Data Analysis and Hypothesis Testing Using the Python ecosystem

t-Test & ANOVAs

Stavros Demetriadis
Assc. Prof., School of Informatics,
Aristotle University of Thessaloniki

sdemetri@csd.auth.gr
http://mlab.csd.auth.gr/sdemetri

Overview

- **D1.2 Statistical controls**
  - Descriptive statistics

- Test for differences between 2 groups
  - t-Test (independent samples)
  - Paired t-test
  - One sample t-test

- Test for differences between 3 or more groups
  - One way ANOVA
  - Write your own ANOVA function
  - Repeated measures ANOVA
  - Two way ANOVA
  - Analysis of Covariance (ANCOVA) *

Descriptive statistics

- Descriptive statistics refer to statistical measures that ‘describe’ the key features of the data distribution. Usually they provide information relevant to:
  - The **shape of the distribution** (i.e. the appearance frequency of individual values and a relevant bar chart plot)
  - The **central tendency** measures (mean, median and mode)
  - The **dispersion** (spread of the values, including variance and standard deviation)

Linking to Scipy

- 1. *Data distribution*
  - Frequency distribution → **value_counts()**
  - Plotting → Bar chart

- 2. *Central tendency*
  - **mean(), median(), std(), var(), min(), max(), skew(), kurt()**
  - **describe()**
  - Correlation: **corr()**

- 3. *Dispersion*
  - Standard deviation → **numpy.std()**
    - → **pandas.DataFrame.std()**

Review exercises

- Descriptive stats

- 1. Construct **rv** as a frozen normal distribution with M=2 and SD = 1.2. Then:

- A) Construct an array **ar1** with 20 randomly selected values from the 'rv' distribution. Following, construct a DataFrame **df1** with ar1 as dataframe values (no special indexes).

- B) Then construct an array **ar2** of shape (1,2) having as items the mean and standard deviation of the ar1 array. Following, construct a **df2** dataframe with ar2 as values (in a row) and column titles: '**Mean**' and '**SD**'

- C) Finally, use **to_excel()** method to save df1 andf2 dataframes in 'output.xlsx' file as follows:
  - Save df1 in spreadsheet with name 'data'
  - Save df2 in spreadsheet with name 'stats'

t-Test (independent samples)

$$t = \frac{M_1 - M_2}{s_p \sqrt{\frac{2}{n}}}$$

- **t-Test for independent samples** compares the mean values of two samples which are independent (performance of one sample does not affect the other)

- To implement independent samples t-test in Scipy you apply the **scipy.stats.ttest_ind** method

- Applying t-Test entails:
- A) Checking the data for the normality and variance criteria
- B) Applying the relevant Scipy method (scipy.stats.**ttest_ind**)
- C) Interpret the outcome

Paired t-Test

- **Paired t-test** is the type of t-test that we apply when we want *to explore whether the two means of two related samples are significantly different*.

- To implement paired t-test in Scipy you apply the **scipy.stats.ttest_rel** method

One sample t-Test

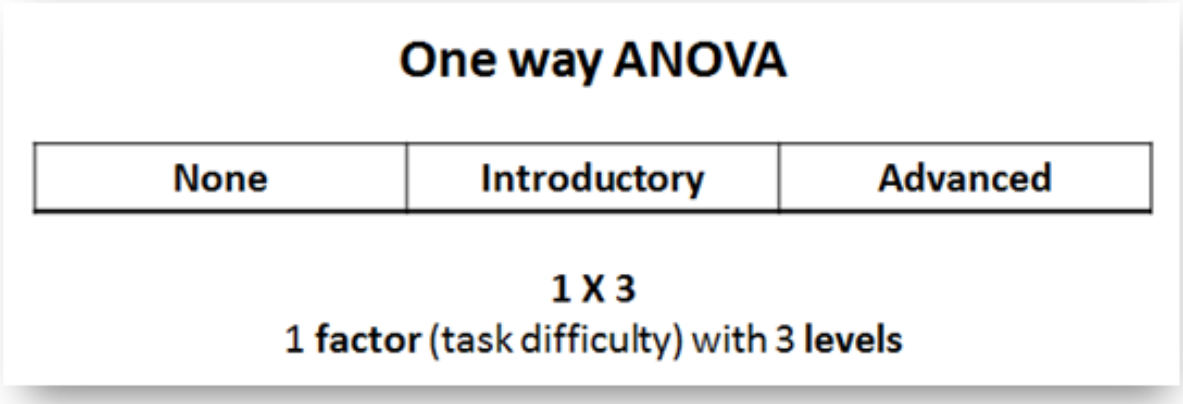
- **One sample t-test** is the type of t-test that we apply when we want *to explore whether the mean of our sample is significantly different from a specific value which usually is the known mean (and standard deviation) of the population*.

- The null hypothesis of a one sample t-test *assumes that there are no statistically significant differences* between the examined sample mean and the known population mean.

- To implement paired t-test in Scipy you apply the **scipy.stats.ttest_1samp** method

ANOVA

- **One way ANOVA** is a test for exploring **the impact of one single factor** on three or more groups



- To implement paired t-test in Scipy you apply the **scipy.stats.f_oneway** method

Review exercises

- t-Test

Independent samples

- 1. Write a function called '**stat_test()**' which implements t-test for independent samples as follows:

- A) The function accepts as input the dataframe **df**.

- B) The function examines whether df has **exactly two columns of numerical data** (use pandas.DataFrame.select_dtypes to check for non-numerical items). In any other case, the function prints out a relevant message and terminates execution.

- C) The function applies sequentially:
  - C.1 Normality & Variance criteria and reports (print out) the result
  - C.2 t-test for independent samples (reports t, p)
  - C.3 the above C.2 but with unequal variances assumed (reports also t, p)

  - *Check the operation of stat_test() passing data from the researchdata.xlsx file*

Paired samples

- 2. Extend the '**stat_test()**' function to apply *either t-test for independent samples or paired t-test* as follows:

- A) The function accepts now an additional *input parameter* in the form of a string with default value 'indep': **test = 'indep'**

- B) If the function is passed a value of : **test='indep'** (or no value at all) then ttest for independent samples is applied as before (see exercise 1, previous slide)

- C)) If the function is passed a value **test='paired'** then paired ttest is applied. The function applies sequentially:
  - C.1 Normality & Variance criteria and reports (print out) the result
  - C.2 t-test for paired samples (reports t, p)

  - *As before, check the operation of stat_test() passing data from the researchdata.xlsx file*

One sample

- 3. As before extend the '**stat_test()**' function to perform also one sample t-test when the input parameter is **test='1sample'**
  - *Similarly, check the operation of stat_test() passing data from the researchdata.xlsx file*

ANOVA

- 4. Extend **stat_test()** so that it performs also one way ANOVA, when the input parameter is **test='anova'**
  - *Similarly, check the operation of stat_test() passing data from the researchdata.xlsx file*