**pyspark.pandas.DataFrame**

pyspark.pandas.DataFrame(data=None, index=None, columns=None, dtype=None, copy=False)

pandas-on-Spark DataFrame that corresponds to pandas DataFrame logically. This holds Spark DataFrame internally.

df2 = ps.DataFrame(np.random.randint(low=0, high=10, size=(5, 5)),

columns=['a', 'b', 'c', 'd', 'e'])

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Methods**   |  |  | | --- | --- | | [**abs**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.abs.html#pyspark.pandas.DataFrame.abs)**()** | Return a Series/DataFrame with absolute numeric value of each element. | | [**add**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.add.html#pyspark.pandas.DataFrame.add)**(other)** | Get Addition of dataframe and other, element-wise (binary operator *+*). | | [**add\_prefix**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.add_prefix.html#pyspark.pandas.DataFrame.add_prefix)**(prefix)** | Prefix labels with string *prefix*. | | [**add\_suffix**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.add_suffix.html#pyspark.pandas.DataFrame.add_suffix)**(suffix)** | Suffix labels with string *suffix*. | | [**agg**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.agg.html#pyspark.pandas.DataFrame.agg)**(func)** | Aggregate using one or more operations over the specified axis. | | [**aggregate**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.aggregate.html#pyspark.pandas.DataFrame.aggregate)**(func)** | Aggregate using one or more operations over the specified axis. | | [**align**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.align.html#pyspark.pandas.DataFrame.align)**(other[, join, axis, copy])** | Align two objects on their axes with the specified join method. | | [**all**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.all.html#pyspark.pandas.DataFrame.all)**([axis])** | Return whether all elements are True. | | [**any**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.any.html#pyspark.pandas.DataFrame.any)**([axis])** | Return whether any element is True. | | [**append**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.append.html#pyspark.pandas.DataFrame.append)**(other[, ignore\_index, …])** | Append rows of other to the end of caller, returning a new object. | | [**apply**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.apply.html#pyspark.pandas.DataFrame.apply)**(func[, axis, args])** | Apply a function along an axis of the DataFrame. | | [**applymap**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.applymap.html#pyspark.pandas.DataFrame.applymap)**(func)** | Apply a function to a Dataframe elementwise. | | [**assign**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.assign.html#pyspark.pandas.DataFrame.assign)**(\*\*kwargs)** | Assign new columns to a DataFrame. | | [**astype**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.astype.html#pyspark.pandas.DataFrame.astype)**(dtype)** | Cast a pandas-on-Spark object to a specified dtype dtype. | | [**at\_time**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.at_time.html#pyspark.pandas.DataFrame.at_time)**(time[, asof, axis])** | Select values at particular time of day (example: 9:30AM). | | [**backfill**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.backfill.html#pyspark.pandas.DataFrame.backfill)**([axis, inplace, limit])** | Synonym for *DataFrame.fillna()* or *Series.fillna()* with method=`bfill`. | | [**between\_time**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.between_time.html#pyspark.pandas.DataFrame.between_time)**(start\_time, end\_time[, …])** | Select values between particular times of the day (example: 9:00-9:30 AM). | | [**bfill**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.bfill.html#pyspark.pandas.DataFrame.bfill)**([axis, inplace, limit])** | Synonym for *DataFrame.fillna()* or *Series.fillna()* with method=`bfill`. | | [**bool**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.bool.html#pyspark.pandas.DataFrame.bool)**()** | Return the bool of a single element in the current object. | | [**clip**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.clip.html#pyspark.pandas.DataFrame.clip)**([lower, upper])** | Trim values at input threshold(s). | | [**copy**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.copy.html#pyspark.pandas.DataFrame.copy)**([deep])** | Make a copy of this object’s indices and data. | | [**corr**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.corr.html#pyspark.pandas.DataFrame.corr)**([method])** | Compute pairwise correlation of columns, excluding NA/null values. | | [**count**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.count.html#pyspark.pandas.DataFrame.count)**([axis, numeric\_only])** | Count non-NA cells for each column. | | [**cummax**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.cummax.html#pyspark.pandas.DataFrame.cummax)**([skipna])** | Return cumulative maximum over a DataFrame or Series axis. | | [**cummin**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.cummin.html#pyspark.pandas.DataFrame.cummin)**([skipna])** | Return cumulative minimum over a DataFrame or Series axis. | | [**cumprod**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.cumprod.html#pyspark.pandas.DataFrame.cumprod)**([skipna])** | Return cumulative product over a DataFrame or Series axis. | | [**cumsum**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.cumsum.html#pyspark.pandas.DataFrame.cumsum)**([skipna])** | Return cumulative sum over a DataFrame or Series axis. | | [**describe**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.describe.html#pyspark.pandas.DataFrame.describe)**([percentiles])** | Generate descriptive statistics that summarize the central tendency, dispersion and shape of a dataset’s distribution, excluding NaN values. | | [**diff**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.diff.html#pyspark.pandas.DataFrame.diff)**([periods, axis])** | First discrete difference of element. | | [**div**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.div.html#pyspark.pandas.DataFrame.div)**(other)** | Get Floating division of dataframe and other, element-wise (binary operator */*). | | **divide(other)** | Get Floating division of dataframe and other, element-wise (binary operator */*). | | [**dot**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.dot.html#pyspark.pandas.DataFrame.dot)**(other)** | Compute the matrix multiplication between the DataFrame and other. | | [**drop**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.drop.html#pyspark.pandas.DataFrame.drop)**([labels, axis, columns])** | Drop specified labels from columns. | | [**drop\_duplicates**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.drop_duplicates.html#pyspark.pandas.DataFrame.drop_duplicates)**([subset, keep, inplace])** | Return DataFrame with duplicate rows removed, optionally only considering certain columns. | | [**droplevel**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.droplevel.html#pyspark.pandas.DataFrame.droplevel)**(level[, axis])** | Return DataFrame with requested index / column level(s) removed. | | [**dropna**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.dropna.html#pyspark.pandas.DataFrame.dropna)**([axis, how, thresh, subset, inplace])** | Remove missing values. | | [**duplicated**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.duplicated.html#pyspark.pandas.DataFrame.duplicated)**([subset, keep])** | Return boolean Series denoting duplicate rows, optionally only considering certain columns. | | [**eq**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.eq.html#pyspark.pandas.DataFrame.eq)**(other)** | Compare if the current value is equal to the other. | | [**equals**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.equals.html#pyspark.pandas.DataFrame.equals)**(other)** | Compare if the current value is equal to the other. | | [**eval**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.eval.html#pyspark.pandas.DataFrame.eval)**(expr[, inplace])** | Evaluate a string describing operations on DataFrame columns. | | [**expanding**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.expanding.html#pyspark.pandas.DataFrame.expanding)**([min\_periods])** | Provide expanding transformations. | | [**explode**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.explode.html#pyspark.pandas.DataFrame.explode)**(column)** | Transform each element of a list-like to a row, replicating index values. | | [**ffill**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.ffill.html#pyspark.pandas.DataFrame.ffill)**([axis, inplace, limit])** | Synonym for *DataFrame.fillna()* or *Series.fillna()* with method=`ffill`. | | [**fillna**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.fillna.html#pyspark.pandas.DataFrame.fillna)**([value, method, axis, inplace, limit])** | Fill