Search the docs ...

Input/output
General functions
Series
DataFrame
pandas.DataFrame
pandas.DataFrame.index
pandas.DataFrame.columns
pandas.DataFrame.dtypes
pandas.DataFrame.select_dtypes
pandas.DataFrame.values
pandas.DataFrame.axes
pandas.DataFrame.ndim
pandas.DataFrame.size
pandas.DataFrame.shape
pandas.DataFrame.memory_usage
pandas.DataFrame.empty
pandas.DataFrame.astype
pandas.DataFrame.convert_dtypes
pandas.DataFrame.infer_objects
pandas.DataFrame.copy

pandas.DataFrame.dropna

`DataFrame.dropna`(*self, axis=0, how='any', thresh=None, subset=None, inplace=False*) [source]

Remove missing values.

See the User Guide for more on which values are considered missing, and how to work with missing data.

Parameters:

axis : *{0 or 'index', 1 or 'columns'}, default 0*

Determine if rows or columns which contain missing values are removed.

- 0, or 'index' : Drop rows which contain missing values.
- 1, or 'columns' : Drop columns which contain missing value.

Changed in version 1.0.0: Pass tuple or list to drop on multiple axes. Only a single axis is allowed.

how : *{'any', 'all'}, default 'any'*

Determine if row or column is removed from DataFrame, when we have at least one NA or all NA.

- 'any' : If any NA values are present, drop that row or column.
- 'all' : If all values are NA, drop that row or column.

thresh : *int, optional*

Require that many non-NA values.

subset : *array-like, optional*

Labels along other axis to consider, e.g. if you are dropping rows these would be a list of columns to include.

inplace : *bool, default False*

If True, do operation inplace and return None.

Returns:

DataFrame

DataFrame with NA entries dropped from it.

See also

`DataFrame.isna`
Indicate missing values.

`DataFrame.notna`
Indicate existing (non-missing) values.

`DataFrame.fillna`
Replace missing values.

`Series.dropna`
Drop missing values.

`Index.dropna`
Drop missing indices.

Examples

```
>>> df = pd.DataFrame({"name": ['Alfred', 'Batman', 'Catwoman'],
...                    "toy": [np.nan, 'Batmobile', 'Bullwhip'],
...                    "born": [pd.NaT, pd.Timestamp("1940-04-25"),
...                             pd.NaT]})
>>> df
       name        toy       born
0    Alfred        NaN        NaT
1    Batman  Batmobile 1940-04-25
2  Catwoman   Bullwhip        NaT
```

Drop the rows where at least one element is missing.

```
>>> df.dropna()
     name        toy       born
1  Batman  Batmobile 1940-04-25
```

Drop the columns where at least one element is missing.

```
>>> df.dropna(axis='columns')
       name
0    Alfred
1    Batman
2  Catwoman
```

Drop the rows where all elements are missing.

```
>>> df.dropna(how='all')
       name        toy       born
0    Alfred        NaN        NaT
1    Batman  Batmobile 1940-04-25
2  Catwoman   Bullwhip        NaT
```

Keep only the rows with at least 2 non-NA values.

```
>>> df.dropna(thresh=2)
       name        toy       born
1    Batman  Batmobile 1940-04-25
2  Catwoman   Bullwhip        NaT
```

Define in which columns to look for missing values.

```
>>> df.dropna(subset=['name', 'born'])
       name        toy       born
1    Batman  Batmobile 1940-04-25
```

Keep the DataFrame with valid entries in the same variable.

```
>>> df.dropna(inplace=True)
>>> df
     name        toy       born
1  Batman  Batmobile 1940-04-25
```

<< pandas.DataFrame.droplevel

pandas.DataFrame.duplicated >>