Applied Statistical Methods

Tests of Means of Numerical Data- Part I

Xuemao Zhang
East Stroudsburg University

February 20, 2023

Outline

- Statistics with Python
- Backgroud
  - Binomial Distribution
  - Nonparametric Statistics and Rank
- Tests of a single mean
  - One Sample t-Test
  - Sign Test
- Comparison of Two groups
  - Paired t-Test
  - t-Test between Independent Groups
  - Wilcoxon rank-sum test
  - Statistical Hypothesis Tests vs Statistical Modeling

Statistics with Python

- Python was not designed for statistical analysis though it has a module `statistics` offering basic mathematical statistics functions.
- In the second part of our course, we mainly use the following Python packages.
  - `numpy`: great at working with with arrays
  - `scipy` is for scientific computing and technical computing. The module `scipy.stats` contains a large number of probability distributions, summary and frequency statistics, correlation functions and statistical tests, masked statistics, kernel density estimation, quasi-Monte Carlo functionality, and more.
    - https://docs.scipy.org/doc/scipy/reference/stats.html
  - `statsmodels` is a Python library that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
    - `statsmodels` is built on top of the numerical libraries `NumPy` and `SciPy`, integrates with `Pandas` for data handling, and uses `Patsy` for an R-like formula interface.
    - https://www.statsmodels.org/stable/index.html
- R users can check the provide R code available in the lecture slides.

Binomial Distribution

Binomial experiment

(1) The experiment consists of a fixed number, n, of **identical** Bernoulli(binary) trials.
(2) Each trial results in one of two outcomes: success, S, or failure, F.
(3) The trials must be independent. (The outcome of any individual trial does not affect the probabilities in the other trials.)
(4) The probability of a success p remains the same in all trials.
(5) We are interested in the number of successes in n trials.

- The binomial random variable is defined as Y = number of successes out of n Bernoulli trials.
- Example: The coin-tossing experiment is a simple example of a binomial random variable. Toss a fair coin $n = 3$ times and record X = number of heads.

Binomial Distribution

Binomial distribution

The probability mass function of the binomial random variable Y is given by

$$p(y) = P(Y = y) = \binom{n}{y} p^y (1-p)^{n-y}, \qquad y = 0, 1, 2, \ldots, n, 0 \leq p \leq 1.$$

- Note: $\binom{n}{y}$ is the number of outcomes with exactly y successes among n trials.

Mean and variance

Let Y be a binomial random variable based on n trials and success probability p. Then

$$\mu = E(Y) = np \text{ and } \sigma^2 = Var(Y) = np(1-p).$$

Nonparametric Statistics

- Parametric tests have requirements about the nature or shape of the populations involved.
- Nonparametric tests do not require that samples come from populations with specified distribution assumptions, for example normal distributions or any other particular distributions, and homogeneous variances. Consequently, nonparametric tests are called distribution-free tests.

Nonparametric Statistics

- Nonparametric methods can be applied to a wide variety of situations because they do not have the more rigid requirements of the corresponding parametric methods.
  - In general, nonparametric methods do not require normally distributed populations.
- Nonparametric methods tend to waste information because exact numerical data are often reduced to a qualitative form.
- Nonparametric tests are not as efficient (powerful) as parametric tests (if assumptions for parametric tests are satisfied), so with a nonparametric test we generally need stronger evidence (such as a larger sample or greater differences) in order to reject a null hypothesis.

Rank

- Data are sorted when they are arranged according to some criterion, such as smallest to the largest or best to worst.
- A rank is a number assigned to an individual sample item according to its order in the sorted list. The first item is assigned a rank of 1, the second is assigned a rank of 2, and so on.
- Nonparametric methods use the rank information in a data set.

Rank

- **Handling Ties in Ranks**: Find the mean of the ranks involved and assign this mean rank to each of the tied items.

