Midterm Review 2023 F

Course Outline

20/10/2023 15:31

By Yushu Zou <- Not A Detailed Oriented Person
Not 100% correct, trust lecture note if there exist conflict :)

- **Prepare Tips**
- **Data Wrangling**
- **Distributions**
- **Confidence Intervals**
- **Hypothesis Tests**
- **Python Code**

Prepare Tips

- Lecture Code & Demo Code
- Homework Problem
- Tutorial Quiz
- Practice Exam

Data Wrangling

- Import library
  `import FULL_NAME as SHORT_NAME`
- Axis : column-wise : axis = 1; row-wise: axis = 0
- `.loc` support boolean index, but `.iloc` does not
- `df.col` or `df['col']` or `df[('col')]` or `df[(('col'))]` give series; `df[['col']]` gives dataframe
- compond function: `&`(and), `|`(or), `==`(equal), `!=`(not equal)
  ex. `df.loc[(cond1)&(cond2), ('col1', 'col2')]`
- Possible Error
  - Forget import package before calling function package
  - Calling wrong package name: if `import FULL_NAME`, then use `FULL_NAME.func`; if `import FULL_NAME as SHORT_NAME`, then use `SHORT_NAME.func`

- Calling wrong function name from the package
- Calling Boolean select column when using `.iloc`

Distribution

Characteristics of a distribution

- Location/Center
  - Mean: `n=len(my_samp); my_samp.sum()/n` or `my_samp.mean()`
  - Median: `np.percentile(my_samp, 50)` or `sorted(my_samp)[int(n/2)]` or `np.quantile(my_samp, 0.5)`
  - Mode: `from collections import Counter Counter(my_samp).most_common()`
- Scale/Spread
  - Range: Maximun - Minimum `my_samp.max() - my_samp.min()`
  - IQR (for boxplot) `np.quantile(my_samp, 0.75) -np.quantile(my_samp, 0.25)`
  - Variance: `my_samp.var(ddof=1)`
  - Standard Deviation: `my_samp.std(ddof=1)`
- Skewness:
  - Left-Skewed (Mean < Median <Mode)
  - Right-Skewed (Mean > Median > Mode)
  - Symmetry (Mode = Median = Mean)
- Modality (Unimodal, Bimodal, Multimodal)

Data Type

Quantitative / Numerical

- Continuous: example: (52.4,23.5)
- Discrete: example(30, 50)

Qualitative / Categorical

- Binary : two level categorical example(Yes, No)
- Ordinal: ordered value, example (Monday, Tuesday,..)
- Nominal: factor example(Countries, Hair Color)

Data Visualization

Barplot

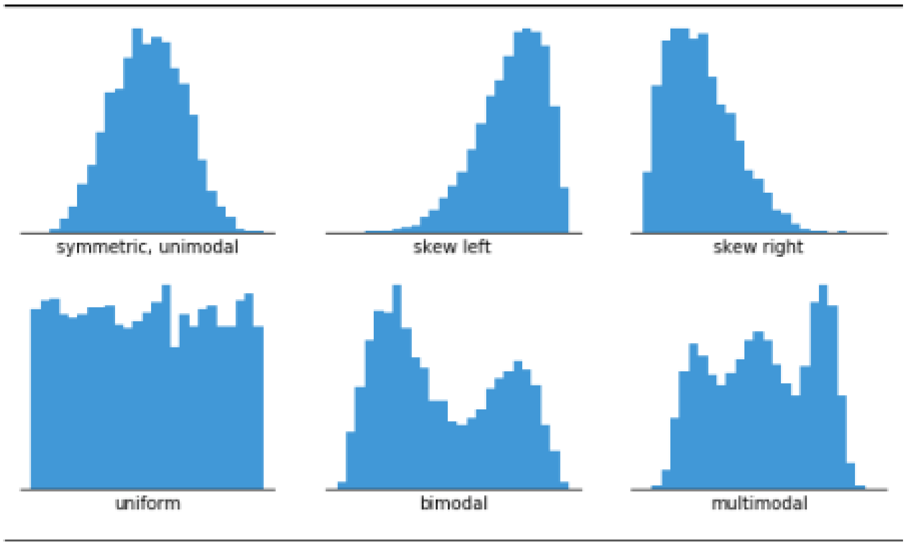
One bar for each **category variable**; order of the bars is arbitrary

- Bar plot can interpret the frequency of the categorical data **[Mode]**

Histogram

Height of each bar counts the number of values of the **numerical variable** in the corresponding bin

- Histogram can interpret **Skewness**, **Modality**



Boxplot

Summarizes the distribution of a numerical variable.

