# pandas.DataFrame.agg

`DataFrame.`**`agg`**(*self*, *func*, *axis=0*, **args*, ***kwargs*) [source]

Aggregate using one or more operations over the specified axis.

*New in version 0.20.0.*

**Parameters:**

**func** : *function, str, list or dict*

Function to use for aggregating the data. If a function, must either work when passed a DataFrame or when passed to DataFrame.apply.

Accepted combinations are:

- function
- string function name
- list of functions and/or function names, e.g. `[np.sum, 'mean']`
- dict of axis labels -> functions, function names or list of such.

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

If 0 or 'index': apply function to each column. If 1 or 'columns': apply function to each row.

***args**

Positional arguments to pass to *func*.

****kwargs**

Keyword arguments to pass to *func*.

**Returns:**

scalar, Series or DataFrame

The return can be:

- scalar : when Series.agg is called with single function
- Series : when DataFrame.agg is called with a single function
- DataFrame : when DataFrame.agg is called with several functions

Return scalar, Series or DataFrame.

The aggregation operations are always performed over an axis, either the index (default) or the column axis. This behavior is different from *numpy* aggregation functions (*mean*, *median*, *prod*, *sum*, *std*, *var*), where the default is to compute the aggregation of the flattened array, e.g., `numpy.mean(arr_2d)` as opposed to `numpy.mean(arr_2d, axis=0)`.

*agg* is an alias for *aggregate*. Use the alias.

**See also:**

**DataFrame.apply**
: Perform any type of operations.

**DataFrame.transform**
: Perform transformation type operations.

**core.groupby.GroupBy**
: Perform operations over groups.

**core.resample.Resampler**
: Perform operations over resampled bins.

**core.window.Rolling**
: Perform operations over rolling window.

**core.window.Expanding**
: Perform operations over expanding window.

**core.window.EWM**
: Perform operation over exponential weighted window.

## Notes

*agg* is an alias for *aggregate*. Use the alias.

A passed user-defined-function will be passed a Series for evaluation.

## Examples

```
>>> df = pd.DataFrame([[1, 2, 3],
...                    [4, 5, 6],
...                    [7, 8, 9],
...                    [np.nan, np.nan, np.nan]],
...                   columns=['A', 'B', 'C'])
```

Aggregate these functions over the rows.

```
>>> df.agg(['sum', 'min'])
        A     B     C
sum  12.0  15.0  18.0
min   1.0   2.0   3.0
```

Different aggregations per column.

```
>>> df.agg({'A' : ['sum', 'min'], 'B' : ['min', 'max']})
        A    B
max   NaN  8.0
min   1.0  2.0
sum  12.0  NaN
```

Aggregate over the columns.

```
>>> df.agg("mean", axis="columns")
0    2.0
1    5.0
2    8.0
3    NaN
dtype: float64
```