NA/NaN values. | | [**filter**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.filter.html#pyspark.pandas.DataFrame.filter)**([items, like, regex, axis])** | Subset rows or columns of dataframe according to labels in the specified index. | | [**first**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.first.html#pyspark.pandas.DataFrame.first)**(offset)** | Select first periods of time series data based on a date offset. | | [**first\_valid\_index**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.first_valid_index.html#pyspark.pandas.DataFrame.first_valid_index)**()** | Retrieves the index of the first valid value. | | [**floordiv**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.floordiv.html#pyspark.pandas.DataFrame.floordiv)**(other)** | Get Integer division of dataframe and other, element-wise (binary operator *//*). | | **from\_dict(data[, orient, dtype, columns])** | Construct DataFrame from dict of array-like or dicts. | | [**from\_records**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.from_records.html#pyspark.pandas.DataFrame.from_records)**(data[, index, exclude, …])** | Convert structured or record ndarray to DataFrame. | | [**ge**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.ge.html#pyspark.pandas.DataFrame.ge)**(other)** | Compare if the current value is greater than or equal to the other. | | [**get**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.get.html#pyspark.pandas.DataFrame.get)**(key[, default])** | Get item from object for given key (DataFrame column, Panel slice, etc.). | | **get\_dtype\_counts()** | Return counts of unique dtypes in this object. | | [**groupby**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.groupby.html#pyspark.pandas.DataFrame.groupby)**(by[, axis, as\_index, dropna])** | Group DataFrame or Series using a Series of columns. | | [**gt**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.gt.html#pyspark.pandas.DataFrame.gt)**(other)** | Compare if the current value is greater than the other. | | [**head**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.head.html#pyspark.pandas.DataFrame.head)**([n])** | Return the first *n* rows. | | [**hist**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.hist.html#pyspark.pandas.DataFrame.hist)**([bins])** | Draw one histogram of the DataFrame’s columns. | | [**idxmax**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.idxmax.html#pyspark.pandas.DataFrame.idxmax)**([axis])** | Return index of first occurrence of maximum over requested axis. | | [**idxmin**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.idxmin.html#pyspark.pandas.DataFrame.idxmin)**([axis])** | Return index of first occurrence of minimum over requested axis. | | [**info**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.info.html#pyspark.pandas.DataFrame.info)**([verbose, buf, max\_cols, null\_counts])** | Print a concise summary of a DataFrame. | | [**insert**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.insert.html#pyspark.pandas.DataFrame.insert)**(loc, column, value[, allow\_duplicates])** | Insert column into DataFrame at specified location. | | [**isin**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.isin.html#pyspark.pandas.DataFrame.isin)**(values)** | Whether each element in the DataFrame is contained in values. | | [**isna**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.isna.html#pyspark.pandas.DataFrame.isna)**()** | Detects missing values for items in the current Dataframe. | | [**isnull**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.isnull.html#pyspark.pandas.DataFrame.isnull)**()** | Detects missing values for items in the current Dataframe. | | [**items**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.items.html#pyspark.pandas.DataFrame.items)**()** | This is an alias of iteritems. | | [**iteritems**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.iteritems.html#pyspark.pandas.DataFrame.iteritems)**()** | Iterator over (column name, Series) pairs. | | [**iterrows**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.iterrows.html#pyspark.pandas.DataFrame.iterrows)**()** | Iterate over DataFrame rows as (index, Series) pairs. | | [**itertuples**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.itertuples.html#pyspark.pandas.DataFrame.itertuples)**([index, name])** | Iterate over DataFrame rows as namedtuples. | | [**join**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.join.html#pyspark.pandas.DataFrame.join)**(right[, on, how, lsuffix, rsuffix])** | Join columns of another DataFrame. | | [**kde**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.kde.html#pyspark.pandas.DataFrame.kde)**([bw\_method, ind])** | Generate Kernel Density Estimate plot using Gaussian kernels. | | [**keys**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.keys.html#pyspark.pandas.DataFrame.keys)**()** | Return alias for columns. | | [**kurt**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.kurt.html#pyspark.pandas.DataFrame.kurt)**([axis, numeric\_only])** | Return unbiased kurtosis using Fisher’s definition of kurtosis (kurtosis of normal == 0.0). | | [**kurtosis**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.kurtosis.html#pyspark.pandas.DataFrame.kurtosis)**([axis, numeric\_only])** | Return unbiased kurtosis using Fisher’s definition of kurtosis (kurtosis of normal == 0.0). | | [**last**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.last.html#pyspark.pandas.DataFrame.last)**(offset)** | Select final periods of time series data based on a date offset. | | [**last\_valid\_index**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.last_valid_index.html#pyspark.pandas.DataFrame.last_valid_index)**()** | Return index for last non-NA/null value. | | [**le**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.le.html#pyspark.pandas.DataFrame.le)**(other)** | Compare if the current value is less than or equal to the other. | | [**lt**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.lt.html#pyspark.pandas.DataFrame.lt)**(other)** | Compare if the current value is less than the other. | | [**mad**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.mad.html#pyspark.pandas.DataFrame.mad)**([axis])** | Return the mean absolute deviation of values. | | [**mask**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.mask.html#pyspark.pandas.DataFrame.mask)**(cond[, other])** | Replace values where the condition is True. | | [**max**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.max.html#pyspark.pandas.DataFrame.max)**([axis, numeric\_only])** | Return the maximum of the values. | | [**mean**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.mean.html#pyspark.pandas.DataFrame.mean)**([axis, numeric\_only])** | Return the mean of the values. | | [**median**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.median.html#pyspark.pandas.DataFrame.median)**([axis, numeric\_only, accuracy])** | Return the median of the values for the requested axis. | | [**melt**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.melt.html#pyspark.pandas.DataFrame.melt)**([id\_vars, value\_vars, var\_name, value\_name])** | Unpivot a DataFrame from wide format to long format, optionally leaving identifier variables set. | | [**merge**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.merge.html#pyspark.pandas.DataFrame.merge)**(right[, how, on, left\_on, right\_on, …])** | Merge DataFrame objects with a database-style join. | | [**min**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.min.html#pyspark.pandas.DataFrame.min)**([axis, numeric\_only])** | Return the minimum of the values. | | [**mod**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.mod.html#pyspark.pandas.DataFrame.mod)**(other)** | Get Modulo of dataframe and other, element-wise (binary operator *%*). | | [**mul**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.mul.html#pyspark.pandas.DataFrame.mul)**(other)** | Get Multiplication of dataframe and other, element-wise (binary operator *\**). | | **multiply(other)** | Get Multiplication of dataframe and other, element-wise (binary operator *\**). | | [**ne**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.ne.html#pyspark.pandas.DataFrame.ne)**(other)** | Compare if the current value is not equal to the other. | | [**nlargest**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.nlargest.html#pyspark.pandas.DataFrame.nlargest)**(n, columns)** | Return the first *n* rows ordered by *columns* in descending order. | | [**notna**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.notna.html#pyspark.pandas.DataFrame.notna)**()** | Detects non-missing values for items in the current