SIMPLE LINEAR AND POLYNOMIAL REGRESSION

Analysis of Variance

Analisis varians (ANOVA) adalah teknik statistik untuk menganalisis variasi dalam variabel respons (variabel acak kontinu) yang diukur dalam kondisi yang ditentukan oleh faktor diskrit (variabel klasifikasi, seringkali dengan level nominal). Seringkali, ANOVA digunakan untuk menguji kesetaraan di antara beberapa cara dengan membandingkan varians antar kelompok relatif terhadap varians dalam kelompok (kesalahan acak).

ANOVA for Regression:

Analysis of Variance (ANOVA) untuk model regresi terdiri dari perhitungan yang memberikan informasi tentang tingkat variabilitas dalam model regresi dan membentuk dasar untuk pengujian signifikansi.

Hipotesis dalam model regresi linier sederhana diberikan sebagai berikut

$$\begin{cases} H_0 : a_1 = 0 \\ H_1 : a_1 \neq 0 \end{cases}$$

Secara umum tabel ANOVA diberikan sebagai berikut ini:

Source	Sum of Sequence	degree of freedom (df)	MS	F value	p-value
SSR	$\sum_{i=1}^{m}(\hat{y}_i - \bar{y})^2$	n	MSR	$F*$	$k*$
SSE	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	$m-n-1$	MSE		
SST	$\sum_{i=1}^{n}(y_i - \bar{y})^2$	$m-1$			

dengan

- $MSR = \dfrac{SSR}{n}$,
- $MSE = \dfrac{SSE}{m-n-1}$,
- $F* = \dfrac{MSR}{MSE}$,
- $k* = P(F_{n,m-n-1} > F*)$ (probability of $F*$ in F distribution) and
- for simple linear regression $n = 1$.

Kesimpulan dalam statistical Hypothesis:

- if p-value ($k*$) < α (significance level), then **REJECT** H_0 hyphotesis
- if p-value ($k*$) > α (significance level), then **DO NOT REJECT** H_0 hyphotesis

Example 1:

Given $F* = 2.79$, with df $n = 4$ and $m = 30$, find the $k*$ and compare with the $\alpha = 0.05$ to make the conclusion

Solusi

In [1]:
```
import scipy
from scipy import stats


Fs = 2.79
n = 4
m = 30
dfn = n
dfd = m-n-1
ks = 1-scipy.stats.f.cdf(Fs, dfn, dfd)
print(ks)
```

```
0.048176948932526664
```

Example 2

Given the following data:

$y = \{22, 42, 44, 52, 45, 37\}$

$\hat{y} = \{52, 33, 8, 47, 43, 32\}$

Please find the $F*$ from the above data and use degree of freedom $n = 1$ and $m = 6$.

Solusi

Membuat tabel ANOVA

In [2]:
```
import numpy as np
import pandas as pd

def ANOVATAB(y,yhat,n,m):
  dfn = n
  dfd = m-n-1
  ybar = np.average(y) #average aja gpp

  SSR = np.sum((yhat-ybar)**2)
  SSE = np.sum(y-yhat)**2
  SST = np.sum((y-ybar)**2)

  MSR = SSR/dfn
  MSE = SSE/dfd
  Fs = MSR/MSE
  ks = 1-scipy.stats.f.cdf(Fs, dfn, dfd)

  data_table= {
    'SS': [SSR, SSE, SST],
    'df': [dfn, dfd,m-1] ,
    'MS': [MSR, MSE,'-'],
    'Fs': [Fs, '-','-'],
    'pval': [ks, '-','-']
  }

  return pd.DataFrame(data_table)
```

Memanggil fungsi ANOVA

In [3]:
```
n =  1
```

```
m = 6
y = np.array([ 22,42,44,52,45,37])
yhat = np.array([52,33,8,47,43,31 ])

print('Tabel ANOVA diberikan sebagai berikut ini')
results = ANOVATAB(y,yhat,n,m)

results
#Terima h0 karena >alfa
```

```
Tabel ANOVA diberikan sebagai berikut ini
```

Out[3]:

	SS	df	MS	Fs	pval
0	1374.000000	1	1374.0	7.010204	0.05712
1	784.000000	4	196.0	-	-
2	521.333333	5	-	-	-

Melihat nilai $F*$

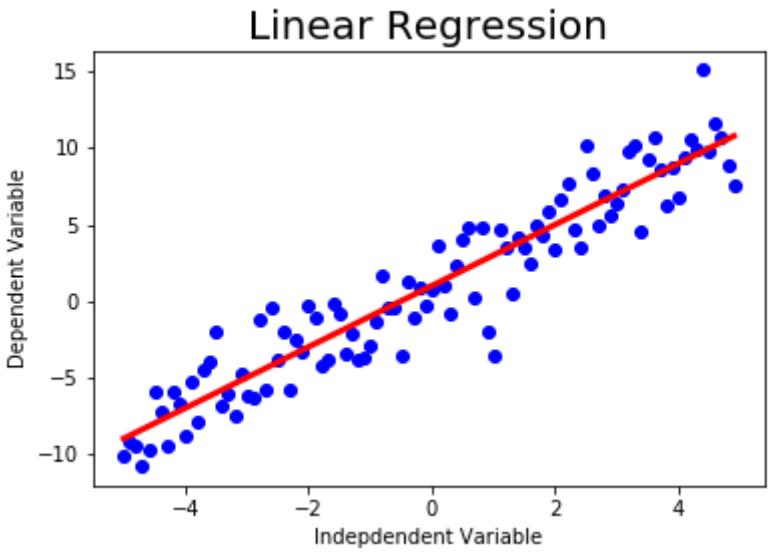
In [4]:

```
Fs = results.Fs[0]
print(Fs)
```

```
7.010204081632653
```

Simple Linear Regression

Regresi linier adalah salah satu teknik statistik dan pembelajaran mesin dasar. Baik Anda ingin melakukan statistik, pembelajaran mesin, atau komputasi ilmiah, ada kemungkinan besar Anda akan membutuhkannya. Dianjurkan untuk mempelajarinya terlebih dahulu dan kemudian melanjutkan ke metode yang lebih kompleks.



Model sederhana regresi linier diberikan sebagai berikut ini:

$y = a_1 x + a_0$

Langkah 1: Menyiapkan data

Dalam demo ini, kita coba menggunakan data biaya memasang iklan di TV, Radio dan Koran serta hasil penjualannya.

