7/2010 14:57	12347	Iceland
1	537626	22729	948	12/7/2010 14:57	12347	Iceland
2	537626	22195	695	12/7/2010 14:57	12347	Iceland
3	542237	22725	636	1/26/2011 14:30	12347	Iceland
4	542237	22729	536	1/26/2011 14:30	12347	Iceland
...	...	...	...	...	...	...
10541	543911	21700	455	2/14/2011 12:46	17829	United Arab Emirates
10542	543911	22111	578	2/14/2011 12:46	17829	United Arab Emirates
10543	543911	22112	163	2/14/2011 12:46	17829	United Arab Emirates
10544	564428	23296	545	8/25/2011 11:27	17844	Canada
10545	564428	23294	643	8/25/2011 11:27	17844	Canada

10546 rows × 6 columns

➔2. data: take the data in the dataset for feature of Qty, Country ("Germany"), month, year ("2011")



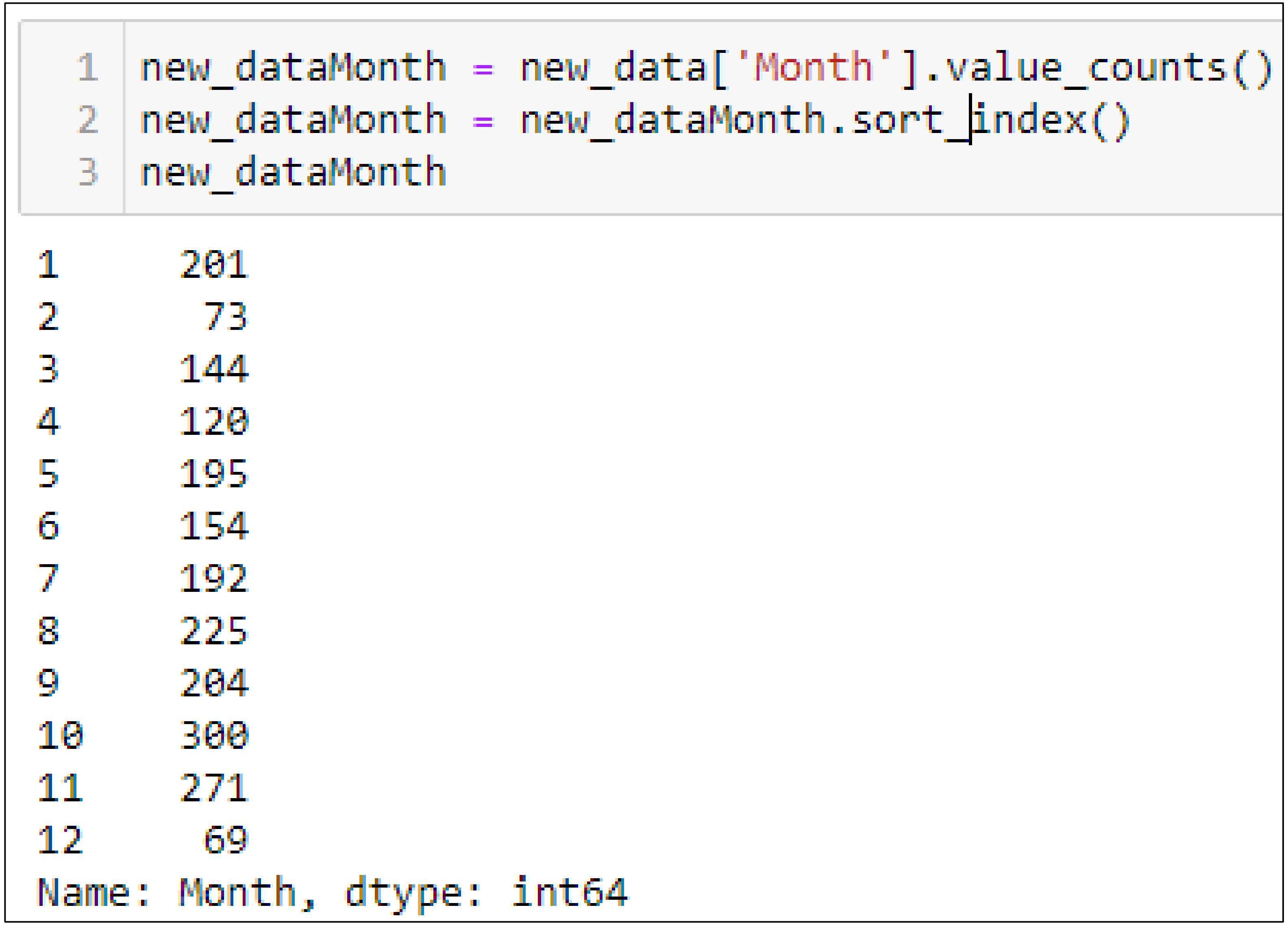
```
new_data = data[['Qty', 'InvoiceDate', 'Country']]
new_data['Month'] = pd.DatetimeIndex(new_data['InvoiceDate']).month
new_data['Year'] = pd.DatetimeIndex(new_data['InvoiceDate']).year
new_data = new_data.loc[(new_data['Country'] == 'Germany') & (new_data['Year'] == 2011)]
new_data = new_data.drop(columns=['InvoiceDate'])
new_data
```