Dataframe. | | [**notnull**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.notnull.html#pyspark.pandas.DataFrame.notnull)**()** | Detects non-missing values for items in the current Dataframe. | | [**nsmallest**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.nsmallest.html#pyspark.pandas.DataFrame.nsmallest)**(n, columns)** | Return the first *n* rows ordered by *columns* in ascending order. | | [**nunique**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.nunique.html#pyspark.pandas.DataFrame.nunique)**([axis, dropna, approx, rsd])** | Return number of unique elements in the object. | | [**pad**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.pad.html#pyspark.pandas.DataFrame.pad)**([axis, inplace, limit])** | Synonym for *DataFrame.fillna()* or *Series.fillna()* with method=`ffill`. | | [**pct\_change**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.pct_change.html#pyspark.pandas.DataFrame.pct_change)**([periods])** | Percentage change between the current and a prior element. | | [**pipe**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.pipe.html#pyspark.pandas.DataFrame.pipe)**(func, \*args, \*\*kwargs)** | Apply func(self, \*args, \*\*kwargs). | | [**pivot**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.pivot.html#pyspark.pandas.DataFrame.pivot)**([index, columns, values])** | Return reshaped DataFrame organized by given index / column values. | | [**pivot\_table**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.pivot_table.html#pyspark.pandas.DataFrame.pivot_table)**([values, index, columns, …])** | Create a spreadsheet-style pivot table as a DataFrame. | | [**pop**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.pop.html#pyspark.pandas.DataFrame.pop)**(item)** | Return item and drop from frame. | | [**pow**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.pow.html#pyspark.pandas.DataFrame.pow)**(other)** | Get Exponential power of series of dataframe and other, element-wise (binary operator *\*\**). | | [**prod**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.prod.html#pyspark.pandas.DataFrame.prod)**([axis, numeric\_only, min\_count])** | Return the product of the values. | | [**product**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.product.html#pyspark.pandas.DataFrame.product)**([axis, numeric\_only, min\_count])** | Return the product of the values. | | [**quantile**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.quantile.html#pyspark.pandas.DataFrame.quantile)**([q, axis, numeric\_only, accuracy])** | Return value at the given quantile. | | [**query**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.query.html#pyspark.pandas.DataFrame.query)**(expr[, inplace])** | Query the columns of a DataFrame with a boolean expression. | | [**radd**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.radd.html#pyspark.pandas.DataFrame.radd)**(other)** | Get Addition of dataframe and other, element-wise (binary operator *+*). | | [**rank**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rank.html#pyspark.pandas.DataFrame.rank)**([method, ascending])** | Compute numerical data ranks (1 through n) along axis. | | [**rdiv**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rdiv.html#pyspark.pandas.DataFrame.rdiv)**(other)** | Get Floating division of dataframe and other, element-wise (binary operator */*). | | [**reindex**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.reindex.html#pyspark.pandas.DataFrame.reindex)**([labels, index, columns, axis, …])** | Conform DataFrame to new index with optional filling logic, placing NA/NaN in locations having no value in the previous index. | | [**reindex\_like**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.reindex_like.html#pyspark.pandas.DataFrame.reindex_like)**(other[, copy])** | Return a DataFrame with matching indices as other object. | | [**rename**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rename.html#pyspark.pandas.DataFrame.rename)**([mapper, index, columns, axis, …])** | Alter axes labels. | | [**rename\_axis**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rename_axis.html#pyspark.pandas.DataFrame.rename_axis)**([mapper, index, columns, axis, …])** | Set the name of the axis for the index or columns. | | [**replace**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.replace.html#pyspark.pandas.DataFrame.replace)**([to\_replace, value, inplace, limit, …])** | Returns a new DataFrame replacing a value with another value. | | [**reset\_index**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.reset_index.html#pyspark.pandas.DataFrame.reset_index)**([level, drop, inplace, …])** | Reset the index, or a level of it. | | [**rfloordiv**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rfloordiv.html#pyspark.pandas.DataFrame.rfloordiv)**(other)** | Get Integer division of dataframe and other, element-wise (binary operator *//*). | | [**rmod**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rmod.html#pyspark.pandas.DataFrame.rmod)**(other)** | Get Modulo of dataframe and other, element-wise (binary operator *%*). | | [**rmul**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rmul.html#pyspark.pandas.DataFrame.rmul)**(other)** | Get Multiplication of dataframe and other, element-wise (binary operator *\**). | | [**rolling**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rolling.html#pyspark.pandas.DataFrame.rolling)**(window[, min\_periods])** | Provide rolling transformations. | | [**round**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.round.html#pyspark.pandas.DataFrame.round)**([decimals])** | Round a DataFrame to a variable number of decimal places. | | [**rpow**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rpow.html#pyspark.pandas.DataFrame.rpow)**(other)** | Get Exponential power of dataframe and other, element-wise (binary operator *\*\**). | | [**rsub**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rsub.html#pyspark.pandas.DataFrame.rsub)**(other)** | Get Subtraction of dataframe and other, element-wise (binary operator *-*). | | [**rtruediv**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rtruediv.html#pyspark.pandas.DataFrame.rtruediv)**(other)** | Get Floating division of dataframe and other, element-wise (binary operator */*). | | [**sample**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.sample.html#pyspark.pandas.DataFrame.sample)**([n, frac, replace, random\_state])** | Return a random sample of items from an axis of object. | | [**select\_dtypes**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.select_dtypes.html#pyspark.pandas.DataFrame.select_dtypes)**([include, exclude])** | Return a subset of the DataFrame’s columns based on the column dtypes. | | [**sem**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.sem.html#pyspark.pandas.DataFrame.sem)**([axis, ddof, numeric\_only])** | Return unbiased standard error of the mean over requested axis. | | [**set\_index**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.set_index.html#pyspark.pandas.DataFrame.set_index)**(keys[, drop, append, inplace])** | Set the DataFrame index (row labels) using one or more existing columns. | | [**shift**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.shift.html#pyspark.pandas.DataFrame.shift)**([periods, fill\_value])** | Shift DataFrame by desired number of periods. | | [**skew**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.skew.html#pyspark.pandas.DataFrame.skew)**([axis, numeric\_only])** | Return unbiased skew normalized by N-1. | | [**sort\_index**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.sort_index.html#pyspark.pandas.DataFrame.sort_index)**([axis, level, ascending, …])** | Sort object by labels (along an axis) | | [**sort\_values**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.sort_values.html#pyspark.pandas.DataFrame.sort_values)**(by[, ascending, inplace, …])** | Sort by the values along either axis. | | [**squeeze**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.squeeze.html#pyspark.pandas.DataFrame.squeeze)**([axis])** | Squeeze 1 dimensional axis objects into scalars. | | [**stack**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.stack.html#pyspark.pandas.DataFrame.stack)**()** | Stack the prescribed level(s) from columns to index. | | [**std**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.std.html#pyspark.pandas.DataFrame.std)**([axis, ddof, numeric\_only])** | Return sample standard deviation. | | [**sub**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.sub.html#pyspark.pandas.DataFrame.sub)**(other)** | Get Subtraction of dataframe and other, element-wise (binary operator *-*). | | **subtract(other)** | Get Subtraction of dataframe and other, element-wise (binary operator *-*). | | [**sum**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.sum.html#pyspark.pandas.DataFrame.sum)**([axis, numeric\_only, min\_count])** | Return the sum of the values. | | [**swapaxes**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.swapaxes.html#pyspark.pandas.DataFrame.swapaxes)**(i, j[, copy])** | Interchange axes and swap values axes appropriately. | | [**swaplevel**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.swaplevel.html#pyspark.pandas.DataFrame.swaplevel)**([i, j, axis])** | Swap levels i and j in a MultiIndex on a particular axis. | | [**tail**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.tail.html#pyspark.pandas.DataFrame.tail)**([n])** | Return the last *n* rows. | | [**take**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.take.html#pyspark.pandas.DataFrame.take)**(indices[, axis])** | Return the elements in the given *positional* indices along an axis. | | [**to\_clipboard**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_clipboard.html#pyspark.pandas.DataFrame.to_clipboard)**([excel, sep])** | Copy object to the system clipboard. | | [**to\_csv**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_csv.html#pyspark.pandas.DataFrame.to_csv)**([path, sep, na\_rep, columns, header, …])** | Write object to a comma-separated values (csv) file. | | [**to\_delta**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_delta.html#pyspark.pandas.DataFrame.to_delta)**(path[, mode, partition\_cols, index\_col])** | Write the DataFrame out as a Delta Lake table. | | [**to\_dict**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_dict.html#pyspark.pandas.DataFrame.to_dict)**([orient, into])** | Convert the DataFrame to a dictionary. | | [**to\_excel**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_excel.html#pyspark.pandas.DataFrame.to_excel)**(excel\_writer[, sheet\_name, na\_rep, …])** | Write object to an Excel sheet. | | [**to\_html**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_html.html#pyspark.pandas.DataFrame.to_html)**([buf, columns, col\_space, header, …])** | Render a DataFrame as an HTML table. | | [**to\_json**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_json.html#pyspark.pandas.DataFrame.to_json)**([path, compression, num\_files, …])** | Convert the object to a JSON string. | | [**to\_latex**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_latex.html#pyspark.pandas.DataFrame.to_latex)**([buf, columns, col\_space, header, …])** | Render an object to a LaTeX tabular environment table. | | [**to\_markdown**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_markdown.html#pyspark.pandas.DataFrame.to_markdown)**([buf, mode])** | Print Series or DataFrame in Markdown-friendly format. | | [**to\_numpy**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_numpy.html#pyspark.pandas.DataFrame.to_numpy)**()** | A NumPy ndarray representing the values in this DataFrame or Series. | | [**to\_orc**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_orc.html#pyspark.pandas.DataFrame.to_orc)**(path[, mode, partition\_cols, index\_col])** | Write the DataFrame out as a ORC file or directory. | | [**to\_pandas**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_pandas.html#pyspark.pandas.DataFrame.to_pandas)**()** | Return a pandas DataFrame. | | [**to\_parquet**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_parquet.html#pyspark.pandas.DataFrame.to_parquet)**(path[, mode, partition\_cols, …])** | Write the DataFrame out as a Parquet file or directory. | | [**to\_records**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_records.html#pyspark.pandas.DataFrame.to_records)**([index, column\_dtypes, index\_dtypes])** | Convert DataFrame to a NumPy record array. | | [**to\_spark**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_spark.html#pyspark.pandas.DataFrame.to_spark)**([index\_col])** | Spark related features. | | [**to\_spark\_io**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_spark_io.html#pyspark.pandas.DataFrame.to_spark_io)**([path, format, mode, …])** | Write the DataFrame out to a Spark data source. | | [**to\_string**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_string.html#pyspark.pandas.DataFrame.to_string)**([buf, columns, col\_space, header, …])** | Render a DataFrame to a console-friendly tabular output. | | [**to\_table**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.to_table.html#pyspark.pandas.DataFrame.to_table)**(name[, format, mode, …])** | Write the DataFrame into a Spark table. | | [**transform**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.transform.html#pyspark.pandas.DataFrame.transform)**(func[, axis])** | Call func on self producing a Series with transformed values and that has the same length as its input. | | [**transpose**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.transpose.html#pyspark.pandas.DataFrame.transpose)**()** | Transpose index and columns. | | [**truediv**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.truediv.html#pyspark.pandas.DataFrame.truediv)**(other)** | Get Floating division of dataframe and other, element-wise (binary operator */*). | | [**truncate**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.truncate.html#pyspark.pandas.DataFrame.truncate)**([before, after, axis, copy])** | Truncate a Series or DataFrame before and after some index value. | | [**unstack**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.unstack.html#pyspark.pandas.DataFrame.unstack)**()** | Pivot the (necessarily hierarchical) index labels. | | [**update**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.update.html#pyspark.pandas.DataFrame.update)**(other[, join, overwrite])** | Modify in place using non-NA values from another DataFrame. | | [**var**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.var.html#pyspark.pandas.DataFrame.var)**([axis, ddof, numeric\_only])** | Return unbiased variance. | | [**where**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.where.html#pyspark.pandas.DataFrame.where)**(cond[, other, axis])** | Replace values where the condition is False. | | [**xs**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.xs.html#pyspark.pandas.DataFrame.xs)**(key[, axis, level])** | Return cross-section from the DataFrame. |   **Attributes**   |  |  | | --- | --- | | [**T**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.T.html#pyspark.pandas.DataFrame.T) | Transpose index and columns. | | [**at**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.at.html#pyspark.pandas.DataFrame.at) | Access a single value for a row/column label pair. | | [**axes**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.axes.html#pyspark.pandas.DataFrame.axes) | Return a list representing the axes of the DataFrame. | | [**columns**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.columns.html#pyspark.pandas.DataFrame.columns) | The column labels of the DataFrame. | | [**dtypes**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.dtypes.html#pyspark.pandas.DataFrame.dtypes) | Return the dtypes in the DataFrame. | | [**empty**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.empty.html#pyspark.pandas.DataFrame.empty) | Returns true if the current DataFrame is empty. | | [**iat**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.iat.html#pyspark.pandas.DataFrame.iat) | Access a single value for a row/column pair by integer position. | | [**iloc**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.iloc.html#pyspark.pandas.DataFrame.iloc) | Purely integer-location based indexing for selection by position. | | [**index**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.index.html#pyspark.pandas.DataFrame.index) | The index (row labels) Column of the DataFrame. | | [**loc**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.loc.html#pyspark.pandas.DataFrame.loc) | Access a group of rows and columns by label(s) or a boolean Series. | | [**ndim**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.ndim.html#pyspark.pandas.DataFrame.ndim) | Return an int representing the number of array dimensions. | | [**shape**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.shape.html#pyspark.pandas.DataFrame.shape) | Return a tuple representing the dimensionality of the DataFrame. | | [**size**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.size.html#pyspark.pandas.DataFrame.size) | Return an int representing the number of elements in this object. | | [**style**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.style.html#pyspark.pandas.DataFrame.style) | Property returning a Styler object containing methods for building a styled HTML representation for the DataFrame. | | [**values**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.values.html#pyspark.pandas.DataFrame.values) | Return a Numpy representation of the DataFrame or the Series. | |

