kuis-komstat-revana

November 25, 2023

Nama : Revana Moza Hendriani NRP : 5003221009 Komputasi Statistika ANomor 1

[11]:
```python
import numpy as np

def generate_normal_distribution(mean, std_dev, size):
    data_yang_dihasilkan   = np.random.normal(mean, std_dev, size)
    return data_yang_dihasilkan

#a. Membangkitkan A~N(10,5) sebanyak 50
result_A= generate_normal_distribution(10, 5, 50)
print("a. ", result_A)
print("\n")
#b. Membangkitkan  ~ (1 * ,5) sebanyak 30 (* adalah angka terakhir di NRP)
#angka terakhir NRP = 9
result_B= generate_normal_distribution(19, 5, 30)
print("a. ", result_B)
print("\n")
```

```
a.  [ 9.97575515 14.572622   13.76517252 10.45575354  6.42880534  6.55874447
  8.80083375 14.00949972 10.73232415  4.50918972 16.05794389 22.82286627
 10.72986058  8.51824262 10.50120131  9.44142603  9.36516137  8.53384888
 18.51337137  5.27275376  8.06819221  7.56076218 10.92946359  6.45680738
  8.33459488  8.80045093 11.81062209 14.95416984  3.65125191 20.27976572
  4.13485597  4.81281183  9.31487692 10.1674603  10.78552912  3.01464796
 17.1994922   0.26595619  3.95334291  4.65978195 15.53620899 12.85429408
  3.73184525 11.12987423 11.47926838 14.92704085  9.93323867 12.0224142
 10.30838803  4.2013265 ]


a.  [21.53289654 23.36272143 20.65602398 18.43480003 19.17366791 19.46349918
 15.88722691 29.85123273 18.23987994 25.4061342  13.29129944 16.56722254
 14.80299873 14.92263717 15.57995372  7.49965175 12.95218058 21.59492661
 23.54687516 22.36404671 13.91634313 16.12467817 23.77378249 13.02709204
 17.55452447 22.98069576 15.0169689  30.97328864 21.68217908 14.92909088]
```

[14]:
```
#c. Menguji kesamaan rata-rata antara A dan B
from scipy.stats import t
import numpy as np

def independent_t_test(result_A, result_B, equal_var=True):  #artinya: rata␣
 ↪rata kedua sampel sama atau tidak?
    n1 = len(result_A)
    n2 = len(result_B)
    x_bar1 = np.mean(result_A)
    x_bar2 = np.mean(result_B)
    if equal_var:
        s_pooled = np.sqrt(((n1 - 1) * np.var(result_A) + (n2 - 1) * np.
 ↪var(result_B)) / (n1 + n2 - 2)) #u/ var sama
    else:
        s1 = np.sqrt(np.var(result_A))
        s2 = np.sqrt(np.var(result_B))
        s_pooled = np.sqrt(((n1 - 1) * s1**2 + (n2 - 1) * s2**2) / (n1 + n2 -␣
 ↪2)) #u/ var beda

    t_statistic = (x_bar1 - x_bar2) / (s_pooled / np.sqrt(n1 + n2))
    df = n1 + n2 - 2 #u/independen ini sampelnya dari 2 kelompok sprt. kelas A␣
 ↪dan kelas B, jadi ditambahkan dulu semua sampelnya lalu dikurangi jml␣
 ↪kelompok

    return t_statistic, df

def calculate_p_value(t_statistic, df, two_tailed=True):
    p_value = 2 * (1 - t.cdf(abs(t_statistic), df)) if two_tailed else 1 - t.
 ↪cdf(abs(t_statistic), df)

    return p_value

t_statistic, df = independent_t_test(result_A, result_B)
p_value = calculate_p_value(t_statistic, df)

print("T statistic:", t_statistic)
print("Degrees of freedom:", df)
print("P-value:", p_value)

alpha = 0.05
if p_value < alpha:
    print("\nKesimpulan: Terdapat perbedaan signifikan antara rata-rata kedua␣
 ↪data.")
else:
    print("\nKesimpulan: Tidak terdapat perbedaan signifikan antara rata-rata␣
 ↪kedua data.")
```

```
T statistic: -16.590838426141786
Degrees of freedom: 78
P-value: 0.0

Kesimpulan: Terdapat perbedaan signifikan antara rata-rata kedua data.
```

[15]:
```python
import numpy as np
import pandas as pd

data1 = pd.read_excel('data kuis.xlsx', sheet_name='Sheet1', header = None)
print(data1)
```

```
     0     1     2
0   95  76.0  86.0
1   86  69.0  39.0
2   76  78.0  82.0
3   78  82.0  75.0
4   87  90.0  19.0
5   81  56.0   9.0
6  100   NaN  26.0
7    9   NaN   NaN
```

