

Nomor 5

```
In [ ]: import pandas as pd
        from scipy.stats import ttest_1samp, t, norm
        import numpy as np
        from statsmodels.stats.proportion import proportions_ztest
        import matplotlib.pyplot as plt
```

```
In [ ]: banana_data = pd.read_csv('banana.csv')
        banana_data.describe()
```

```
Out[ ]:
```

	Unnamed: 0	Acidity	Weight	Length	Appearance	Tannin	R
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000
mean	999.500000	8.014830	150.011549	49.950434	4.965595	7.965435	6
std	577.494589	1.105781	1.194980	0.894599	1.014863	1.217188	0
min	0.000000	4.456118	146.060922	46.418052	1.775864	4.291274	4
25%	499.750000	7.259942	149.227116	49.346508	4.258210	7.167241	6
50%	999.500000	8.005347	150.022865	49.923682	4.979534	8.022448	6
75%	1499.250000	8.758361	150.827613	50.572027	5.653875	8.792184	7
max	1999.000000	11.418636	154.070370	53.065151	8.233968	12.416177	9

```
In [ ]: def one_sample_t_test_sf(sample, popmean):
        """
        Parameter:
            sample: sample data
            popmean: hipotesis rata rata populasi
        """
        sample_mean = np.mean(sample)
        sample_std = np.std(sample, ddof=1)
        n = len(sample)
        t_stat = (sample_mean - popmean) / (sample_std / np.sqrt(n))
        df = n - 1
        p_value = 2 * t.sf(t_stat, df)
        return t_stat, p_value

        def t_critical(df, alpha, tail='one'):
            step = 0.0001 # Tingkat kepresisian
            prob_target = 1 - alpha if tail == 'one' else 1 - alpha / 2
            cumulative_prob = 0
            t_val = 0

            while cumulative_prob < prob_target:
```

```

        t_val += step
        cumulative_prob = t.cdf(t_val, df)

    return t_val

def proportion_test(count, nobs, prop0):
    """
    Manually performs a z-test for proportions.

    Parameters:
        count (int): The number of successes.
        nobs (int): The number of observations.
        prop0 (float): The hypothesized proportion.

    Returns:
        z_stat (float): The calculated z statistic.
        p_value (float): The p-value from the z statistic.
    """
    prop = count / nobs
    se = np.sqrt(prop0 * (1 - prop0) / nobs)
    z_stat = (prop - prop0) / se
    p_value = norm.sf(abs(z_stat)) * 2 # two-tailed
    return z_stat, p_value

```

5.1 Mean Acidity > 6?

```

In [ ]: sample_data_acidity = banana_data['Acidity']

# Langkah 1: tentukan hipotesis nol
# H0:  $\mu = 6$  (rata-rata acidity adalah 6)

# Langkah 2: pilih hipotesis Alternatif
# H1:  $\mu > 6$  (rata-rata acidity lebih dari 6)

# Langkah 3: tentukan tingkat signifikansi
alpha = 0.05

# Langkah 4: tentukan uji statistik dan daerah kritis
# Menggunakan t-test untuk satu sampel
n = len(sample_data_acidity) # jumlah sampel
df = n - 1 # derajat kebebasan
t_critical_scipy = t.ppf(1 - alpha, df) # nilai kritis untuk tes satu arah
print(f"Nilai t-kritis (one-tailed, right): {t_critical_scipy}")

# Langkah 5: hitung nilai uji statistik dari data sample

# Menggunakan fungsi buatan sendiri
t_stat_manual, p_value_manual = one_sample_t_test_sf(sample_data_acidity, 6)

# Menggunakan fungsi dari SciPy
t_stat_scipy, p_value_scipy = ttest_1samp(sample_data_acidity, 6)

# Langkah 6: 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'

```

```

conclusion_manual = 'Rata-rata acidity lebih dari 6' if decision_manual == 'Tolak H
conclusion_scipy = 'Rata-rata acidity lebih dari 6' if decision_scipy == 'Tolak H0'

print("\nHasil tes menggunakan fungsi manual:")
print(f"t-statistic: {t_stat_manual}, p-value: {round(p_value_manual, 10)}, Keputus
print("Kesimpulan:", conclusion_manual)

print("\nHasil tes menggunakan fungsi SciPy:")
print(f"t-statistic: {t_stat_scipy}, p-value: {round(p_value_scipy, 10)}, Keputusan
print("Kesimpulan:", conclusion_scipy)