In [5]:
```
import pandas as pd

url = 'http://bit.ly/Test-PHN'
data = pd.read_csv(url)

data
```

Out[5]:

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...	...	...	...	...	...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

200 rows × 5 columns

In [6]:
```
data.describe()
```

Out[6]:

	Unnamed: 0	TV	radio	newspaper	sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

Langkah 2: Menyiapkan data untuk variabel independent dan dependent

$y = a_1 x + a_0$

y = variabel dependent

x = variabel independent

a_0 dan a_1 = konstanta yang harus dicari

Dalam kasus ini akan dicoba untuk mencari hubungan antara Iklan di TV (x) dan Jumlah penjualan/ sales (y)

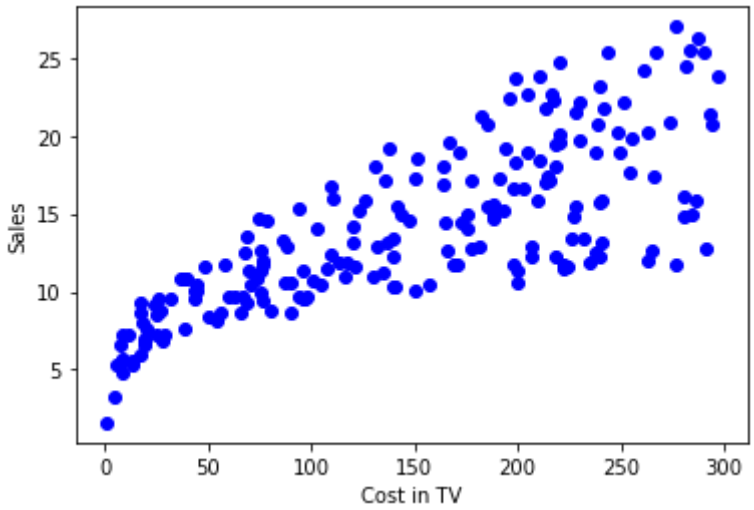
In [7]:
```
cdf = data[['TV', 'sales']]
cdf.head(9)
```

Out[7]:

	TV	sales
0	230.1	22.1
1	44.5	10.4
2	17.2	9.3
3	151.5	18.5
4	180.8	12.9
5	8.7	7.2
6	57.5	11.8
7	120.2	13.2
8	8.6	4.8

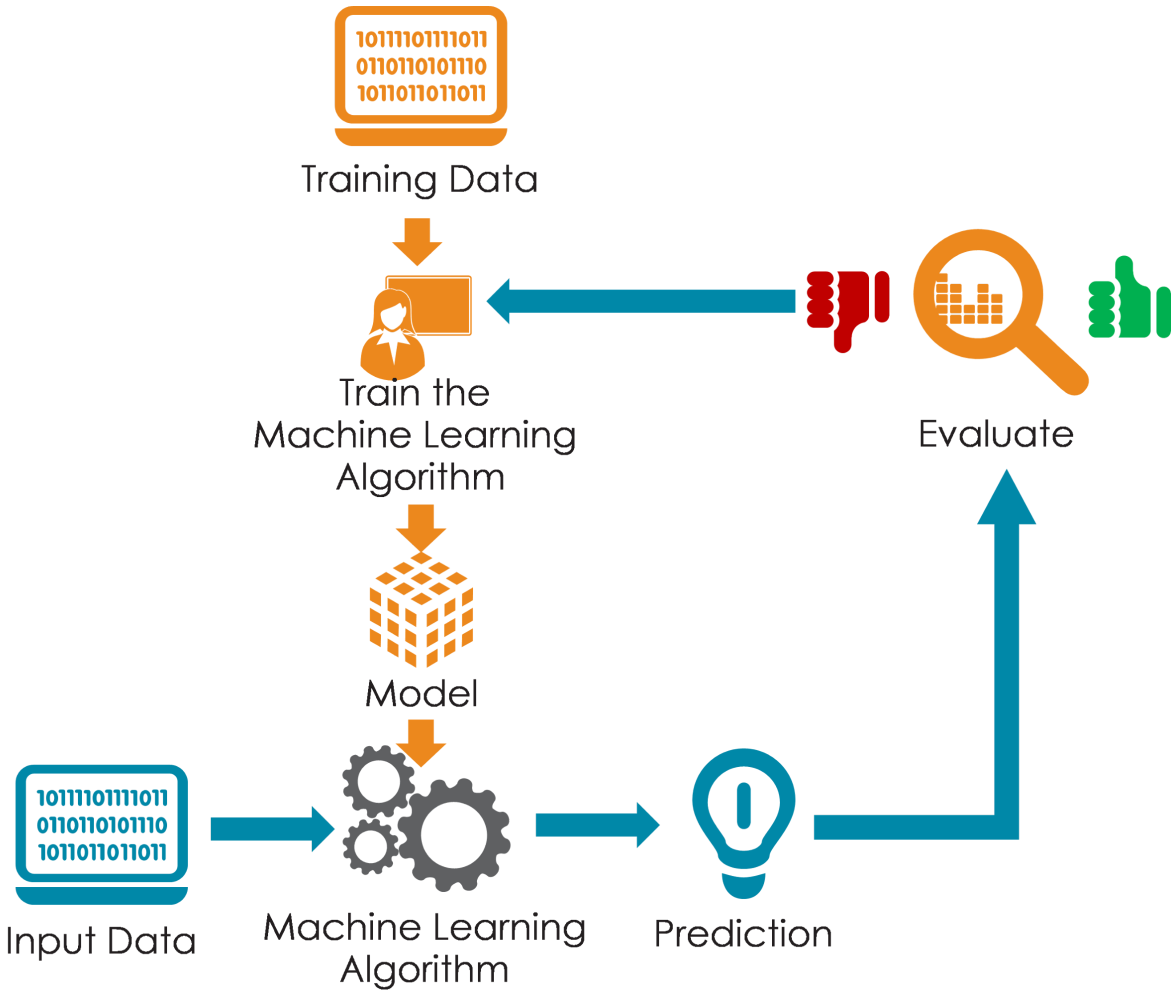
Plot data untuk melihat sebaran atau profil data

In [8]:
```
import matplotlib.pyplot as plt
plt.scatter(data['TV'],data['sales'], color='blue')
plt.xlabel("Cost in TV")
plt.ylabel("Sales")
plt.show()
```



Langkah 3: Menentukan data training dan dan data testing

Pada langkah ini kita akan menggunakan 80% data sebagai training dan 20% sebagai testing.



credit: https://medium.com/@tekaround/train-validation-test-set-in-machine-learning-how-to-understand-6cdd98d4a764

In [9]:
```
import numpy as np
msk = np.random.rand(len(data)) < 0.8
train = cdf[msk]
test = cdf[~msk]

x = train.TV
y = train.sales
```

Langkah 4: Membangun Model Regresi Linier

Untuk memperkirakan data dengan garis, pertimbangkan terlebih dahulu tentang persamaan garis seperti berikut ini:

$$y(x) = a_1 x + a_0$$

dengan x adalah biaya publikasi di TV dan $y(x)$ nilai penjualan.

