MULTIPLE AND MULTIVARIATE LINEAR REGRESSION

Analysis of Variance for Multiple Linear Regression (MLR)

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 beganda (MLR) diberikan sebagai berikut

$$\begin{cases} H_0 : a_1 = a_2 = \cdots = a_k = 0 \\ H_1 : a_j \neq 0, \ \text{(paling tidak ada satu j)} \end{cases}$$

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 $n = 1$.

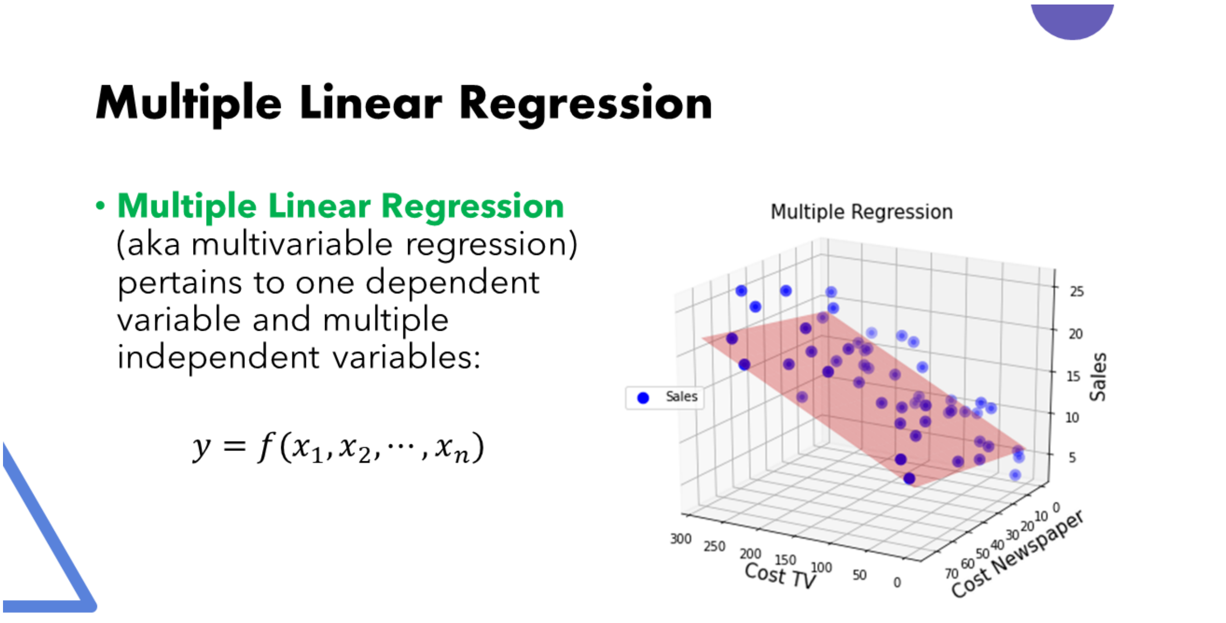
Kesimpulan dalam statistical Hypothesis:

- 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

In [47]:

```
import numpy as np
import pandas as pd
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)
```

Multiple Linear Regression



Multiple Linear Regression

- **Multiple Linear Regression** (aka multivariable regression) pertains to one dependent variable and multiple independent variables:

$$y = f(x_1, x_2, \cdots, x_n)$$

Least Squared Method

Langkah 1: Membentuk data

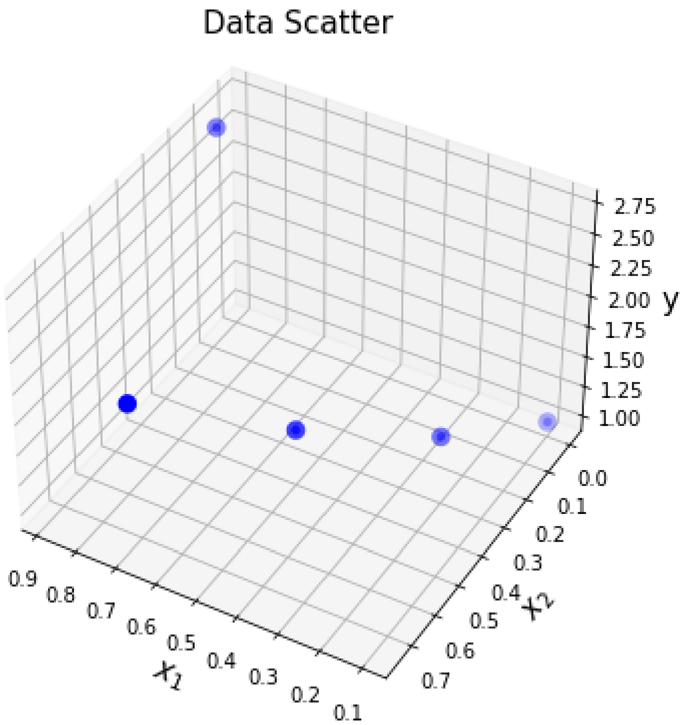
In [48]:

```
x1 = np.array([0.1,0.23,0.44,0.69,0.88])
x2 = np.array([0.,0.25,0.5,0.75,0.1])
y = np.array([1.0,1.284,1.6486,2.1170,2.7183])
```

Plot scatter untuk melihat pola data

In [49]:

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(40, 120)
ax.scatter(x1, x2, y,  color='blue',lw=5)
#plt.legend(('$\hat{y}$',), loc='center left')
ax.set_xlabel('$x_1$',fontsize=15)
ax.set_ylabel('$x_2$',fontsize=15)
ax.set_zlabel('y', fontsize=15)
ax.set_title('Data Scatter', fontsize=15)
plt.show()
```



Langkah 2: Menentuka Matrix A dan vektor y

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, A = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix},$$

In [50]:

```
m = 5  #number of row
A= np.transpose(np.array([np.ones(m),x1,x2]))
print(A)
print(y)
```

```
[[1.   0.1  0.  ]
 [1.   0.23 0.25]
 [1.   0.44 0.5 ]
 [1.   0.69 0.75]
 [1.   0.88 0.1 ]]
[1.     1.284  1.6486 2.117  2.7183]
```

Langkah 3: Mencari koefisien C dengan Least Squared

$$C = (A'A)^{-1}A'y$$

In [51]:

```
c =np.linalg.inv(np.transpose(A)@A)@np.transpose(A)@y
print(c)
```

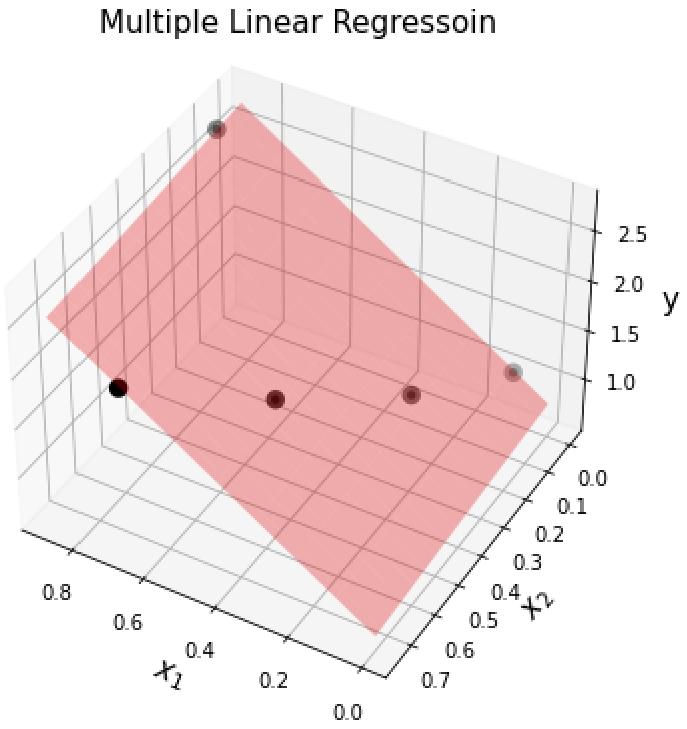
```
[ 0.80777744  2.19613823 -0.25621917]
```

Visualisasikan data dan daerah hampiran

In [52]:

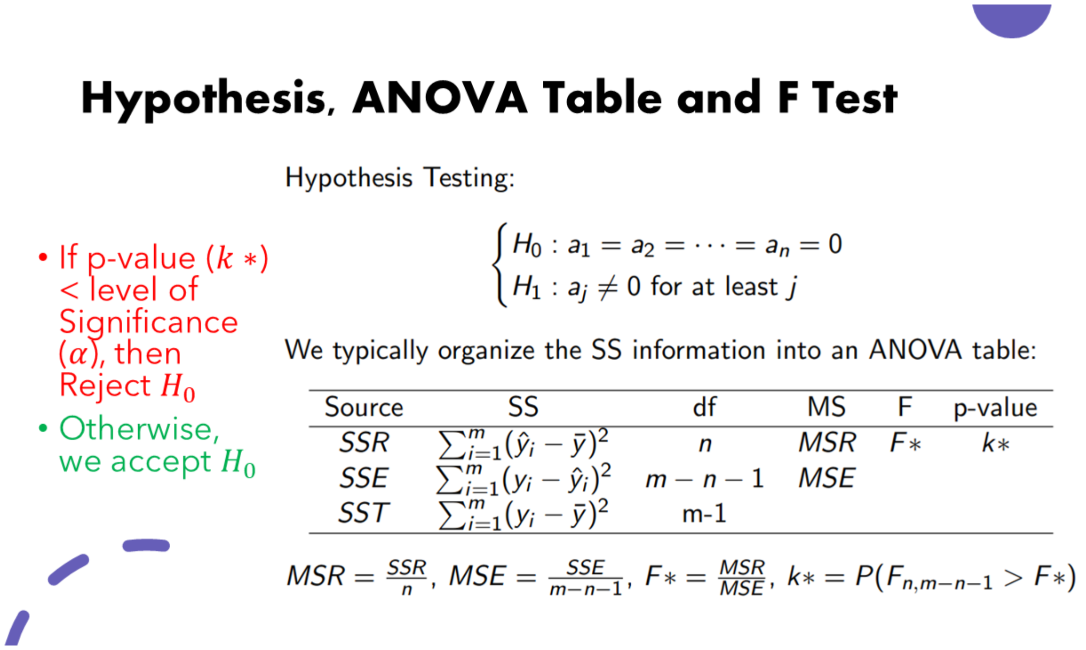
```
from matplotlib import cm
```