Example The numbers 4, 5, 5, 5, 10, 11, 12, and 12 are given ranks of 1, 3, 3, 3, 5, 6, 7.5, and 7.5, respectively. The table below illustrates the procedure for handling ties.

Sorted Data	Preliminary Ranking		Rank
4	1		**1**
5	2		**3**
5	3	Mean is 3.	**3**
5	4		**3**
10	5		**5**
11	6		**6**
12	7		**7.5**
12	8	Mean is 7.5.	**7.5**

One Sample t-Test

- When $\overline{Y}$ is the mean of a random sample of size n from a **normal distribution** with mean μ, the random variable

$$T = \frac{\overline{Y} - \mu}{S/\sqrt{n}}$$

  has a t-distribution with $n - 1$ degrees of freedom (df).

- If the population is not normally distributed, but the sample size n is large (> 30), then the statistics T above is approximately t-distributed with $n - 1$ df.

- The t-distribution of T is robust to small or even moderate departures from normality unless the sample size n is quite small.

One Sample t-Test

- Let $Y_1, \ldots, Y_n$ be a random sample from a normal population with mean μ.
  (1) A two-sided $1-\alpha$ CI of μ is $\overline{Y} \pm t_{\alpha/2}\left(\frac{S}{\sqrt{n}}\right)$.
  (2) A one-sided $1-\alpha$ CI of μ is $[\overline{Y} - t_{\alpha}\left(\frac{S}{\sqrt{n}}\right), \infty)$.
  (3) A one-sided $1-\alpha$ CI of μ is $(-\infty, \overline{Y} + t_{\alpha}\left(\frac{S}{\sqrt{n}}\right)]$.
  where $t_{\alpha/2}$ is determined from the t distribution with $df = n-1$.

One Sample t-Test

$H_0 : \mu = \mu_0$

$$H_a : \begin{cases} \mu > \mu_0, & \text{upper-tail alternative;} \\ \mu < \mu_0, & \text{lower-tail alternative;} \\ \mu \neq \mu_0, & \text{two-tailed alternative.} \end{cases}$$

Test Statistic: $t_0 = \dfrac{\overline{Y} - \mu_0}{S/\sqrt{n}}$

$$\text{Rejection Region: } RR = \begin{cases} \{t : t \geq t_\alpha\}, & \text{upper-tail RR;} \\ \{t : t \leq -t_\alpha\}, & \text{lower-tail RR;} \\ \{t : |t| \geq t_{\alpha/2}\}, & \text{two-tailed RR.} \end{cases}$$

where the t-distribution has $df = n - 1$.

- Confidence interval method: We reject $H_0 : \mu = \mu_0$ at significance level α if μ_0 lies outside the interval. Especially
  - $\overline{Y} \pm t_{\alpha/2}\left(\frac{S}{\sqrt{n}}\right)$ is for the two-sided test: $H_a : \mu \neq \mu_0$
  - $[\overline{Y} - t_\alpha\left(\frac{S}{\sqrt{n}}\right), \infty)$ is for $H_a : \mu > \mu_0$
  - $(-\infty, \overline{Y} + t_\alpha\left(\frac{S}{\sqrt{n}}\right)]$ is for $H_a : \mu < \mu_0$

One Sample t-Test

- p-value method:
  - Test statistic: $t_0 = \frac{\overline{Y}-\mu_0}{S/\sqrt{n}}$
  - $H_a : \mu > \mu_0$: p-value$=P(t \geq t_0)$
  - $H_a : \mu < \mu_0$: p-value$= P(t \leq t_0)$
  - $H_a : \mu \neq \mu_0$: p-value$= 2P(t \geq |t_0|)$

- 1-sample t-test in Python

```
import pandas as pd
from scipy import stats
mtcars=pd.read_csv("../data/mtcars.csv")
test1=stats.ttest_1samp(mtcars['mpg'], popmean=18, alternative = "greater")
test1.statistic
```

```
## 1.962247030028023
```

```
test1.pvalue
#statistic, pvalue=stats.ttest_1samp(mtcars['mpg'], popmean=18, alternative
#print(statistic)
#print(pvalue)
```

```
## 0.029382112157780 55
```