|  |
| --- |
| pyspark.pandas.DataFrame.index  The index (row labels) Column of the DataFrame. |

|  |
| --- |
| pyspark.pandas.DataFrame.columns |

|  |
| --- |
| pyspark.pandas.DataFrame.empty  Returns true if the current DataFrame is empty. Otherwise, returns false.  ps.DataFrame({}, index=list('abc')).empty |

|  |
| --- |
| pyspark.pandas.DataFrame.dtypes  df.dtypes  a object  b int64  c int8  d float64  e bool  f datetime64[ns] |

|  |
| --- |
| pyspark.pandas.DataFrame.shape  df.shape  (2, 3) |

|  |
| --- |
| pyspark.pandas.DataFrame.axes  Return a list representing the axes of the DataFrame.  It has the row axis labels and column axis labels as the only members. They are returned in that order.  df.axes  [Int64Index([0, 1], dtype='int64'), Index(['col1', 'col2'], dtype='object')] |

|  |
| --- |
| pyspark.pandas.DataFrame.ndim  Return an int representing the number of array dimensions.  return 2 for DataFrame. |

|  |
| --- |
| pyspark.pandas.DataFrame.size  Return an int representing the number of elements in this object.  Return the number of rows if Series. Otherwise return the number of rows times number of columns if DataFrame. |

|  |
| --- |
| pyspark.pandas.DataFrame.select\_dtypes(include: Union[str, List[str], None] = None, exclude: Union[str, List[str], None] = None) → pyspark.pandas.frame.DataFrame  Return a subset of the DataFrame’s columns based on the column dtypes.  df.select\_dtypes(include='bool')  b  0 True  1 False  2 True  3 False  4 True  5 False  df.select\_dtypes(exclude=['int']) |

|  |
| --- |
| pyspark.pandas.DataFrame.values  Return a Numpy representation of the DataFrame or the Series.  Returns: numpy.ndarray  ps.Series([1, 2, 3]).values  ps.Series(list('aabc')).values  >>>df2.values  array([['parrot', 24.0, 'second'],  ['lion', 80.5, 'first'],  ['monkey', nan, None]], dtype=object) |

|  |
| --- |
| pyspark.pandas.DataFrame.copy(deep: bool = True) → pyspark.pandas.frame.DataFrame  df\_copy = df.copy()  df\_copy |

|  |
| --- |
| pyspark.pandas.DataFrame.isna() → pyspark.pandas.frame.DataFrame  Detects missing values for items in the current Dataframe.  Return a boolean same-sized Dataframe indicating if the values are NA. NA values, such as None or numpy.NaN, gets mapped to True values. Everything else gets mapped to False values.  df.isnull() |

|  |
| --- |
| pyspark.pandas.DataFrame.astype  df = ps.DataFrame({'a': [1, 2, 3], 'b': [1, 2, 3]}, dtype='int64')  df.astype('float')  df.astype({'a': float}) |