	Qty	Country	Month	Year
1185	628	Germany	5	2011
1186	981	Germany	5	2011
1187	212	Germany	5	2011
1188	910	Germany	5	2011
1189	668	Germany	5	2011
...	...	...	...	...
8339	562	Germany	9	2011
8340	692	Germany	9	2011
8341	400	Germany	9	2011
8342	769	Germany	11	2011
8343	842	Germany	11	2011

2148 rows × 4 columns

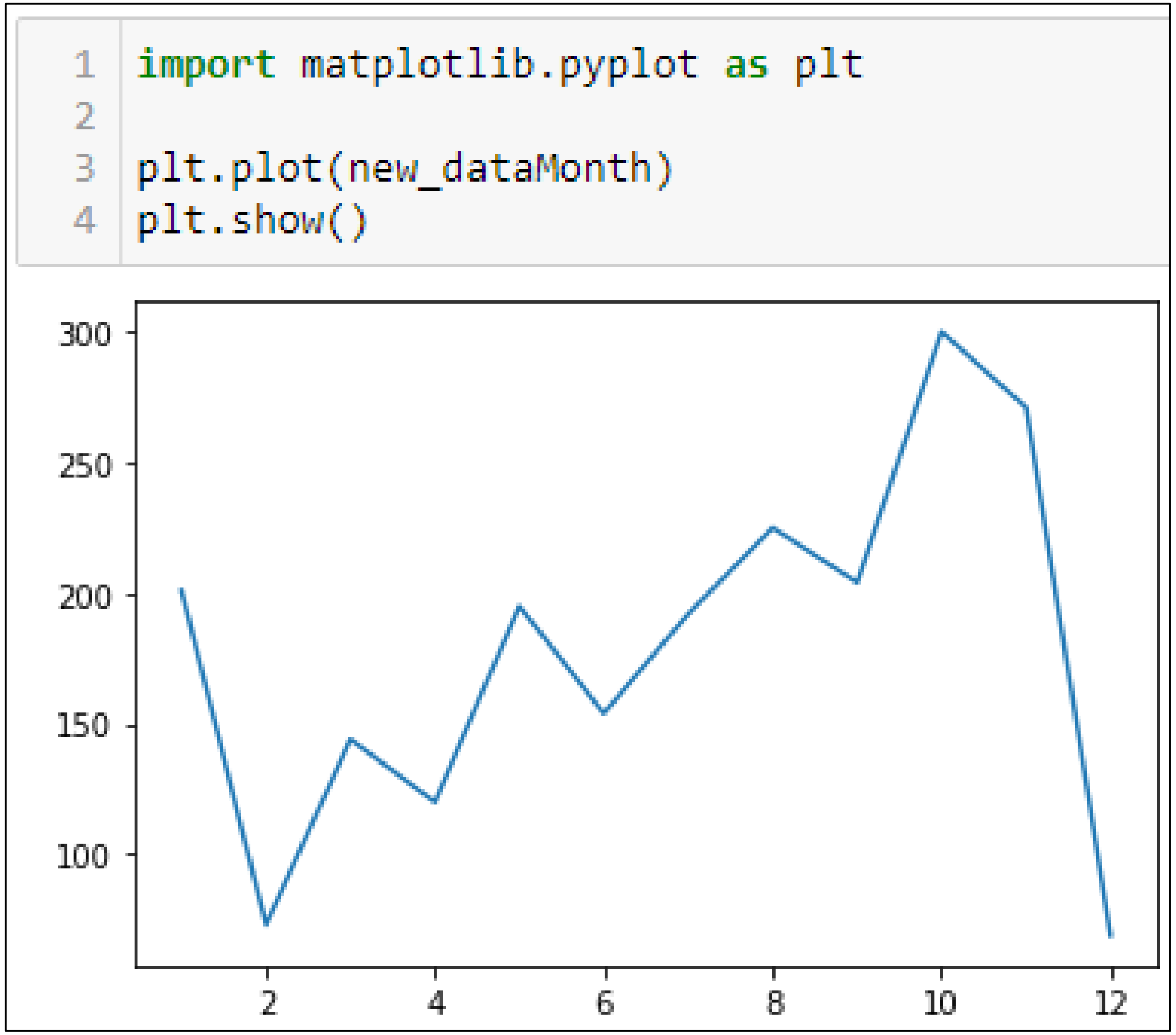
➔3. TotalQty: take Month from the data and accumulated Qty in the same month, and show it



```python
new_dataMonth = new_data['Month'].value_counts()
new_dataMonth = new_dataMonth.sort_index()
new_dataMonth
```

```
1     201
2      73
3     144
4     120
5     195
6     154
7     192
8     225
9     204
10    300
11    271
12     69
Name: Month, dtype: int64
```