One Sample t-Test

- R code:

```
t.test(mtcars$mpg, alternative = "greater", mu=18)
```

Sign Test

- Let θ be the population median, the location parameter. When testing the population center, median instead of mean is generally used.
- We test a population median $H_0 : \theta = \theta_0$. The negative and positive signs are based on the claimed value of the median θ_0 under the null hypothesis.
- The signs are the signs of the observations minus θ_0. That is, the observations greater than θ_0 have positive signs and the observations less than θ_0 have negative signs.
- The basic idea underlying the sign test is to analyze the frequencies of the plus and minus signs to determine whether they are significantly different.

Sign Test

- The null hypothesis $H_0 : \theta = \theta_0$ could be rejected only if there are more plus signs or more minus signs.
  - $X =$ the number of times the positive sign occurs
  - $n =$ the total number of positive and negative signs combined
  - $p =$ the proportion of positive (success) signs
  - Under H_0, $X \sim binomial(n, 0.5)$

- Test statistics: x_0 the observed number of positive signs
  - $H_a : \theta > \theta_0$, then p-value$= p_+ = P(X \geq x_0)$
  - $H_a : \theta < \theta_0$, then p-value$= p_- = P(X \leq x_0)$
  - $H_a : \theta \neq \theta_0$, then p-value$= 2min(p_+, p_-)$

Sign Test

- Example: $H_0 : \theta = 18$ vs $H_a : \theta > 18$
- Function `binom_test()` in `scipy` can be used.
  - https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.binomtest.html

```
from scipy.stats import binomtest
m1 = sum(mtcars.mpg>18)
m2 = sum(mtcars.mpg<18)
m=m1+m2
test=binomtest(k=m1, n=m, p=0.5, alternative="greater")
test.statistic
```

```
## 0.59375
```

```
test.pvalue
```

```
## 0.18854279373772442
```

Sign Test

- R code

```
#m1 = sum(ifelse(mtcars$mpg>18, 1, 0))
#m2 = sum(ifelse(mtcars$mpg<18, 1, 0))
m1 = sum(mtcars$mpg>18)
m2 = sum(mtcars$mpg<18)
m=m1+m2
binom.test(x=m1, n=m, p = 0.5, alternative = c("greater"))
```

Sign Test

- The package `statsmodels` `sign_test` can perform 1-sample sign test for us.
  - https://www.statsmodels.org/stable/generated/statsmodels.stats.descriptivestats.sign_test.html
  - But it's for 2-sided test only.

```
import statsmodels.stats.descriptivestats as smsd
test2=smsd.sign_test(mtcars['mpg'], mu0=18)
print(test2)
```

```
## (3.0, 0.37708558747544885)
```

Paired t-Test

- Two samples are said to be paired or matched samples when for each data value collected from one sample there is a corresponding data value collected from the second sample, and both these data values are collected from the same source.
  - $(x_{1i}, x_{2i}), i = 1, \ldots, n$
- We can eliminate unwanted variability in the experiment by analyzing only the differences $d_i = x_{1i} - x_{2i}, i = 1, \ldots, n$ to see if there is a difference in the two population means, $\mu_1 - \mu_2$.
- Thus, the two-sample inference problem is reduced to one-sample inference problem.

