

Uji Hipotesis

One sample z test

```
#Menggunakan package yang tersedia  
from statsmodels.stats.weightstats import ztest as ztest
```

```
#enter data  
data = [88, 92, 94, 94, 96, 97, 97, 97, 99, 99,  
        105, 109, 109, 109, 110, 112, 112, 113, 114, 115]
```

```
#perform one sample z-test test value is 100  
ztest(data, value=100)
```

One sample z test

```
import numpy as np
```

```
def one_sample_z_test(sample, mu):  
    n = len(sample)  
    x_bar = np.mean(sample)  
    s = np.sqrt(np.sum((sample - x_bar)**2) / (n - 1))  
  
    z_statistic = (x_bar - mu) / (s / np.sqrt(n))  
  
    return z_statistic
```

```
# Example usage
```

```
sample = np.array([10, 12, 14, 15, 16, 17, 18, 19, 20])  
mu = 15
```

```
z_statistic = one_sample_z_test(sample, mu)  
print("Z statistic:", z_statistic)
```

Tugas 1: Tambahkan p perhitungan p value

One sample t test

```
#menggunakan package yang tersedia  
import scipy.stats as stats
```

```
sample = [2, 3, 4, 5, 6, 7, 8, 9, 10]  
population_mean = 5
```

```
t_statistic, p_value = stats.ttest_1samp(sample,  
population_mean)
```

One sample t test

```
import numpy as np
```

```
def one_sample_t_test(sample, mu):  
    n = len(sample)  
    x_bar = np.mean(sample)  
    s = np.sqrt(np.sum((sample - x_bar)**2) / (n - 1))  
    t_statistic = (x_bar - mu) / (s / np.sqrt(n))  
    df = n - 1  
  
    return t_statistic, df
```

```
# Example usage
```

```
sample = np.array([10, 12, 14, 15, 16, 17, 18, 19, 20])  
mu = 15
```

```
t_statistic, df = one_sample_t_test(sample, mu)  
print("T statistic:", t_statistic)  
print("Degrees of freedom:", df)
```

Paired t test

```
import numpy as np

def paired_t_test(sample1, sample2):
    n = len(sample1)
    d = sample1 - sample2
    d_bar = np.mean(d)
    sd = np.sqrt(np.sum((d - d_bar)**2) / (n - 1))

    t_statistic = d_bar / (sd / np.sqrt(n))
    df = n - 1

    return t_statistic, df
```

```
# Example usage
sample1 = np.array([10, 12, 14, 15, 16, 17, 18, 19, 20])
sample2 = np.array([12, 14, 16, 18, 20, 22, 24, 26, 28])

t_statistic, df = paired_t_test(sample1, sample2)
print("T statistic:", t_statistic)
print("Degrees of freedom:", df)
```

Independent t test

```
import numpy as np

def independent_t_test(sample1, sample2, equal_var=True):
    n1 = len(sample1)
    n2 = len(sample2)

    x_bar1 = np.mean(sample1)
    x_bar2 = np.mean(sample2)

    if equal_var:
        s_pooled = np.sqrt(((n1 - 1) * np.var(sample1) + (n2 - 1) * np.var(sample2)) /
(n1 + n2 - 2))
    else:
        s1 = np.sqrt(np.var(sample1))
        s2 = np.sqrt(np.var(sample2))
        s_pooled = np.sqrt(((n1 - 1) * s1**2 + (n2 - 1) * s2**2) / (n1 + n2 - 2))

    t_statistic = (x_bar1 - x_bar2) / (s_pooled / np.sqrt(n1 + n2))
    df = n1 + n2 - 2

    return t_statistic, df
```

Example usage

```
sample1 = np.array([10, 12, 14, 15, 16])
sample2 = np.array([13, 15, 17, 19, 21])
```

```
t_statistic, df = independent_t_test(sample1, sample2)
print("T statistic:", t_statistic)
print("Degrees of freedom:", df)
```

P value

```
import scipy.stats as st
```

```
def t_to_p(t_statistic, df, tails='two-sided'):
    if tails == 'two-sided':
        p_value = 2 * st.t.cdf(-abs(t_statistic), df)
    elif tails == 'one-sided':
        if t_statistic < 0:
            p_value = st.t.cdf(t_statistic, df)
        else:
            p_value = 1 - st.t.cdf(t_statistic, df)
    else:
        raise ValueError("Invalid value for 'tails' parameter.")

    return p_value
```

```
# Example usage
t_statistic = 2.345
df = 10
tails = 'two-sided'
```

```
p_value = t_to_p(t_statistic, df, tails)
print("P-value:", p_value)
```

Tugas 2: Tambahkan perhitungan p value untuk one sample t test, paired t test, dan independent t test

Tugas

3. Buat syntax untuk menguji hipotesis 2 proporsi
4. Buat syntax untuk ANOVA