

Python Reference

Created by Nishant Kheterpal and Jessica Hu

Modified by Yanay Rosen

Table Functions and Methods

In the examples in the left column, `np` refers to the NumPy module, as usual. Everything else is a function, a method, an example of an argument to a function or method, or an example of an object we might call the method on. For example, `tbl` refers to a table, `array` refers to an array, and `num` refers to a number. `array.item(0)` is an example call for the method `item`, and in that example, `array` is the name previously given to some array.

Name	Chapter	Description	Input	Output
<code>Table()</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	Create an empty table, usually to extend with data	None	An empty Table
<code>Table().read_table(filename)</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	Create a table from a data file	string : the name of the file	Table with the contents of the data file
<code>tbl.with_columns(name, values)</code> <code>tbl.with_columns(n1, v1, n2, v2,...)</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	A table with an additional or replaced column or columns. <code>name</code> is a string for the name of a column, <code>values</code> is an array	1. string : the name of the new column; 2. array : the values in that column	Table : a copy of the original Table with the new columns added
<code>tbl.column(column_name_or_index)</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	The values of a column (an array)	string or int : the column name or index	array : the values in that column
<code>tbl.row(i)</code>	17.3 (https://inferentialthinking.com/chapters/17/3/Rows_of_Tables.html)	Returns a row object from the table	int : the index of the row to return	row : a row object
<code>tbl.num_rows</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	Compute the number of rows in a table	None	int : the number of rows in the table
<code>tbl.num_columns</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	Compute the number of columns in a table	None	int : the number of columns in the table

Name	Chapter	Description	Input	Output
<code>tbl.labels</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	Lists the column labels in a table	None	array: the names of each column (as strings) in the table
<code>tbl.select(col1, col2, ...)</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	Create a copy of a table with only some of the columns. Each column is the column name or index.	string or int: column name(s) or index(es)	Table with the selected columns
<code>tbl.drop(col1, col2, ...)</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	Create a copy of a table without some of the columns. Each column is the column name or index.	string or int: column name(s) or index(es)	Table without the selected columns
<code>tbl.relabeled(old_label, new_label)</code>	6 (https://www.inferentialthinking.com/chapters/06/Tables.html)	Modifies the existing table <i>in place</i> , changing the column heading in the first argument to the second	1. string: the old column name 2. string: the new column name	Table: a copy of the original with the changed label
<code>tbl.show(n)</code>	6.1 (https://www.inferentialthinking.com/chapters/06/1/Sorting_Rows.html)	Display <code>n</code> rows of a table. If no argument is specified, defaults to displaying the entire table.	(Optional) int: number of rows you want to display	None: displays a table with <code>n</code> rows
<code>tbl.sort(column_name_or_index)</code>	6.1 (https://www.inferentialthinking.com/chapters/06/1/Sorting_Rows.html)	Create a copy of a table sorted by the values in a column. Defaults to ascending order unless <code>descending = True</code> is included.	1. string or int: column index or name 2. (Optional) <code>descending = True</code>	Table: a copy of the original with the column sorted
<code>tbl.where(column, predicate)</code>	6.2 (https://www.inferentialthinking.com/chapters/06/2/Selecting_Rows.html)	Create a copy of a table with only the rows that match some <i>predicate</i> . See <code>Table.where</code> predicates below.	1. string or int: column name or index 2. <code>are(...)</code> predicate or array of boolean values corresponding to rows to include	Table: a copy of the original table with only the rows that match the predicate
<code>tbl.take(row_indices)</code>	6.2 (https://www.inferentialthinking.com/chapters/06/2/Selecting_Rows.html)	A table with only the rows at the given indices. <code>row_indices</code> is either an array of indices or an integer corresponding to one index.	array of ints: the indices of the rows to be included in the Table OR int: the index of the row to be included	Table: a copy of the original table with only the rows at the given indices

