

Deprecate .plotting

The `pandas.tools.plotting` module has been deprecated, in favor of the top level `pandas.plotting` module. All the public plotting functions are now available from `pandas.plotting` ([GH12548](#)).

Furthermore, the top-level `pandas.scatter_matrix` and `pandas.plot_params` are deprecated. Users can import these from `pandas.plotting` as well.

Previous script:

```
pd.tools.plotting.scatter_matrix(df)
pd.scatter_matrix(df)
```

Should be changed to:

```
pd.plotting.scatter_matrix(df)
```

Other deprecations

- `SparseArray.to_dense()` has deprecated the `fill` parameter, as that parameter was not being respected ([GH14647](#))
- `SparseSeries.to_dense()` has deprecated the `sparse_only` parameter ([GH14647](#))
- `Series.repeat()` has deprecated the `reps` parameter in favor of `repeats` ([GH12662](#))
- The `Series` constructor and `.astype` method have deprecated accepting timestamp dtypes without a frequency (e.g. `np.datetime64`) for the `dtype` parameter ([GH15524](#))
- `Index.repeat()` and `MultiIndex.repeat()` have deprecated the `n` parameter in favor of `repeats` ([GH12662](#))
- `Categorical.searchsorted()` and `Series.searchsorted()` have deprecated the `v` parameter in favor of `value` ([GH12662](#))
- `TimedeltaIndex.searchsorted()`, `DatetimeIndex.searchsorted()`, and `PeriodIndex.searchsorted()` have deprecated the `key` parameter in favor of `value` ([GH12662](#))
- `DataFrame.astype()` has deprecated the `raise_on_error` parameter in favor of `errors` ([GH14878](#))
- `Series.sortlevel` and `DataFrame.sortlevel` have been deprecated in favor of `Series.sort_index` and `DataFrame.sort_index` ([GH15099](#))
- importing `concat` from `pandas.tools.merge` has been deprecated in favor of imports from the `pandas` namespace. This should only affect explicit imports ([GH15358](#))
- `Series/DataFrame/Panel consolidate()` been deprecated as a public method. ([GH15483](#))
- The `as_indexer` keyword of `Series.str.match()` has been deprecated (ignored keyword) ([GH15257](#)).
- The following top-level `pandas` functions have been deprecated and will be removed in a future version ([GH13790](#), [GH15940](#))
 - `pd.pnow()`, replaced by `Period.now()`
 - `pd.Term`, is removed, as it is not applicable to user code. Instead use in-line string expressions in the `where` clause when searching in `HDFStore`
 - `pd.Expr`, is removed, as it is not applicable to user code.
 - `pd.match()`, is removed.

- `pd.groupby()`, replaced by using the `.groupby()` method directly on a `Series/DataFrame`
- `pd.get_store()`, replaced by a direct call to `pd.HDFStore(...)`
- `is_any_int_dtype`, `is_floating_dtype`, and `is_sequence` are deprecated from `pandas.api.types` ([GH16042](#))

Removal of prior version deprecations/changes

- The `pandas.rpy` module is removed. Similar functionality can be accessed through the `rpy2` project. See the [R interfacing docs](#) for more details.
- The `pandas.io.ga` module with a `google-analytics` interface is removed ([GH11308](#)). Similar functionality can be found in the [Google2Pandas](#) package.
- `pd.to_datetime` and `pd.to_timedelta` have dropped the `coerce` parameter in favor of errors ([GH13602](#))
- `pandas.stats.fama_macbeth`, `pandas.stats.ols`, `pandas.stats.plm` and `pandas.stats.var`, as well as the top-level `pandas.fama_macbeth` and `pandas.ols` routines are removed. Similar functionality can be found in the [statsmodels](#) package. ([GH11898](#))
- The `TimeSeries` and `SparseTimeSeries` classes, aliases of `Series` and `SparseSeries`, are removed ([GH10890](#), [GH15098](#)).
- `Series.is_time_series` is dropped in favor of `Series.index.is_all_dates` ([GH15098](#))
- The deprecated `irow`, `icol`, `iget` and `iget_value` methods are removed in favor of `iloc` and `iat` as explained [here](#) ([GH10711](#)).
- The deprecated `DataFrame.iterkv()` has been removed in favor of `DataFrame.iteritems()` ([GH10711](#))
- The `Categorical` constructor has dropped the `name` parameter ([GH10632](#))
- `Categorical` has dropped support for `NaN` categories ([GH10748](#))
- The `take_last` parameter has been dropped from `uplicated()`, `drop_duplicates()`, `nlargest()`, and `nsmallest()` methods ([GH10236](#), [GH10792](#), [GH10920](#))
- `Series`, `Index`, and `DataFrame` have dropped the `sort` and `order` methods ([GH10726](#))
- Where clauses in `pytables` are only accepted as strings and expressions types and not other data-types ([GH12027](#))
- `DataFrame` has dropped the `combineAdd` and `combineMult` methods in favor of `add` and `mul` respectively ([GH10735](#))

