

Nomor 6

```
In [ ]: import numpy as np
import pandas as pd
from scipy.stats import t, ttest_ind, norm
```

```
In [ ]: """
Melakukan uji t dua sampel secara manual untuk perbandingan rata-rata dua populasi.

Args:
- sample1: array-like, sampel pertama
- sample2: array-like, sampel kedua
- alpha: float, tingkat signifikansi

Returns:
- t_statistic: nilai t-statistic dari uji t dua sampel
- p_value: nilai p-value dari uji t dua sampel
"""

# Fungsi untuk melakukan uji t dua sampel secara manual
def two_sample_t_test_manual(sample1, sample2, alpha):
    mean1 = np.mean(sample1)
    mean2 = np.mean(sample2)
    std1 = np.std(sample1, ddof=1)
    std2 = np.std(sample2, ddof=1)
    n1 = len(sample1)
    n2 = len(sample2)
    pooled_std = np.sqrt(((n1 - 1) * std1**2 + (n2 - 1) * std2**2) / (n1 + n2 - 2))
    t_statistic = (mean1 - mean2) / (pooled_std * np.sqrt(1/n1 + 1/n2))
    df = n1 + n2 - 2

    # Hitung p-value
    p_value = 2 * (1 - t.cdf(abs(t_statistic), df))

    return t_statistic, p_value

def two_proportion_z_test(sample1, sample2, alpha):
    n1 = len(sample1)
    n2 = len(sample2)
    p1 = np.sum(sample1 > 8) / n1
    p2 = np.sum(sample2 > 8) / n2
    p_pool = (np.sum(sample1 > 8) + np.sum(sample2 > 8)) / (n1 + n2)
    z_statistic = (p1 - p2) / np.sqrt(p_pool * (1 - p_pool) * (1/n1 + 1/n2))
    p_value = 1 - norm.cdf(z_statistic) # uji satu ekor

    return z_statistic, p_value
```

```
In [ ]: from scipy.stats import bartlett, chi2
```

```
"""
Melakukan uji Bartlett secara manual untuk perbandingan variansi dua populasi.

Args:
```

```

- sample1: array-like, sampel pertama
- sample2: array-like, sampel kedua

Returns:
- bartlett_stat: nilai Bartlett's statistic
- p_value: nilai p-value dari uji Bartlett
"""
# def bartlett_test_manual(sample1, sample2):
#     n1, n2 = len(sample1), len(sample2)
#     var1, var2 = np.var(sample1, ddof=1), np.var(sample2, ddof=1)
#     s_pooled = ((n1 - 1) * var1 + (n2 - 1) * var2) / (n1 + n2 - 2)
#     bartlett_stat = (n1 * np.log(var1) + n2 * np.log(var2) - (n1 + n2) * np.log(s
#     df = 1 # Degree of freedom for Bartlett test between two samples

#     # Menghitung p-value
#     p_value = 1 - chi2.cdf(bartlett_stat, df)

#     return bartlett_stat, p_value

def bartlett_test_manual(sample1, sample2):
    n1, n2 = len(sample1), len(sample2)
    var1, var2 = np.var(sample1, ddof=1), np.var(sample2, ddof=1)

    s_pooled = ((n1 - 1) * var1 + (n2 - 1) * var2) / (n1 + n2 - 2)

    N = n1 + n2
    k = 2 # Number of groups

    numerator = (N * np.log(s_pooled)) - (n1 * np.log(var1) + n2 * np.log(var2))
    denominator = 1 + (1 / (3 * (k - 1))) * ((1 / (n1 - 1)) + (1 / (n2 - 1)) - (1 /

    bartlett_stat = numerator / denominator
    df = 1 # Degree of freedom for Bartlett test between two samples

    # Menghitung p-value
    p_value = 1 - chi2.cdf(bartlett_stat, df)

    return bartlett_stat, p_value