```

Nilai t-kritis (one-tailed, right): 1.645616248187262

Hasil tes menggunakan fungsi manual:

t-statistic: 81.48620233757643, p-value: 0.0, Keputusan: Tolak H0

Kesimpulan: Rata-rata acidity lebih dari 6

Hasil tes menggunakan fungsi SciPy:

t-statistic: 81.48620233757642, p-value: 0.0, Keputusan: Tolak H0

Kesimpulan: Rata-rata acidity lebih dari 6

5.2 Mean Weight ≠ 150 gram?

```

In [ ]: sample_data_weight = banana_data['Weight']

# Langkah 1: tentukan hipotesis nol
# H0:  $\mu = 150$  (rata-rata berat adalah 150 gram)

# Langkah 2: pilih hipotesis alternatif
# H1:  $\mu \neq 150$  (rata-rata berat bukan 150 gram)

# Langkah 3: tentukan tingkat signifikansi
alpha = 0.05

# Langkah 4: tentukan uji statistik dan daerah kritis
# Menggunakan t-test untuk satu sampel
n = len(sample_data_weight) # jumlah sample
df = n - 1 # derajat kebebasan
t_critical_scipy = t.ppf(1 - alpha, df) # nilai kritis untuk tes satu arah
print(f"Nilai t-kritis (one-tailed, right): {t_critical_scipy}")

# Langkah 5: hitung nilai uji statistik dari data sample
# Menggunakan fungsi buatan sendiri
t_stat_manual, p_value_manual = one_sample_t_test_sf(sample_data_weight, 150)

# Menggunakan fungsi dari SciPy
t_stat_scipy, p_value_scipy = ttest_1samp(sample_data_weight, 150)

# Langkah 6: 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'

conclusion_manual = 'Rata-rata berat bukan 150 gram' if decision_manual == 'Tolak H

```

```

conclusion_scipy = 'Rata-rata berat bukan 150 gram' if decision_scipy == 'Tolak H0'

print("\nHasil tes menggunakan fungsi manual:")
print(f"t-statistic: {t_stat_manual}, p-value: {round(p_value_manual, 10)}, Keputusan")
print("Kesimpulan: ", conclusion_manual)

print("\nHasil tes menggunakan fungsi SciPy:")
print(f"t-statistic: {t_stat_scipy}, p-value: {round(p_value_scipy, 10)}, Keputusan")
print("Kesimpulan: ", conclusion_scipy)

```

Nilai t-kritis (one-tailed, right): 1.645616248187262

Hasil tes menggunakan fungsi manual:

t-statistic: 0.4322079045945881, p-value: 0.6656369317, Keputusan: Gagal Tolak H0

Kesimpulan: Rata-rata berat 150 gram

Hasil tes menggunakan fungsi SciPy:

t-statistic: 0.432207904594588, p-value: 0.6656369317, Keputusan: Gagal Tolak H0

Kesimpulan: Rata-rata berat 150 gram

5.3 Last 10 Row Mean Length \neq 49

```

In [ ]: sample_data_length = banana_data['Length'].tail(10)

# Langkah 1: tentukan hipotesis nol
# H0:  $\mu = 150$  (rata-rata berat adalah 150 gram)

# Langkah 2: pilih hipotesis alternatif
# H1:  $\mu \neq 150$  (rata-rata berat bukan 150 gram)

# Langkah 3: tentukan tingkat signifikansi
alpha = 0.05

# Langkah 4: tentukan uji statistik dan daerah kritis
# Menggunakan t-test untuk satu sampel
n = len(sample_data_length) # jumlah sample
df = n - 1 # derajat kebebasan
t_critical_scipy = t.ppf(1 - alpha, df) # nilai kritis untuk tes satu arah
print(f"Nilai t-kritis (one-tailed, right): {t_critical_scipy}")

# Langkah 5: hitung nilai uji statistik dari data sample
# Menggunakan fungsi buatan sendiri
t_stat_manual, p_value_manual = one_sample_t_test_sf(sample_data_length, 49)

# Menggunakan fungsi dari SciPy
t_stat_scipy, p_value_scipy = ttest_1samp(sample_data_length, 49)

# Langkah 6: 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'

conclusion_manual = 'rata-rata panjang pisang tidak sama dengan 49' if p_value_manu
conclusion_scipy = 'rata-rata panjang pisang tidak sama dengan 49' if p_value_scipy

print("\nHasil tes menggunakan fungsi manual:")