Performance improvements

- Improved performance of `pd.wide_to_long()` ([GH14779](#))
- Improved performance of `pd.factorize()` by releasing the GIL with `object` dtype when inferred as strings ([GH14859](#), [GH16057](#))
- Improved performance of timeseries plotting with an irregular `DatetimeIndex` (or with `compat_x=True`) ([GH15073](#)).
- Improved performance of `groupby().cummin()` and `groupby().cummax()` ([GH15048](#), [GH15109](#), [GH15561](#), [GH15635](#))
- Improved performance and reduced memory when indexing with a `MultiIndex` ([GH15245](#))

- When reading buffer object in `read_sas()` method without specified format, filepath string is inferred rather than buffer object. (GH14947)
- Improved performance of `.rank()` for categorical data (GH15498)
- Improved performance when using `.unstack()` (GH15503)
- Improved performance of merge/join on category columns (GH10409)
- Improved performance of `drop_duplicates()` on bool columns (GH12963)
- Improve performance of `pd.core.groupby.GroupBy.apply` when the applied function used the `.name` attribute of the group DataFrame (GH15062).
- Improved performance of `iloc` indexing with a list or array (GH15504).
- Improved performance of `Series.sort_index()` with a monotonic index (GH15694)
- Improved performance in `pd.read_csv()` on some platforms with buffered reads (GH16039)

Bug fixes

Conversion

- Bug in `Timestamp.replace` now raises `TypeError` when incorrect argument names are given; previously this raised `ValueError` (GH15240)
- Bug in `Timestamp.replace` with `compat` for passing long integers (GH15030)
- Bug in `Timestamp` returning UTC based time/date attributes when a timezone was provided (GH13303, GH6538)
- Bug in `Timestamp` incorrectly localizing timezones during construction (GH11481, GH15777)
- Bug in `TimedeltaIndex` addition where overflow was being allowed without error (GH14816)
- Bug in `TimedeltaIndex` raising a `ValueError` when boolean indexing with `loc` (GH14946)
- Bug in catching an overflow in `Timestamp + Timedelta/Offset` operations (GH15126)
- Bug in `DatetimeIndex.round()` and `Timestamp.round()` floating point accuracy when rounding by milliseconds or less (GH14440, GH15578)
- Bug in `astype()` where `inf` values were incorrectly converted to integers. Now raises error now with `astype()` for Series and DataFrames (GH14265)
- Bug in `DataFrame(...).apply(to_numeric)` when values are of type `decimal.Decimal`. (GH14827)
- Bug in `describe()` when passing a numpy array which does not contain the median to the percentiles keyword argument (GH14908)
- Cleaned up `PeriodIndex` constructor, including raising on floats more consistently (GH13277)
- Bug in using `__deepcopy__` on empty NDFrame objects (GH15370)
- Bug in `.replace()` may result in incorrect dtypes. (GH12747, GH15765)
- Bug in `Series.replace` and `DataFrame.replace` which failed on empty replacement dicts (GH15289)
- Bug in `Series.replace` which replaced a numeric by string (GH15743)
- Bug in `Index` construction with NaN elements and integer dtype specified (GH15187)
- Bug in `Series` construction with a `datetimetz` (GH14928)
- Bug in `Series.dt.round()` inconsistent behaviour on NaT 's with different arguments (GH14940)

- Bug in `Series` constructor when both `copy=True` and `dtype` arguments are provided ([GH15125](#))
- Incorrect dtypes `Series` was returned by comparison methods (e.g., `lt`, `gt`, ...) against a constant for an empty `DataFrame` ([GH15077](#))
- Bug in `Series.fillna()` with mixed dtypes containing tz-aware datetimes. ([GH14956](#))
- Bug in `DataFrame.fillna()` where the argument `downcast` was ignored when `fillna` value was of type `dict` ([GH15277](#))
- Bug in `.asfreq()`, where frequency was not set for empty `Series` ([GH14320](#))
- Bug in `DataFrame` construction with nulls and datetimes in a list-like ([GH15869](#))
- Bug in `DataFrame.fillna()` with tz-aware datetimes ([GH15855](#))
- Bug in `is_string_dtype`, `is_timedelta64_ns_dtype`, and `is_string_like_dtype` in which an error was raised when `None` was passed in ([GH15941](#))
- Bug in the return type of `pd.unique` on a `Categorical`, which was returning an `ndarray` and not a `Categorical` ([GH15903](#))
- Bug in `Index.to_series()` where the index was not copied (and so mutating later would change the original), ([GH15949](#))
- Bug in indexing with partial string indexing with a len-1 `DataFrame` ([GH16071](#))
- Bug in `Series` construction where passing invalid dtype didn't raise an error. ([GH15520](#))