Menggunakan **Least Square Method** untuk setiap data ($\forall i \in \{1, 2, \cdots, m\}$), dan total error didefinisikan sebagai berikut

$$E(a_0, a_1) = \sum_{i=1}^{m} |y_i - (a_1 x_i + a_0)|$$

Karena masalah ini bertujuan untuk meminimalisir eror, maka fungsi eror $E(a_0, a_1)$ perlu diminimalkan dengan menerapkan turunan parsial untuk a_0 dan a_1. Masalah bahwa persamaan

di atas tidak dapat dibedakan karena fungsi nilai absolut tidak dapat terdiferensiasi pada nol, oleh karena itu eror total diberikan sebagai berikut ini

$$E(a_0, a_1) = \sum_{i=1}^{m} |y_i - (a_1 x_i + a_0)|^2$$

Sehingga meminimukan error menjadi,

$$\frac{\partial E}{\partial a_0} = 2 \sum_{i=1}^{m} |y_i - (a_1 x_i + a_0)|(-1) = 0$$

dan

$$\frac{\partial E}{\partial a_1} = 2 \sum_{i=1}^{m} |y_i - (a_1 x_i + a_0)|(-x_i) = 0$$

Akhirnya, sistem persamaan didapatkan

$$a_0 \sum_{i=1}^{m} 1 + a_1 \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} y_i$$
$$a_0 \sum_{i=1}^{m} x_i + a_1 \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} y_i x_i$$

atau dapat ditulis sebagai bentuk matrix dan vektor

$$\begin{bmatrix} m & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} y_i x_i \end{bmatrix}$$

Definisikan matriks A dan vektor $\vec{b}$ dan mencari koefisien $\vec{c}$ dengan

$$A \cdot \vec{c} = \vec{b}$$

In [10]:

```
a11 =  len(x)
a12 =  sum(x)
a21 =  sum(x)
a22 =  sum(x**2)

b1 =  sum(y)
b2 =  sum(y*x)

A = np.array([[a11, a12], [a21, a22]])
print('Matrix A:')
print(A)
b = np.array([ b1, b2])
print('Vector b')
print(b)

c = np.linalg.inv(A)@b   # c = inv(A) b
print('Coefficients:')
print(c)
```

```
Matrix A:
[[1.62000000e+02 2.40252000e+04]
 [2.40252000e+04 4.68021934e+06]]
Vector b
[  2284.6  393567.21]
```

```
Coefficients:
[6.83424667 0.04900904]
```

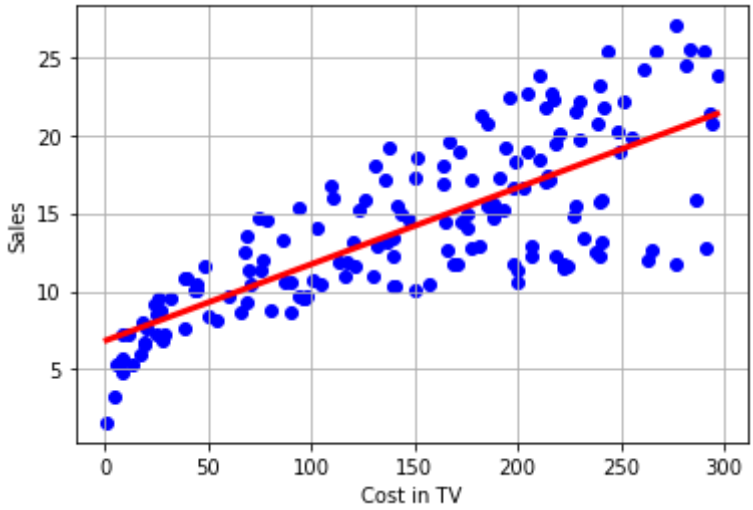
In [11]:
```
print(c[0],c[1])
```

```
6.834246670128039 0.04900904214904589
```

Plot fungsi hampiran dan data Training

In [12]:
```
xp = np.linspace(min(x), max(x), 100)
yp = c[1]*xp + c[0]


plt.plot(xp,yp, color = 'red', linewidth=3)
plt.scatter(x,y,  color='blue')
plt.xlabel("Cost in TV")
plt.ylabel("Sales")
plt.grid()
plt.show()
```



Langkah 5: Mengevaluasi model yang didapatkan

Untuk mengevaluasi model regresi, terdapat dua matriks yakni

1. Mean Squared Error : model yang bagus adalah model yang memeiliki nilai MSE mendekati 0.
2. R2 Score : model yang yang bagus adalah model yang memiliki R2 score mendekati 1.

Kita akan menguji model dengan data Testing

MSE dihitung dengan

$$MSE = \frac{1}{m-n-1}\sum_{i=1}^{m}|y_i - \hat{y}_i|^2$$

In [13]:
```
x = test.TV
y = test.sales
yhat = c[1]*x +c[0]

MSE = sum((y-yhat)**2)/(len(x)-1-1)

print(MSE)
```

```
9.140790891569583
```

R^2 dihitung dengan

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}$$

dengan

- y_i adalah data asli (test)
- $\hat{y}_i$ adalah data hampiran
- $\bar{y}$ adalah rata-rata data asli (test)

In [14]:
```
ybar = np.average(y)
SSE = sum((y-yhat)**2)
SST = sum((y-ybar)**2)
R2 = 1 - SSE/SST
print('R-squared =',R2)
```

```
R-squared = 0.6542007596579454
```

Langkah 6 Membuat kesimpulan dari Tabel ANOVA

Pada langkah ini, kita menentukan tabel ANOVA terlebih dahulu, lalu menentukan kesimpulan dari p-value yang didapatkan.

Pada contoh ini digunakan Hypothesis testing:

$$\begin{cases} H_0 : a_1 = 0 \\ H_1 : a_1 \neq 0 \end{cases}$$

dengan level of significance $\alpha = 5\%$.

In [15]:
```
n = 1
m = len(x)

results = ANOVATAB(y,yhat,n,m)

results
#Pvalue 0, TOLAK H0
```

Out[15]:

	SS	df	MS	Fs	pval
0	836.851059	1	836.851059	2284.1192	0.0
1	13.189609	36	0.366378	-	-
2	951.617105	37	-	-	-

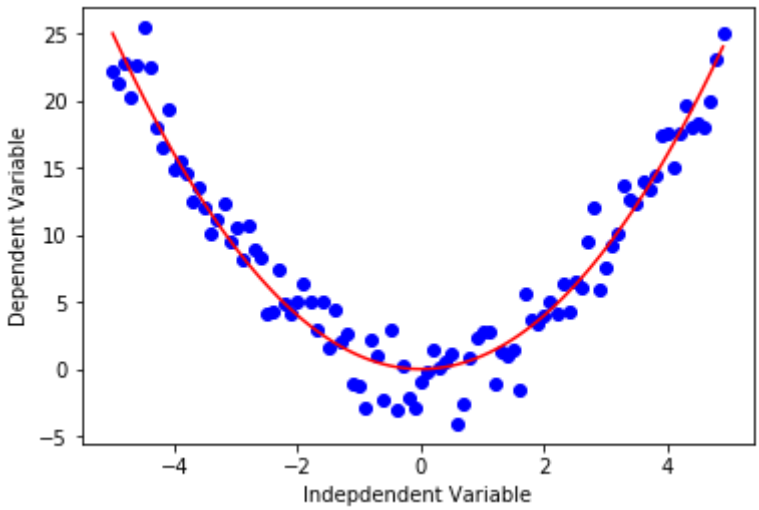
Kesimpulan:

Karena P-val < α, Maka tolak H_0

Hal ini berarti data dapat dimodelkan dengan garis linier.

