

Jawaban Tugas 3 STDA4101-24.1 (jawaban 1 dan 2)

December 10, 2024

Nama: Moh Evan Alsadik | Nim: 053779188 | STDA4101

```
[54]: from scipy.stats import linregress
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels.api as sm
file_tugas = "Tugas 3 STDA4101-2024.1.xlsx"
tugas_df = pd.read_excel(file_tugas)
# tugas_df.describe()

age = tugas_df["AGE"]
body_mass_index = tugas_df["BMI"]
average_blood_pressure = tugas_df["BP"]
total_serum_cholesterol = tugas_df["S1"]
perkembangan_penyakit = tugas_df["Y"]
print("\n")
```

Berdasarkan data ukuran kuantitatif perkembangan penyakit setelah satu tahun amatan (Y), usia dalam tahun (AGE), body mass index (BMI), average blood pressure (BP), dan total serum kolesterol (S1) terlampir, lakukan analisis statistik deskriptif dengan cara:

0.0.1 Buatlah visualisasi data multivariat menggunakan Scatter Plot.

Data multivariat merupakan data yang dikumpulkan dari dua atau lebih pengamatan dan diukur menggunakan beberapa karakteristik. Singkatnya, data ini melibatkan lebih dari satu variabel baik independent maupun dependent yang dilakukan analisis secara bersamaan.

Scatterplot adalah salah satu grafik yang umum digunakan dalam menemukan suatu pola hubungan antara 2 variabel. Komponen pada grafik ini terdiri dari garis memanjang (x) sebagai variabel independent, garis melebar untuk variabel dependen, serta titik yang merepresentasikan nilai x dan y.

```
[52]: # Grid 2x2
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

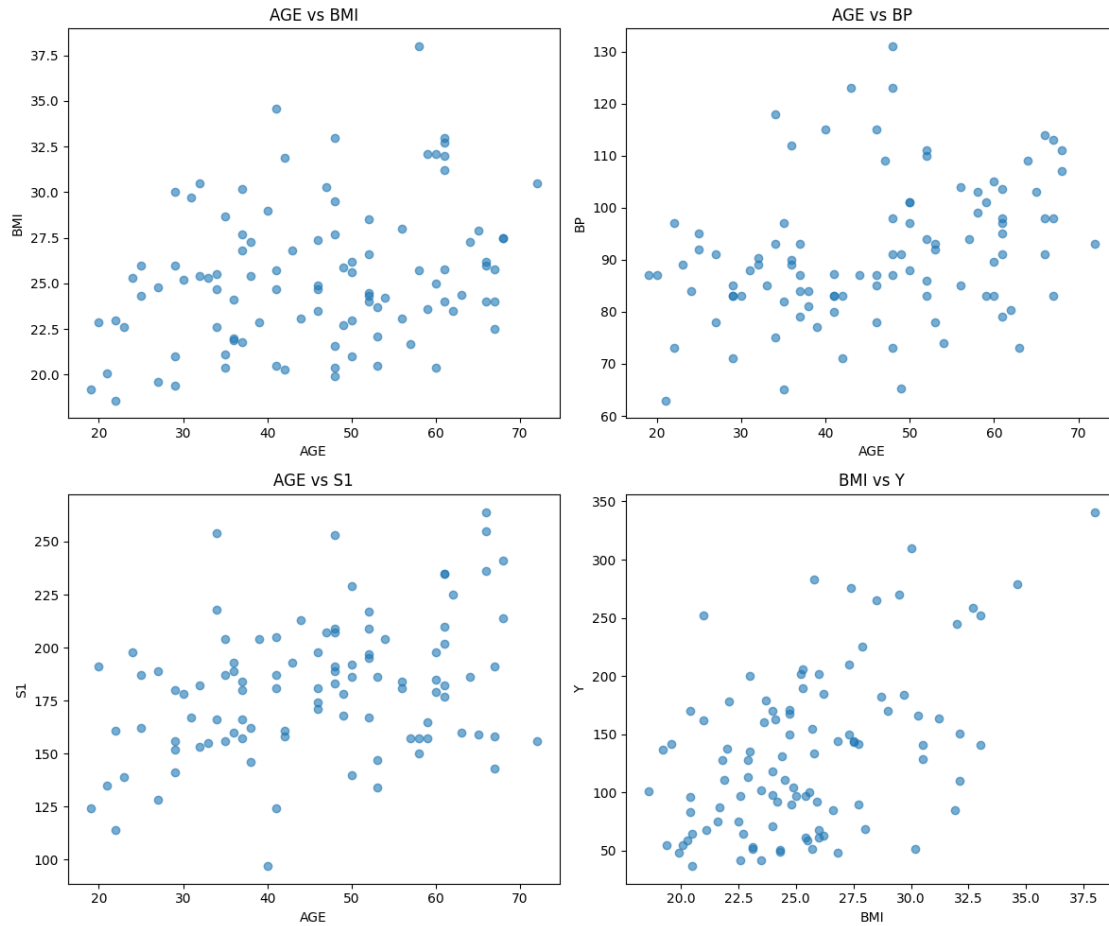
# Plot AGE dengan BMI
axes[0, 0].scatter(age, body_mass_index, alpha=0.6)
axes[0, 0].set_title('AGE vs BMI')
axes[0, 0].set_xlabel('AGE')
axes[0, 0].set_ylabel('BMI')

# Plot AGE dengan BP
axes[0, 1].scatter(age, average_blood_pressure, alpha=0.6)
axes[0, 1].set_title('AGE vs BP')
axes[0, 1].set_xlabel('AGE')
axes[0, 1].set_ylabel('BP')

# Plot AGE dengan S1
axes[1, 0].scatter(age, total_serum_cholesterol, alpha=0.6)
axes[1, 0].set_title('AGE vs S1')
axes[1, 0].set_xlabel('AGE')
axes[1, 0].set_ylabel('S1')

# Plot BMI dengan Y
axes[1, 1].scatter(body_mass_index, perkembangan_penyakit, alpha=0.6)
axes[1, 1].set_title('BMI vs Y')
axes[1, 1].set_xlabel('BMI')
axes[1, 1].set_ylabel('Y')

plt.tight_layout() # Atur tata letak supaya tidak tumpang tindih
plt.show()
print("\n")
```