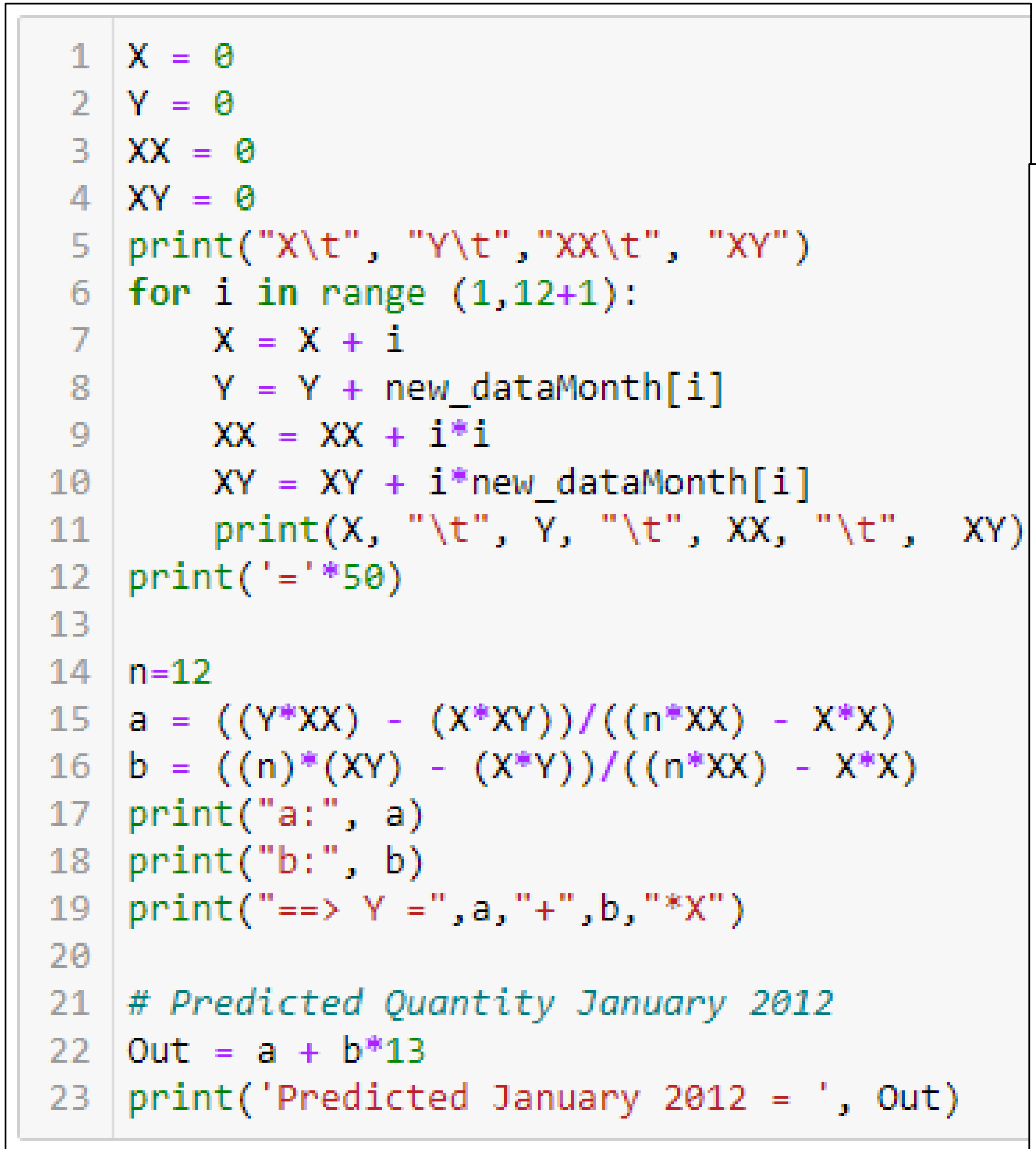
➔4. Visualize the movement of TotalQty values where the x axis = Month and the y axis = TotalQty



```python
import matplotlib.pyplot as plt

plt.plot(new_dataMonth)
plt.show()
```