Name	Chapter	Description	Input	Output
<code>tbl.scatter(x_column, y_column, group=)</code>	7 (https://www.inferentialthinking.com/chapters/07/Visualization.html)	Draws a scatter plot consisting of one point for each row of the table. Note that <code>x_column</code> and <code>y_column</code> must be strings specifying column names.	1. string : name of the column on the x-axis 2. string : name of the column on the y-axis 3. string : name of categorical column to color points by.	None: draws a scatter plot
<code>tbl.plot(x_column, y_column)</code>	7 (https://www.inferentialthinking.com/chapters/07/Visualization.html)	Draw a line graph consisting of one point for each row of the table. Draws different color lines for each other numerical column if only the x axis label is specified.	1. string : name of the column on the x-axis 2. string : name of the column on the y-axis.	None: draws a line graph
<code>tbl.barh(categories)</code> <code>tbl.barh(categories, values)</code>	7.1 (https://www.inferentialthinking.com/chapters/07/1/Visualizing_Categorical_Distributions.html)	Displays a bar chart with bars for each category in a column, with height proportional to the corresponding frequency. If the values argument is not specified, bars will be drawn with different colors for each other column in the table.	1. string : name of the column with categories 2. (Optional) string : the name of the column with values for corresponding categories	None: draws a bar chart
<code>tbl.hist(column, unit=, bins=, group=)</code>	7.2 (https://www.inferentialthinking.com/chapters/07/2/Visualizing_Numerical_Distributions.html)	Generates a histogram of the numerical values in a column. <code>unit</code> and <code>bins</code> are optional arguments, used to label the axes and group the values into intervals (bins), respectively. Bins have the form <code>[a, b)</code> , where <code>a</code> is included in the bin and <code>b</code> is not.	1. string : name of the column with categories 2. (Optional) string : units of x-axis 3. (Optional) array of ints/floats denoting bin boundaries 4. string : name of categorical column to draw separate overlaid histograms for.	None: draws a histogram

Name	Chapter	Description	Input	Output
<code>tbl.bin(column, bins=)</code>	7.2 (https://www.inferentialthinking.com/chapters/07/2/Visualizing_Numerical_Distributions.html)	Generates a binned table for the numerical values in a column. <code>bins</code> is an optional arguments, used to group the values into intervals (bins). Bins have the form <code>[a, b)</code> , where <code>a</code> is included in the bin and <code>b</code> is not.	1. string : name of the column with categories 2. (Optional) array of ints/floats denoting bin boundaries.	Table : a table with the columns <code>bin</code> , containing the left ends of each bin, and <code>count</code> , containing the counts of rows that had <code>column</code> values fall into the corresponding bin.
<code>tbl.apply(function, col1, col2, ...)</code>	8.1 (https://www.inferentialthinking.com/chapters/08/1/Applying_a_Function_to_a_Column.html)	Returns an array of values resulting from applying a function to each item in a column.	1. function : function to apply to column 2. (Optional) string : name of the column to apply function to (if you have multiple columns, the respective column's values will be passed as the corresponding argument to the function), and if there is no argument, your function will be applied to every row in <code>tbl</code>	array : contains an element for each value in the original column after applying the function to it
<code>tbl.group(column_or_columns, func)</code>	8.2 (https://www.inferentialthinking.com/chapters/08/2/Classifying_by_One_Variable.html)	Group rows by unique values or combinations of values in a column(s). Multiple columns must be entered in array or list form. Other values aggregated by count (default) or optional argument <code>func</code> .	1. string or array of strings : column(s) on which to group 2. (Optional) function : function to aggregate values in cells (defaults to count)	Table : a new Table

