# Statistics Library Documentation

Author: Vishanth Hari Raj

GitHub Repository: https://github.com/vishanth10/statistics\_library.git

## Function Documentation

### 1. read\_data

The read\_data function reads data from either a DataFrame or a file path. If the input is a DataFrame, it makes a copy. If it's a file path, it checks the file extension (.xls, .xlsx, .csv) and reads the data accordingly. It then applies pandas' to\_numeric function to convert the data to numeric types, with non-numeric values coerced to NaN.

Usage Example:

data = read\_data('path\_to\_file.csv') # For CSV files  
data = read\_data('path\_to\_file.xlsx') # For Excel files  
data = read\_data(dataframe) # For existing DataFrame

### 2. compare\_dimensions

This function compares the dimensions of the observed and expected datasets. If the dimensions don't match, it raises a ValueError. This ensures that further statistical calculations are valid.

Usage Example:

compare\_dimensions(observed\_data, expected\_data) # Checks if dimensions of observed and expected data match

### 3. calculate\_chi\_squared

Calculates the chi-squared statistic from observed and expected datasets. This is done by summing the squared differences between observed and expected values, divided by the expected values.

Usage Example:

chi\_squared = calculate\_chi\_squared(observed\_data, expected\_data) # Returns the chi-squared statistic

### 4. calculate\_expected

Computes expected frequencies for a contingency table based on the observed data. It uses row and column sums to calculate these frequencies, assuming independence between rows and columns.

Usage Example:

expected\_data = calculate\_expected(observed\_data) # Returns the expected frequencies

### 5. calculate\_chi\_abs

This function calculates the chi absolute statistic for observed and expected data. If expected data is not provided, it calculates the expected data based on the observed data.

Usage Example:

chi\_abs = calculate\_chi\_abs(observed\_data, expected\_data) # Calculates the chi absolute statistic

### 6. chi\_abs\_stat

A wrapper function for calculate\_chi\_abs. It allows the user to provide only observed data, with an option to provide expected data. It handles dimension comparison and error logging.

Usage Example:

chi\_abs = chi\_abs\_stat(observed\_data, expected\_data) # Calculates chi absolute statistic with optional expected data

### 7. calculate\_p\_value

Calculates the p-value from a chi-squared statistic and degrees of freedom using the chi-squared survival function.

Usage Example:

p\_value = calculate\_p\_value(chi\_squared, dof) # Returns the p-value

### 8. chi\_squared\_stat

Reads observed and expected data, checks their dimensions, calculates the chi-squared statistic, and then computes the p-value.

Usage Example:

chi\_squared\_value = chi\_squared\_stat(observed\_data, expected\_data) # Calculates chi-squared statistic

### 9. p\_value\_stat

Reads observed and expected data, checks dimensions, calculates the chi-squared statistic, and then computes the p-value.

Usage Example:

p\_value = p\_value\_stat(observed\_data, expected\_data) # Calculates the p-value for chi-squared statistic

### 10. convert\_df\_to\_numpy

Converts DataFrame data to numpy arrays for use in functions, particularly for bootstrapping. It returns a tuple of numpy arrays for observed and expected data.

Usage Example:

observed\_array, expected\_array = convert\_df\_to\_numpy(df\_observed, df\_expected) # Converts DataFrame to numpy arrays

### 11. bootstrap\_chi\_abs\_distribution

Generates a bootstrap distribution of the chi absolute statistic for an n\*n contingency table. Simulates new datasets and calculates the chi absolute for each, returning an array of these statistics.

Usage Example:

simulated\_chi\_abs = bootstrap\_chi\_abs\_distribution(observed\_data) # Returns array of simulated chi absolute statistics

### 12. calculate\_p\_value\_bootstrap

Calculates the p-value for the chi absolute statistic using bootstrap methods. Compares the observed chi absolute statistic against the distribution of simulated chi absolute values to compute the p-value.

Usage Example:

p\_value = calculate\_p\_value\_bootstrap(observed\_chi\_abs, simulated\_chi\_abs) # Calculates p-value using bootstrap method

### 13. plot\_chi\_abs\_distribution

Plots the distribution of simulated chi absolute values along with the observed chi absolute value. Shows the calculated p-value, providing a visual representation of the statistical analysis.

Usage Example:

plot\_chi\_abs\_distribution(simulated\_data, observed\_data, p\_value) # Plots distribution of chi absolute values

**APPENDIX - I**

## bootstrap\_chi\_abs\_distribution :

The bootstrap\_chi\_abs\_distribution function is designed for generating a bootstrap distribution of the chi absolute statistic for an n\*n contingency table. This is crucial in statistical analysis, especially when assessing the significance of observed frequencies against expected frequencies in categorical data.