Paired t-Test

- Test statistic $t_0 = \dfrac{\bar{d} - 0}{s_d/\sqrt{n}}$. The p-value formulas follows the 1-sample t-test.
- Example with function `scipy.stats.ttest_rel()`
  - https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html

```
x1=[189, 173, 183, 180, 179]
x2=[170, 185, 175, 180, 178]
statistic,pvalue=stats.ttest_rel(x1,x2)
statistic
```

```
## 0.628297009258132 6
```

```
pvalue
```

```
## 0.563899355999518 9
```

Paired t-Test

- Or just conduct 1-sample t-test for the data of differences

```
import numpy as np
x1=np.array([189, 173, 183, 180, 179])
x2=np.array([170, 185, 175, 180, 178])
statistic,pvalue=stats.ttest_1samp(x1-x2,  popmean=0)
statistic
```

```
## 0.628297009258132 6
```

```
pvalue
```

```
## 0.5638993559995189
```

Paired t-Test

- R code

```
x1=c(189, 173, 183, 180, 179)
x2=c(170, 185, 175, 180, 178)
t.test(x1,x2, alternative='two.sided', paired=TRUE)
```

t-Test between Independent Groups

- Suppose there are two independent samples selected from two populations. The two data sets are of the form $x_1, \ldots, x_{n1}$ and $y_1, \ldots, y_{n2}$
- Let $X_1, \ldots, X_{n_1}$ be a random sample from a **normal population** with mean μ_1.Let $Y_1, \ldots, Y_{n_2}$ be a random sample from a **normal population** with mean μ_2. Assume $\sigma_1 = \sigma_2$. If the two samples are **independent**, then

$$\frac{(\overline{X} - \overline{Y}) - (\mu_1 - \mu_2)}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

follows the t-distribution with $df = n_1 + n_2 - 2$, where

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$

t-Test between Independent Groups

- If $\sigma_1 = \sigma_2$ is not true, then

$$\frac{(\overline{X} - \overline{Y}) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_x^2}{n_1} + \frac{s_y^2}{n_2}}}$$

follows the t-distribution with df determined by Welch-Scatterthwaite equation

$$df = \frac{\left(\frac{s_x^2}{n_1} + \frac{s_y^2}{n_2}\right)^2}{\frac{1}{n_1 - 1}\left(\frac{s_x^2}{n_1}\right)^2 + \frac{1}{n_2 - 1}\left(\frac{s_y^2}{n_2}\right)^2}$$

t-Test between Independent Groups

- Under $H_0 : \mu_1 - \mu_2 = 0$, the test statisitc is depending on our assumption.
- If $\sigma_1 = \sigma_2$,

$$t_0 = \frac{\overline{X} - \overline{Y}}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

- If $\sigma_1 = \sigma_2$ is not true,

$$t_0 = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{s_x^2}{n_1} + \frac{s_y^2}{n_2}}}$$

t-Test between Independent Groups

- `scipy.stats.ttest_ind()`: 2-sample t-test
  - https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html
- Example: $H_a : \mu_1 < \mu_2$

```
from scipy import stats
x1=[2,9,21,3,6,10,18,11,6,25,23,6,2,15,32,25,11,8,17,19,5,15,0,26]
x2=[17,6,13,12,13,33,59,10,7,63,9,25,36,15]
statistic,pvalue=stats.ttest_ind(x1,x2, equal_var=False, alternative
statistic
```

```
## -1.8101264569194846
```

```
pvalue
```

```
## 0.044216366759026016
```

t-Test between Independent Groups

- R code

```
x1=c(2,9,21,3,6,10,18,11,6,25,23,6,2,15,32,25,11,8,17,19,5,
15,0,26)
x2=c(17,6,13,12,13,33,59,10,7,63,9,25,36,15)
#var.test(x1, x2, alternative = "two.sided")
#Test of equality of variances
t.test(x1,x2, alternative='less', var.equal=FALSE)
```

Wilcoxon rank-sum test

- The Mann-Whitney(-Wilcoxon) test is a nonparametric test that uses ranks of sample data from two independent populations. It is also called Wilcoxon rank-sum test.
- There is no requirement that the data come from normal populations.
- It is used to test the null hypothesis that the two independent samples come from populations with equal medians, $H_0 : \theta_1 = \theta_2$.