```

```

In [ ]: from scipy.stats import norm
from statsmodels.stats.proportion import proportions_ztest

"""
Melakukan uji proporsi dua sampel secara manual.

Args:
- sample1: array-like, sampel pertama
- sample2: array-like, sampel kedua
- alpha: float, tingkat signifikansi

Returns:
- z_statistic: nilai z-statistic dari uji proporsi dua sampel
- p_value: nilai p-value dari uji proporsi dua sampel
"""
def two_proportion_z_test(sample1, sample2, alpha):
    n1 = len(sample1)

```

```

n2 = len(sample2)
p1 = np.sum(sample1) / n1
p2 = np.sum(sample2) / n2
p_pool = (np.sum(sample1) + np.sum(sample2)) / (n1 + n2)
z_statistic = (p1 - p2) / np.sqrt(p_pool * (1 - p_pool) * (1/n1 + 1/n2))
p_value = 1 - norm.cdf(z_statistic) # uji satu ekor

return z_statistic, p_value

```

```

In [ ]: # Fungsi untuk melakukan uji t dua sampel satu arah secara manual
def one_sample_t_test_manual(sample1, sample2, delta, alpha):
    mean1 = np.mean(sample1)
    mean2 = np.mean(sample2)
    std1 = np.std(sample1, ddof=1)
    std2 = np.std(sample2, ddof=1)
    n1 = len(sample1)
    n2 = len(sample2)
    pooled_std = np.sqrt(((n1 - 1) * std1**2 + (n2 - 1) * std2**2) / (n1 + n2 - 2))
    t_statistic = ((mean1 - mean2) - delta) / (pooled_std * np.sqrt(1/n1 + 1/n2))
    df = n1 + n2 - 2

    # Hitung p-value untuk uji satu ekor
    p_value = t.cdf(t_statistic, df)

    return t_statistic, p_value

```

```

In [ ]: banana_data = pd.read_csv("banana.csv")

```

6.1 Acidity Mean of First Half = Last Half?

```

In [ ]: # Bagi dataset menjadi dua bagian
n = len(banana_data)
n_half = n // 2
first_half_acidity = banana_data['Acidity'][:n_half]
second_half_acidity = banana_data['Acidity'][n_half:]

# Tentukan Hipotesis Nol dan Alternatif
# H0: Rata-rata acidity pada paruh awal sama dengan rata-rata acidity pada paruh ak
# H1: Rata-rata acidity pada paruh awal berbeda dengan rata-rata acidity pada paruh

# Tentukan Tingkat Signifikansi
alpha = 0.05

# Hitung nilai uji statistik dan p-value menggunakan metode manual
t_stat_manual, p_value_manual = two_sample_t_test_manual(first_half_acidity, second_half_acidity)

# Hitung nilai uji statistik dan p-value menggunakan SciPy
t_stat_scipy, p_value_scipy = ttest_ind(first_half_acidity, second_half_acidity)

# Ambil keputusan
decision_manual = 'Tolak H0' if p_value_manual < alpha else 'Gagal Tolak H0'
decision_scipy = 'Tolak H0' if p_value_scipy < alpha else 'Gagal Tolak H0'

# Kesimpulan berdasarkan keputusan

```

```

conclusion_scipy = 'Rata-rata acidity pada paruh awal berbeda dengan rata-rata acid
conclusion_manual = 'Rata-rata acidity pada paruh awal berbeda dengan rata-rata aci

# Tampilkan hasil
print("Hasil tes menggunakan perhitungan manual:")
print(f"t-statistic: {t_stat_manual}, p-value: {p_value_manual}, Keputusan: {decisi
print("Keputusan:", conclusion_manual)

print("\nHasil tes menggunakan library SciPy:")
print(f"t-statistic: {t_stat_scipy}, p-value: {p_value_scipy}, Keputusan: {decision
print("Kesimpulan:", conclusion_scipy)

```

Hasil tes menggunakan perhitungan manual:

t-statistic: -1.2604879619118594, p-value: 0.20764057121149349, Keputusan: Gagal Tolak H_0

Keputusan: Rata-rata acidity pada paruh awal sama dengan rata-rata acidity pada paruh akhir

Hasil tes menggunakan library SciPy:

t-statistic: -1.2604879619118592, p-value: 0.2076405712114936, Keputusan: Gagal Tolak H_0