```
import matplotlib as mpl

def f(x1,x2):
  return c[0]+c[1]*x1+c[2]*x2

xp = np.linspace(0,max(x1),100)
yp = np.linspace(0,max(x2),100)

X,Y = np.meshgrid(xp,yp)
Z = f(X,Y)

fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(40, 120)
ax.scatter(x1, x2, y,  color='black',lw=5)
ax.plot_surface(X, Y, Z, alpha=0.3, color='red', rstride=6, cstride=12)
#plt.legend(('$\hat{y}_1$','$\hat{y}_2$'), loc='center left')
ax.set_xlabel('$x_1$',fontsize=15)
ax.set_ylabel('$x_2$',fontsize=15)
ax.set_zlabel('y', fontsize=15)
ax.set_title('Multiple Linear Regressoin', fontsize=15)
plt.show()
```



Langkah 4: Evaluasi model

1. Menentukan tabel ANOVA dari Hypothesi testing



Hypothesis, ANOVA Table and F Test

- If p-value ($k*$) < level of Significance (α), then Reject H_0
- Otherwise, we accept H_0

Hypothesis Testing:

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

We typically organize the SS information into an ANOVA table:

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

$$MSR = \frac{SSR}{n},\ MSE = \frac{SSE}{m-n-1},\ F* = \frac{MSR}{MSE},\ k* = P(F_{n,m-n-1} > F*)$$

In [53]:
```
n=2
m=len(y)
yhat = A@c
tabel = ANOVATAB(y,yhat,n,m)
tabel
```

Out[53]:

	SS	df	MS	Fs	pval
0	1.859967	2	0.929983	845.0813	0.001182
1	0.002201	2	0.0011	-	-
2	1.862168	4	-	-	-

Dengan level of siginificance (α) = 5%, maka kita mendapatkan, $pval < 5\%$. sehingga keputusan, tolak H_0

1. Mengevaluasi R^2

$$R^2 = 1 - \frac{SSE}{SST}$$

In [54]:
```
R_s = 1-tabel.SS[1]/tabel.SS[2]
print('Nilai R^2 adalah', R_s)
```

```
Nilai R^2 adalah 0.9988180804853403
```

Machine Learning Approach

Pada bagian ini akan dijelaskan bagaimana menerapkan Multiple Linear Regression pada Machine Learning

Langkah 1: Membuat atau memanggil data

Pada contoh kali ini, kita menggunakan data advertising.

In [55]:
```
import pandas as pd
```

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

data
```

Out[55]:

	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

Langkah 2: Menentukan data Training 80% dan Testing 20%

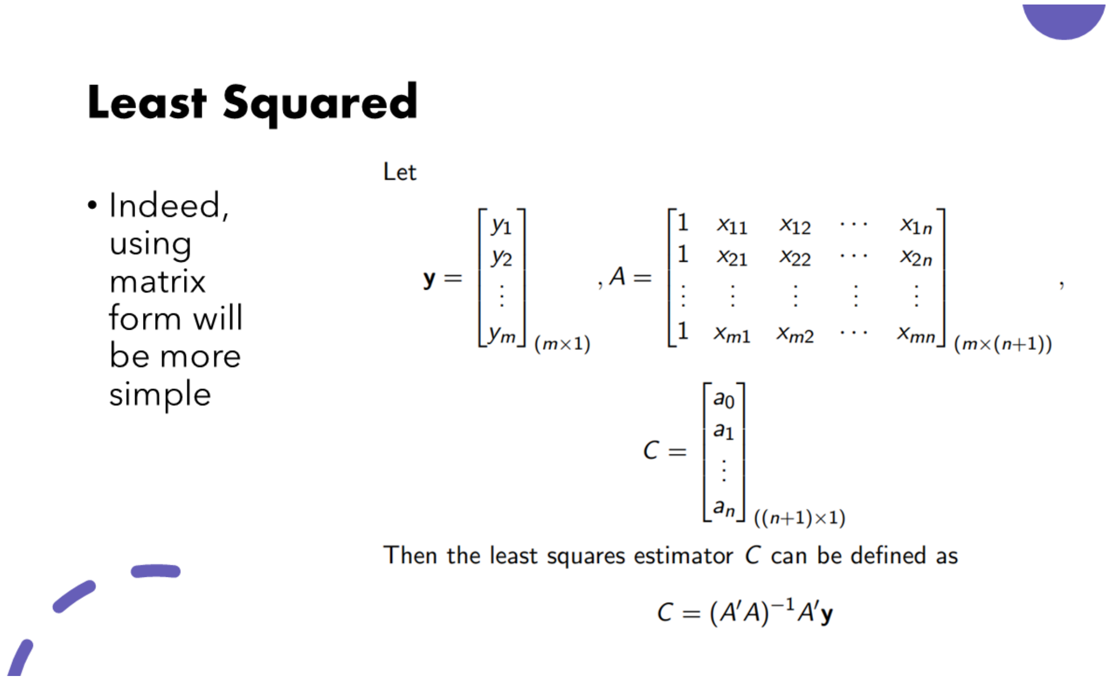
In [56]:

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

Out[56]:

	TV	radio	newspaper	sales
11	66.1	5.8	24.2	8.6
15	204.1	32.9	46.0	19.0
18	281.4	39.6	55.8	24.4
19	69.2	20.5	18.3	11.3
28	240.1	16.7	22.9	15.9

Langkah 3: Mentukan matriks A dan vektor y



Least Squared

- Indeed, using matrix form will be more simple

Let

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}_{(m\times 1)}, A = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{(m\times(n+1))},$$

$$C = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}_{((n+1)\times 1)}$$

Then the least squares estimator C can be defined as

$$C = (A'A)^{-1}A'\mathbf{y}$$

Menentukan matriks A, dengan x_1 TV dan x_2 newspaper

In [57]:

```
m = len(train.TV) #number of rows data

A = np.asanyarray(train[["TV","newspaper"]])
print(A[0:5,:]) #print sampai baris ke 5
```

```
[[230.1  69.2]
 [ 44.5  45.1]
 [ 17.2  69.3]
 [151.5  58.5]
 [180.8  58.4]]
```

Menambahkan vektor 1 pada kolom ke 0

In [58]:

```
A = np.insert(A,0,np.ones(m),1)
print (A[0:5,:]) #print sampai baris ke 5
```

```
[[  1.   230.1  69.2]
 [  1.    44.5  45.1]
 [  1.    17.2  69.3]
 [  1.   151.5  58.5]
 [  1.   180.8  58.4]]
```

Menentukan vektor y

In [59]:

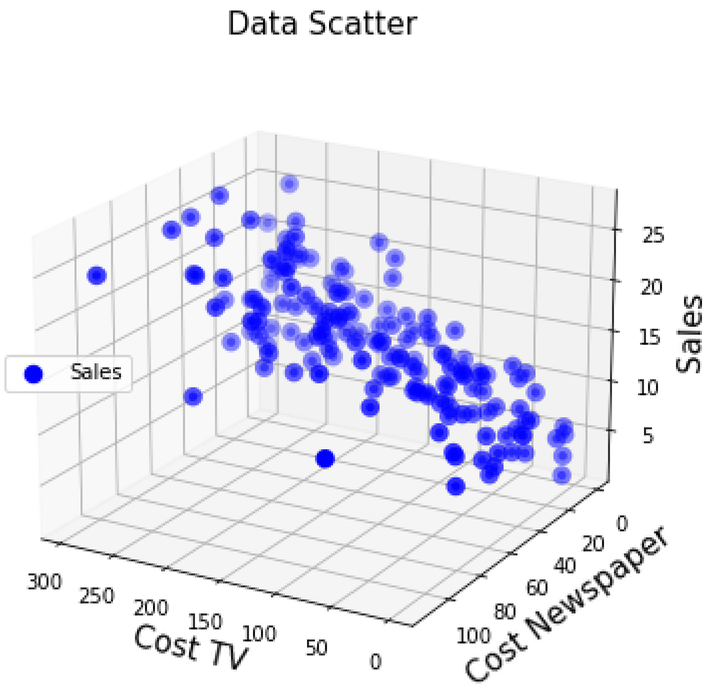
```
y = np.asanyarray(train[['sales']])
print(y[0:5])
```

```
[[22.1]
 [10.4]
 [ 9.3]
 [18.5]
 [12.9]]
```