### Parameters:

- observed\_data: Observed frequencies in an n\*n contingency table, either a numpy array or pandas DataFrame.  
- num\_simulations: Number of bootstrap samples to generate.  
- with\_replacement: Boolean indicating whether sampling should be with replacement.

### Process:

1. Convert DataFrame to Numpy Array: If observed\_data is a DataFrame, it's converted to a numpy array.  
2. Determine Dimensions and Calculate Expected Frequencies: Calculates total rows, total columns, and expected frequencies.  
3. Initialize Results Array: An array to store chi absolute statistics for each simulation.  
4. Calculate Total Counts Per Column: Computes the total count of observations per column.  
5. Create Pooled Data Array: Concatenates all categories for resampling.  
6. Bootstrap Simulation Loop: Iterates num\_simulations times, creating and analyzing simulated datasets.  
7. Calculate Chi Absolute Statistic: Computes chi absolute statistic for each simulated dataset.  
8. Return Results: Returns the array containing chi absolute statistics from all simulations.

### Usage:

This function is used in statistical hypothesis testing to assess the significance of the observed data. By comparing the observed chi absolute statistic to a distribution generated through bootstrapping, one can infer the likelihood of observing such a statistic under the null hypothesis.

# Extended:- Relative Risk Analysis Documentation

This documentation provides an overview and usage guide for a set of Python functions designed to calculate the relative risk between two treatments, resample data for statistical analysis, calculate confidence intervals, and plot the distribution of relative risks. These functions are intended for statistical analysis in lifesciences research or any field requiring comparative risk assessment.

## 1. calculate\_relative\_risk\_two\_treatments

Calculates the relative risk (RR) of an event occurring between two treatments. RR is a measure of the strength of association between an exposure and an outcome.

Parameters:

- observed\_data: 2D array of observed data.  
- event\_row\_index: Index of the row corresponding to the event.  
- treatment1\_index: Column index for the first treatment.  
- treatment2\_index: Column index for the second treatment.

Logic:

The function sums the total occurrences for each treatment and calculates the probability of the event for each treatment. The relative risk is then the ratio of these probabilities.

Usage Example:

relative\_risk = calculate\_relative\_risk\_two\_treatments(observed\_data, 0, 1, 2)

## 2. resample\_and\_calculate\_rr

Performs resampling to calculate a distribution of relative risks through bootstrapping.

Parameters:

- observed\_data: 2D array of observed data.  
- event\_row\_index: Index of the event row.  
- reference\_treatment\_index: Index for the reference treatment column (default is 0).  
- num\_simulations: Number of bootstrap simulations to perform.

Logic:

Generates simulated datasets by resampling from the observed data, maintaining the original proportions. Calculates the relative risk for each simulated dataset to create a distribution of relative risks.

Usage Example:

simulated\_rr = resample\_and\_calculate\_rr(observed\_data, 0, num\_simulations=10000)

## 3. calculate\_confidence\_interval

Calculates the 95% confidence interval for the relative risk from a distribution of simulated relative risks.

Parameters:

- simulated\_rr (np.array): Array of simulated relative risks.  
- percentile (float, optional): The percentile for the confidence interval, defaulting to 95%.

Logic:

The function calculates the lower and upper bounds of the confidence interval based on the specified percentile using numpy's percentile function. This provides an estimate of the interval within which the true relative risk is expected to lie with 95% certainty.

Usage Example:

lower\_bound, upper\_bound = calculate\_confidence\_interval(simulated\_rr, 95)

## 4. calculate\_probabilities\_for\_each\_treatment

Calculates the probability of an event occurring for each treatment group.

Parameters:

- observed\_data (np.array or pd.DataFrame): The observed data.  
- event\_row\_index (int): The row index for the event.

Logic:

For each treatment group, the function calculates the total counts and then determines the probability of the event. These probabilities are stored in a dictionary, providing a quick reference for each treatment's event probability.

Usage Example:

probabilities = calculate\_probabilities\_for\_each\_treatment(observed\_data, 0)

## 5. plot\_relative\_risk\_distribution

Plots the distribution of simulated relative risks against the observed relative risk, including confidence intervals.

Parameters:

- simulated\_rr (np.array): Array of simulated relative risks.  
- observed\_rr (float): The observed relative risk.

Logic:

This function visualizes the distribution of simulated relative risks, highlighting the observed relative risk and marking the confidence intervals. It uses seaborn and matplotlib for plotting.

Usage Example:

plot\_relative\_risk\_distribution(simulated\_rr, observed\_rr)

**APPENDIX II**