|  |
| --- |
| pyspark.pandas.DataFrame.isnull() → pyspark.pandas.frame.DataFrame  Detects missing values for items in the current Dataframe.  df = ps.DataFrame([(.2, .3), (.0, None), (.6, None), (.2, .1)])  df.isnull() |

|  |
| --- |
| pyspark.pandas.DataFrame.notna() → pyspark.pandas.frame.DataFrame  pyspark.pandas.DataFrame.notnull()  Detects non-missing values for items in the current Dataframe.  This function takes a dataframe and indicates whether it’s values are valid (not missing, which is NaN in numeric datatypes, None or NaN in objects and NaT in datetimelike).  df = ps.DataFrame([(.2, .3), (.0, None), (.6, None), (.2, .1)])  df.notnull() |

|  |
| --- |
| pyspark.pandas.DataFrame.bool()  Return the bool of a single element in the current object.  This must be a boolean scalar value, either True or False. Raise a ValueError if the object does not have exactly 1 element, or that element is not boolean  >>>ps.DataFrame({'a': [True]}).bool()  True |

|  |
| --- |
| pyspark.pandas.DataFrame.at  Access a single value for a row/column label pair. If the index is not unique, all matching pairs are returned as an array. Similar to loc, in that both provide label-based lookups. Use at if you only need to get a single value in a DataFrame or Series.  >>>psdf  A B C  4 0 2 3  5 0 4 1  5 10 20 30  >>>psdf.at[4, 'B']  2 |

|  |
| --- |
| pyspark.pandas.DataFrame.iat  Access a single value for a row/column pair by integer position.  Similar to iloc, in that both provide integer-based lookups. Use iat if you only need to get or set a single value in a DataFrame or Series.  >>>df.iat[1, 2]  1  >>>psser.iat[1]  2 |

|  |
| --- |
| pyspark.pandas.DataFrame.head(n)  df.head(3) |

|  |
| --- |
| pyspark.pandas.DataFrame.idxmax(axis: Union[int, str] = 0) → Series  Return index of first occurrence of maximum over requested axis. NA/null values are excluded.  pyspark.pandas.DataFrame.idxmin(axis: Union[int, str] = 0) → Series  Return index of first occurrence of minimum over requested axis. NA/null values are excluded. |

|  |
| --- |
| pyspark.pandas.DataFrame.loc  Access a group of rows and columns by label(s) or a boolean Series.  max\_speed shield  cobra 1 2  viper 4 5  sidewinder 7 8  >>>df.loc['viper']  max\_speed 4  shield 5  >>>df.loc[['viper', 'sidewinder']]  max\_speed shield  viper 4 5  sidewinder 7 8  >>>df.loc[['sidewinder', 'viper']]  max\_speed shield  viper 4 5  sidewinder 7 8  >>>df.loc['cobra', 'shield']  2  >>>df.loc[['cobra'], 'shield']  cobra 2  Name: shield, dtype: int64  >>>df.loc['cobra', ['shield']]  shield 2  Name: cobra, dtype: int64  >>>df.loc[['cobra'], ['shield']]  shield  cobra 2  >>>df.loc['cobra':'viper', 'max\_speed']  cobra 1  viper 4  Name: max\_speed, dtype: int64  >>>df.loc[df['shield'] > 6]  max\_speed shield  sidewinder 7 8  etc… |

|  |
| --- |
| pyspark.pandas.DataFrame.iloc  >>>df  a b c d  0 1 2 3 4  1 100 200 300 400  2 1000 2000 3000 4000  >>>df.iloc[1]  a 100  b 200  c 300  d 400  >>>df.iloc[[0]]  a b c d  0 1 2 3 4  >>>df.iloc[:3]  a b c d  0 1 2 3 4  1 100 200 300 400  2 1000 2000 3000 4000  >>>df.iloc[:2, [1, 3]]  b d  0 2 4  1 200 400  >>>df.iloc[:2, 0:3]  a b c  0 1 2 3  1 100 200 300 |

|  |
| --- |
| pyspark.pandas.DataFrame.items()→ Iterator[Tuple[Union[Any, Tuple[Any, …]], Series]]  This is an alias of iteritems. |

|  |
| --- |
| pyspark.pandas.DataFrame.iteritems() → Iterator[Tuple[Union[Any, Tuple[Any, …]], Series]]  Iterator over (column name, Series) pairs.  Iterates over the DataFrame columns, returning a tuple with the column name and the content as a Series.  for label, content in df.iteritems():  print('label:', label)  print('content:', content.to\_string()) |

|  |
| --- |
| pyspark.pandas.DataFrame.iterrows()  Because iterrows returns a Series for each row, it does not preserve dtypes across the rows (dtypes are preserved across columns for DataFrames).  >>>df = ps.DataFrame([[1, 1.5]], columns=['int', 'float'])  >>>row = next(df.iterrows())[1]  >>>row  int 1.0  float 1.5  Name: 0, dtype: float64  >>>print(row['int'].dtype)  float64  >>>print(df['int'].dtype)  int64 |

|  |
| --- |
| pyspark.pandas.DataFrame.itertuples(index: bool = True, name: Optional[str] = 'PandasOnSpark') → Iterator[Tuple]  Iterate over DataFrame rows as namedtuples.  >>>for row in df.itertuples():  print(row)  PandasOnSpark(Index='dog', num\_legs=4, num\_wings=0)  PandasOnSpark(Index='hawk', num\_legs=2, num\_wings=2)  >>>for row in df.itertuples(index=False):  print(row)  PandasOnSpark(num\_legs=4, num\_wings=0)  PandasOnSpark(num\_legs=2, num\_wings=2)  >>>for row in df.itertuples(name='Animal'):  print(row)  Animal(Index='dog', num\_legs=4, num\_wings=0)  Animal(Index='hawk', num\_legs=2, num\_wings=2) |

|  |
| --- |
| pyspark.pandas.DataFrame.keys()  Return alias for columns.  >>>df.keys()  Index(['max\_speed', 'shield'], dtype='object') |

|  |
| --- |
| pyspark.pandas.DataFrame.pop(item: Union[Any, Tuple[Any, …]])  Return item and drop from frame. Raise KeyError if not found.  df.pop('class') #’class’ is column |