Plot sebaran data yang akan dihampiri

In [60]:

```
fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(train.TV, train.newspaper, train.sales,  color='blue',lw=5)
plt.legend(('Sales',), loc='center left')
ax.set_xlabel('Cost TV',fontsize=15)
ax.set_ylabel('Cost Newspaper',fontsize=15)
ax.set_zlabel('Sales', fontsize=15)
ax.set_title('Data Scatter', fontsize=15)
plt.show()
```



Langkah 4: Mencari koefisien C

In [61]:

```
C = np.linalg.inv(np.transpose(A)@A)@np.transpose(A)@y
print(C)
```

```
[[5.58137741]
 [0.04511553]
 [0.05126001]]
```

Langkah 5: Menghitung hampiran $\hat{y}$ untuk data testing

Menentukan matriks A dan vektor y untuk data testing

In [62]:

```
m = len(test.TV)
A = np.asanyarray(test[['TV','newspaper']])
A = np.insert(A,0,np.ones(m),1)  #insert vector 1

y = np.asanyarray(test[['sales']]) #untuk evaluasi
```

Mencari nilai hampiran $\hat{y}$

In [63]:

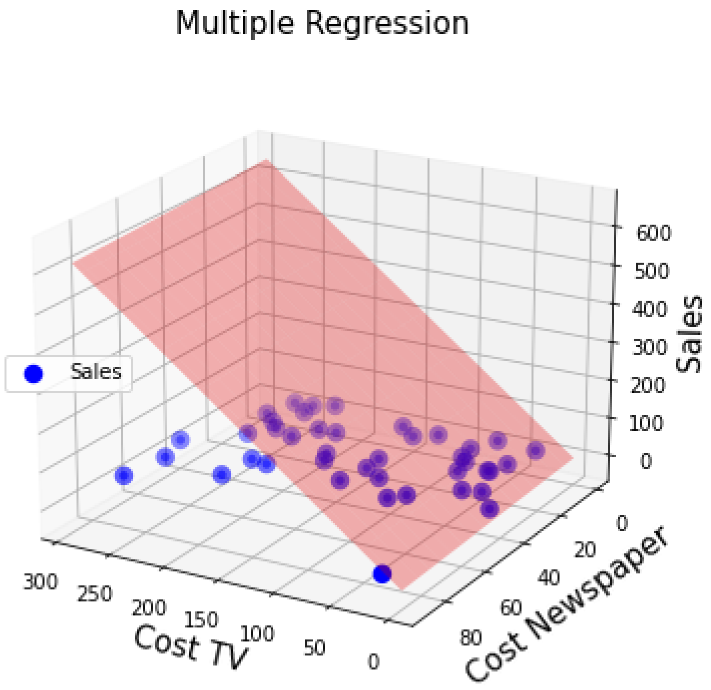
```
yhat = A@C
print(yhat[0:5])
```

```
[[ 9.80400625]
 [17.14741783]
 [21.13719654]
 [ 9.64143036]
 [17.58747084]]
```

Visualisasikan data testing dan daerah hampiran.

In [64]:

```python
def f(x1,x2):
  return c[0]+c[1]*x1+c[2]*x2
xp = np.linspace(0,max(test.TV),100)
yp = np.linspace(0,max(test.newspaper),100)

X,Y = np.meshgrid(xp,yp)
Z = f(X,Y)

fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(test.TV, test.newspaper, test.sales,  color='blue',lw=5)
ax.plot_surface(X, Y, Z, alpha=0.3, color='red', rstride=6, cstride=12)
plt.legend(('Sales',), loc='center left')
ax.set_xlabel('Cost TV',fontsize=15)
ax.set_ylabel('Cost Newspaper',fontsize=15)
ax.set_zlabel('Sales', fontsize=15)
ax.set_title('Multiple Regression', fontsize=15)
plt.show()
```



Langkah 6: Evaluasi model

1. Menentukan TABEL ANOVA membuat keputusan Hypothesis dan melihat MSE

In [65]:

```python
n=2
tabel = ANOVATAB(y,yhat,n,m)
tabel
```

Out[65]:

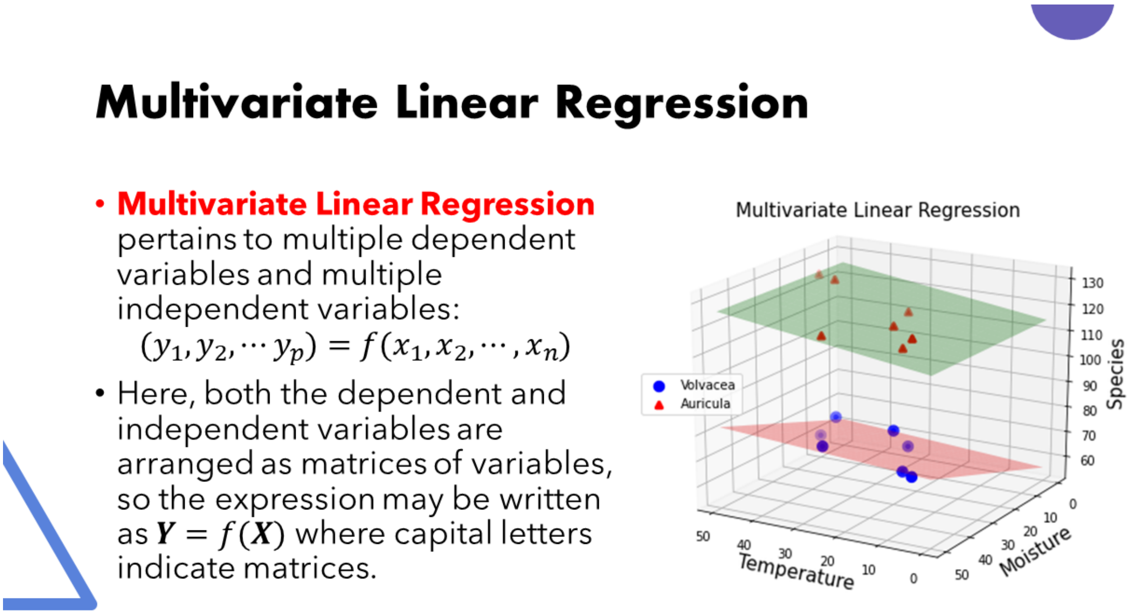
	SS	df	MS	Fs	pval
0	[744.8267629209878]	2	[372.4133814604939]	[36.39305053171656]	[1.4813853477235739e-09]
1	[388.8574408777728]	38	[10.233090549415072]	-	-
2	[1182.1756097560979]	40	-	-	-

1. Mengevaluasi R^2

```
In [66]:  R_s = 1 - tabel.SS[1]/tabel.SS[2]
          print('Nilai R^2 adalah', R_s)
```

```
Nilai R^2 adalah [0.67106626]
```

Multivariate Regression

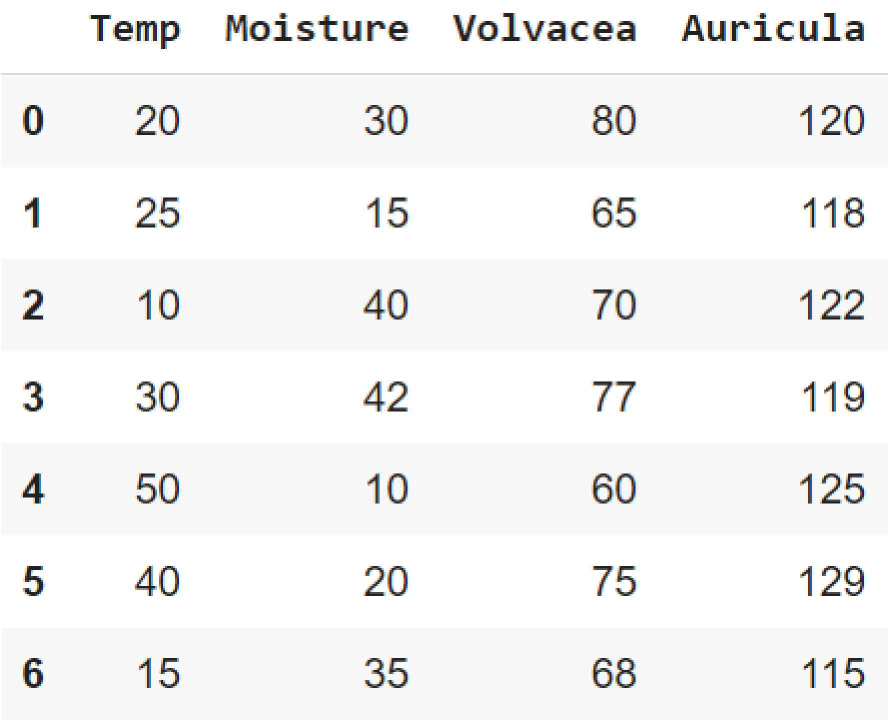


Multivariate Linear Regression

- **Multivariate Linear Regression** pertains to multiple dependent variables and multiple independent variables:
  $(y_1, y_2, \cdots y_p) = f(x_1, x_2, \cdots, x_n)$
- Here, both the dependent and independent variables are arranged as matrices of variables, so the expression may be written as $\boldsymbol{Y} = f(\boldsymbol{X})$ where capital letters indicate matrices.