Nomor 2

[20]:
```python
X = np.array(data1.iloc[:, -1])
Y = np.array(data1.iloc[:, 0])
Z = np.array(data1.iloc[:, 1])

#mengubah NaN ke nilai 0 agar dapat menghasilkan nilai t stat dan p value pada␣
 ↪uji independent t test
X[np.isnan(X)] = 0
Y[np.isnan(Y)] = 0
Z[np.isnan(Z)] = 0

print("Nilai X:")
print(X)

print("\nNilai Y:")
print(Y)

print("\nNilai Z:")
print(Z)
```

```
Nilai X:
[86. 39. 82. 75. 19.  9. 26.  0.]

Nilai Y:
[ 95  86  76  78  87  81 100   9]
```

```
Nilai Z:
[76. 69. 78. 82. 90. 56.  0.  0.]
```

[22]:
```python
from scipy.stats import t
import numpy as np

def independent_t_test(X, Y, Z, equal_var=True):  #artinya: rata rata ketiga␣
 ↪sampel sama atau tidak?
    n1 = len(X)
    n2 = len(Y)
    n3 = len(Z)
    x_bar1 = np.mean(X)
    x_bar2 = np.mean(Y)
    x_bar3 = np.mean(Z)
    if equal_var:
        s_pooled = np.sqrt(((n1 - 1) * np.var(X) + (n2 - 1) * np.var(Y) + (n3 -␣
 ↪1) * np.var(Z)) / (n1 + n2 + n3 - 3)) #u/ var sama
    else:
        s1 = np.sqrt(np.var(X))
        s2 = np.sqrt(np.var(Y))
        s2 = np.sqrt(np.var(Z))
        s_pooled = np.sqrt(((n1 - 1) * s1**2 + (n2 - 1) * s2**2 + (n3 - 1) *␣
 ↪s3**2 ) / (n1 + n2 + n3 - 2)) #u/ var beda

    t_statistic = (x_bar1 - x_bar2) / (s_pooled / np.sqrt(n1 + n2 + n3))
    df = n1 + n2 + n3 - 3

    return t_statistic, df

def calculate_p_value(t_statistic, df, two_tailed=True):
    p_value = 2 * (1 - t.cdf(abs(t_statistic), df)) if two_tailed else 1 - t.
 ↪cdf(abs(t_statistic), df)

    return p_value

t_statistic, df = independent_t_test(X, Y, Z)
p_value = calculate_p_value(t_statistic, df)

print("T statistic:", t_statistic)
print("Degrees of freedom:", df)
print("P-value:", p_value)

alpha = 0.05
if p_value < alpha:
    print("\nKesimpulan: Terdapat perbedaan signifikan antara rata-rata␣
 ↪kelompok 1, 2, dan 3.")
else:
    print("\nKesimpulan: Tidak terdapat perbedaan signifikan antara rata-rata␣
↪kelompok 1, 2, dan 3.")
```

```
T statistic: -5.447618950321652
Degrees of freedom: 21
P-value: 2.0996828458796912e-05

Kesimpulan: Terdapat perbedaan signifikan antara rata-rata kelompok 1, 2, dan 3.
```

Nomor 3

[24]:
```
import numpy as np
import pandas as pd

data = pd.read_excel('data kuis 2.xlsx', sheet_name='Sheet1', header = None)
print(data)
```

```
      0
0    95
1    86
2    76
3    78
4    87
5    81
6   100
7     9
8    76
9    69
10   78
11   82
12   90
13   56
14   86
15   39
16   82
17   75
18   19
19    9
20   26
```

[28]:
```
Nilai = np.array(data.iloc[:, -1])
print("Data Variabel Nilai:")
print(Nilai)
```

```
Data Variabel Nilai:
[ 95  86  76  78  87  81 100   9  76  69  78  82  90  56  86  39  82  75
  19   9  26]
```

[32]:
```python
import numpy as np

def is_inner_outlier(Nilai):
    # Menghitung Q1, Q3, dan IQR
    q1 = np.percentile(Nilai, 25)
    q3 = np.percentile(Nilai, 75)
    iqr = q3 - q1

    # Mengecek apakah terdapat pengamatan inner outlier
    inner_outlier_1 = any((q3 + 1.5 * iqr < x < q3 + 3 * iqr) for x in Nilai)
    inner_outlier_2 = any((q1 + 3 * iqr < x < q1 + 1.5 * iqr) for x in Nilai)
    return inner_outlier_1, inner_outlier_2

# Memanggil fungsi untuk mendeteksi inner outlier
hasil_deteksi = is_inner_outlier(Nilai)

# Menampilkan hasil deteksi
if hasil_deteksi:
    print("Terdapat pengamatan inner outlier pada data variabel Nilai : ")
else:
    print("Tidak terdapat pengamatan inner outlier pada data variabel Nilai.")
```

```
Terdapat pengamatan inner outlier pada data variabel Nilai :  (False, False)
```

[ ]:
```
```