Polynomial Regression

Polynomial Regresi adalah model matematika darri sebaran data dalam bentuk persamaan polinomial. Model dapat berbentuk polinomial orde lebih besar dari 2.



Secara umum untuk menghampiri himpunan data, $\{(x_i, y_i)|i = 1, 2, \cdots, m\}$, dapat menggunakan model polinomial berikut ini,

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

dengan derajat $n < m - 1$, dan model dapat dibentuk mengguanakn metode least squared method.

Langkah 1: Menyiapkan data

Dalam demo ini, kita coba menggunakan data biaya memasang iklan di TV, Radio dan Koran serta hasil penjualannya.

In [16]:
```
import pandas as pd

url = 'http://bit.ly/Test-PHN'
data = pd.read_csv(url, index_col=0)

data.head()
```

Out[16]:

	TV	radio	newspaper	sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

Langkah 2: Menyiapkan data untuk variabel independent dan dependent

Dalam kasus ini akan dicoba untuk mencari hubungan antara Iklan di TV (x) dan Jumlah penjualan/ sales (y)

In [17]:
```
cdf = data[['TV', 'sales']]
cdf.head()
```

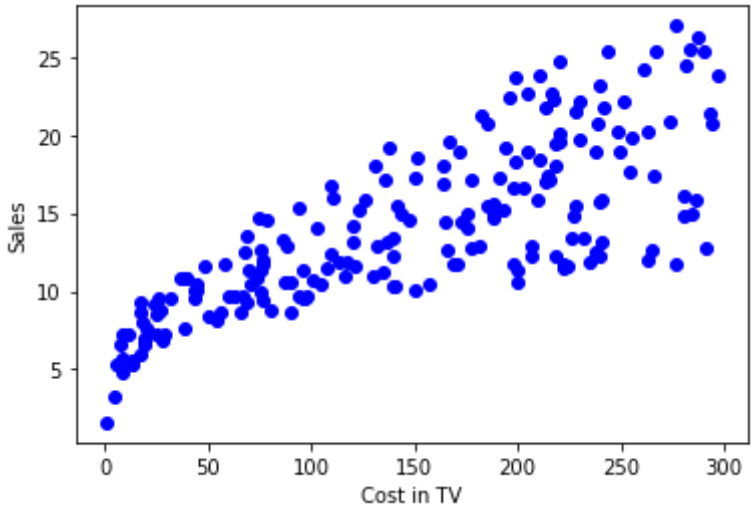
Out[17]:

	TV	sales
1	230.1	22.1
2	44.5	10.4
3	17.2	9.3
4	151.5	18.5
5	180.8	12.9

Plot data

In [18]:

```
import matplotlib.pyplot as plt
plt.scatter(cdf.TV, cdf.sales,  color='blue')
plt.xlabel("Cost in TV")
plt.ylabel("Sales")
plt.show()
```



Langkah 3: Menentukan data training dan dan data testing

Pada langkah ini kita akan menggunakan 80% data sebagai training dan 20% sebagai testing.

In [19]:

```
import numpy as np
msk = np.random.rand(len(data)) < 0.8
train = cdf[msk]
test = cdf[~msk]
test.head()

x =  train.TV
y =  train.sales
```

Langkah 4: Membangun Model Regresi Polinomial

The coefficients $a_0, a_1, \cdots, a_n$ are obtained by minimizing the least square error,

$$
\begin{aligned}
E &= \sum_{i=1}^{m}(y_i - P_n(x_i))^2 \\
&= \sum_{i=1}^{m} y_i^2 - 2\sum_{i=1}^{m} P_n(x_i)y_i + \sum_{i=1}^{m}(P_n(x_i))^2 \\
&= \sum_{i=1}^{m} y_i^2 - 2\sum_{j=0}^{n} a_j \left(\sum_{i=1}^{m} y_i x_i^j\right) + \sum_{j=0}^{n}\sum_{k=0}^{n} a_j a_k \left(\sum_{i=1}^{m} x_i^{j+k}\right)
\end{aligned}
$$

As in the linear case, for E to be minimized it is necessary that $\partial E/\partial a_j = 0$, for each $j = 0, 1, \cdots, n$. Thus, for each j,

$$
\frac{\partial E}{\partial a_j} = -2\sum_{i=1}^{m} y_i x_i^j + 2\sum_{k=0}^{n} a_k \sum_{i=1}^{m} x_i^{k+j} = 0
$$

This gives $n+1$ **normal equations** in the $n+1$ unknowns a_j. Resulting,

$$
\sum_{k=0}^{n} a_k \sum_{i=1}^{m} x_i^{k+j} = \sum_{i=1}^{m} y_i x_i^j
$$

Or it can be written as

$$
\begin{aligned}
a_0\sum_{i=1}^{m} x_i^0 + a_1\sum_{i=1}^{m} x_i^1 + a_2\sum_{i=1}^{m} x_i^2 + \cdots + a_n\sum_{i=1}^{m} x_i^n &= \sum_{i=1}^{m} y_i x_i^0, \\
a_0\sum_{i=1}^{m} x_i^1 + a_1\sum_{i=1}^{m} x_i^2 + a_2\sum_{i=1}^{m} x_i^3 + \cdots + a_n\sum_{i=1}^{m} x_i^{n+1} &= \sum_{i=1}^{m} y_i x_i^1, \\
&\vdots \\
a_0\sum_{i=1}^{m} x_i^n + a_1\sum_{i=1}^{m} x_i^{n+1} + a_2\sum_{i=1}^{m} x_i^{n+2} + \cdots + a_n\sum_{i=1}^{m} x_i^{2n} &= \sum_{i=1}^{m} y_i x_i^n
\end{aligned}
$$

Untuk membentuk model regresi polinomial orde dua P_2, maka model yang diinginkan adalah

$$
y(x) = a_2 x^2 + a_1 x + a_0
$$

dengan x adalah biaya publikasi di TV dan $y(x)$ nilai penjualan.