- Boxplot can interpret **Median**; **IQR (**$75^{th} - 25^{th}$**)**; **Outliers**, **Skewness**, etc
- We cannot interpret mean and variance from boxplot

Scatterplot

Each point is determined by the values of 2 **numerical variables**: one on x axis, the other on y axis

Sample Statistics

sample size = n

- Sample Mean : $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Sample Variance $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$
- Sample Standard Deviation: $s = \sqrt{s^2}$

Confidence Interval

There's a xxx% chance that this xxx% confidence interval construction procedure **captured the true parameter value**

The purpose of confidence interval is to obtain an estimate the parameter that reflects sampling variability.

A larger confidence level (e.g., instead of 95%, use 98%) would ensure that we capture the population parameter in more samples. This would give a wider confidence interval, extending to more of the bootstrap sampling distribution.

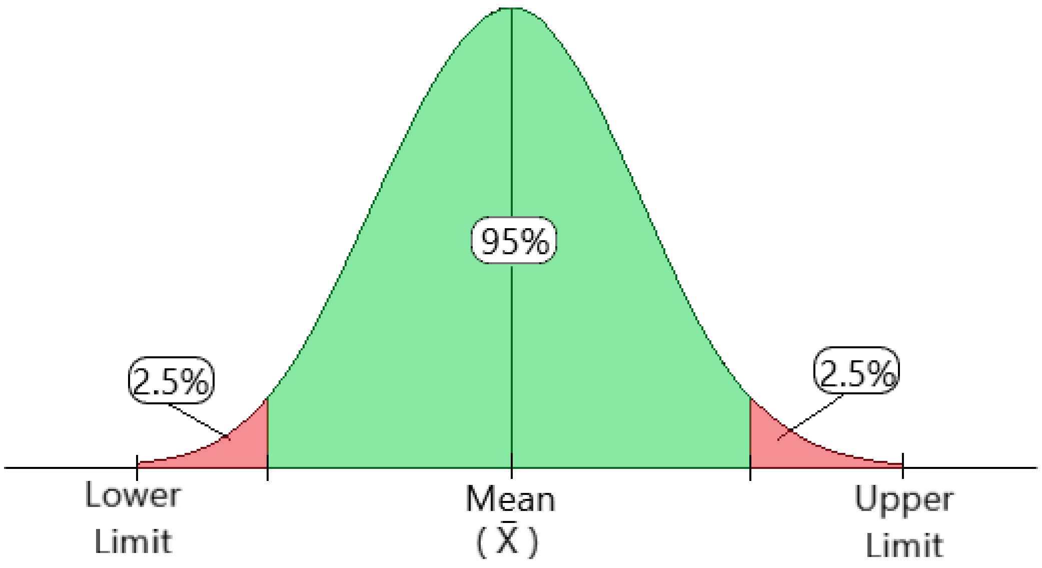
The probability of committing a type I error (rejecting the null when it is actually true) is called α; i.e.,the level of statistical significance. For 99% CI, α=0.01 (confidence level). This means that there is a 1% probability of making a type I error.

Bootstrapping

1. Obtain data sample $x_1, x_2, \ldots, x_n$ drawn from a distribution F.
2. Define u – statistic computed from the sample (mean, median, etc).
3. Sample $x_1^*, x_2^*, \ldots, x_n^*$ with replacement from the original data sample. Let it be F* – the empirical distribution. Repeat N times (N is bootstrap iterations).
4. Compute u* – the statistic calculated from each resample.

if you are still confuse about the concept https://www.youtube.com/watch?v=Xz0x-8-cgaQ

Bootstrapping --> Confidence Interval



```
# Create list variable to store the sample statistics
bootstrapped_means = []
# Start boostrapping, iteration
for i in range(reps):
# Drawing sample from the Sample with replacement <- independent sample
    bootstrap_sample = np.random.choice(x, n, replace = True)
# Calculate the sample statistics from the sample and append to the list
    bootstrapped_means += [bootstrap_sample.mean()]
```

```
# Constructing 95% Confidence Interval based on the sample statistics
np.quantile(bootstrapped_means,(0.025, .975))
```