|  |
| --- |
| pyspark.pandas.DataFrame.tail(n) |

|  |
| --- |
| pyspark.pandas.DataFrame.get(key: Any, default: Optional[Any] = None)  Get item from object for given key (DataFrame column, Panel slice, etc.). Returns default value if not found.  >>>df  x y z  10 0 a a  20 1 b b  20 2 b b  >>>df.get(['x', 'y'])  x y  10 0 a  20 1 b  20 2 b  >>>df.get('x')  10 0  20 1  20 2  >>>df.x.get(10)  0  >>>df.x.get(15, -1)  -1 |

|  |
| --- |
| pyspark.pandas.DataFrame.where  df1.where(df1 > 0).sort\_index()  df1.where(df1 > 1, 10).sort\_index()  df1.where(df1 > 1, df1 + 100).sort\_index() |

|  |
| --- |
| pyspark.pandas.DataFrame.mask(cond: Union[DataFrame, Series], other: Union[DataFrame, Series, Any] = nan)  Replace values where the condition is True.  from pyspark.pandas.config import set\_option, reset\_option  set\_option("compute.ops\_on\_diff\_frames", True)  df1 = ps.DataFrame({'A': [0, 1, 2, 3, 4], 'B':[100, 200, 300, 400, 500]})  df2 = ps.DataFrame({'A': [0, -1, -2, -3, -4], 'B':[-100, -200, -300, -400, -500]})  df1.mask(df1 > 0).sort\_index()  df1.mask(df1 > 1, 10).sort\_index()  df1.mask(df1 > 1, df1 + 100).sort\_index()  df1.mask(df1 > 1, df2).sort\_index()  reset\_option("compute.ops\_on\_diff\_frames") |

|  |
| --- |
| pyspark.pandas.DataFrame.query  Query the columns of a DataFrame with a boolean expression.  >>>df = ps.DataFrame({'A': range(1, 6),  'B': range(10, 0, -2),  'C C': range(10, 5, -1)})  >>>df.query('A > B')  A B C C  4 5 2 6 |

|  |
| --- |
| pyspark.pandas.DataFrame.add  Get Addition of dataframe and other, element-wise (binary operator +).  Equivalent to dataframe + other. With reverse version, radd.  #Add a scalar with operator version which return the same results. Also reverse version.  df + 1 #df.add(1)  df.add(df)  df.add(df)  df + df + df  df.radd(1)  df / 10 #df.div(10)  df.sub(1)  df.mul(1)  df \*\* 2 #df.pow(2)   * [pyspark.pandas.DataFrame.add](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.add.html)(6) * [pyspark.pandas.DataFrame.radd](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.radd.html) * [pyspark.pandas.DataFrame.div](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.div.html)(7) * [pyspark.pandas.DataFrame.rdiv](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rdiv.html) * [pyspark.pandas.DataFrame.truediv](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.truediv.html) * [pyspark.pandas.DataFrame.rtruediv](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rtruediv.html) * [pyspark.pandas.DataFrame.mul](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.mul.html)(8) * [pyspark.pandas.DataFrame.rmul](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rmul.html) * [pyspark.pandas.DataFrame.sub](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.sub.html)(9) * [pyspark.pandas.DataFrame.rsub](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rsub.html) * [pyspark.pandas.DataFrame.pow](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.pow.html)(2) * [pyspark.pandas.DataFrame.rpow](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rpow.html) * [pyspark.pandas.DataFrame.mod](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.mod.html)(3) * [pyspark.pandas.DataFrame.rmod](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rmod.html) * [pyspark.pandas.DataFrame.floordiv](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.floordiv.html) * [pyspark.pandas.DataFrame.rfloordiv](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.rfloordiv.html) * [pyspark.pandas.DataFrame.lt](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.lt.html)(4) * [pyspark.pandas.DataFrame.gt](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.gt.html)(5) * [pyspark.pandas.DataFrame.le](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.le.html)(4.5) * [pyspark.pandas.DataFrame.ge](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.ge.html) * [pyspark.pandas.DataFrame.ne](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.ne.html) * [pyspark.pandas.DataFrame.eq](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.eq.html)(10) * [pyspark.pandas.DataFrame.dot](https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.dot.html)(df2) |

|  |
| --- |
| pyspark.pandas.DataFrame.apply(func: Callable, axis: Union[int, str] = 0, args: Sequence[Any] = (), \*\*kwds: Any)  Apply a function along an axis of the DataFrame.  def sqrt(x) -> ps.Series[float]:  return np.sqrt(x)  df.apply(sqrt, axis=0)  df.apply(np.sqrt, axis=0)  #When axis is 1 or ‘columns’, it applies the function for each row.  df.apply(summation, axis=1)  df.apply(lambda x: [1, 2], axis=1)  def plus\_two(a, b, c) -> ps.DataFrame[np.int64, np.int64]:  return a + b + c  df.apply(plus\_two, axis=1, args=(1,), c=3) |

|  |
| --- |
| pyspark.pandas.DataFrame.applymap  Apply a function to a Dataframe elementwise.  This method applies a function that accepts and returns a scalar to every element of a DataFrame. |

|  |
| --- |
| pyspark.pandas.DataFrame.pipe  Apply func(self, \*args, \*\*kwargs).  Use .pipe when chaining together functions that expect Series, DataFrames or GroupBy objects.  #multiply(add\_one(keep\_category\_a(df), column="col1"), column1="col2", column2="col3")  #Instaed use Pipe to pipe multiple number of functions  (df.pipe(keep\_category\_a)  .pipe(add\_one, column="col1")  .pipe(multiply, column1="col2", column2="col3")  ) |

|  |
| --- |
| pyspark.pandas.DataFrame.agg  >>>df  A B C  0 1.0 2.0 3.0  1 4.0 5.0 6.0  2 7.0 8.0 9.0  3 NaN NaN NaN  >>>df.agg(['sum', 'min'])[['A', 'B', 'C']].sort\_index()  A B C  min 1.0 2.0 3.0  sum 12.0 15.0 18.0  >>>df.agg({'A' : ['sum', 'min'], 'B' : ['min', 'max']})[['A', 'B']].sort\_index()  A B  max NaN 8.0  min 1.0 2.0  sum 12.0 NaN |