In [20]:

```
a11 =  len(x)
a12 =  sum(x)
a13 =  sum(x**2)

a21 =  sum(x)
a22 =  sum(x**2)
a23 =  sum(x**3)

a31 =  sum(x**2)
a32 =  sum(x**3)
a33 =  sum(x**4)

b1 =  sum(y)
b2 =  sum(y*x)
b3 =  sum(y*(x**2))

A = np.array([[a11, a12, a13], [a21, a22, a23], [a31, a32, a33]])

b = np.array([ b1, b2, b3])
```

```
c = np.linalg.inv(A)@b
print('Koefisien model:')
print(c)
```

```
Koefisien model:
[ 5.96021321e+00  7.31793816e-02 -8.92593416e-05]
```

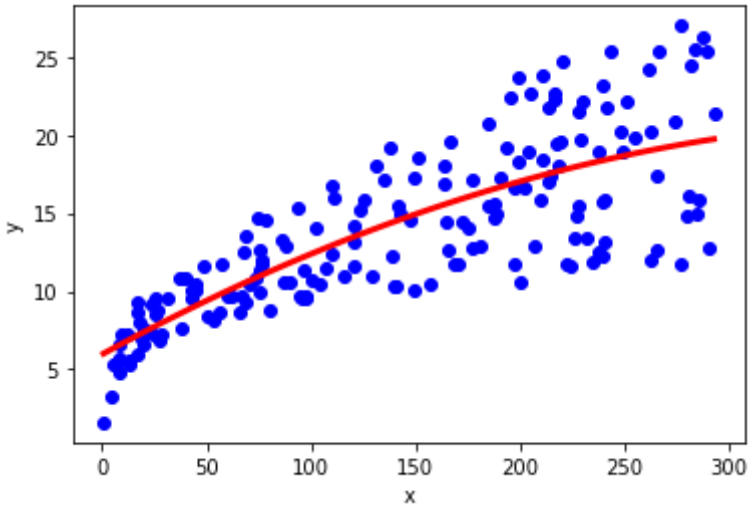
Plot the results

In [21]:

```
import matplotlib.pyplot as plt

xp= np.linspace(min(x),max(x),100)
yp = c[2]*xp**2 + c[1]*xp + c[0]

plt.scatter(x, y,  color='blue')
plt.plot(xp, yp , color = 'red', linewidth=3)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



Langkah 5: Mengevaluasi model yang didapatkan

Untuk mengevaluasi model regresi, terdapat dua matriks yakni

1. Mean Squared Error : model yang bagus adalah model yang memeiliki nilai MSE mendekati 0.
2. R2 Score : model yang yang bagus adalah model yang memiliki R2 score mendekati 1.

Kita akan menguji model dengan data Testing

MSE Untuk Polynomial P_2 maka MSE dihitung dengan $n = 2$,

$$MSE = \frac{1}{m-n-1}\sum_{i=1}^{m}|y_i - \hat{y}_i|^2$$

In [22]:

```
x = test.TV
y = test.sales
yhat = c[2]*x**2 + c[1]*x + c[0]

MSE = sum((y-yhat)**2)/(len(x)-1-1)

print(MSE)
```

```
10.14697611826456
```

R^2 dihitung dengan

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}$$

dengan

- y_i adalah data asli (test)
- $\hat{y}_i$ adalah data hampiran
- $\bar{y}$ adalah rata-rata data asli (test)

In [23]:
```
ybar = np.average(y)
SSE = sum((y-yhat)**2)
SST = sum((y-ybar)**2)
R2 = 1 - SSE/SST
print('R-squared =',R2)
```

```
R-squared = 0.5216545130618442
```

Langkah 6: Membuat Kesimpulan dengan Tabel ANOVA

Pada langkah ini, kita menentukan tabel ANOVA terlebih dahulu, lalu menentukan kesimpulan dari p-value yang didapatkan.

Pada contoh ini digunakan Hypothesis testing:

$$\begin{cases} H_0 : a_1 = a_2 = \cdots = a_n = 0 \\ H_1 : a_j \neq 0 \text{ for at least } j \end{cases}$$

dengan level of significance $\alpha = 5\%$.

In [24]:
```
n = 2
m = len(x)

results = ANOVATAB(y,yhat,n,m)

results
```

Out[24]:

	SS	df	MS	Fs	pval
0	246.502311	2	123.251155	5.010988	0.016625
1	516.519797	21	24.596181	-	-
2	466.678333	23	-	-	-

Kesimpulan:

Karena P-val < α, Maka tolak H_0

Hal ini berarti bahwa koefisien polynomial tidak nol dan data dapat dimodelkan dengan polinomial orde 2 (P_2).

Homework

Please solve the following problems:

1. Define a linear regression from data 'http://bit.ly/Test-PHN2' and analyze the performances
2. Define a Polynomial P_2 linear regression of cost in Newspaper vs sales and compute the performances
3. Define a Polynomial P_3 linear regression from data 'http://bit.ly/Test-PHN2' and analyze the performance

1. Modelkan data Day dan Temperature (https://bit.ly/3cXpKbJ) dengan regresi linier sederhana.

Input data

In [61]:
```
import scipy
from scipy import stats
```

In [73]:
```
import pandas as pd
url = 'https://bit.ly/3cXpKbJ'
data = pd.read_csv(url)
data
```

Out[73]:

	Day	Temperature
0	1	1.3
1	2	3.5
2	3	4.2
3	4	5.0
4	5	7.0
5	6	8.8
6	7	10.1
7	8	12.5
8	9	13.0
9	10	15.6

In [74]:
```
cdf = data[['Day','Temperature']]
cdf
```

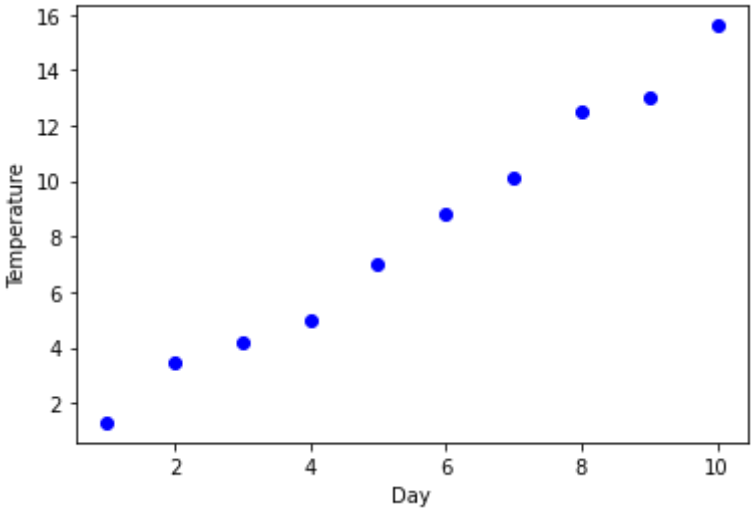
Out[74]:

	Day	Temperature
0	1	1.3
1	2	3.5
2	3	4.2
3	4	5.0
4	5	7.0
5	6	8.8
6	7	10.1

	Day	Temperature
7	8	12.5
8	9	13.0
9	10	15.6

Plot untuk melihat pesebaran data

In [75]:
```
import matplotlib.pyplot as plt
plt.scatter(cdf.Day, cdf.Temperature,  color='blue')
plt.xlabel("Day")
plt.ylabel("Temperature")
plt.show()
```



Pembagian data train 80% dan data test 20%

In [81]:
```
import numpy as np
msk = np.random.rand(len(data)) < 0.8
train = cdf[msk]
test = cdf[~msk]
x = train.Day
y = train.Temperature
```

Model regresi linear

In [82]:
```
a11 = len(x)
a12 = sum(x)
a21 = sum(x)
a22 = sum(x**2)

b1 = sum(y)
b2 = sum(y*x)