```

```
print(f"t-statistic: {t_stat_manual}, p-value: {round(p_value_manual, 10)}, Keputusan
print(f"Kesimpulan: {conclusion_manual}")

print("\nHasil tes menggunakan fungsi SciPy:")
print(f"t-statistic: {t_stat_scipy}, p-value: {round(p_value_scipy, 10)}, Keputusan
print(f"Kesimpulan: {conclusion_scipy}")
```

Nilai t-kritis (one-tailed, right): 1.8331129326536333

Hasil tes menggunakan fungsi manual:

t-statistic: 1.2862564266596896, p-value: 0.2304557295, Keputusan: Gagal Tolak H₀
Kesimpulan: rata-rata panjang pisang sama dengan 49

Hasil tes menggunakan fungsi SciPy:

t-statistic: 1.2862564266596894, p-value: 0.2304557295, Keputusan: Gagal Tolak H₀
Kesimpulan: rata-rata panjang pisang sama dengan 49

5.4 Tannin Value > 8 ≠ 55% of Total Dataset

```
In [ ]: total_tannin = len(banana_data)
count_tannin_gt8 = np.sum(banana_data['Tannin'] > 8)

# Langkah 1: tentukan hipotesis nol
# H0: proporsi nilai Tannin yang lebih besar dari 8 = 55%

# Langkah 2: tentukan hipotesis alternatif
# H1: proporsi nilai Tannin yang lebih besar dari 8 ≠ 55%

# Langkah 3: tentukan tingkat signifikansi
alpha = 0.05

# Langkah 4: tentukan uji statistik dan daerah kritis
# Menggunakan t-test untuk satu sampel
n = count_tannin_gt8 # jumlah sample
df = n - 1 # derajat kebebasan
t_critical_scipy = t.ppf(1 - alpha, df) # nilai kritis untuk tes satu arah
print(f"Nilai t-kritis (one-tailed, right): {t_critical_scipy}")

# Langkah 5: hitung nilai uji statistik dari data sample
# Menggunakan fungsi buatan sendiri
z_stat_manual, p_value_manual = proportion_test(count_tannin_gt8, total_tannin, 0.5)

# Menggunakan fungsi dari library
z_stat_library, p_value_library = proportions_ztest(count_tannin_gt8, total_tannin,

# Decision making based on alpha
decision_manual = 'Tolak H0' if p_value_manual < alpha else 'Gagal Tolak H0'
decision_library = 'Tolak H0' if p_value_library < alpha else 'Gagal Tolak H0'
conclusion_manual = 'Proporsi nilai Tannin yang lebih besar dari 8 tidak sama denga
conclusion_library = 'Proporsi nilai Tannin yang lebih besar dari 8 tidak sama deng
print("\nHasil tes menggunakan fungsi manual:")
print(f"z-statistic: {z_stat_manual}, p-value: {p_value_manual}, Keputusan: {decisi
print(f"Kesimpulan: {conclusion_manual}")

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

```
print(f"z-statistic: {z_stat_library}, p-value: {p_value_library}, Keputusan: {decision_library}")  
print(f"Kesimpulan: {conclusion_library}")
```

Nilai t-kritis (one-tailed, right): 1.6463607153610087

Hasil tes menggunakan fungsi manual:

z-statistic: -3.9103592022868128, p-value: 9.215896806326626e-05, Keputusan: Tolak H_0

Kesimpulan: Proporsi nilai Tannin yang lebih besar dari 8 tidak sama dengan 55%

Hasil tes menggunakan library Statsmodels:

z-statistic: -3.891087091601724, p-value: 9.97961019109872e-05, Keputusan: Tolak H_0

Kesimpulan: Proporsi nilai Tannin yang lebih besar dari 8 tidak sama dengan 55%