Example

Seorang peneliti menemukan bahwa jumlah panen jenis jamur Volvacea dan Auricula bergantung pada beberapa faktor eksternal yaitu suhu, kelembaban dan curah hujan seperti tabel di bawah ini,



	Temp	Moisture	Volvacea	Auricula
0	20	30	80	120
1	25	15	65	118
2	10	40	70	122
3	30	42	77	119
4	50	10	60	125
5	40	20	75	129
6	15	35	68	115

Temukan fungsi estimasi jumlah panen jamur Volvacea dan Auricula menggunakan regresi linier multivariat

solusi

Langkah 1: menentukan dataframe untuk data pada tabel di atas

In [67]:

```
import pandas as pd

data_exercise= {
    'Temp': [20,25,10,30,50,40,15],
    'Moisture': [30,15,40,42,10,20,35],
    'Volvacea': [80,65,70,77,60,75,68],
    'Auricula': [120,118,122,119,125,129,115]
    }
Data = pd.DataFrame(data_exercise)

Data.head()
```

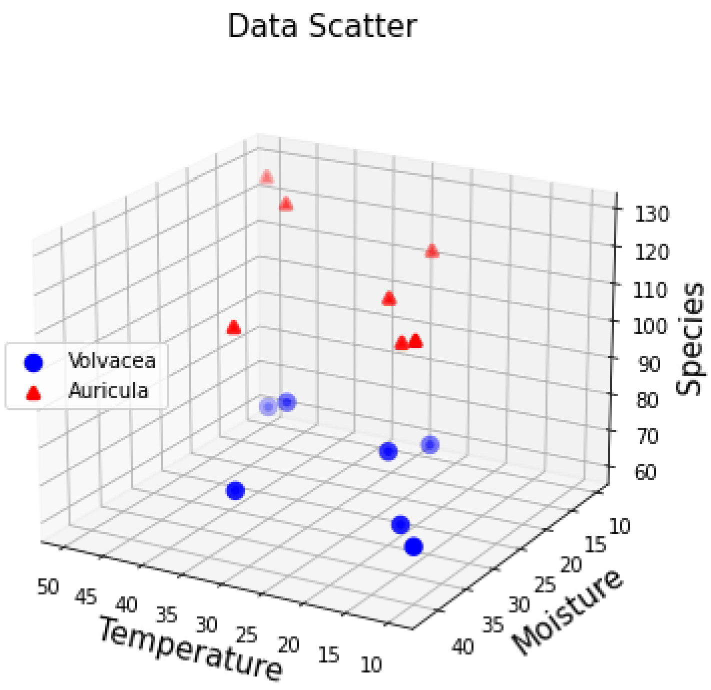
Out[67]:

	Temp	Moisture	Volvacea	Auricula
0	20	30	80	120
1	25	15	65	118
2	10	40	70	122
3	30	42	77	119
4	50	10	60	125

memvisualisasikan sebaran data

In [68]:

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(Data.Temp, Data.Moisture, Data.Volvacea,  color='blue',lw=5)
ax.scatter(Data.Temp, Data.Moisture, Data.Auricula, color='red', marker='^', linewid
plt.legend(('Volvacea','Auricula'), loc='center left')
ax.set_xlabel('Temperature',fontsize=15)
ax.set_ylabel('Moisture',fontsize=15)
ax.set_zlabel('Species', fontsize=15)
ax.set_title('Data Scatter', fontsize=15)
plt.show()
```



Langkah 2: Menentukan matriks $\mathbf{A}$ dan $\mathbf{Y}$

Matriks A

$$\mathbf{A} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{(m\times n+1)} \qquad ,$$

vektor y

$$\mathbf{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1p} \\ y_{21} & \cdots & y_{2p} \\ \vdots & \vdots & \vdots \\ y_{m1} & \cdots & y_{mp} \end{bmatrix}_{(m\times p)} \qquad .$$

In [69]:

```
m = len(Data.Temp)
A = np.asanyarray(Data[['Temp','Moisture']])
A = np.insert(A,0,np.ones(m),1) #insert vektors 1 at column 0
Y = np.asanyarray(Data[['Volvacea','Auricula']])
```

Langkah 3: Mencari koefisien C

Permasalahan Least Squared,

$$\min_{\mathbf{C}\in\mathbb{R}^{(n+1)\times p}} ||\mathbf{Y} - \hat{\mathbf{Y}}||^2,$$

atau

$$\min_{\mathbf{C}\in\mathbb{R}^{(n+1)\times p}} ||\mathbf{Y} - \mathbf{A}\mathbf{C}||^2.$$

dengan $||\cdot||$ menotasikan Frobenius norm.

solusinya adalah

$$\mathbf{C} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{Y}$$

In [70]:

```
C = np.linalg.inv((np.transpose(A)@A))@np.transpose(A)@Y
print(C)
```

```
[[5.62388851e+01 1.14711682e+02]
 [1.11467091e-01 2.24143503e-01]
 [4.17443005e-01 1.26612639e-02]]
```

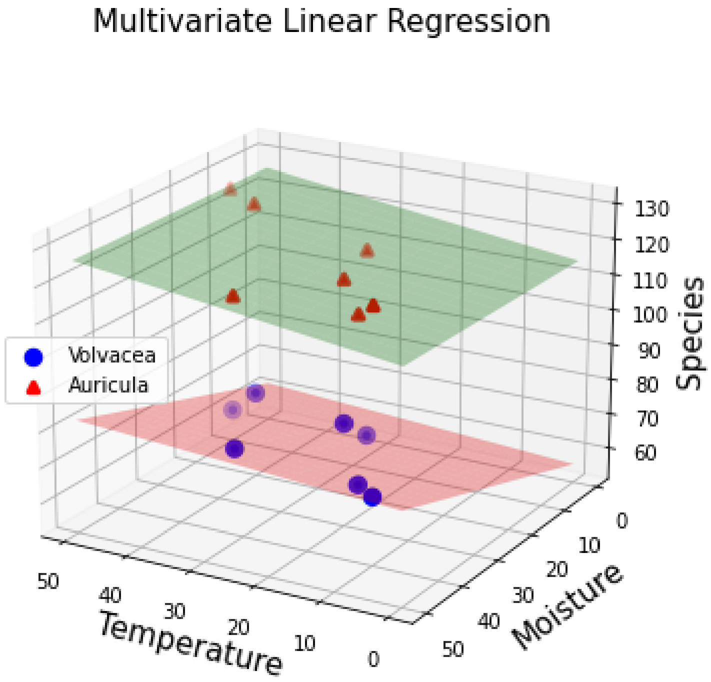
Memvisualisaikan data

In [71]:

```
def fVolvacea(x1,x2):
  return C[0][0] + C[1][0]*x1 + C[2][0]*x2

def fAuricula(x1,x2):
  return C[0][1] + C[1][1]*x1 + C[2][1]*x2
```

In [72]:

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

xp = np.linspace(0,50,100)
yp = np.linspace(0,50,100)

Xp,Yp = np.meshgrid(xp,yp)
Z1 = fVolvacea(Xp,Yp)
Z2 = fAuricula(Xp,Yp)


fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(Data.Temp, Data.Moisture, Data.Volvacea,  color='blue',lw=5)
ax.scatter(Data.Temp, Data.Moisture, Data.Auricula, color='red', marker='^', linewid
ax.plot_surface(Xp, Yp, Z1, alpha=0.3, color='red', rstride=6, cstride=12)
ax.plot_surface(Xp, Yp, Z2, alpha=0.3, color='green', rstride=6, cstride=12)
plt.legend(('Volvacea','Auricula'), loc='center left')
ax.set_xlabel('Temperature',fontsize=15)
ax.set_ylabel('Moisture',fontsize=15)
ax.set_zlabel('Species', fontsize=15)
ax.set_title('Multivariate Linear Regression', fontsize=15)
plt.show()
```



