Lab 2: Hypothesis Testing

Outline

1. Population Proportion Test
    1. One population proportion case
    2. Two population proportion case
2. Hypothesis Testing with Excel
    1. One population mean test with Excel
    2. Two population means test with Excel
3. Hypothesis Testing with python
    1. One population mean test with python
    2. Two population means test with python
4. Nonparametric Testing with Python
    1. Sign test with python
    2. Mann-Whitney rank test with python
    3. Wilcoxon signed rank test with python

Testing Population Proportion: Motivating Example

- Over the past year, the ratio of male students to female students in a student club at SJTU was 4:1.
- A special promotion at the beginning of this term is designed to
- attract female members.
- One month later, the club president wants to know whether the
- proportion of female members had increased.

Testing Population Proportion

- Suppose the population proportion is p, and we want to test whether

p is greater than p_0

- Step 1: this is an upper-tailed hypothesis testing:
  - $H_0: p \leq p_0$, versus $H_a: p > p_0$
- Steps 2-5 to design test procedures.
  - Specify the significant level, test statistic and the rejection rule.
  - Collect a random sample of n members and count the sample

  proportion $\overline{x}$.

Testing Population Proportion

- For the i-th member, let $X_i = 1$ for female and $X_i = 0$ for male. Then
  - $X_i \sim$ Bernoulli(p), for $i = 1,2,\dots,n$
- Step 3: We construct a test statistic whose distribution is known
  - $\overline{p} = \frac{1}{n}\sum_{i=1}^{n} X_i$, where $\sum_{i=1}^{n} X_i \sim B(n,p)$.
- Step 2 and 4: Denote the rejection region as [c, 1], and the significance level is $\sup_{p \le p_0} P($ Type I error $) = P(\overline{p} \ge c \,|p = p_0) = \sum_{k=nc}^{n} \binom{n}{k} p_0^k (1 - p_0)^{n-k} = \alpha$.
- It is very complicated to solve the critical value c for given α.

Central Limit Theorem

- Recall that the Central Limit Theorem states that for i.i.d. random variables $X_1, \ldots, X_n$ with mean μ and variance σ^2, we have:

$$\frac{\frac{1}{n}\sum_{i=1}^{n} X_i - \mu}{\frac{\sigma}{\sqrt{n}}} \sim N(0,1)$$

- In our case, when $p = p_0$, X_i ~ Bernoulli(p_0), and

$$\frac{\bar{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \sim N(0,1)$$

- Rule of Thumb: $np_0 \geq 5$ and $n(1-p_0) \geq 5$

Testing Population Proportion: Normal Approximation

- Based on the central limit theorem, we have an easier way to get the

critical value c is via normal approximation for large n.

- Step 3: Modify the test statistic to $z = \frac{\bar{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$

- Step 4: Using the Z-test for the upper-tailed hypothesis testing

$$\text{Reject } H_0\text{: } p \leq p\_0, \text{ if } \frac{\bar{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \geq z_\alpha$$

Extensions to Other Hypotheses for Population Proportion

- An overview of the lower tail, upper tail and two-tailed tests is as follows.
- The test statistic is the same, but the rejection regions are not.

	Lower-tailed Test	Upper-tailed Test	Two-tailed Test
Hypotheses	$H_0 : p \geq p_0$ $H_a : p < p_0$	$H_0 : p \leq p_0$ $H_a : p > p_0$	$H_0 : p = p_0$ $H_a : p \neq p_0$
Test Statistic	$z = \frac{\bar{p}-p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$	$z = \frac{\bar{p}-p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$	$z = \frac{\bar{p}-p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$
Rejection rule: Critical value approach	Reject H_0 if $z \leq -z_\alpha$	Reject H_0 if $z \geq z_\alpha$	Reject H_0 if $\|z\| \geq -z_{\alpha/2}$
Rejection rule: p-value approach	Reject H_0 if p-value $\leq \alpha$	Reject H_0 if p-value $\leq \alpha$	Reject H_0 if p-value $\leq \alpha$

Testing for Two Proportions: Motivating Example

- A tax preparation firm want to compare the quality of work at two regional offices.
- By randomly sampling tax returns prepared at each office, the firm can estimate the proportion of erroneous returns prepared at each office.
- Step 1: The firm want to test if there is difference between these proportions.

$$H_0\colon p_1 = p_2, \text{ versus } H_a\colon p_1 \neq p_2.$$

- This is a two-tailed test for two (independent) population proportions.
- In general, the test could be two-tailed, upper-tailed and lower-tailed.