Wilcoxon rank-sum test

- We select two independent random samples from each population. Designate each of the observations from population 1 as an "A" and each of the observations from population 2 as a "B".
- If H_0 is true, when we rank all the values in both samples from smallest to largest, the A's and B's should be randomly mixed in the rankings(with labels indicating which population an observation belongs to). In this case if we summed the ranks of the A measurements and the ranks of the B measurements, the sums would be similar.
- If the observations come from populations with two different medians, with population 1 lying to the left/right of population 2. In this case the sum of the ranks of the B observations would be larger/smaller than that for the A observations.

Wilcoxon rank-sum test

- `scipy.stats.ranksums()`: Compute the Wilcoxon rank-sum statistic for two samples.
  - https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ranksums.html
- Example

```
from scipy import stats
mpg0=mtcars[mtcars.am==0].mpg
mpg1=mtcars[mtcars.am==1].mpg
statistic,pvalue=stats.ranksums(mpg0,mpg1,alternative="two-sided")
statistic
```

```
## -3.1271073130324774
```

```
pvalue
```

```
## 0.0017653547692792252
```

Wilcoxon rank-sum test

- R code

```
mtcars$am=as.factor(mtcars$am)
wilcox.test(mpg ~ am, alternative="two.sided", data=mtcars)
```

Wilcoxon rank-sum test

- compared with t-test

```
statistic,pvalue=stats.ttest_ind(mpg0,mpg1, equal_var=False, alternative='t
statistic
```

```
## -3.767123145144923
```

```
pvalue
```

```
## 0.0013736383330710345
```

- R code

```
t.test(mpg ~ am, alternative="two.sided", data=mtcars)
```

Statistical Hypothesis Tests vs Statistical Modeling

- Most problems can be viewed from two perspectives
  - Statistical Hypothesis Tests: make a statistical hypothesis and verify or falsify that hypothesis
    - ⋆ Statistical Modeling: make a statistical model, and analyze the significance of the model parameters
- Let's use a classical t-test as an example
- Classical t-Test

```
import numpy as np
from scipy import stats
np.random.seed(123)
x1=np.round(np.random.randn(20)*10+90)
x2=np.round(np.random.randn(20)*10+85)
statistic,pvalue=stats.ttest_rel(x1,x2)
statistic
```

```
## 2.3040209271929544
```

```
pvalue
```

```
## 0.032682085532223897
```

Statistical Hypothesis Tests vs Statistical Modeling

- Statistical modeling: we assume that $\mu_1 - \mu_2$ is a constant and then $H_0 : \mu_1 - \mu_2 = 0$. This model has one parameter: the constant value.

```
import pandas as pd
import statsmodels.formula.api as sm
df=pd.DataFrame({"x1":x1, "x2":x2})
fit1=sm.ols(formula='I(x2-x1)~1', data=df).fit()
```

Statistical Hypothesis Tests vs Statistical Modeling

```
print(fit1.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:              I(x2 - x1)   R-squared:                       -0.000
## Model:                             OLS   Adj. R-squared:                  -0.000
## Method:                  Least Squares   F-statistic:                       nan
## Date:                 Mon, 13 Feb 2023   Prob (F-statistic):                nan
## Time:                         17:58:16   Log-Likelihood:                -85.296
## No. Observations:                   20   AIC:                             172.6
## Df Residuals:                       19   BIC:                             173.6
## Df Model:                            0
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept     -9.1000      3.950     -2.304      0.033     -17.367      -0.833
## ==============================================================================
## Omnibus:                        0.894   Durbin-Watson:                   2.009
## Prob(Omnibus):                  0.639   Jarque-Bera (JB):                0.793
## Skew:                           0.428   Prob(JB):                        0.673
## Kurtosis:                       2.532   Cond. No.                         1.00
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly
```

Statistical Hypothesis Tests vs Statistical Modeling

- R code

```
set.seed(1)
x1=round(rnorm(20)*10+90)
x2=round(rnorm(20)*10+85)
t.test(x1,x2, alternative = "two.sided")
fit2=lm(formula = I(x2-x1)~1)
summary(fit2)
```

License

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.