A = np.array([[a11, a12], [a21, a22]])
print('Matrix A adalah:')
print(A)

b = np.array([ b1, b2])
print('Vector b adalah:')
print(b)

c = np.linalg.inv(A)@b
```

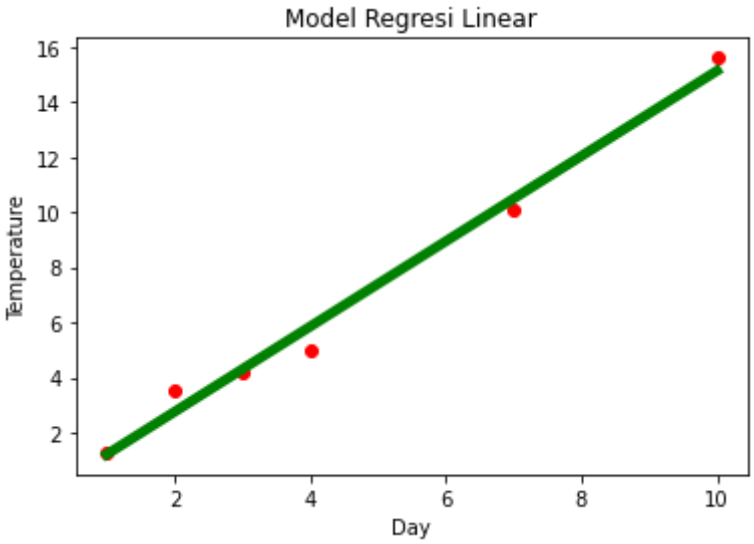
```
print('Koefisien (c) adalah:')
print(c)
```

```
Matrix A adalah:
[[  6  27]
 [ 27 179]]
Vector b adalah:
[ 39.7 267.6]
Koefisien (c) adalah:
[-0.34463768  1.54695652]
```

In [83]:

```
xp = np.linspace(min(x), max(x), 100)
yp = c[1]*xp + c[0]


plt.plot(xp,yp, color = 'green', linewidth=5)
plt.scatter(x,y, color='red')
plt.xlabel("Day")
plt.ylabel("Temperature")
plt.title("Model Regresi Linear")
plt.show()
```



Evaluasi Model

In [84]:

```
x = test.Day
y = test.Temperature
yhat = c[1]*x +c[0]

MSE = sum((y-yhat)**2)/(len(x)-1-1)

print(MSE)
```

```
0.36250388573829184
```

In [85]:

```
ybar = np.average(y)
SSE = sum((y-yhat)**2)
SST = sum((y-ybar)**2)
R2 = 1 - SSE/SST
print('R-squared =',R2)
n = 1
m = len(x)
results = ANOVATAB(y,yhat,n,m)
results
```

```
R-squared = 0.9713067073720556
```

Out[85]:

	SS	df	MS	Fs	pval
0	24.031943	1	24.031943	66.294303	0.014751
1	0.725008	2	0.362504	-	-
2	25.267500	3	-	-	-

Karena P-val < α, Maka tolak H_0

Hal ini berarti data dapat dimodelkan dengan garis linier.

2. Buatlah Model dan ukur model regresi liner sederhana menggunakan data newspaper dan sales

In [86]:

```
import pandas as pd
url = 'http://bit.ly/Test-PHN'
data = pd.read_csv(url,index_col=0)
data
```

Out[86]:

	TV	radio	newspaper	sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
...	...	...	...	...
196	38.2	3.7	13.8	7.6
197	94.2	4.9	8.1	9.7
198	177.0	9.3	6.4	12.8
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

200 rows × 4 columns

In [87]:

```
cdf = data[['newspaper', 'sales']]
cdf.head(9)
```

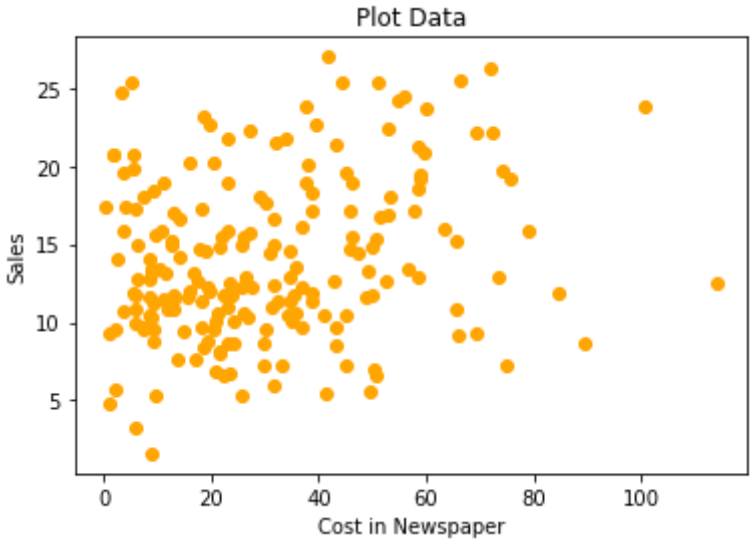
Out[87]:

	newspaper	sales
1	69.2	22.1
2	45.1	10.4
3	69.3	9.3
4	58.5	18.5
5	58.4	12.9
6	75.0	7.2

	newspaper	sales
7	23.5	11.8
8	11.6	13.2
9	1.0	4.8

In [88]:

```
import matplotlib.pyplot as plt
plt.scatter(cdf.newspaper, cdf.sales, color='orange')
plt.title("Plot Data")
plt.xlabel("Cost in Newspaper")
plt.ylabel("Sales")
plt.show()
```



In [89]:

```
import numpy as np
msk = np.random.rand(len(data)) < 0.8
train = cdf[msk]
test = cdf[~msk]
x = train.newspaper
y = train.sales
```

In [90]:

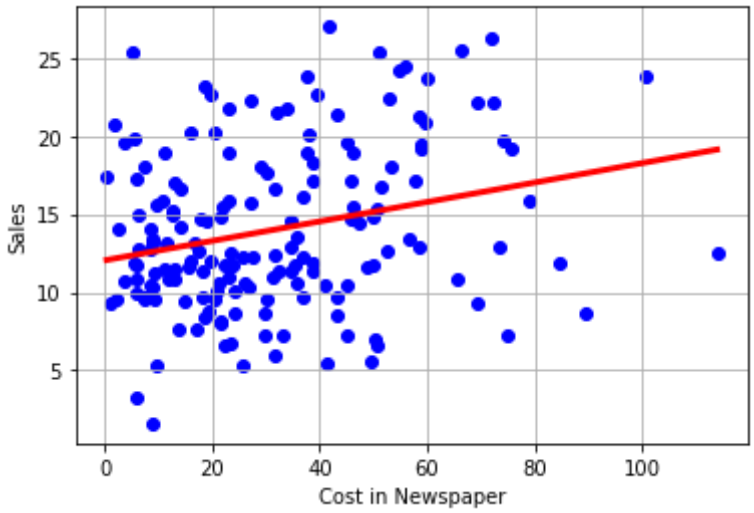
```
a11 = len(x)
a12 = sum(x)
a21 = sum(x)
a22 = sum(x**2)
b1 = sum(y)
b2 = sum(y*x)
A = np.array([[a11, a12], [a21, a22]])
print('Matrix A adalah:')
print(A)
b = np.array([ b1, b2])
print('Vector b adalah:')
print(b)
c = np.linalg.inv(A)@b # c = inv(A) b
print('Koefisien (c) adalah:')
print(c)
```