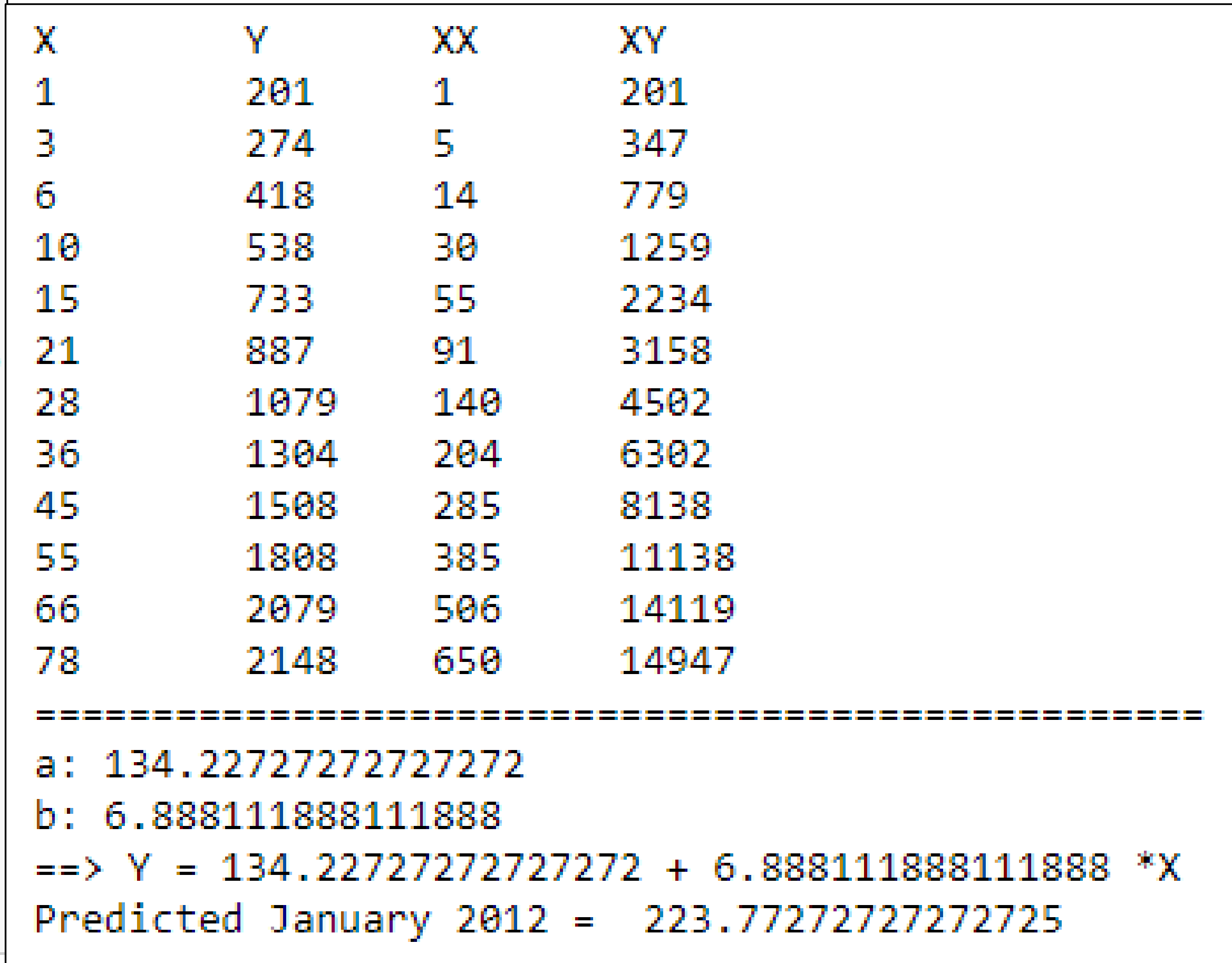
➔5. PredictedQty: predict the total Qty of TotalQty in January 2012 with **Linear Regression**



```python
X = 0
Y = 0
XX = 0
XY = 0
print("X\t", "Y\t","XX\t", "XY")
for i in range (1,12+1):
    X = X + i
    Y = Y + new_dataMonth[i]
    XX = XX + i*i
    XY = XY + i*new_dataMonth[i]
    print(X, "\t", Y, "\t", XX, "\t",  XY)
print('='*50)

n=12
a = ((Y*XX) - (X*XY))/((n*XX) - X*X)
b = ((n)*(XY) - (X*Y))/((n*XX) - X*X)
print("a:", a)
print("b:", b)
print("==> Y =",a,"+",b,"*X")

# Predicted Quantity January 2012
Out = a + b*13
print('Predicted January 2012 = ', Out)
```



```
X        Y       XX      XY
1        201     1       201
3        274     5       347
6        418     14      779
10       538     30      1259
15       733     55      2234
21       887     91      3158
28       1079    140     4502
36       1304    204     6302
45       1508    285     8138
55       1808    385     11138
66       2079    506     14119
78       2148    650     14947
==================================================
a: 134.22727272727272
b: 6.888111888111888
==> Y = 134.22727272727272 + 6.888111888111888 *X
Predicted January 2012 =  223.77272727272725
```