Langkah 4: Mengevaluasi data

Menentukan data hampiran, matriks $\hat{\mathbf{Y}}$, yaitu

$$\hat{\mathbf{Y}} = \mathbf{AC} + \mathbf{E}$$

dengan

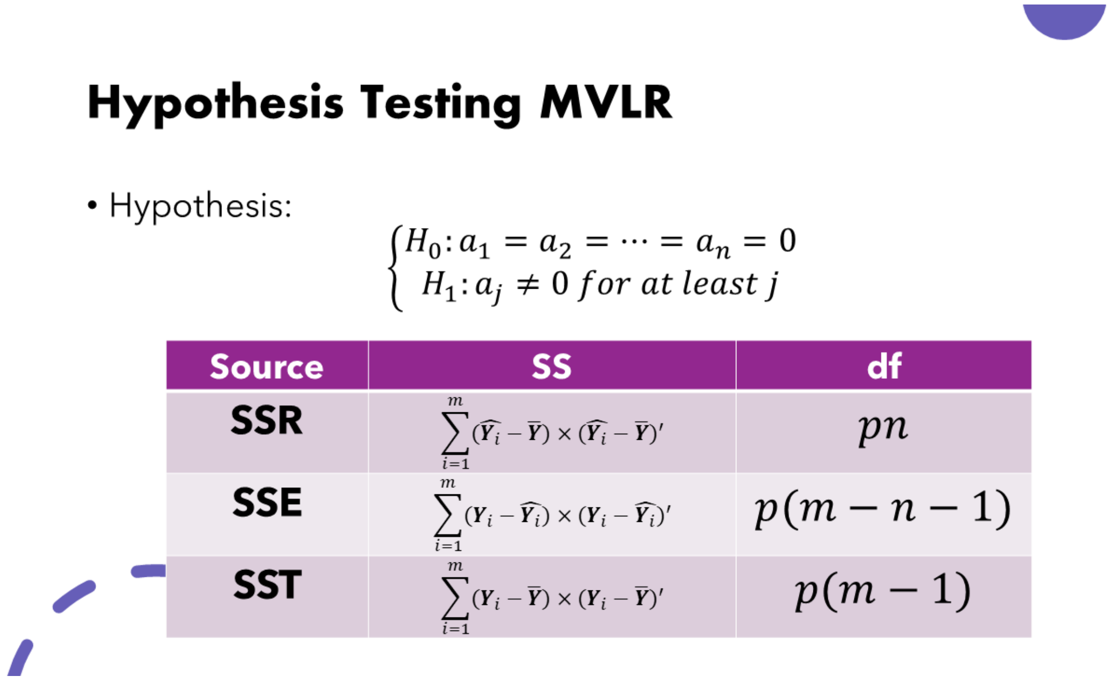
$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{y}_{11} & \cdots & \hat{y}_{1p} \\ \hat{y}_{21} & \cdots & \hat{y}_{2p} \\ \vdots & \vdots & \vdots \\ \hat{y}_{m1} & \cdots & \hat{y}_{mp} \end{bmatrix}_{(m \times p)}$$

In [73]:

```
Yhat = A@C
print(Yhat)
```

```
[[ 70.99151708 119.57438966]
 [ 65.28720745 120.50518821]
 [ 74.05127623 117.45956727]
 [ 77.11550405 121.96775985]
 [ 65.9866697  126.04546946]
 [ 69.04642884 123.93064708]
 [ 72.52139665 118.51697846]]
```

Hypothesis testing and Tabel ANOVA untuk multivariate linear regression



Hypothesis Testing MVLR

- Hypothesis:

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

Source	SS	df
SSR	$\sum_{i=1}^{m} (\widehat{Y}_i - \bar{Y}) \times (\widehat{Y}_i - \bar{Y})'$	pn
SSE	$\sum_{i=1}^{m} (Y_i - \widehat{Y}_i) \times (Y_i - \widehat{Y}_i)'$	$p(m-n-1)$
SST	$\sum_{i=1}^{m} (Y_i - \bar{Y}) \times (Y_i - \bar{Y})'$	$p(m-1)$

Membuat fungsi TABEL ANOVA MVLR

In [74]:

```
import numpy as np
import pandas as pd
import scipy
from scipy import stats

def ANOVATAB_MVLR(y,yhat,n,p,m):
  dfn = p*n
  dfd = p*(m-n-1)
  ybar = np.average(y)

  SSR = np.sum((yhat-ybar)@np.transpose((yhat-ybar)))
  SSE = np.sum((y-yhat)@np.transpose((y-yhat)))
  SST = np.sum((y-ybar)@np.transpose((y-ybar)))
  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)
```

1. Memanggil tabel ANOVA untuk kesimpulan dan melihat MSE

In [75]:

```
n=2
p=2
tabel = ANOVATAB_MVLR(Y,Yhat,n,p,m)
tabel
```

Out[75]:

	SS	df	MS	Fs	pval

	SS	df	MS	Fs	pval
0	6.230450e+04	4	15576.125	-438428942161726 5152.0	1.0
1	-2.842171e-14	8	-0.0	-	-
2	6.230450e+04	6	-	-	-

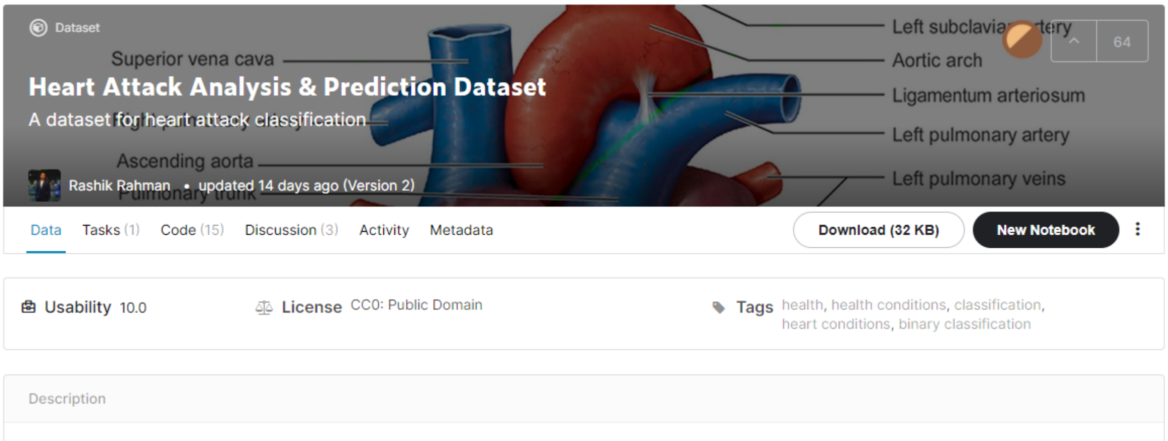
1. mengevaluasi R^2

In [76]:
```
R_s = 1 - tabel.SS[1]/tabel.SS[2]
print('Nilai R^2 adalah', R_s)
```

```
Nilai R^2 adalah 1.0
```

Machine Learning Approach

Pada contoh kali ini kita akan menggunakan data Heart Attack Analysis yang diambil dari situs kaggle.com



Dataset

Heart Attack Analysis & Prediction Dataset

A dataset for heart attack classification

Rashik Rahman • updated 14 days ago (Version 2)

Usability 10.0

License CC0: Public Domain

Tags health, health conditions, classification, heart conditions, binary classification

Description

Langkah 1: Menentukan data

Pertama kita download terlebih dahulu data Heart Attack Analysis & Prediction Dataset dan upload ke folder Files di Colab (menu sebelah kiri)

https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset?select=heart.csv

Atau dengan menggunakan link dibawah ini

In [77]:
```
import pandas as pd

url ='https://bit.ly/3fsXQaF'
data = pd.read_csv(url)

data
```

Out[77]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

Mencoba memprediksi (y_1, y_2):

1. maximum heart rate achieved (thalachh)
2. Previous Peak (oldpeak)

dengan predictors (x_1, x_2):

1. resting blood pressure (trtbps)
2. cholestoral (chol)

In [78]:

```
DataNew = data[['trtbps','chol','thalachh','oldpeak']]
DataNew
```

Out[78]:

	trtbps	chol	thalachh	oldpeak
0	145	233	150	2.3
1	130	250	187	3.5
2	130	204	172	1.4
3	120	236	178	0.8
4	120	354	163	0.6
...	...	...	...	...
298	140	241	123	0.2
299	110	264	132	1.2
300	144	193	141	3.4
301	130	131	115	1.2
302	130	236	174	0.0

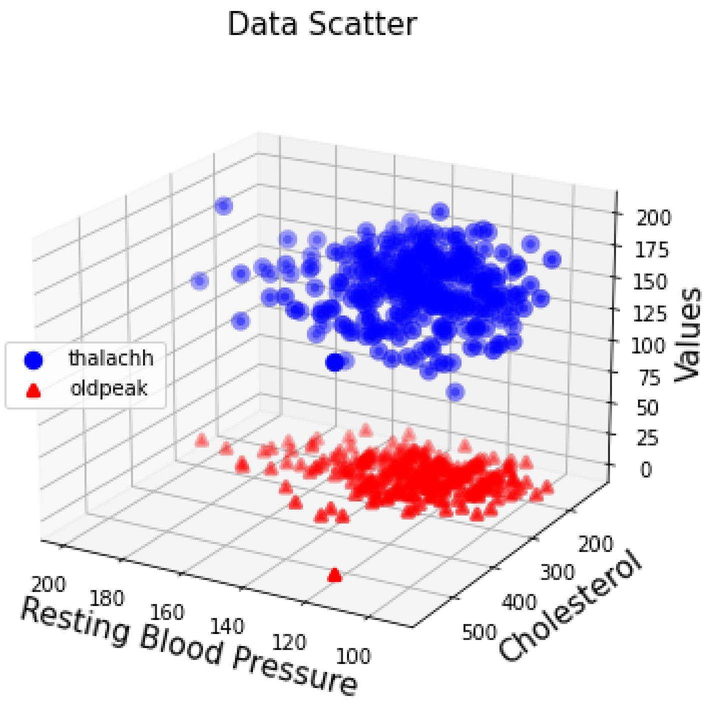
303 rows × 4 columns

memvisualisasikan data

In [79]:

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
```

```
fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(DataNew.trtbps, DataNew.chol, DataNew.thalachh,  color='blue',lw=5)
ax.scatter(DataNew.trtbps, DataNew.chol, DataNew.oldpeak, color='red', marker='^', l
plt.legend(('thalachh','oldpeak'), loc='center left')
ax.set_xlabel('Resting Blood Pressure',fontsize=15)
ax.set_ylabel('Cholesterol',fontsize=15)
ax.set_zlabel('Values', fontsize=15)
ax.set_title('Data Scatter', fontsize=15)
plt.show()
```



Langkah 2: Menentukan data training dan testing

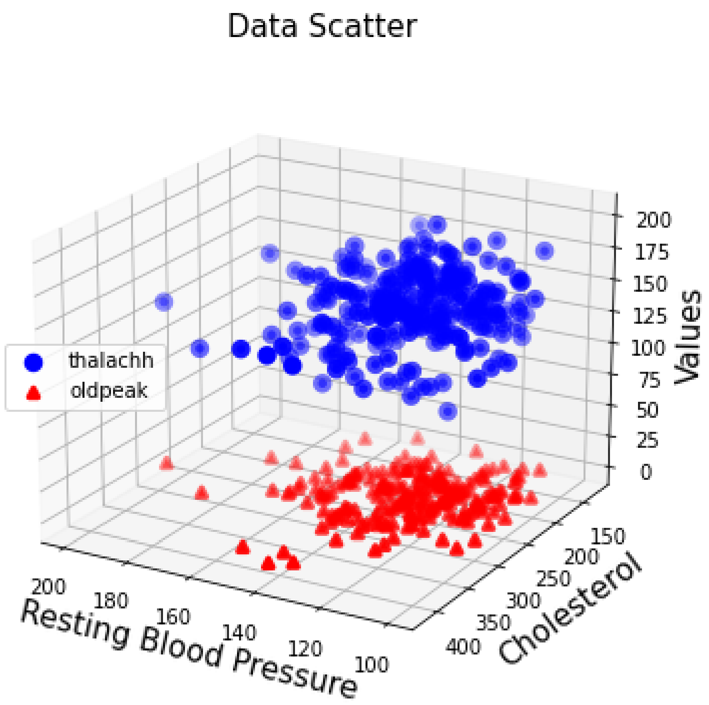
In [80]:

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

memvisualisasikan data training

In [81]:

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(train.trtbps, train.chol, train.thalachh,  color='blue',lw=5)
ax.scatter(train.trtbps, train.chol, train.oldpeak, color='red', marker='^', linewid
plt.legend(('thalachh','oldpeak'), loc='center left')
ax.set_xlabel('Resting Blood Pressure',fontsize=15)
ax.set_ylabel('Cholesterol',fontsize=15)
ax.set_zlabel('Values', fontsize=15)
ax.set_title('Data Scatter', fontsize=15)
plt.show()
```



Langkah 3: Menentukan Matriks $\mathbf{A}$ dan Matriks $\mathbf{Y}$

In [82]:

```
m = len(train.trtbps)
A = np.asanyarray(train[['trtbps','chol']])
A = np.insert(A,0,np.ones(m),1) #insert vector 1 in column 0
Y = np.asanyarray(train[['thalachh','oldpeak']])
```

Langkah 4: Menentukan Koefisien $\mathbf{C}$

In [83]:

```
C = np.linalg.inv((np.transpose(A)@A))@np.transpose(A)@Y
print(C)
```

```
[[ 1.71307273e+02 -1.20038351e+00]
 [-1.69976459e-01  1.83519079e-02]
 [-1.20580503e-03 -6.57916284e-04]]
```

Langkah 5: Menghitung hampiran data, matriks $\hat{\mathbf{Y}}$

Membuat matriks $\mathbf{A}$ dan $\mathbf{Y}$ dari data testing

In [84]:

```
m = len(test.trtbps)
print(m)
A = np.asanyarray(test[['trtbps','chol']])
A = np.insert(A, 0, np.ones(m), 1)
Y = np.asanyarray(test[['thalachh','oldpeak']])
```

```
71
```

memvisualisasikan daerah hampiran

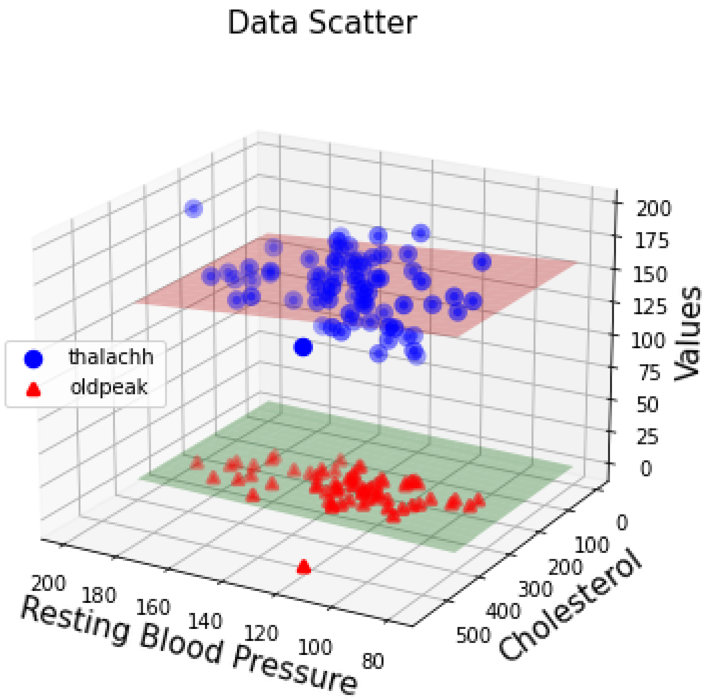
In [85]:

```
def fthalachh(x1,x2):
  return C[0][0] + C[1][0]*x1 + C[2][0]*x2

def foldpeak(x1,x2):
  return C[0][1] + C[1][1]*x1 + C[2][1]*x2
```

In [86]:

```
from mpl_toolkits.mplot3d import Axes3D
```

```
import matplotlib.pyplot as plt

xp = np.linspace(80,200,100)
yp = np.linspace(0,400,100)

Xp,Yp = np.meshgrid(xp,yp)
Z1 = fthalachh(Xp,Yp)
Z2 = foldpeak(Xp,Yp)

fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(test.trtbps, test.chol, test.thalachh,  color='blue',lw=5)
ax.scatter(test.trtbps, test.chol, test.oldpeak, color='red', marker='^', linewidth=
ax.plot_surface(Xp, Yp, Z1, alpha=0.3, color='red', rstride=6, cstride=12)
ax.plot_surface(Xp, Yp, Z2, alpha=0.3, color='green', rstride=6, cstride=12)
plt.legend(('thalachh','oldpeak'), loc='center left')
ax.set_xlabel('Resting Blood Pressure',fontsize=15)
ax.set_ylabel('Cholesterol',fontsize=15)
ax.set_zlabel('Values', fontsize=15)
ax.set_title('Data Scatter', fontsize=15)
plt.show()
```



Menghitung $\hat{\mathbf{Y}}$

In [87]:

```
Yhat = A@C
print(Yhat[:,0])
```

```
[148.964349   147.15606196 141.83136676 145.6082288  148.88958908
 147.2380567  143.74688634 148.94626192 146.95718823 143.67694965
 148.89200069 151.02577099 147.25614377 154.04193826 155.05576799
 147.19585352 148.05899967 147.24408572 151.07990606 152.66886274
 140.72589585 150.65567295 143.82888109 140.31962366 150.67737744
 147.52374839 155.08953053 150.58573625 150.69546451 146.15549609
 147.58524445 150.33863032 148.45799498 147.18741289 148.90164713
 145.53949792 151.92022602 149.90944784 148.84256269 149.28983225
 150.68340646 146.32064932 142.01818244 149.75394104 140.38111971
 148.45920079 148.81241756 148.05417645 147.51048453 142.06400303
 149.69365079 149.55020204 141.43112359 149.8998014  138.33054996
 148.57254646 149.23798264 145.51658762 152.62545376 154.0274686
 152.27826601 147.80582267 147.26096699 144.85115145 152.20591771
 145.90597855 151.54651056 142.13996874 145.21522047 147.28508309
 148.92576323]
```

Langkah 6: Mengevaluasi model

1. Menentukan tabel ANOVA

In [88]:

```
n=2
p=2
tabel = ANOVATAB_MVLR(Y,Yhat,n,p,m)
tabel
```

Out[88]:

	SS	df	MS	Fs	pval
0	5.431316e+07	4	13578290.320104	19734.487162	0.0
1	9.357464e+04	136	688.048806	-	-
2	5.754499e+07	70	-	-	-

1. Menetukan R^2

In [89]:

```
R_s = 1 - tabel.SS[1]/tabel.SS[2]
print('Nilai R^2 adalah', R_s)
```

```
Nilai R^2 adalah 0.9983738873827012
```

Homework

1. Buat Program Multiple Linear Regression yang menggunakan dua data cost TV, radio untuk memprediksi sales. (plot scatter dan bidang regresi)
2. Buat Program Multiple Linear Regression yang menggunakan tiga data cost TV, radio, newspaper untuk memprediksi sales.
3. Diberikan data Air Quality, pada link ini url = 'https://bit.ly/31xnBhR', hampiri 'T', 'RH' menggunakan kolom 'CO(GT)', 'C6H6(GT)'. Gunakan Multivariate analysis.

dengan

- T: Temperature in Â°C
- RH: Relative Humidity (%)
- CO(GT): True hourly averaged concentration CO in mg/m^3 (reference analyzer)
- C6H6(GT): True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)

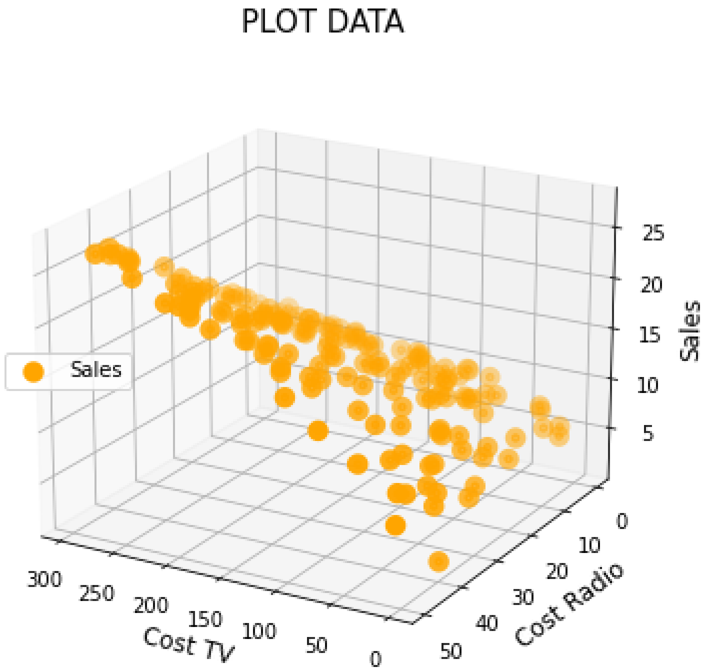
HINT: Gunakan syntax Python berikut untuk mengubah nama kolom pada dataFrame

DataNew = DataNew.rename(columns={'CO(GT)': 'A1', 'C6H6(GT)': 'A2', 'T':'Y1','RH':'Y2'})

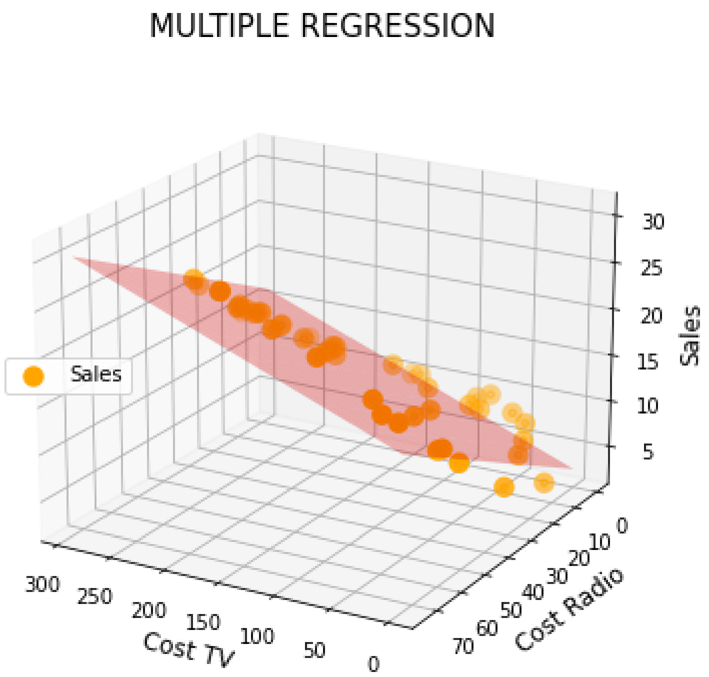
1. Buat Program Multiple Linear Regression yang menggunakan dua data cost TV, radio untuk memprediksi sales. (plot scatter dan bidang regresi)

In [98]:

```
import pandas as pd
import numpy as np
from matplotlib import cm
import matplotlib as mpl
import scipy
from scipy import stats
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

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

#Menentukan pembagian data training dan testing
msk = np.random.rand(len(data)) < 0.8
train = data[msk]
test = data[~msk]

#Menentukan matriks A dan vektor y
m = len(train.sales)
A = np.asanyarray(train[['TV', 'radio']])
A = np.insert(A, 0, np.ones(m), 1) #Menambahkan vektor 1 pada kolom 0
y = np.asanyarray(train[['sales']])

#Membuat plot pesebaran data
fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(train.TV, train.radio, train.sales,  color='orange',lw=6)
plt.legend(('Sales',), loc='center left')
ax.set_xlabel('Cost TV',fontsize=12)
ax.set_ylabel('Cost Radio',fontsize=12)
ax.set_zlabel('Sales', fontsize=12)
ax.set_title('PLOT DATA', fontsize=15)
plt.show()

#Melakukan pencarian koefisien C
C = (np.linalg.inv(np.transpose(A)@A))@np.transpose(A)@y
print('Koefisien : ', C)

#Menghitung hampiran y hat untuk data testing
m = len(test.sales)
A = np.asanyarray(test[['TV','radio']])
A = np.insert(A, 0, np.ones(m), 1)  #insert vector 1
y = np.asanyarray(test[['sales']]) #untuk evaluasi
yhat = A@C

#Membuat visualisasi data
def f(x1,x2):
  return C[0] + C[1]*x1 + C[2]*x2

xp = np.linspace(0,max(test.TV),100)
yp = np.linspace(0,max(test.newspaper),100)

X,Y = np.meshgrid(xp,yp)
Z = f(X,Y)

fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(test.TV, test.radio, test.sales,  color='orange',lw=6)
ax.plot_surface(X, Y, Z, alpha=0.3, color='red', rstride=6, cstride=12)
plt.legend(('Sales',), loc='center left')
ax.set_xlabel('Cost TV',fontsize=12)
ax.set_ylabel('Cost Radio',fontsize=12)
ax.set_zlabel('Sales', fontsize=12)
ax.set_title('MULTIPLE REGRESSION', fontsize=15)
plt.show()

#Pembuatan evaluasi model hampiran
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=2

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



```
Koefisien :  [[2.84168389]
 [0.04578054]
 [0.18994256]]
```



Out[98]:

	SS	df	MS	Fs	pval
0	[1196.2656845272527]	2	[598.1328422636263]	[280.67060033857405]	[1.1102230246251565e-16]
1	[89.50556041412263]	42	[2.1310847717648245]	-	-
2	[1244.4720000000002]	44	-	-	-

In [99]:

```
R_s = 1 - tabel.SS[1]/tabel.SS[2]
print('Nilai R^2 yaitu:', R_s)
```

```
Nilai R^2 yaitu: [0.92807748]
```

Kesimpulan : Dipilih nilai $\alpha = 0.05$.

- Untuk p-value < α, maka H_0 akan ditolak.
- Nilai R^2 mendekati 1 yang menandakan bahwa model cukup bagus

2. Buat Program Multiple Linear Regression yang menggunakan tiga data cost TV, radio, newspaper untuk memprediksi sales.

In [102…

```
import pandas as pd
import numpy as np
import scipy
from scipy import stats

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

#Menentukan pembagian data training dan testing
msk = np.random.rand(len(data)) < 0.8
train = data[msk]
test = data[~msk]

#Menentukan matriks A dan vektor y
m = len(train.sales) #number of rows data
```

```
A = np.asanyarray(train[['TV', 'radio', 'newspaper']])
A = np.insert(A, 0, np.ones(m), 1)
y = np.asanyarray(train[['sales']])

#Mencari nilai koefisien C
C = (np.linalg.inv(np.transpose(A)@A))@np.transpose(A)@y
print('Koefisien : ', C)

#Menghitung hampiran yhat untuk data testing
m = len(test.sales)
A = np.asanyarray(test[['TV','radio', 'newspaper']])
A = np.insert(A, 0, np.ones(m), 1)  #insert vector 1
y = np.asanyarray(test[['sales']]) #untuk evaluasi
yhat = A@C

#Mendefinisikan model
def f(x1,x2,x3):
  return C[0] + C[1]*x1 + C[2]*x2 + C[3]*x3

#Mengevaluasi model hampiran
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)