Hypothesis Testing

A **statistic** is any function of a sample of data
p-value : The probability of a test statistic being as or more extreme than the observed test statistic if H_0 was true

Process

1. State null hypothesis H_0 and alternative hypothesis H_1: H_0 is FALSE
2. Choose the α-significance level at which H_0 will be rejected (for p-values smaller than α)
3. Simulate a chosen number of samples (e.g. 10,000) from the sampling distribution of the test statistic under the assumption that H_0 is true.

- for one-sample hypothesis test, the test statistic will be same as the statistic (mean, median, proportion ...) of the group
- for two-sample hypothesis test, the test statistic will be the difference between the statistic of the two groups
- for paired hypothesis test, the test statistic will be the

5. Compute the observed test statistic and the p-value of the observed test statistic relative to the above sampling distribution.
   - collect the proportion of cases that are more extreme than the oberseved statistics

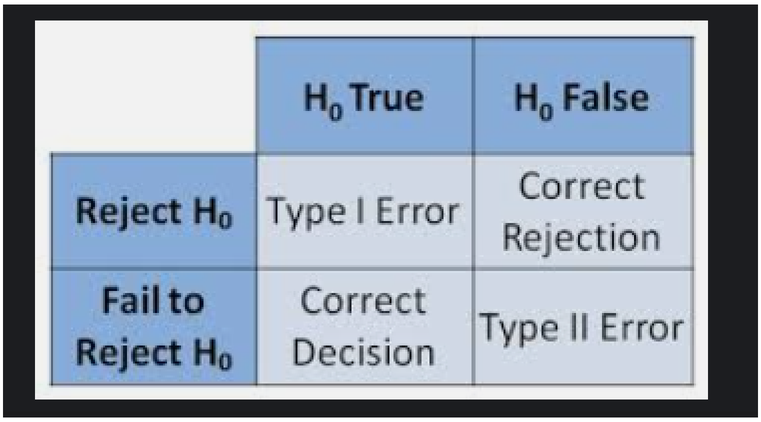
   ```
   (abs(simulated_statistics - null_hypothesis) >= abs(observed_statistics - null_hypothesis)).sum()/num_simulation
   ```

6. Conclusion: “Reject H_0 at α-significance level” if the p-value is less than α; Otherwise, “fail to reject H_0 at α-significance level”.

Type 1 and Type 2 Error

Type 1: Reject H_0 when H_0 is true

Type 2: Do not reject H_0 when H_0 is false



	H_0 True	H_0 False
Reject H_0	Type I Error	Correct Rejection
Fail to Reject H_0	Correct Decision	Type II Error

Python Code (Does not contain all the code Professor show on the lecture!)

import Numpy as np
import Pandas as pd

1. Reading csv file
   `pd.read_csv(.... FILENAME...)`
2. Finding variable type in dataframe
   `df.dtypes` or `type(...VARIABLE_NAME...)`
3. Finding the number of rows and columns in the dataframe
   `df.shape`
4. Summary variable in dataframe
   `df.describe()
5. Checking NULL value in the dataframe
   `df.isnull().sum()
6. Show subset of dataframe
   `df.head()` same as `df.iloc[:5,:]`
   <-- Return first 5 rows
7. Select columns from dataframe
   one column: `df['col']`
   multiple column: `df[['col1', 'col2']]`; `.iloc[,:]`; `df.loc[:, CONDITION]`
   based on condition : `df.loc[boolean_selection_column, ('col1', 'col2')]`
8. Group by and aggregation function
   `df .groupby('one column').aggregation function`
   aggregation function: `.sort_values(ascending=True/False)`; `.size()`; `.mean()`....
9. Drawing histogram/bar plot

```
import plotly.express as px
fig = px.histogram(df, x="col", nbins=n)
```

```
fig.show()
```

8. Drawing boxplot

```
fig = px.box(df, y="col") // y is numeric
// if want to create boxplot by categorical variable col2
fig = px.box(df, y="col1", x = "col2")
fig.show()
```