Testing the difference between Two Population Proportions

- Intuitively, we would reject H_0 if $|p_1 - p_2|$ is sufficiently large.
- To derive a test statistic for the hypothesis tests, we need the sampling distribution of $\overline{p_1} - \overline{p_2}$.
- As in the 1-sample tests, we shall use normal approximation.
- Step 3: If H_0 is true and $p_1 = p_2 = p$, then

$$\frac{\overline{p_1} - \overline{p_2}}{\sqrt{p(1-p)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \sim N(0,1)$$

where p can be estimated by the pooled estimation

$$\overline{p} = \frac{n_1\overline{p_1} + n_2\overline{p_2}}{n_1 + n_2}$$

Testing the difference between Two Population Proportions

- Step 1: Hypothesis

$$H_0: p_1 = p_2, \text{ versus } H_a: p_1 \neq p_2$$

- Step 2: Choose level of significance α.
- Step 3: Test Statistic

$$z = \frac{\overline{p_1} - \overline{p_2}}{\sqrt{p(1-p)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

- Step 4: Rejection Rule

Reject H_0, if $|z| \geq z_{\frac{\alpha}{2}}$ or p-value $\leq \alpha$.

Hypothesis Testing with Excel

- One Population mean z-test with Excel
  - Hyp Sigma Known.xlsx

	A	B	C	D	E
1	**Yards**		**Hypothesis Test About a Population Mean:**		
2	303		**σ Known Case**		
3	282				
4	289		**Sample Size**	=COUNT(A2:A51)	
5	298		**Sample Mean**	=AVERAGE(A2:A51)	
6	283				
7	317		**Population Standard Deviation**	12	
8	297		**Hypothesized Value**	295	
9	308				
10	317		**Standard Error**	=D7/SQRT(D4)	
11	293		**Test Statistic z**	=(D5-D8)/D10	
12	284				
13	290		***p*-value (Lower Tail)**	=NORM.S.DIST(D11,TRUE)	
14	304		***p*-value (Upper Tail)**	=1-D13	
15	290		***p*-value (Two Tail)**	=2*(MIN(D13,D14))	

Hypothesis Testing with Excel

- One Population mean t-test with Excel
- Hyp Sigma Unknown.xlsx

	A	B	C	D	E
1	**Rating**		**Hypothesis Test About a Population Mean**		
2	5		**With σ Unknown**		
3	7				
4	8		**Sample Size**	=COUNT(A2:A61)	
5	7		**Sample Mean**	=AVERAGE(A2:A61)	
6	8		**Sample Std. Deviation**	=STDEV.S(A2:A61)	
7	8				
8	8		**Hypothesized Value**	7	
9	7				
10	8		**Standard Error**	=D6/SQRT(D4)	
11	10		**Test Statistic *t***	=(D5-D8)/D10	
12	6		**Degrees of Freedom**	=D4-1	
13	7				
14	8		***p*-value (Lower Tail)**	=IF(D11<0,TDIST(-D11,D12,1),1-TDIST(D11,D12,1))	
15	8		***p*-value (Upper Tail)**	=1-D14	
16	9		***p*-value (Two Tail)**	=2*(MIN(D14,D15))	
17	7				

Hyphothesis Testing with Excel

- We use Data analysis module in Excel
  - Installation
- Difference Between Two Population Means: σ_1 and σ_2 Known.
  - ExamScores.xlsx

Hyphothesis Testing with Excel

- Difference Between Two Population Means: σ_1 and σ_2 Unnown.
- SoftwareTest.xlsx

Hyphothesis Testing with Excel

- Difference Between Two Population Means with Matched Samples
- Matched.xlsx

Python Package Needed—— Scipy

- SciPy
- **Scipy** is a python open source mathematical computing library, which can be used in mathematics, science and engineering fields. It is a scientific computing library based on numpy.
- Modules of Scipy:
  - Special functions (scipy.special)
  - Integration (scipy.integrate)
  - Optimization (scipy.optimize)
  - Statistics (scipy.stats)
  - …
- Import scipy
- **Version newer than or equal to 1.6.0 is needed**