Kesimpulan: Rata-rata acidity pada paruh awal sama dengan rata-rata acidity pada paruh akhir

6.2 Mean Appearance First Half is 0.1 Greater than Last Half?

```

In [ ]: banana_data = pd.read_csv("banana.csv")

# Bagi dataset menjadi dua bagian
n = len(banana_data)
n_half = n // 2
first_half_appearance = banana_data['Appearance'][:n_half]
second_half_appearance = banana_data['Appearance'][n_half:]

# Tentukan Hipotesis Nol dan Alternatif
# H0: Rata-rata appearance pada paruh awal tidak lebih besar dari rata-rata appearance
# H1: Rata-rata appearance pada paruh awal lebih besar dari rata-rata appearance pa

# Tentukan Tingkat Signifikansi
alpha = 0.05

# Hitung nilai uji statistik dan p-value menggunakan metode manual
t_stat_manual, p_value_manual = two_sample_t_test_manual(first_half_appearance, sec

# Hitung nilai uji statistik dan p-value menggunakan SciPy
t_stat_scipy, p_value_scipy = ttest_ind(first_half_appearance, second_half_appearan

# Ambil keputusan
decision_manual = 'Tolak H0' if p_value_manual < alpha else 'Gagal Tolak H0'
decision_scipy = 'Tolak H0' if p_value_scipy < alpha else 'Gagal Tolak H0'

# Kesimpulan berdasarkan keputusan
conclusion_manual = 'Rata-rata appearance pada paruh awal lebih besar dari paruh ak
conclusion_scipy = 'Rata-rata appearance pada paruh awal lebih besar dari paruh akh

```

```
# Tampilkan hasil
print("Hasil tes menggunakan perhitungan manual:")
print(f"t-statistic: {t_stat_manual}, p-value: {p_value_manual}, Keputusan: {decision_manual}")
print("Kesimpulan:", conclusion_manual)

print("\nHasil tes menggunakan library SciPy:")
print(f"t-statistic: {t_stat_scipy}, p-value: {p_value_scipy}, Keputusan: {decision_scipy}")
print("Kesimpulan:", conclusion_scipy)
```

Hasil tes menggunakan perhitungan manual:

t-statistic: -0.047847812034888026, p-value: 0.9618423137383616, Keputusan: Gagal Tolak H_0

Kesimpulan: Rata-rata appearance pada paruh awal tidak lebih besar dari paruh akhir sebesar 0.1 unit

Hasil tes menggunakan library SciPy:

t-statistic: -0.04784781203488802, p-value: 0.5190788431308192, Keputusan: Gagal Tolak H_0

Kesimpulan: Rata-rata appearance pada paruh awal tidak lebih besar dari paruh akhir sebesar 0.1 unit

6.3 Variance of Banana Length are Equal Between First Half and Last Half?

```
In [ ]: n = len(banana_data)
n_half = n // 2
first_half_length = banana_data['Length'][:n_half]
second_half_length = banana_data['Length'][n_half:]

# Tentukan Hipotesis Nol dan Alternatif
# H0: Variansi panjang pisang pada paruh awal sama dengan variansi panjang pisang pada paruh akhir
# H1: Variansi panjang pisang pada paruh awal tidak sama dengan variansi panjang pisang pada paruh akhir

# Tentukan Tingkat Signifikansi
alpha = 0.05

# Hitung nilai uji statistik dan p-value menggunakan metode manual
bartlett_stat_manual, p_value_manual = two_sample_t_test_manual(first_half_length, second_half_length)

# Hitung nilai uji statistik dan p-value menggunakan SciPy
bartlett_stat_scipy, p_value_scipy = bartlett(first_half_length, second_half_length)

# Ambil keputusan
decision_manual = 'Tolak H0' if p_value_manual < alpha else 'Gagal Tolak H0'
decision_scipy = 'Tolak H0' if p_value_scipy < alpha else 'Gagal Tolak H0'

# Kesimpulan berdasarkan keputusan
conclusion_manual = 'Variansi panjang pisang pada paruh awal tidak sama dengan variansi panjang pisang pada paruh akhir'
conclusion_scipy = 'Variansi panjang pisang pada paruh awal tidak sama dengan variansi panjang pisang pada paruh akhir'

# Tampilkan hasil
print("Hasil tes menggunakan perhitungan manual:")
print(f"Bartlett's stat: {bartlett_stat_manual}, p-value: {p_value_manual}, Keputusan: {decision_manual}")
print("Kesimpulan:", conclusion_manual)
```

```
print("\nHasil tes menggunakan library SciPy:")
print(f"Bartlett's stat: {bartlett_stat_scipy}, p-value: {p_value_scipy}, Keputusan")
print("Kesimpulan:", conclusion_scipy)
```

