Comprehensive Lecture on Statistical Testing and Exploratory Data Analysis with Python

Introduction & Agenda

- The session began with technical checks and student attendance.

- Agenda covered: Chi-square test, ANOVA, application in Python, and exploratory data analysis (EDA) with coding demonstrations.

Chi-square Test

- Introduced as a non-parametric test to find relationships between categorical columns (e.g., gender and result).

- No distribution assumptions required.

- Explained the construction of a contingency table (e.g., gender vs. pass/fail counts).

- Demonstrated cross-tabulation in pandas for categorical data count aggregation.

- Hypothesis testing steps: null hypothesis (no relationship), alternate (relationship exists).

- Calculated totals, expected values using the formula: (Row Total * Column Total) / Grand Total.

- Step-by-step chi-square statistic calculation: sum of ((observed-expected)^2/expected) for all cells.

- Compared calculated value to tabulated value (for given significance level and degrees of freedom) to decide on rejecting the null hypothesis.

ANOVA (Analysis of Variance)

- Defined as a method to compare more than two groups.
- Types: one-way (one factor), two-way (two factors), N-way (N factors).
- Example problem: Comparing scores across three schools, explained null (no variation) versus alternate hypothesis (variation exists).
- Illustrated calculations: mean of columns, between-groups sum of squares (SSC), within-groups sum of squares (SSE), degrees of freedom.
- Calculation of mean squares and F-statistic (MSC/MSE), and interpretation via ANOVA table with Python.
- Comparison of calculated F with tabulated value to accept/reject hypotheses.

T-tests

- Examples of one-sample, two-sample (independent), and paired-sample t-tests.
- Key steps: hypothesis setup, calculation of t-statistic and p-value, and decision based on p-value (<0.05 to reject null).
- Demonstrated question-response on hypothesis creation and interpretation.

Python Implementation of Statistical Tests

- Importing relevant libraries (scipy, pandas, numpy, statsmodels).
- Conducting chi-square, t-test (one-sample, two-sample, paired), and ANOVA via function calls.

- Fetching and interpreting p-values and test statistics.

- DataFrame operations for analysis: creating/test tables, cross-tabs, aggregation, groupby mechanics.

- Error handling and practical caveats (e.g., categorical vs numerical aggregation).

Exploratory Data Analysis (EDA) in Python

- Loading data (pandas.read_csv), viewing with head/tail, accessing info and describe.

- Filtering, slicing, creating and modifying columns, sorting and indexing.

- Value counts and normalization for frequency analysis.

- Aggregations: sum, mean, count, standard deviation, custom calculations (e.g., premium = sold price - base price).

- Groupby operations with multiple columns and aggregation functions.

- Visualization: bar plots, count plots, box plots, scatterplots, KDE plots using seaborn/matplotlib.

- Extracting statistical properties (IQR, median, quartiles) from plots and DataFrames.

- Emphasis on data type correctness for proper analysis.

Practice, Assignments, and EDA Guidance

- Assignments set: replicate statistics and plots, create cross-tabulations, perform groupby/aggregate tasks, apply filtering, implement EDA with new datasets.

- Advice: Use documentation (e.g., pandas, matplotlib), practice coding for proficiency, focus on critical thinking in EDA (always start with questions), and practice both statistical and programming techniques.

- EDA seen as an open-ended, infinite loop iterative process; accuracy improves with questioning but is data-dependent.

Closing & Next Steps

- Encouraged student initiative and practice (with code and documentation review).

- Next scheduled topics: linear algebra and introduction to machine learning.

- Class concluded with remarks on practice importance and helping students access assignments and data.