POLINOMIAL REGRESSION

Mohan M J

Introduction

Sometimes, a plot of the residuals versus a predictor may suggest there is a nonlinear relationship. One way to try to account for such a relationship is through a **polynomial regression** model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_h X^h$$

h is called the **degree** of the polynomial

(*h* = 2 is called **quadratic**, *h* = 3 is called **cubic**, *h* = 4 is called **quartic**, and so on)

polynomial regression is still considered linear regression since it is linear in the regression coefficients

Contd..

Simple Linear Regression	$y = b_0 + b_1 x_1$
Multiple Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$
Polynomial Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$

Exercise

We are about to make an offer to the new employee. It is time to negotiate on salary. The employee is asking for 160k.

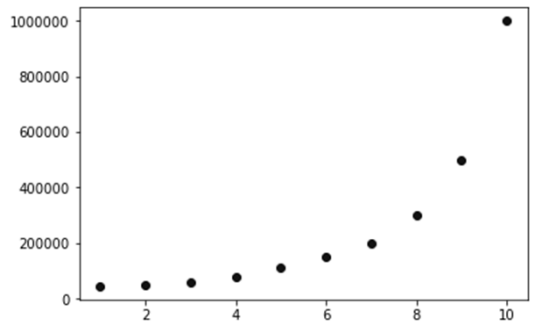
Salary of employees at different levels in an organization is given. Find out the actual salary for the employee that can be offered from the available data.

Python Code:

```
import numpy as np
import matplotlib.pyplot as myPlot
import pandas as pd
# Importing the dataset
myData = pd.read_csv('Position_Salaries.csv')
myData
X = myData.iloc[:, 1:2].values
X
y = myData.iloc[:, 2].values
y
```

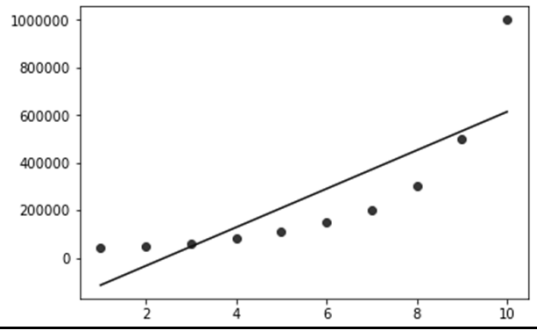
Python Code:

```
myPlot.scatter(X,y, color='b')
myPlot.show()
```



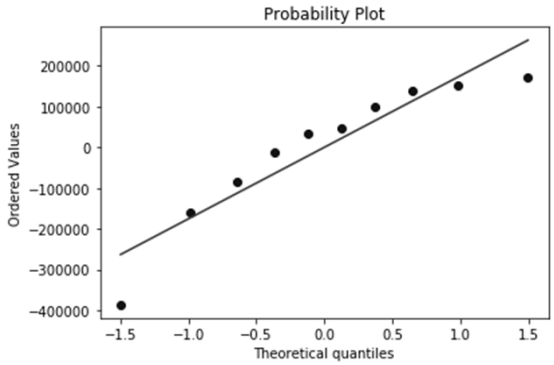
Python Code:

```
from sklearn.linear_model import LinearRegression
myModel_linear = LinearRegression().fit(X,y)
myPlot.scatter(X,y, color='r')
myPlot.plot(X,myModel_linear.predict(X), color='b')
myPlot.show()
```



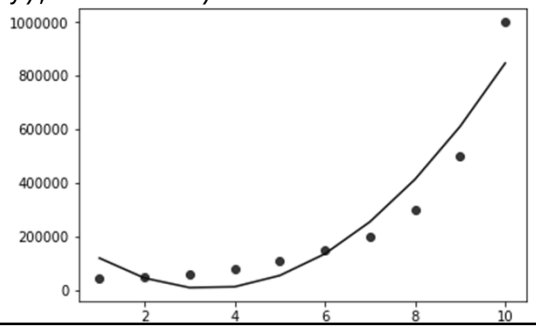
Python Code:

```
from scipy import stats as mystats
res = myModel_linear.predict(X) -y
mystats.probplot(res,plot=myPlot)
myPlot.show()
```



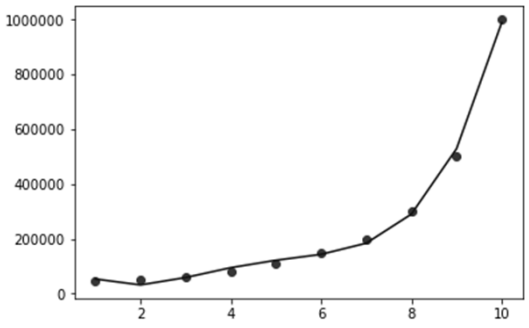
Python Code:

```
from sklearn.preprocessing import PolynomialFeatures
X_poly = PolynomialFeatures(degree=2).fit_transform(X)
X_poly
myModel_poly = LinearRegression().fit(X_poly,y)
myPlot.scatter(X,y, color='r')
myPlot.plot(X,myModel_poly.predict(X_poly), color='b')
myPlot.show()
```



Python Code:

```
X_poly = PolynomialFeatures(degree=4).fit_transform(X)
myModel_poly = LinearRegression().fit(X_poly,y)
myPlot.scatter(X,y, color='r')
myPlot.plot(X,myModel_poly.predict(X_poly), color='b')
myPlot.show()
```



THANK YOU