

Creating, Reading, Writing

`df = pd.DataFrame({"col0": [val0, val1], "col1": [val0, val1]}, index=[0, 1])` Create a dataframe

`series = pd.Series(["val0", "val1", "val2"], index=[0, 1, 2, 3], name="name")` Create a series

`read = pd.read_csv("../folder/folder/file.csv", index_col=0)` Read a csv

`save.to_csv("file.csv")` Save an existing dataframe as a csv

Indexing, Selecting, Assigning

`table.head` Show first 5 rows of a dataframe

`table["col"]` Select the col from table

`table.col.iloc[0]` Select 1st value of a col from table

`table.iloc[0]` Select 1st row of data from table

`table.col.iloc[:10]` Select 1st 10 values from col in table (index-based select)

`table.col.loc[:10]` Select 1st 10 values from col in table (label-based select)

`table.loc[indices, cols]` Select certain rows from certain cols

`table[table.col == 'val']` Select cols have a certain val (conditional select)

`table.col.isin(['val1', 'val2'])` Select cols have certain vals (conditional select)

Summary Functions & Maps

`table.col.describe()` Get high-lvl summary of given col's attributes

`table.col.mean()` Get mean of a col with numerical vals

`table.col.unique()` Get each unique val of a col w/ no dupes

`table.col.value_counts()` Get frequency of each val in col

Summary Functions & Maps (cont)

`table.col.map(lambda p: p - s)` Map function to remap a Series of point vals (p) by using a transformation (p-s) -> returns new Series

`table.apply(func, axis='col')` Apply function to transform entire df by calling custom method (func taking a row) on each row (umns')

Grouping & Sorting

`table.groupby('col').col.count()` Group data w/ same vals in the given col -> count frequency of given col (same as `value_counts()`)

`table.groupby('col').size()` Same as above

`table.groupby('col').apply(lambda df: df.title.iloc[0])` Select name (title) of the 1st thing in col

`table.col.idxmax()` Get index of max val in col

`table.groupby(['col0', 'col1']).col1.agg([f1, f2, f3])` `agg()` runs diff. funcs. simultaneously on a df

`table.groupby(['col0', 'col1']).col2.agg([len])` Multi-index output has tiered structure. Require 2 levels of labels to retrieve a val

`df.reset_index()` Multi-index method used to converting back to regular index

`df.sort_values(by='col')` Sort rows of data by vals in col (ascending)

`df.sort_values(by='col', ascending=False)` Sort rows of data by vals in col (descending)

`df.sort_values(by=['col0', 'col1'])` Sort rows by more than 1 col at a time

`df.sort_index()` Sort rows by index (default order; ascending)



By [blueberryshortcake](#)

Not published yet.

Last updated 25th June, 2024.

Page 1 of 2.

Sponsored by [CrosswordCheats.com](#)

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Data Types & Missing Values

<code>table.col.dtype</code>	Get data type of a col
<code>table.dtypes</code>	Get data types of each col in table
<code>table.col.astype('datatype')</code>	Convert col to datatype if allowed (e.g, int64 -> float64)
<code>table.index.dtype</code>	Number indices are int64
<code>table[pd.isnull(table.col)]</code>	Select NaN entries in a col
<code>table.col.fillna("filler")</code>	Replace all NaN vals in a col with a sentinel val ("Unknown", "Undisclosed", "Invalid") or non-null val
<code>table.col.replace("init_val", "new_val")</code>	Replace, in col, all existing vals with new_vals

Renaming & Combining

<code>table.rename(columns={'init': 'new'})</code>	Rename col or index col names
<code>table.rename(index={0: 'firstEntry', 1: 'secondEntry'})</code>	Rename index or col vals by specifying an index or col param
<code>table.rename_axis("name", axis='rows').rename_axis("name1", axis='columns')</code>	Rename row index &/or col index
<code>pd.concat(list, of, els)</code>	Smush together the list of elements along an axis
<code>left.join(right, lsuffix='strL', rsuffix='strR')</code>	Combine diff df objects that have an index in common. left and right are df.s defined beforehand



By [blueberryshortcake](#)

Not published yet.
Last updated 25th June, 2024.
Page 2 of 2.

Sponsored by [CrosswordCheats.com](#)
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>