Linear Regresion – Pemrograman Simulasi

Fadly Faturrohman (152017076)

Link Github : https://github.com/FadlyFatur/Linear-regresi-manual-python

File **.ipynb** (dibukan menggukan jupyter notebook/ google collaborator)

Screenshot Program

Proses input Menggunakan file csv	In [6]: ``` data = pd.read_csv("data2.csv", sep=';') data ``` Out[6]: Medical_expenses Age 0 100 15 1 135 20 2 135 25 3 150 37 4 250 40 5 270 45 6 290 48 7 360 50 8 375 55 9 400 61 10 500 64 11 1000 67 12 1500 70
Import lib Numpy → untuk menghitung perhitungan sigma, square, square root Matplotlib → membuat bagan Pandas → import file csv	In [5]: ``` # Fadly Faturrohman - 152017076 # Pemrograman Simulasi import numpy as np import matplotlib.pyplot as plt import pandas as pd ```
Mengubah data menjadi bentuk array x dan array y	In [9]: ``` #x = data.as_matrix(columns=data.columns[1:]) x = data["Medical_expenses"].to_numpy() y = data["Age"].to_numpy() n = np.size(x) print(x) print(y) ``` [100 135 135 150 250 270 290 360 375 400 500 1000 1500] [15 20 25 37 40 45 48 50 55 61 64 67 70]

Melakukan perhitungan untuk :
(Σy) (Σx²) – (Σx) (Σxy)

In [10]:
```
# (Σy) (Σx²) - (Σx) (Σxy)

sum_x, sum_y = np.sum(x), np.sum(y)
print('(Σx) = ',sum_x, '\n(Σy) = ',sum_y)

pgkt_x= np.sum(np.square(x))
pgkt_y= np.sum(np.square(y))
print('(Σx²) = ', pgkt_x,'\n(Σy²) = ',pgkt_y)

sum_xy = np.sum(np.multiply(x,y))
print('(Σxy) = ', sum_xy)

print('(Σy)(Σx²) = ',sum_y*pgkt_x)
print('(Σx)(Σxy) = ',sum_x*sum_xy)
print('(Σy)(Σx²) - (Σx)(Σxy) = ',sum_y*pgkt_x-sum_x*sum_xy)
```

```
(Σx) =  5465
(Σy) =  597
(Σx²) =  4218675
(Σy²) =  31279
(Σxy) =  316220
(Σy)(Σx²) =  2518548975
(Σx)(Σxy) =  1728142300
(Σy)(Σx²) - (Σx)(Σxy) =  790406675
```

Melakukan perhitungan untuk :
n(Σx²) – (Σx)²

In [11]:
```
# n(Σx²) - (Σx)²

sum_nX2 = n*pgkt_x
print('n(Σx²) = ', sum_nX2)

sum_x2 = np.square(sum_x)
print('(Σx)² = ', sum_x2)
print('n(Σx²)-(Σx)² = ',sum_nX2-sum_x2)
```

```
n(Σx²) =  54842775
(Σx)² =  29866225
n(Σx²)-(Σx)² =  24976550
```

Menghitung nilai a

In [12]:
```
# a = (Σy) (Σx²) - (Σx) (Σxy) / n(Σx²) - (Σx)²

x1 = sum_y*pgkt_x
x2 = sum_x*sum_xy
x3 = x1-x2
x4 = sum_nX2-sum_x2

a_2 = (sum_y*pgkt_x - sum_x*sum_xy) / (sum_nX2-sum_x2)
# a = (x3/x4)
print('a = (Σy) (Σx²) - (Σx) (Σxy) / n(Σx²) - (Σx)²')
print('a = ', a_2)
```

```
a = (Σy) (Σx²) - (Σx) (Σxy) / n(Σx²) - (Σx)²
a =  31.645950901946026
```

Melakukan perhitungan untuk :
n(Σxy) – (Σx) (Σy)

In [13]:
```
# n(Σxy) - (Σx) (Σy)

b_x1 = n*sum_xy
b_x2 = sum_x*sum_y

print('n(Σxy) = ', b_x1, '\n(Σx)(Σy) = ',b_x2)
print('n(Σxy) - (Σx) (Σy) = ',b_x1-b_x2)
```

```
n(Σxy) =  4110860
(Σx)(Σy) =  3262605
n(Σxy) - (Σx) (Σy) =  848255
```

Melakukan perhitungan untuk : $n(\Sigma x) - (\Sigma x)^2$	In [14]: ``` # n(Σx) - (Σx)² b_x3 = n*sum_x b_x4 = np.square(sum_x) b_x5 = n*pgkt_x print('n(Σx) = ',b_x3,'\n(Σx)² = ', b_x4) print('n(Σx) - (Σx)² = ', b_x3-b_x4) ``` ``` n(Σx) = 71045 (Σx)² = 29866225 n(Σx) - (Σx)² = -29795180 ```
Menhitung nilai b	In [15]: ``` # b = n(Σxy) - (Σx) (Σy) / n(Σx) - (Σx)² b = (b_x1 - b_x2)/(b_x3 - b_x4) print('b = ',b) print('Model Regresi : Y = a + b * x = ',a_2,'+',b,'*x') ``` ``` b = -0.02846953769032441 Model Regresi : Y = a + b * x = 31.645950901946026 + -0.02846953769032441 *x ```
Menghitung nilai r	In [29]: ``` # r = nΣXiYi - (Σxi)(Σyi) /√(nΣXi^2-(ΣXi)^2)(nΣYi^2 - (Σyi)^2) # x1 x2 y1 y2 y3 y4 x1 = b_x1 x2 = b_x2 y1 = sum_nX2 y2 = sum_x2 y3 = n*pgkt_y y4 = np.square(sum_y) x_tot = x1-x2 print('nΣXiYi-(Σxi)(Σyi) = ',x_tot) print('(nΣXi^2-(ΣXi)^2) = ',y1-y2) print('(nΣYi^2-(Σyi)^2) = ',y3-y4) y_tot = (y1-y2)*(y3-y4) print('(nΣXi^2-(ΣXi)^2)(nΣYi^2 - (Σyi)^2) = ',y_tot) y_tot2 = np.sqrt(y_tot) print('√(nΣXi^2-(ΣXi)^2)(nΣYi^2 - (Σyi)^2) = ',y_tot2) r = x_tot/y_tot2 print('r = ',r) ``` ``` nΣXiYi-(Σxi)(Σyi) = 848255 (nΣXi^2-(ΣXi)^2) = 24976550 (nΣYi^2-(Σyi)^2) = 50218 (nΣXi^2-(ΣXi)^2)(nΣYi^2 - (Σyi)^2) = 1254272387900 √(nΣXi^2-(ΣXi)^2)(nΣYi^2 - (Σyi)^2) = 1119943.0288635222 r = 0.7574090629063325 ```
Menghitung nilai koefisien determinasi	In [27]: ``` # koefisien determinasi koef = np.square(r)*1 print(koef) ``` ``` 0.5736684885726487 ```
Menghitung 100% - koefisien determinasi	In [28]: ``` # 100% - Koefisien determinasion hasil = 1-koef print(hasil) ``` ``` 0.42633151142735126 ```