

USER MANUAL FOR CLI STATISTICS CALCULATOR

Designed for users who want to run your CLI Statistical Calculator from start to finish with **zero confusion**.

Introduction

The CLI Statistical Calculator is a command-line application written in Python that reads numerical data from CSV files and computes:

- Descriptive statistics (mean, median, mode, std dev, etc.)
- Correlation analysis (Pearson)
- Hypothesis testing (t-tests, chi-square)
- Frequency distributions and text-based histograms
- Printable reports with interpretation guidelines

It supports error handling, object-oriented design, and robust input validation.

FEATURES

Module	Description
Descriptive Stats	Mean, median, mode, std dev, variance, skewness, kurtosis
Correlation	Pearson correlation between two variables
Hypothesis Testing	One-sample t-test, two-sample t-test, chi-square
Reports	Auto-generated reports saved as text
Error Handling	Graceful response to missing or bad input
Input Validation	Ensures only valid numerical columns are used
Unit Tests	test_all.py validates each module independently

Installation Requirements

To run this project, you need:

Python 3.8+

Install required libraries using pip:

pip install pandas numpy scipy

File Structure:

project/

- cli_calculator.py
- stats_module.py
- correlation_module.py
- hypothesis_module.py
- file_handler.py
- report_writer.py
- test_suite/ test_all.py
- sample_data/ input_data.csv
- results/ analysis_report.txt

How to run the app

From the terminal, navigate to your project folder and run:

python cli_calculator.py

You'll be prompted with options like:

- Enter path to CSV file
- Choose a numeric column
- Select statistical analysis type
- Choose to export report or not

Using Descriptive Statistics

Sample input:

Enter column for analysis: MathScore

Choose operation: mean, median, std_dev, etc.

Sample output:

Mean: 77.3

Standard Deviation: 5.4

Skewness: -0.23

You will also get an option to:

Save full analysis report to results/analysis_report.txt?

Correlation Analysis**You'll be asked:**

Enter first column (e.g., MathScore):

Enter second column (e.g., EnglishScore):

Result:

Pearson Correlation Coefficient: 0.88 → Strong Positive Correlation

Hypothesis testing**One-sample t-test**

Test if 'MathScore' has mean = 75

Result: $t = 1.34$, $p = 0.21$ → Fail to reject null hypothesis

Two-sample t-test:

Compare 'MathScore' and 'EnglishScore'

Result: t = -1.12, p = 0.29

Chi-square test:

Test gender distribution in 'Gender' column

Chi² = 0.4, p = 0.52 → No significant difference

Frequency Distribution and Histograms

Outputs:

70.0: 1 72.0: 2 75.0: 3

Histogram: 70.0:  72.0:  75.0: 


Useful for visualizing score trends and data distribution directly in terminal.

Understanding the Output

Metric	Interpretation
Mean	Average
Std Dev	Spread of Data
Skewness	Asymmetry of distribution
Kurtosis	Heaviness of tails
Correlation	Strength of linear relationship
P-value	< 0.05 → Statistically significant

Sample Report

Auto-generated text file:

 results/analysis_report.txt

Includes:

- Date & time
- Summary statistics
- Frequency table
- Text-based histogram
- Interpretation prompts

Error and Troubleshooting

Problem	Cause	Solution
FileNotFoundError	Wrong path	Double-check path to CSV
ValueError	Non-numeric column	Choose only numeric columns
ZeroDivisionError	Insufficient data	Use at least 2 values
TypeError	Wrong input format	Enter valid numbers only

Unit Testing

Run all tests:

`python -m unittest test_suite/test_all.py`

Confirms the core logic is:

- Mathematically accurate
- Stable under edge cases
- Valid for skew/kurtosis interpretation

Best Practices

- Clean CSVs before using
- Always include header row
- Use lowercase or CamelCase column names (no special characters)
- Validate large datasets in chunks

Credits and Contracts

Python Version: 3.10+

Dependencies: pandas, numpy, scipy