```
Matrix A adalah:
[[1.6900000e+02 5.3055000e+03]
 [5.3055000e+03 2.4732943e+05]]
Vector b adalah:
```

```
[ 2365.3 79275.3]
Koefisien (c) adalah:
[12.0446198   0.06215423]
```

In [91]:

```
xp = np.linspace(min(x), max(x), 100)
yp = c[1]*xp + c[0]
plt.plot(xp,yp, color = 'red', linewidth=3)
plt.scatter(x,y, color='blue')
plt.xlabel("Cost in Newspaper")
plt.ylabel("Sales")
plt.grid()
plt.show()
```



In [92]:

```
x = test.newspaper
y = test.sales
yhat = c[1]*x +c[0]
MSE = sum((y-yhat)**2)/(len(x)-1-1)
print("MSE = ",MSE)
```

```
MSE =  26.971920665529435
```

In [93]:

```
ybar = np.average(y)
SSE = sum((y-yhat)**2)
SST = sum((y-ybar)**2)
R2 = 1 - SSE/SST
print('R-squared =',R2)
```

```
R-squared = -0.049592836236723725
```

In [94]:

```
import scipy
from scipy import stats
def ANOVATAB(y,yhat,n,m):
    dfn = n
    dfd = m-n-1
    ybar = np.average(y)
    SSR = sum((yhat - ybar)**2)
    SSE = sum((y - yhat)**2)
    SST = sum((y - ybar)**2)
    MSR = SSR/dfn
    MSE = SSE/dfd
    Fs = MSR/MSE
    ks = 1-scipy.stats.f.cdf(Fs, dfn, dfd)
    data_table= {
        'SS': [SSR, SSE, SST],
```

```
        'df': [dfn, dfd,m-1] ,
        'MS': [MSR, MSE,'-'],
        'Fs': [Fs, '-','-'],
        'pval': [ks, '-','-']}
    return pd.DataFrame(data_table)
n = 1
m = len(x)
results = ANOVATAB(y,yhat,n,m)
results
```

Out[94]:

	SS	df	MS	Fs	pval
0	57.648779	1	57.648779	2.137363	0.154502
1	782.185699	29	26.971921	-	-
2	745.227742	30	-	-	-

Karena P-val > α, Maka tolak H_0

Hal ini berarti data tidak dapat dimodelkan dengan garis linier.

3. Define a Polinomyal P_2 linear regression of cost in Radio vs sales and compute the performances

In [95]:

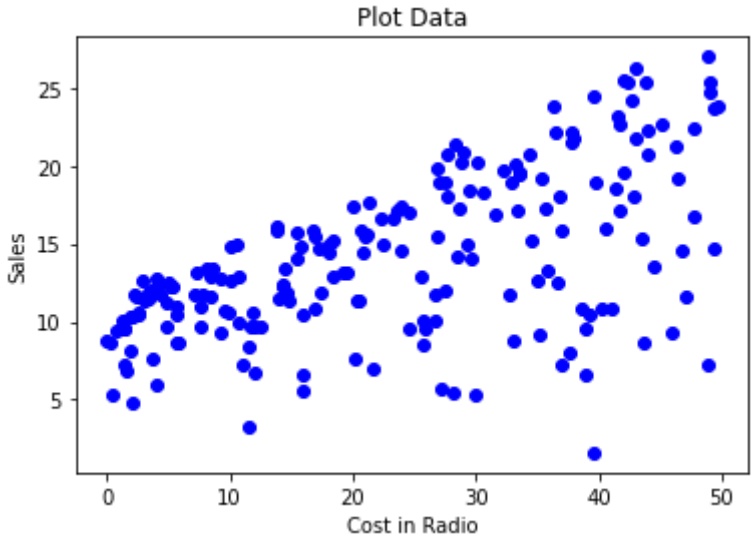
```
cdf = data[['radio', 'sales']]
cdf.head()
```

Out[95]:

	radio	sales
1	37.8	22.1
2	39.3	10.4
3	45.9	9.3
4	41.3	18.5
5	10.8	12.9

In [96]:

```
import matplotlib.pyplot as plt
plt.scatter(cdf.radio, cdf.sales, color='blue')
plt.title('Plot Data')
plt.xlabel("Cost in Radio")
plt.ylabel("Sales")
plt.show()
```



In [97]:

```
import numpy as np
msk = np.random.rand(len(data)) < 0.8
train = cdf[msk]
test = cdf[~msk]
test.head()
x = train.radio
y = train.sales
```

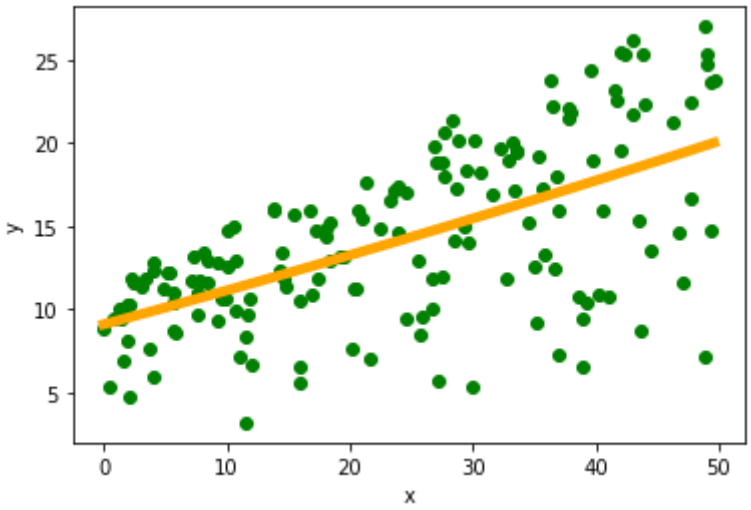
In [98]:

```
a11 = len(x)
a12 = sum(x)
a13 = sum(x**2)
a21 = sum(x)
a22 = sum(x**2)
a23 = sum(x**3)
a31 = sum(x**2)
a32 = sum(x**3)
a33 = sum(x**4)
b1 = sum(y)
b2 = sum(y*x)
b3 = sum(y*x*x)
A = np.array([[a11, a12, a13], [a21, a22, a23], [a31, a32, a33]])
b = np.array([ b1, b2, b3])
c = np.linalg.inv(A)@b
print('Koefisien model adalah:')
print(c)
```