Hasil tes menggunakan perhitungan manual:

Bartlett's stat: 0.5614300115943567, p-value: 0.5745673935621842, Keputusan: Gagal Tolak H_0

Kesimpulan: Variansi panjang pisang pada paruh awal sama dengan paruh akhir

Hasil tes menggunakan library SciPy:

Bartlett's stat: 0.2063368284484766, p-value: 0.6496538375068803, Keputusan: Gagal Tolak H_0

Kesimpulan: Variansi panjang pisang pada paruh awal sama dengan paruh akhir

6.4 Proportion of Banana that More Than 150 in The First Half is Greater than the Proportion of Last Half?

```
In [ ]: # Bagi dataset menjadi dua bagian
first_half_weight = banana_data['Weight'][:n_half]
second_half_weight = banana_data['Weight'][n_half:]

# Tentukan berat pisang yang lebih dari 150
first_half_over_150 = first_half_weight > 150
second_half_over_150 = second_half_weight > 150

# Tentukan Hipotesis Nol dan Alternatif
# H0: Proporsi berat pisang yang lebih dari 150 pada paruh awal tidak lebih besar dari
# H1: Proporsi berat pisang yang lebih dari 150 pada paruh awal lebih besar dari paruh akhir

# Tentukan Tingkat Signifikansi
alpha = 0.05

# Hitung nilai uji statistik dan p-value menggunakan metode manual
z_stat_manual, p_value_manual = two_proportion_z_test(first_half_over_150, second_half_over_150, n_half)

# Hitung nilai uji statistik dan p-value menggunakan SciPy
count = np.array([np.sum(first_half_over_150), np.sum(second_half_over_150)])
nobs = np.array([n_half, len(banana_data) - n_half])
z_stat_scipy, p_value_scipy = proportions_ztest(count, nobs, alternative='larger')

# Ambil keputusan
decision_manual = 'Tolak H0' if p_value_manual < alpha else 'Gagal Tolak H0'
decision_scipy = 'Tolak H0' if p_value_scipy < alpha else 'Gagal Tolak H0'

# Kesimpulan berdasarkan keputusan
conclusion_manual = 'Proporsi berat pisang yang lebih dari 150 pada paruh awal lebih besar dari paruh akhir'
conclusion_scipy = 'Proporsi berat pisang yang lebih dari 150 pada paruh awal lebih besar dari paruh akhir'

# Tampilkan hasil
print("Hasil tes menggunakan perhitungan manual:")
print(f"z-statistic: {z_stat_manual}, p-value: {p_value_manual}, Keputusan: {decision_manual}")
print("Kesimpulan:", conclusion_manual)

print("\nHasil tes menggunakan library SciPy:")
```

```
print(f"z-statistic: {z_stat_scipy}, p-value: {p_value_scipy}, Keputusan: {decision_scipy}")  
print("Kesimpulan:", conclusion_scipy)
```

Hasil tes menggunakan perhitungan manual:

z-statistic: 1.3865625188118174, p-value: 0.0827876020366598, Keputusan: Gagal Tolak H_0

Kesimpulan: Proporsi berat pisang yang lebih dari 150 pada paruh awal tidak lebih besar daripada paruh akhir

Hasil tes menggunakan library SciPy:

z-statistic: 1.3865625188118174, p-value: 0.08278760203665986, Keputusan: Gagal Tolak H_0

Kesimpulan: Proporsi berat pisang yang lebih dari 150 pada paruh awal tidak lebih besar daripada paruh akhir