|  |
| --- |
| pyspark.pandas.DataFrame.groupby  A groupby operation involves some combination of splitting the object, applying a function, and combining the results. This can be used to group large amounts of data and compute operations on these groups.  df.groupby(['Animal']).mean().sort\_index()  df.groupby(['Animal'], as\_index=False).mean().sort\_values('Animal')  df.groupby(by=["b"]).sum().sort\_index()  df.groupby(by=["b"], dropna=False).sum().sort\_index() |

|  |
| --- |
| pyspark.pandas.DataFrame.rolling |

|  |
| --- |
| pyspark.pandas.DataFrame.expanding |

|  |
| --- |
| pyspark.pandas.DataFrame.transform  Call func on self producing a Series with transformed values and that has the same length as its input.  def square(x) -> ps.Series[np.int32]:  return x \*\* 2  df.transform(square)  A B  0 0 1  1 1 4  2 4 9  def calculation(x, y, z) -> ps.Series[int]:  return x \*\* y + z  df.transform(calculation, y=10, z=20) |

|  |
| --- |
| pyspark.pandas.DataFrame.abs()  Return a Series/DataFrame with absolute numeric value of each element.  s.abs()  df.abs() |

|  |
| --- |
| pyspark.pandas.DataFrame.all  Return whether all elements are True.  Returns True unless there is at least one element within a series that is False or equivalent (e.g. zero or empty)  df = ps.DataFrame({  'col1': [True, True, True],  'col2': [True, False, False],  'col3': [0, 0, 0],  'col4': [1, 2, 3],  'col5': [True, True, None],  'col6': [True, False, None]},  columns=['col1', 'col2', 'col3', 'col4', 'col5', 'col6'])  >>>df.all()  col1 True  col2 False  col3 False  col4 True  col5 True  col6 False |

|  |
| --- |
| pyspark.pandas.DataFrame.any  Return whether any element is True.  Returns False unless there is at least one element within a series that is True or equivalent (e.g. non-zero or non-empty).  >>>df.any()  col1 False  col2 True  col3 True  col4 True  col5 False  col6 True |

|  |
| --- |
| pyspark.pandas.DataFrame.corr  df.corr('pearson')  df.corr('spearman') |

|  |
| --- |
| pyspark.pandas.DataFrame.count  Count non-NA cells for each column.  df.count()  df['Person'].count() |

|  |
| --- |
| pyspark.pandas.DataFrame.describe  Generate descriptive statistics that summarize the central tendency, dispersion and shape of a dataset’s distribution, excluding NaN values.  s = ps.Series([1, 2, 3])  s.describe() |

|  |
| --- |
| pyspark.pandas.DataFrame.kurt  Return unbiased kurtosis using Fisher’s definition of kurtosis (kurtosis of normal == 0.0). Normalized by N-1.  df.kurtosis()  df['a'].kurtosis() |

|  |
| --- |
| pyspark.pandas.DataFrame.kurtosis  Return unbiased kurtosis using Fisher’s definition of kurtosis (kurtosis of normal == 0.0). Normalized by N-1. |

|  |
| --- |
| pyspark.pandas.DataFrame.mad  #Return the mean absolute deviation of values.  pyspark.pandas.DataFrame.max  pyspark.pandas.DataFrame.min  pyspark.pandas.DataFrame.mean  pyspark.pandas.DataFrame.median |

|  |
| --- |
| pyspark.pandas.DataFrame.pct\_change  Percentage change between the current and a prior element. |

|  |
| --- |
| pyspark.pandas.DataFrame.prod  pyspark.pandas.DataFrame.product  >>>ps.Series([1, 2, 3, 4, 5]).prod()  120  >>>psdf.prod()  A 120  B 1200000 |

|  |
| --- |
| pyspark.pandas.DataFrame.quantile  Return value at the given quantile.  >>>psdf  a b  0 1 6  1 2 7  2 3 8  3 4 9  4 5 0  >>>psdf.quantile(.5)  a 3.0  b 7.0  >>>psdf.quantile([.25, .5, .75])  a b  0.25 2.0 6.0  0.50 3.0 7.0  0.75 4.0 8.0 |

|  |
| --- |
| pyspark.pandas.DataFrame.nunique  #Return number of unique elements in the object.  pyspark.pandas.DataFrame.skew  #Return unbiased skew normalized by N-1.  pyspark.pandas.DataFrame.sum  pyspark.pandas.DataFrame.std  pyspark.pandas.DataFrame.var  pyspark.pandas.DataFrame.cummin  pyspark.pandas.DataFrame.cummax  pyspark.pandas.DataFrame.cumsum  pyspark.pandas.DataFrame.cumprod |

|  |
| --- |
| pyspark.pandas.DataFrame.round  df.round(2)  df.round({'A': 1, 'C': 2})  decimals = ps.Series([1, 0, 2], index=['A', 'B', 'C'])  df.round(decimals) |

|  |
| --- |
| pyspark.pandas.DataFrame.diff  pyspark.pandas.DataFrame.eval |

|  |
| --- |
| pyspark.pandas.DataFrame.add\_prefix  pyspark.pandas.DataFrame.add\_suffix  For Series, the row labels are prefixed. For DataFrame, the column labels are prefixed.  df.add\_prefix('my\_')  my\_A my\_B  0 1 3  1 2 4 |

|  |
| --- |
| pyspark.pandas.DataFrame.align  Align two objects on their axes with the specified join method.  Join method is specified for each axis Index. |

|  |
| --- |
| pyspark.pandas.DataFrame.at\_time  Select values at particular time of day (example: 9:30AM).  >>>idx = pd.date\_range('2018-04-09', periods=4, freq='12H')  >>>psdf = ps.DataFrame({'A': [1, 2, 3, 4]}, index=idx)  >>>psdf  A  2018-04-09 00:00:00 1  2018-04-09 12:00:00 2  2018-04-10 00:00:00 3  2018-04-10 12:00:00 4  >>>psdf.at\_time('12:00')  A  2018-04-09 12:00:00 2  2018-04-10 12:00:00 4 |

|  |
| --- |
| pyspark.pandas.DataFrame.between\_time |

|  |
| --- |
| pyspark.pandas.DataFrame.drop  >>>df  x y z w  0 1 3 5 7  >>>df.drop('x', axis=1)  >>>df.drop(['y', 'z'], axis=1)  >>>df.drop(columns=['y', 'z']) |

|  |
| --- |
| Etc… |