```
Koefisien model adalah:
[9.10517322e+00 1.99622080e-01 4.11654417e-04]
```

In [99]:

```
import matplotlib.pyplot as plt
xp= np.linspace(min(x),max(x),100)
yp = c[2]*xp**2 + c[1]*xp + c[0]
plt.scatter(x, y, color='green')
plt.plot(xp, yp , color = 'orange', linewidth=5)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



In [100…

```
x = test.radio
y = test.sales
yhat = c[2]*x**2 + c[1]*x + c[0]
MSE = sum((y-yhat)**2)/(len(x)-1-1)
print( "MSE = ", MSE)
```

```
MSE =  22.33502720717391
```

In [101…

```
ybar = np.average(y)
SSE = sum((y-yhat)**2)
SST = sum((y-ybar)**2)
R2 = 1 - SSE/SST
print('R-squared =',R2)
```

```
R-squared = 0.1200046882459056
```

In [102…

```
n = 2
m = len(x)
results = ANOVATAB(y,yhat,n,m)
results
```

Out[102…

	SS	df	MS	Fs	pval
0	442.612828	2	221.306414	9.633255	0.000464
1	804.060979	35	22.973171	-	-
2	913.710526	37	-	-	-

Karena P-val < α, Maka tolak H_0

Hal ini berarti data dapat dimodelkan dengan garis linier.

4. Define a Polynomial P_3 linear regression from data 'http://bit.ly/Test-PHN2' and analyze the performance

In [103…

```
data={'xi':[0,0.25,0.50,0.75,1.0], 'yi':[1.0000,1.2840,1.6487,2.1170,2.7183]}
df = pd.DataFrame(data)
df
```

Out[103…

	xi	yi

	xi	yi
0	0.00	1.0000
1	0.25	1.2840
2	0.50	1.6487
3	0.75	2.1170
4	1.00	2.7183

In [104…

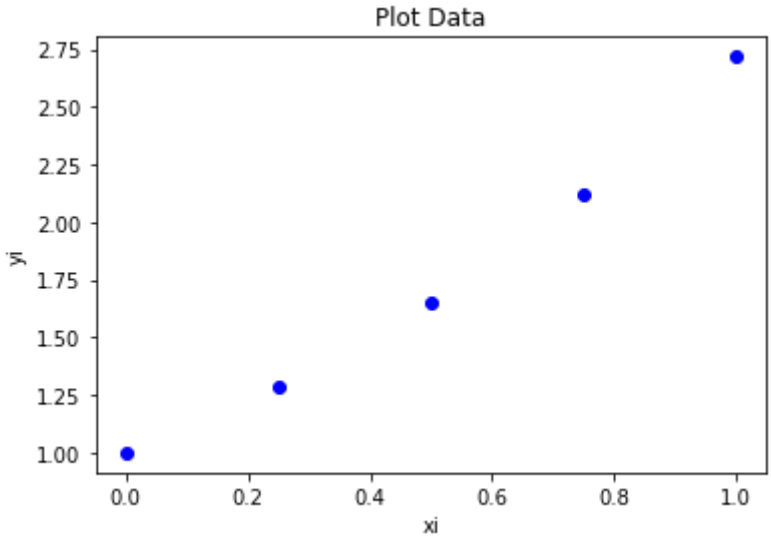
```
cdf = df[['xi', 'yi']]
cdf.head()
```

Out[104…

	xi	yi
0	0.00	1.0000
1	0.25	1.2840
2	0.50	1.6487
3	0.75	2.1170
4	1.00	2.7183

In [105…

```
import matplotlib.pyplot as plt
plt.scatter(cdf.xi, cdf.yi, color='blue')
plt.title("Plot Data")
plt.xlabel("xi")
plt.ylabel("yi")
plt.show()
```



In [106…

```
a11 = len(x)
a12 = sum(x)
a13 = sum(x**2)

a21 = sum(x)
a22 = sum(x**2)
a23 = sum(x**3)

a31 = sum(x**2)
```

```
a32 = sum(x**3)
a33 = sum(x**4)

b1 = sum(y)
b2 = sum(y*x)
b3 = sum(y*x*x)

A = np.array([[a11, a12, a13], [a21, a22, a23], [a31, a32, a33]])
b = np.array([ b1, b2, b3])
c = np.linalg.inv(A)@b

print('Koefisien model adalah:')
print(c)
```

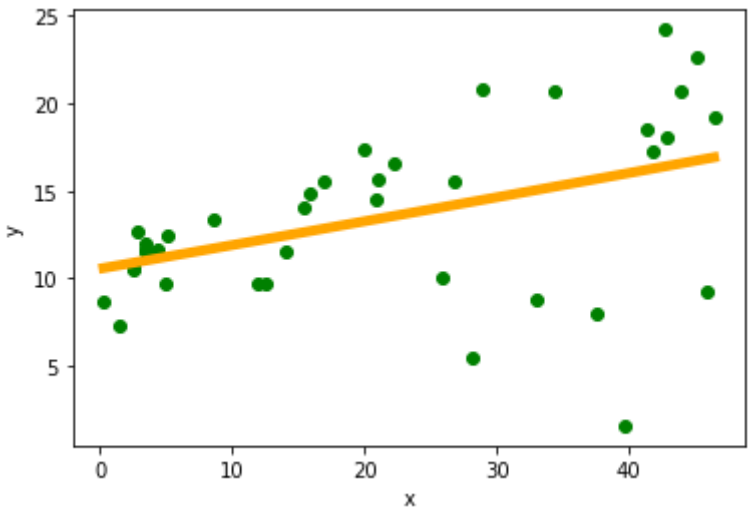
```
Koefisien model adalah:
[1.05536150e+01 1.34026382e-01 6.39448734e-05]
```

In [107…

```
import matplotlib.pyplot as plt

xp= np.linspace(min(x),max(x),100)
yp = c[2]*xp**2 + c[1]*xp + c[0]

plt.scatter(x, y,  color='green')
plt.plot(xp, yp , color = 'orange', linewidth=5)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



In [108…

```
ybar = np.average(y)
SSE = sum((y-yhat)**2)
SST = sum((y-ybar)**2)

R2 = 1 - SSE/SST
print('R-squared =',R2)
```

```
R-squared = 0.1200046882459056
```

In [109…

```
n = 2
m = len(x)

results = ANOVATAB(y,yhat,n,m)
results
```

Out[109…

	SS	df	MS	Fs	pval

	SS	df	MS	Fs	pval
0	442.612828	2	221.306414	9.633255	0.000464
1	804.060979	35	22.973171	-	-
2	913.710526	37	-	-	-

Karena P-val < α, Maka tolak H_0

Hal ini berarti data dapat dimodelkan dengan garis linier.