R_s = 1 - tabel.SS[1]/tabel.SS[2]
print('Nilai R^2 adalah', R_s)

n=3
tabel = ANOVATAB(y,yhat,n,m)
tabel
```

```
Koefisien :  [[2.88118847e+00]
 [4.51053506e-02]
 [1.89952519e-01]
 [2.11330402e-03]]
Nilai R^2 adalah [0.92807748]
```

Out[102…

	SS	df	MS	Fs	pval
0	[1183.6577680000526]	3	[394.55258933335085]	[176.95477659178846]	[1.1102230246251565e-16]
1	[86.95753389860593]	39	[2.229680356374511]	-	-
2	[1305.6176744186046]	42	-	-	-

Kesimpulan : Dipilih nilai $\alpha = 0.05$.

- Jika nilai p-value < α, maka H_0 ditolak.
- Nilai R^2 mendekati 1 yang menandakan bahwa model cukup bagus

3. Diberikan data Air Quality, pada link ini url = 'https://bit.ly/31xnBhR', hampiri 'T', 'RH' menggunakan kolom 'CO(GT)', 'C6H6(GT)'. Gunakan Multivariate analysis.

dengan

- T: Temperature in Â°C
- RH: Relative Humidity (%)
- CO(GT): True hourly averaged concentration CO in mg/m^3 (reference analyzer)
- C6H6(GT): True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)

HINT: Gunakan syntax Python berikut untuk mengubah nama kolom pada dataFrame

DataNew = DataNew.rename(columns={'CO(GT)': 'A1', 'C6H6(GT)': 'A2', 'T':'Y1','RH':'Y2'})

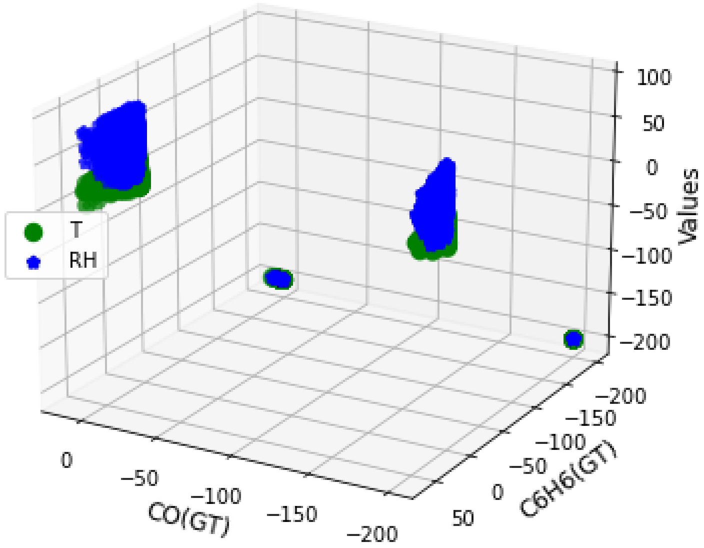
In [103…

```
import pandas as pd
import numpy as np
import scipy
from scipy import stats
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

#Menyiapkan data
url = 'https://bit.ly/31xnBhR'
data = pd.read_csv(url, index_col=0)
DataNew = data[['CO(GT)', 'C6H6(GT)', 'T','RH']]
DataNew = DataNew.rename(columns={'CO(GT)': 'A1', 'C6H6(GT)': 'A2', 'T':'Y1',
                                    'RH':'Y2'})

#Plot pesebaran data
fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(DataNew.A1, DataNew.A2, DataNew.Y1,  color='green',lw=5)
ax.scatter(DataNew.A1, DataNew.A2, DataNew.Y2, color='blue', marker='*',
           linewidth=3)
plt.legend(('T','RH'), loc='center left')
ax.set_xlabel('CO(GT)',fontsize=12)
ax.set_ylabel('C6H6(GT)',fontsize=12)
ax.set_zlabel('Values', fontsize=12)
ax.set_title('PLOT DATA', fontsize=15)
plt.show()

#Menentukan pembagian data training dan testing
msk = np.random.rand(len(DataNew)) < 0.8
train = DataNew[msk]
test = DataNew[~msk]

#Menentukan matriks A dan matriks Y
m = len(train.A1)
A = np.asanyarray(train[['A1','A2']])
A = np.insert(A, 0, np.ones(m), 1) #insert vector 1 in column 0
Y = np.asanyarray(train[['Y1','Y2']])

#Menentukan koefisien C
C = (np.linalg.inv(np.transpose(A)@A))@np.transpose(A)@Y
```

```
print('Koefisien : ', C)

#Menghitung hampiran yhat untuk data testing
m = len(test.A1)
A = np.asanyarray(test[['A1','A2']])
A = np.insert(A, 0, np.ones(m), 1)
Y = np.asanyarray(test[['Y1','Y2']])
Yhat = A@C
print('Yhat :')
print(Yhat[:,0])

#Melakukan visualisasi daerah hampiran
def fY1(x1,x2):
  return C[0][0] + C[1][0]*x1 + C[2][0]*x2

def fY2(x1,x2):
  return C[0][1] + C[1][1]*x1 + C[2][1]*x2

xp = np.linspace(80,200,100)
yp = np.linspace(0,400,100)

Xp,Yp = np.meshgrid(xp,yp)
Z1 = fY1(Xp,Yp)
Z2 = fY2(Xp,Yp)

#Membuat plot data
fig = plt.figure(figsize=(7,6))
ax = fig.add_subplot(111, projection='3d')
ax.view_init(20, 120)
ax.scatter(test.A1, test.A2, test.Y1, color='red',lw=5)
ax.scatter(test.A1, test.A2, test.Y2, color='orange', marker='*', linewidth=3)
ax.plot_surface(Xp, Yp, Z1, alpha=0.3, color='orange', rstride=6, cstride=12)
ax.plot_surface(Xp, Yp, Z2, alpha=0.3, color='purple', rstride=6, cstride=12)
plt.legend(('T','RH'), loc='center left')
ax.set_xlabel('CO(GT)',fontsize=12)
ax.set_ylabel('C6H6(GT)',fontsize=12)
ax.set_zlabel('Values', fontsize=12)
ax.set_title('MULTIVARIATE LINEAR REGRESSION', fontsize=15)
plt.show()

#Mengevaluasi model regresi
def ANOVATAB_MVLR(y,yhat,n,p,m):
  dfn = p*n
  dfd = p*(m-n-1)
  ybar = np.average(y)

  SSR = np.sum((yhat-ybar)@np.transpose(yhat-ybar))
  SSE = np.sum((y-yhat)@np.transpose(y-yhat))
  SST = np.sum((y-ybar)@np.transpose(y-ybar))
  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)
```

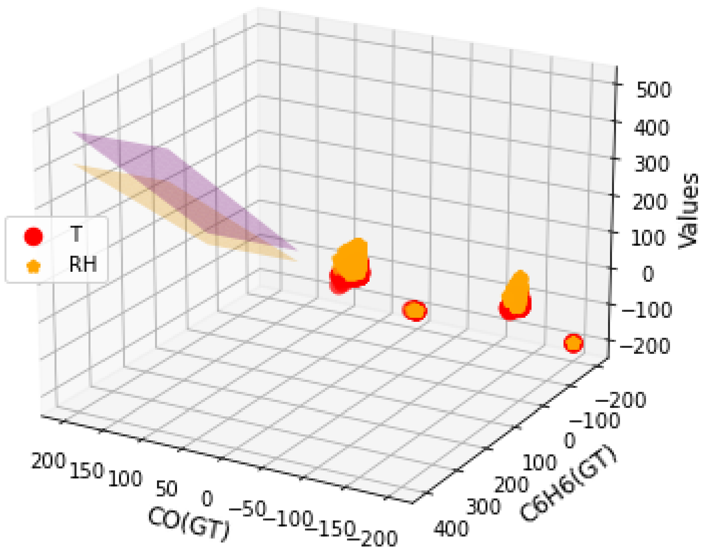
```
R_s = 1 - tabel.SS[1]/tabel.SS[2]
print('Nilai R^2 yaitu: ', R_s)

n=2
p=2
tabel = ANOVATAB_MVLR(Y,Yhat,n,p,m)
tabel
```

PLOT DATA



```
Koefisien :  [[ 7.21891106e+00  3.69667668e+01]
 [-2.14169963e-02 -1.36629681e-02]
 [ 1.01540808e+00  1.14679991e+00]]
Yhat :
[16.31046638 11.96562864 10.84867975 ... 11.462208    8.73131468
 12.31463678]
```

MULTIVARIATE LINEAR REGRESSION



```
Nilai R^2 yaitu:  [0.9333974]
```

Out[103…

	SS	df	MS	Fs	pval
0	1.546540e+09	4	386635029.234073	1320553.903014	0.0

	SS	df	MS	Fs	pval
1	1.083881e+06	3702	292.782467	-	-
2	1.534741e+09	1853	-	-	-

Kesimpulan : Dipilih nilai $\alpha = 0.05$.

- Jika nilai p-value < α, maka H_0 ditolak.
- Nilai R^2 mendekati 1 yang menandakan bahwa model cukup bagus