Linear Regression from Y to X

$$Y = a + b * X$$

where:

Y = dependent variable

X = independent variable

a = intercept

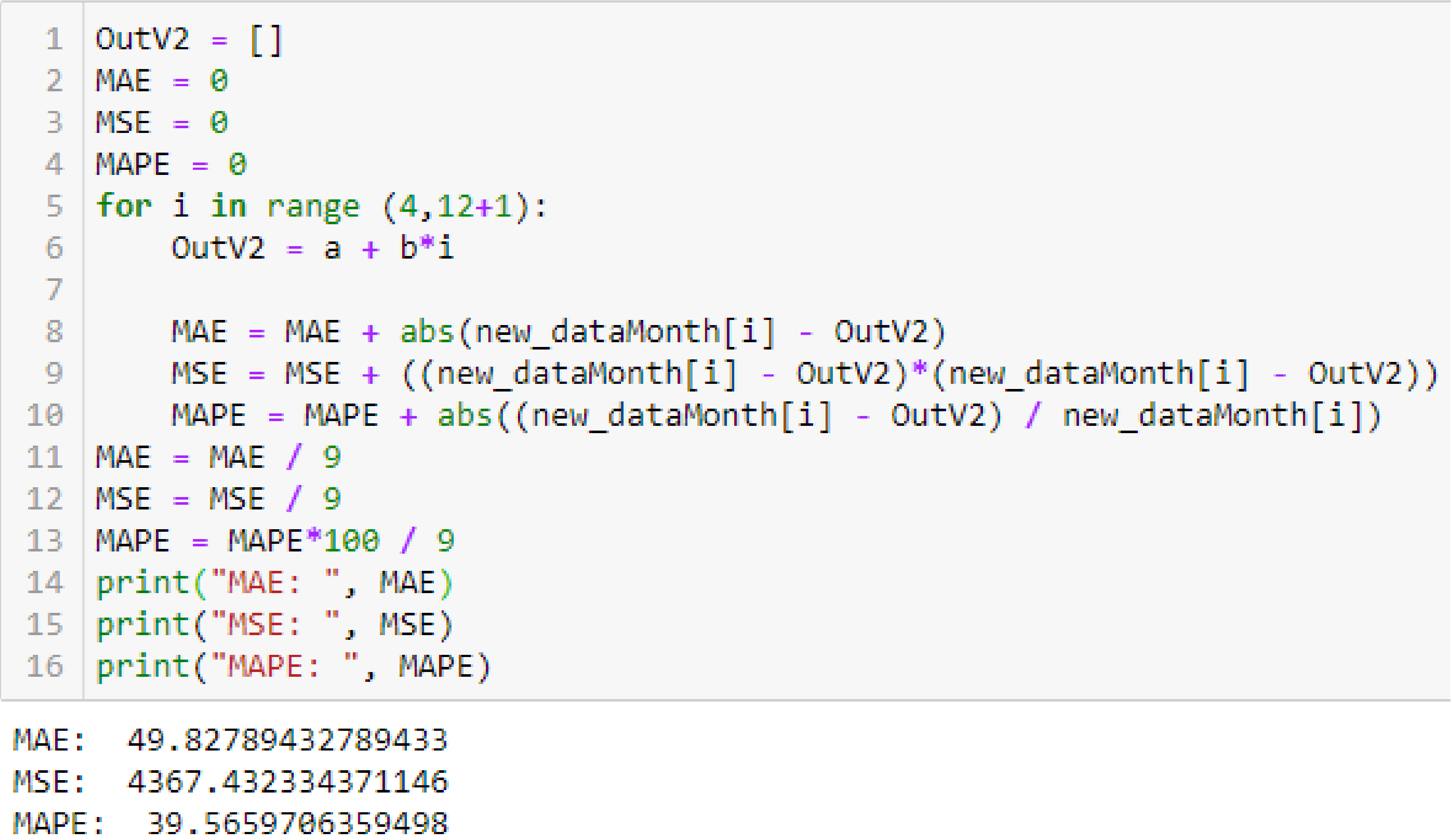
b = slope (regression coefficient)

$$a = \frac{(\Sigma Y)(\Sigma X^2) - (\Sigma X)(\Sigma XY)}{(n)(\Sigma X^2) - (\Sigma X)^2}$$

$$b = \frac{(n)(\Sigma XY) - (\Sigma X)(\Sigma Y)}{(n)(\Sigma X^2) - (\Sigma X)^2}$$

➔6. Calculate the MAE, MSE and MAPE for within last 9 months

Predict



```python
OutV2 = []
MAE = 0
MSE = 0
MAPE = 0
for i in range (4,12+1):
    OutV2 = a + b*i

    MAE = MAE + abs(new_dataMonth[i] - OutV2)
    MSE = MSE + ((new_dataMonth[i] - OutV2)*(new_dataMonth[i] - OutV2))
    MAPE = MAPE + abs((new_dataMonth[i] - OutV2) / new_dataMonth[i])
MAE = MAE / 9
MSE = MSE / 9
MAPE = MAPE*100 / 9
print("MAE: ", MAE)
print("MSE: ", MSE)
print("MAPE: ", MAPE)
```

```
MAE:  49.82789432789433
MSE:  4367.432334371146
MAPE:  39.5659706359498
```

Prediction Evaluation

$$\text{Mean Absolute Error (MAE)} = \frac{\sum_{t=1}^{N} \left| d_t - d_t' \right|}{N}$$

$$\text{Mean Squared Error (MSE)} = \frac{\sum_{t=1}^{N} \left(d_t - d_t' \right)^2}{N}$$

$$\text{Mean Absolute Percent Error (MAPE)} = \frac{100}{\mathrm{N}} \sum_{\mathrm{t}=1}^{\mathrm{N}} \left[\left| \frac{\mathrm{d}_\mathrm{t} - \mathrm{d}_\mathrm{t}'}{\mathrm{d}_\mathrm{t}} \right| \right]$$