Name	Chapter	Description	Input	Output
<code>tbl.pivot(col1, col2, values, collect)</code> <code>tbl.pivot(col1, col2)</code>	8.3 (https://www.inferentialthinking.com/chapters/08/3/Classifying_by_More_than_One_Variable.html)	A pivot table where each unique value in <code>col1</code> has its own column and each unique value in <code>col2</code> has its own row. Count or aggregate values from a third column, collect with some function. Default values and collect return counts in cells.	1. string : name of column whose unique values will make up columns of pivot table 2. string : name of column whose unique values will make up rows of pivot table 3. (Optional) string : name of column that describes the values of cell 4. (Optional) function : how the values are collected, e.g. <code>sum</code> or <code>np.mean</code>	Table : a new Table
<code>tblA.join(colA, tblB, colB)</code> <code>tblA.join(colA, tblB)</code>	8.4 (https://www.inferentialthinking.com/chapters/08/4/Joining_Tables_by_Columns.html)	Generate a table with the columns of <code>tblA</code> and <code>tblB</code> , containing rows for all values of a column that appear in both tables. Default <code>colB</code> is <code>colA</code> . <code>colA</code> and <code>colB</code> must be strings specifying column names.	1. string : name of column in <code>tblA</code> with values to join on 2. Table : other Table 3. (Optional) string : if column names are different between Tables, the name of the shared column in <code>tblB</code>	Table : a new Table
<code>tbl.sample(n)</code> <code>tbl.sample(n, with_replacement)</code>	10 (https://www.inferentialthinking.com/chapters/10/Sampling_and_Empirical_Distributions.html)	A new table where <code>n</code> rows are randomly sampled from the original table; by default, <code>n=tbl.num_rows</code> . Default is with replacement. For sampling without replacement, use argument <code>with_replacement=False</code> . For a non-uniform sample, provide a third argument <code>weights=distribution</code> where <code>distribution</code> is an array or list containing the probability of each row.	1. int : sample size 2. (Optional) <code>with_replacement=True</code>	Table : a new Table with <code>n</code> rows

Name	Chapter	Description
<code>str.split(separator)</code>	N/A	Splits the string (<code>str</code>) into a list based on the <code>separator</code> that is passed in
<code>str.join(array)</code>	N/A	Combines each element of <code>array</code> into one string, with <code>str</code> being in-between each element
<code>str.replace(old_string, new_string)</code>	4.2.1 (https://www.inferentialthinking.com/chapters/04/2/1/String_Methods.html)	Replaces each occurrence of <code>old_string</code> in <code>str</code> with the value of <code>new_string</code>

Array Functions and Methods

Name	Chapter	Description
<code>max(array)</code>	3.3 (https://www.inferentialthinking.com/chapters/03/3/Calls.html)	Returns the maximum value of an array
<code>min(array)</code>	3.3 (https://www.inferentialthinking.com/chapters/03/3/Calls.html)	Returns the minimum value of an array
<code>sum(array)</code>	3.3 (https://www.inferentialthinking.com/chapters/03/3/Calls.html)	Returns the sum of the values in an array
<code>abs(num)</code> , <code>np.abs(array)</code>	3.3 (https://www.inferentialthinking.com/chapters/03/3/Calls.html)	Take the absolute value of number or each number in an array.
<code>round(num)</code> , <code>np.round(array)</code>	3.3 (https://www.inferentialthinking.com/chapters/03/3/Calls.html)	Round number or array of numbers to the nearest integer.
<code>len(array)</code>	3.3 (https://www.inferentialthinking.com/chapters/03/3/Calls.html)	Returns the length (number of elements) of an array
<code>make_array(val1, val2, ...)</code>	5 (https://www.inferentialthinking.com/chapters/05/Sequences.html)	Makes a numpy array with the values passed in
<code>np.average(array)</code> <code>np.mean(array)</code>	5.1 (https://www.inferentialthinking.com/chapters/05/1/Arrays.html)	Returns the mean value of an array
<code>np.median(array)</code>	10.3 (https://inferentialthinking.com/chapters/10/3/Empirical_Distribution_of_a_Statistic.html#parameter)	Returns the median, 'middle', value of an array
<code>np.std(array)</code>	14.2 (https://www.inferentialthinking.com/chapters/14/2/Variability.html#Standard-Deviation)	Returns the standard deviation of an array
<code>np.diff(array)</code>	5.1 (https://www.inferentialthinking.com/chapters/05/1/Arrays.html)	Returns a new array of size <code>len(arr)-1</code> with elements equal to the difference between adjacent elements; <code>val_2 - val_1</code> , <code>val_3 - val_2</code> , etc.
<code>np.cumsum(array)</code>	5.1 (https://www.inferentialthinking.com/chapters/05/1/Arrays.html)	Returns a new array of size <code>len(arr)</code> with elements equal to the cumulative sum of previous elements; <code>val_1</code> , <code>val_1 + val_2</code> , <code>val_1 + val_2 + val_3</code> etc.
<code>np.sqrt(array)</code>	5.1 (https://www.inferentialthinking.com/chapters/05/1/Arrays.html)	Returns an array with the square root of each element