Python Package Needed—— Statsmodels

- [Introduction — statsmodels]
- **statsmodels** is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
- Modules of statsmodels:
  - Regression and Linear Models
  - Time Series Analysis
  - Statistics and Tools
  - Data Sets
- Examples:
  - Ordinary Least Squares, Univariate Kernel Density Estimator, Copulas…
- import statsmodels

Dataset: Wine recognition dataset

- Load the data from sklearn.datasets

```python
#load the dataset
dataset = sklearn.datasets.load_wine()
#show the document of the dataset
print(dataset['DESCR'])
```

Python

- Show the info of the dataset from dataset[‘DESCR’]:
  - Number of Instances: 178 (50 in each of three classes)
  - Number of Attributes: 13 numeric
  - Attribute Information:
    - Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline
  - dataset['target_names’]:
    - array(['class_0', 'class_1', 'class_2'], dtype='<U7')

Dataset: Wine recognition dataset

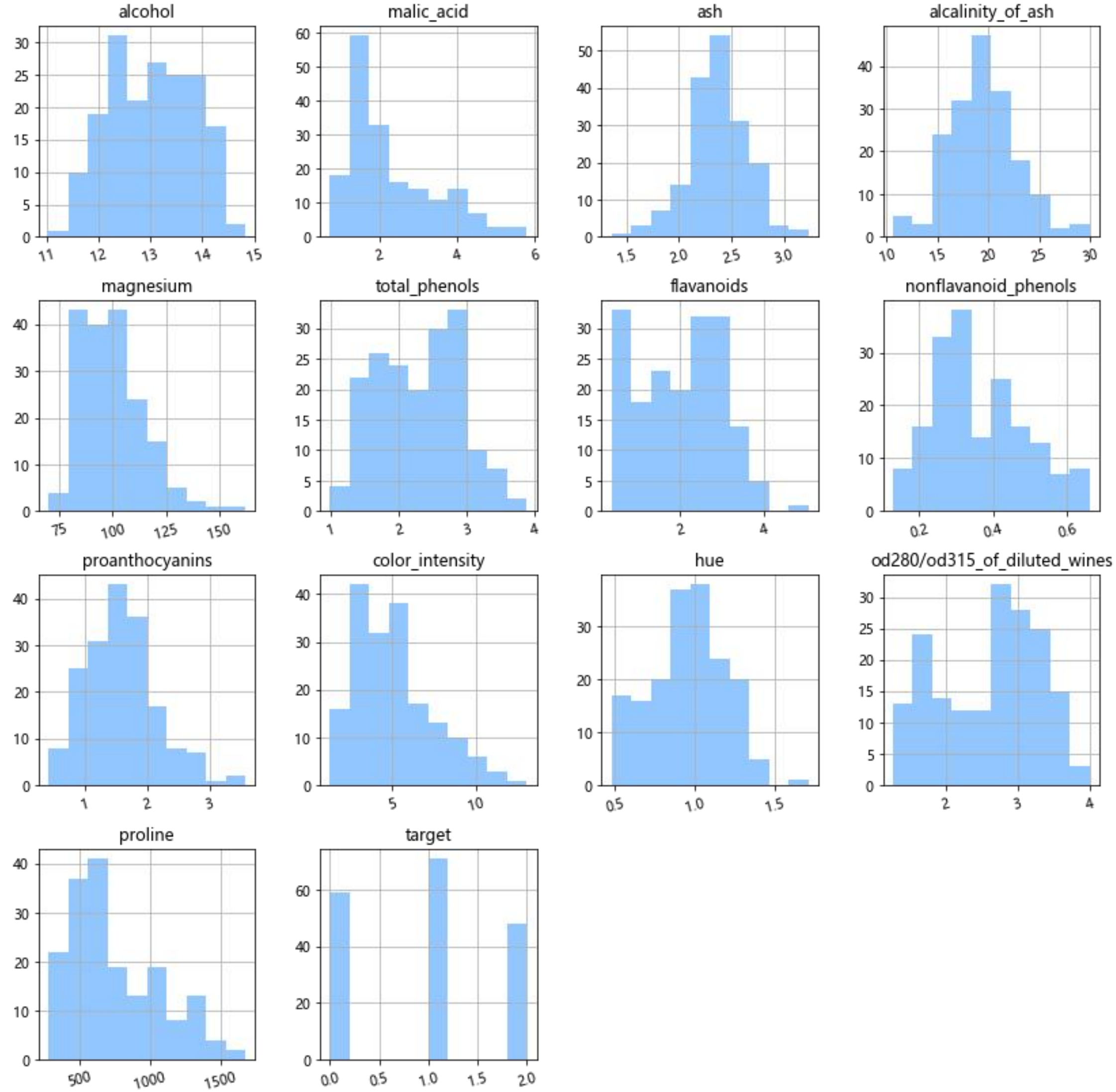
- Form the object with **dataframe** as type:

```
# form the object with dataframe as type
data = pd.DataFrame(dataset['data'],columns = dataset['feature_names'])
data['target'] = dataset['target']
data.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocy
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	

Dataset: Wine recognition dataset

- Plot the distribution of attributes of the dataset:

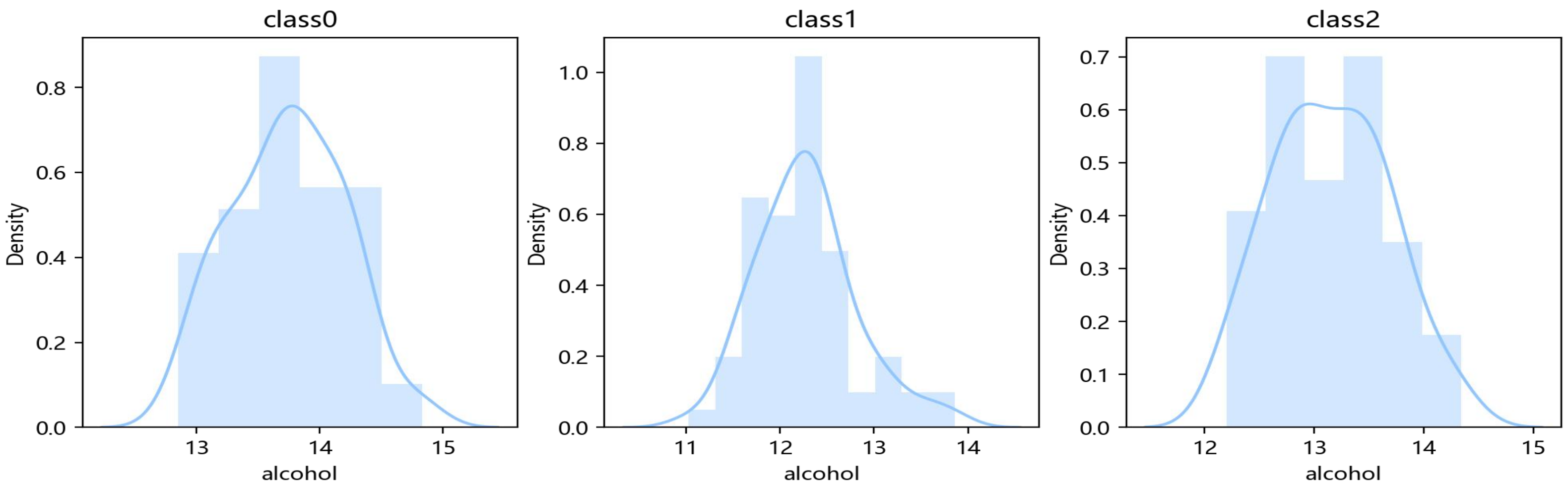


Dataset: Wine recognition dataset

- We want to know whether the distribution of attributes changes with different classes.

- Plot the distplot of attribute “alcohol” with three classes respectively:

```
# shows the alcohol content grouped by 'class'
plt.figure(figsize = (12,4),dpi=200)
for i in range(3):
    plt.subplot(1,3,i+1)
    sns.distplot(data[data['target']==i]['alcohol'],kde = True)
    plt.title('class'+str(i))
plt.show()
```



One Population Mean Z-test with python

- We use function from statsmodels:
- `statsmodels.stats.weightstats.ztest(`

  `x_1,value=mu,alternative='two-sided’)`
  - This function checks one population mean z-test $H_0: \mu_1 = \mu$, $H_a: \mu_1 \neq \mu$.
  - It takes two-sided test when “Alternative” =‘two_sided’.
  - To take lower-tailed mean z-test, make sure “Alternative” =‘larger’
  - To take upper-tailed mean z-test, make sure “Alternative” =‘smaller’
  - Return (z,pval)

```
print(data_1['alcohol'].mean())
```

✓ 0.3s

```
12.278732394366198
```

- We focus on the alcohol content of class-0 cases and check whether the expectation of alcohol content is 12 with one population mean z-test:
- $H_0: \mu = 12$ $H_a: \mu \neq 12$

One Population Mean Z-test with Python

```
print(data_1['alcohol'].mean())
'''
H_0: the average alcohol content of class_1-wine is 12
H_a: the average alcohol content of class_1-wine is not equal to 12
'''
import statsmodels.stats.weightstats

z,pval = statsmodels.stats.weightstats.ztest(data_1['alcohol'],value = 12,alternative='two-sided')
print(z,pval)

'''
p = 1.2666192509733668e-05 < .05, reject H_0
'''
```

One Population Mean T-test with Python

- **scipy.stats.ttest_1samp**(*x_1*, *popmean*, *axis=0*, *nan_policy='propagate'*, *alternative='two-sided')*
  - X_1: data used
  - Popmean: Expected value in null hypothesis.
  - Alternative:
    - 'two-sided' when taking two-sided test
    - 'less' when taking upper-tailed test.
    - 'greater' when taking lower-tailed test.
- $H_0: \mu = 12$ $H_a: \mu \neq 12$
- `scipy.stats.ttest_1samp(data_1['alcohol'],popmean=12)`
- `Return (t,pval).`
- This function achieves two-sided one population mean t-test.

One Population Mean T-test with Python

```
import scipy.stats
t,pval = scipy.stats.ttest_1samp(data_1['alcohol'],popmean=12)
print(t,pval)
'''
p = 4.287629382972247e-05 < .05, reject H_0
'''
```

```
4.3657938005865145 4.287629382972247e-05
```

Two Population Means t-test with Python

- We want to check whether the expectation of alcohol content of class-0 cases is the same as that of class-1 cases:
- H_0: $\mu_1 = \mu_2$, H_a: $\mu_1 \neq \mu_2$.
- We use function scipy.stats.ttest_ind(x_1,x_2, equal_var=True, alternative='two-sided')

```
'''
H_0: the alcohol content of class_1-wine is equal to that of class_2-wine
H_a: the alcohol content of class_1-wine differs from that of class_2-wine
'''
import scipy.stats
t,pval = scipy.stats.ttest_ind(data_1['alcohol'],data_2['alcohol'],alternative = 'two-sided')
print(t,pval)
'''
p < 0.05, reject H_0, it is considered that the alcohol content is different between the two
'''
```

✓ 0.3s

Nonparametric Testing with Python

- **Sign Test**
  - $H_0: median = 12, H_a: median \neq 12$
  - We define a function ***sign_test*** by definition to take the test:

```
def sign_test(data,median):
    k=min(len(data[data>median]), len(data[data<median]))
    pval=2*scipy.stats.binom.cdf(k, len(data), 0.5)
    return pval

print(sign_test(data_1['alcohol'],12))
```

✓ 0.3s

```
0.00011226864689498
```

Nonparametric Testing with Python

➢ Mann-Whitney rank test

- ➢ Again, use dataset wine recognition, $H_0: \mu_1 = \mu_2$, $H_a: \mu_1 \neq \mu_2$.
- ➢ **scipy.stats.mannwhitneyu(x_1, x_2, alternative='two-sided')**

```
scipy.stats.mannwhitneyu(data_1['alcohol'],data_2['alcohol'],alternative='two-sided')
```

✓ 0.3s

```
MannwhitneyuResult(statistic=410.0, pvalue=2.4196781051850865e-12)
```

➢ Wilcoxon signed rank test

- ➢ **scipy.stats.Wilcoxon(x_1,x_2, correction=False, alternative='two-sided', mode='auto')**