Statastical

`Statastic` is a Typst library designed to provide various statistical functions for numerical data. It offers functionalities like extracting specific columns from datasets, converting array elements to different data types, and computing various statistical measures such as average, median, mode, variance, standard deviation, and percentiles.

All functions

arrayStats

Computes a set of statistical measures for an array. Includes: average, median, integer mode, variance, standard deviation, and some percentiles.

Parameters

```
arrayStats(arr: array) -> dictionary
```

arr `array`

Array of numbers.

stats

Computes a set of statistical measures for a specific column in a dataset. Includes: average, median, integer mode, variance, standard deviation, and some percentiles.

Parameters

```
stats(
  data: array,
  colId: int
) -> dictionary
```

data `array`

The dataset.

colId `int`

The identifier for the column.

extractColumn

Extracts a specific column from the given dataset based on the column.

Parameters

```
extractColumn(
  data: array,
  colId: int
) -> array
```

data `array`

The dataset.

colId `int`

The identifier for the column to be extracted.

tofloatArray

Converts an array's elements to floating point numbers.

Parameters

tofloatArray(arr: `array`) -> `array`

arr `array`

Array with elements to be converted.

toIntArray

Converts an array's elements to integers.

Parameters

toIntArray(arr: `array`) -> `array`

arr `array`

Array with elements to be converted.

isInt

Determines if a given value is an integer.

Parameters

isInt(val: `mixed`) -> `boolean`

val `mixed`

The value to be checked.

lerp

Calculates a value between two numbers at a specific fraction.

Parameters

```
lerp(
  lower: float,
  upper: float,
  fraction: float
) -> float
```

lower `float`

The lower number.

upper `float`

The upper number.

fraction `float`

The fraction between the two numbers.

arrayAvg

Calculates the average of an array’s elements.

Parameters

```
arrayAvg(arr: array) -> float
```

arr `array`

Array of numbers.

avg

Calculates the average of a specific column in a dataset.

Parameters

```
avg(
  data: array,
  colId: int
) -> float
```

data `array`

The dataset.

colId `int`

The identifier for the column.

arrayMedian

Calculates the median of an array's elements.

Parameters

```
arrayMedian(arr: array) -> float
```

arr `array`

Array of numbers.

median

Calculates the median of a specific column in a dataset.

Parameters

```
median(
  data: array,
  colId: int
) -> float
```

data `array`

The dataset.

colId `int`

The identifier for the column.

arrayIntMode

Calculates the mode of an integer array. Converts all floats to integers.

Parameters

```
arrayIntMode(arr: array) -> array
```

arr `array`

Array of integers.

intMode

Calculates the integer mode of a specific column in a dataset. Converts all floats to integers.

Parameters

```
intMode(
  data: array,
  colId: int
) -> array
```

data `array`

The dataset.

colId `int`

The identifier for the column.

arrayVar

Calculates the variance of an array's elements.

Parameters

```
arrayVar(arr: array) -> float
```

arr `array`

Array of numbers.

var

Calculates the variance of a specific column in a dataset.

Parameters

```
var(
  data: array,
  colId: int
) -> float
```

data `array`

The dataset.

colId `int`

The identifier for the column.

arrayStd

Calculates the standard deviation of an array's elements.

Parameters

```
arrayStd(arr: array) -> float
```

arr `array`

Array of numbers.

std

Calculates the standard deviation of a specific column in a dataset.

Parameters

```
std(
  data: array,
  colId: int
) -> float
```

data `array`

The dataset.

colId `int`

The identifier for the column.

arrayPercentile

Calculates a specific percentile of an array's elements.

Parameters

```
arrayPercentile(
  arr: array,
  p: float
) -> float
```

arr `array`

Array of numbers.

p `float`

The desired percentile (between 0 and 1).

percentile

Calculates a specific percentile of a column in a dataset.

Parameters

```
percentile(
  data: array,
  colId: int,
  p: float
) -> float
```

data `array`

The dataset.

colId `int`

The identifier for the column.

p `float`

The desired percentile (between 0 and 1).