0.0.2 Tentukan model persamaan regresi linear.

```
[53]: # Variabel independen (AGE, BMI, BP, S1)
X = tugas_df[['AGE', 'BMI', 'BP', 'S1']]

# Menambahkan konstanta (intercept) ke variabel independen
X = sm.add_constant(X) # Tambahkan kolom konstanta (1) agar ada intercept

# Variabel dependen (Y)
Y = tugas_df['Y']

# Model regresi
model = sm.OLS(Y, X).fit()

# Ringkasan hasil regresi
```

```

print(model.summary())
print("\n")

# Menampilkan hasil regresi
print("Persamaan Regresi Linear:")
print(f"Y = {intercept:.2f} + {slope:.2f} * X")
print(f"Koefisien Korelasi (r): {r_value:.2f}") # Menunjukkan kekuatan dan
↪ arah hubungan
print(f"p-value: \t\t{p_value:.4f}") # Menguji signifikansi hubungan antar
↪ variabel
print(f"Standard Error: \t{std_err:.2f}") # Mengukur ketidakpastian dalam
↪ estimasi slope.
print("\n")

```

OLS Regression Results

=====						
Dep. Variable:	Y	R-squared:	0.270			
Model:	OLS	Adj. R-squared:	0.239			
Method:	Least Squares	F-statistic:	8.768			
Date:	Tue, 10 Dec 2024	Prob (F-statistic):	4.55e-06			
Time:	11:42:49	Log-Likelihood:	-549.34			
No. Observations:	100	AIC:	1109.			
Df Residuals:	95	BIC:	1122.			
Df Model:	4					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

const	-119.8912	53.293	-2.250	0.027	-225.690	-14.092
AGE	-0.0748	0.493	-0.152	0.880	-1.054	0.905
BMI	8.3701	1.728	4.843	0.000	4.939	11.801
BP	0.8582	0.517	1.659	0.100	-0.169	1.885
S1	-0.1886	0.213	-0.886	0.378	-0.611	0.234
=====						
Omnibus:	2.415	Durbin-Watson:	1.986			
Prob(Omnibus):	0.299	Jarque-Bera (JB):	2.400			
Skew:	0.329	Prob(JB):	0.301			
Kurtosis:	2.620	Cond. No.	1.87e+03			
=====						

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 1.87e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Persamaan Regresi Linear:

$$Y = -94.10 + 8.96 * X$$

Koefisien Korelasi (r): 0.50

p-value: 0.0000

Standard Error: 1.58

```
[55]: # Data untuk BMI dan Y
bmi = tugas_df['BMI']
y = tugas_df['Y']

# Komponen persamaan regresi
intercept = -94.10
slope = 8.96

# Membuat garis regresi berdasarkan persamaan yang telah didapat
bmi_sorted = np.sort(bmi)
y_pred = intercept + slope * bmi_sorted

# Membuat plot
plt.figure(figsize=(10, 6))
sns.set(style="whitegrid")

# Scatter Plot
plt.scatter(bmi, y, color='blue', alpha=0.6, label="Data Observasi",
            edgecolors="w", s=100)

# Garis Regresi
plt.plot(bmi_sorted, y_pred, color='red', linewidth=3, label="Garis Regresi")

# Label dan Grid
plt.title("Hubungan antara BMI dan Perkembangan Penyakit (Y)", fontsize=16)
plt.xlabel("Body Mass Index (BMI)", fontsize=14)
plt.ylabel("Perkembangan Penyakit (Y)", fontsize=14)
plt.legend()
plt.grid(True)
plt.tight_layout()

# Menampilkan plot
plt.show()
print("\n")
```