Name	Chapter	Description
np.arange(start, stop, step) np.arange(start, stop) np.arange(stop)	5.2 (https://www.inferentialthinking.com/chapters/05/2/Ranges.html)	An array of numbers starting with <code>start</code> , going up in increments of <code>step</code> , and going up to but excluding <code>stop</code> . When <code>start</code> and/or <code>step</code> are left out, default values are used in their place. Default step is 1; default start is 0.
array.item(index)	5.3 (https://www.inferentialthinking.com/chapters/05/3/More_On_Arrays.html)	Returns the i-th item in an array (remember Python indices start at 0!)
np.random.choice(array, n) np.random.choice(array) np.random.choice(array, n, replace)	9 (https://www.inferentialthinking.com/chapters/09/Randomness.html)	Picks one (by default) or some number 'n' of items from an array at random. Default is with replacement. For sampling without replacement, use argument <code>replace=False</code> .
np.count_nonzero(array)	9 (https://www.inferentialthinking.com/chapters/09/Randomness.html)	Returns the number of non-zero (or <code>True</code>) elements in an array.
np.append(array, item)	9.2 (https://www.inferentialthinking.com/chapters/09/2/Iteration.html)	Returns a copy of the input array with <code>item</code> (must be the same type as the other entries in the array) appended to the end.
percentile(percentile, array)	13.1 (https://www.inferentialthinking.com/chapters/13/1/Percentiles.html)	Returns the corresponding percentile of an array.

Table.where Predicates

Any of these predicates can be negated by adding `not_` in front of them, e.g. `are.not_equal_to(Z)` or `are.not_containing(S)` .

Predicate	Description
<code>are.equal_to(Z)</code>	Equal to <code>z</code>
<code>are.above(x)</code>	Greater than <code>x</code>
<code>are.above_or_equal_to(x)</code>	Greater than or equal to <code>x</code>
<code>are.below(x)</code>	Less than <code>x</code>
<code>are.below_or_equal_to(x)</code>	Less than or equal to <code>x</code>
<code>are.between(x,y)</code>	Greater than or equal to <code>x</code> and less than <code>y</code>
<code>are.between_or_equal_to(x,y)</code>	Greater than or equal to <code>x</code> , and less than or equal to <code>y</code>
<code>are.contained_in(A)</code>	Is a substring of <code>A</code> (if <code>A</code> is a string) or an element of <code>A</code> (if <code>A</code> is a list/array)
<code>are.containing(S)</code>	Contains the string <code>s</code>
<code>are.strictly_between(x,y)</code>	Greater than <code>x</code> and less than <code>y</code>

Miscellaneous Functions

These are functions in the `datascience` library that are used in the course that don't fall into any of the categories above.

Name	Chapter	Description	Input	Output
<code>sample_proportions(sample_size, model_proportions)</code>	11.1 (https://www.inferentialthinking.com/chapters/11/1/Assessing_Models.html)	<code>sample_size</code> should be an integer, <code>model_proportions</code> an array of probabilities that sum up to 1. The function samples <code>sample_size</code> objects from the distribution specified by <code>model_proportions</code> . It returns an array with the same size as <code>model_proportions</code> . Each item in the array corresponds to the proportion of times it was sampled out of the <code>sample_size</code> times.	1. int: sample size 2. array: an array of proportions that should sum to 1	array: each item corresponds to the proportion of times that corresponding item was sampled from model_proportions in sample_size draws, should sum to 1
<code>minimize(function)</code>	15.4 (https://www.inferentialthinking.com/chapters/15/4/Least_Squares_Regression)	Returns an array of values such that if each value in the array was passed into <code>function</code> as arguments, it would minimize the output value of <code>function</code> .	function: name of a function that will be minimized.	array: An array in which each element corresponds to an argument that minimizes the output of the function. Values in the array are listed based on the order they are passed into the function; the first element in the array is also going to be the first value passed into the function.