Indexing

- Bug in `Index` power operations with reversed operands ([GH14973](#))
- Bug in `DataFrame.sort_values()` when sorting by multiple columns where one column is of type `int64` and contains `NaT` ([GH14922](#))
- Bug in `DataFrame.reindex()` in which method was ignored when passing columns ([GH14992](#))
- Bug in `DataFrame.loc` with indexing a `MultiIndex` with a `Series` indexer ([GH14730](#), [GH15424](#))
- Bug in `DataFrame.loc` with indexing a `MultiIndex` with a numpy array ([GH15434](#))
- Bug in `Series.asof` which raised if the series contained all `np.nan` ([GH15713](#))
- Bug in `.at` when selecting from a tz-aware column ([GH15822](#))
- Bug in `Series.where()` and `DataFrame.where()` where array-like conditionals were being rejected ([GH15414](#))
- Bug in `Series.where()` where TZ-aware data was converted to float representation ([GH15701](#))
- Bug in `.loc` that would not return the correct dtype for scalar access for a `DataFrame` ([GH11617](#))
- Bug in output formatting of a `MultiIndex` when names are integers ([GH12223](#), [GH15262](#))
- Bug in `Categorical.searchsorted()` where alphabetical instead of the provided categorical order was used ([GH14522](#))
- Bug in `Series.iloc` where a `Categorical` object for list-like indexes input was returned, where a `Series` was expected. ([GH14580](#))
- Bug in `DataFrame.isin` comparing datetimelike to empty frame ([GH15473](#))
- Bug in `.reset_index()` when an all `NaN` level of a `MultiIndex` would fail ([GH6322](#))

- Bug in `.reset_index()` when raising error for index name already present in `MultiIndex` columns ([GH16120](#))
- Bug in creating a `MultiIndex` with tuples and not passing a list of names; this will now raise `ValueError` ([GH15110](#))
- Bug in the HTML display with with a `MultiIndex` and truncation ([GH14882](#))
- Bug in the display of `.info()` where a qualifier (+) would always be displayed with a `MultiIndex` that contains only non-strings ([GH15245](#))
- Bug in `pd.concat()` where the names of `MultiIndex` of resulting `DataFrame` are not handled correctly when `None` is presented in the names of `MultiIndex` of input `DataFrame` ([GH15787](#))
- Bug in `DataFrame.sort_index()` and `Series.sort_index()` where `na_position` doesn't work with a `MultiIndex` ([GH14784](#), [GH16604](#))
- Bug in in `pd.concat()` when combining objects with a `CategoricalIndex` ([GH16111](#))
- Bug in indexing with a scalar and a `CategoricalIndex` ([GH16123](#))

I/O

- Bug in `pd.to_numeric()` in which float and unsigned integer elements were being improperly casted ([GH14941](#), [GH15005](#))
- Bug in `pd.read_fwf()` where the `skiprows` parameter was not being respected during column width inference ([GH11256](#))
- Bug in `pd.read_csv()` in which the `dialect` parameter was not being verified before processing ([GH14898](#))
- Bug in `pd.read_csv()` in which missing data was being improperly handled with `usecols` ([GH6710](#))
- Bug in `pd.read_csv()` in which a file containing a row with many columns followed by rows with fewer columns would cause a crash ([GH14125](#))
- Bug in `pd.read_csv()` for the C engine where `usecols` were being indexed incorrectly with `parse_dates` ([GH14792](#))
- Bug in `pd.read_csv()` with `parse_dates` when multi-line headers are specified ([GH15376](#))
- Bug in `pd.read_csv()` with `float_precision='round_trip'` which caused a segfault when a text entry is parsed ([GH15140](#))
- Bug in `pd.read_csv()` when an index was specified and no values were specified as null values ([GH15835](#))
- Bug in `pd.read_csv()` in which certain invalid file objects caused the Python interpreter to crash ([GH15337](#))
- Bug in `pd.read_csv()` in which invalid values for `nrows` and `chunksize` were allowed ([GH15767](#))
- Bug in `pd.read_csv()` for the Python engine in which unhelpful error messages were being raised when parsing errors occurred ([GH15910](#))
- Bug in `pd.read_csv()` in which the `skipfooter` parameter was not being properly validated ([GH15925](#))
- Bug in `pd.to_csv()` in which there was numeric overflow when a timestamp index was being written ([GH15982](#))
- Bug in `pd.util.hashing.hash_pandas_object()` in which hashing of categoricals depended on the ordering of categories, instead of just their values. ([GH15143](#))
- Bug in `.to_json()` where `lines=True` and contents (keys or values) contain escaped characters ([GH15096](#))

- Bug in `.to_json()` causing single byte ascii characters to be expanded to four byte unicode ([GH15344](#))
- Bug in `.to_json()` for the C engine where rollover was not correctly handled for case where frac is odd and diff is exactly 0.5 ([GH15716](#), [GH15864](#))
- Bug in `pd.read_json()` for Python 2 where `lines=True` and contents contain non-ascii unicode characters ([GH15132](#))
- Bug in `pd.read_msgpack()` in which Series categoricals were being improperly processed ([GH14901](#))
- Bug in `pd.read_msgpack()` which did not allow loading of a dataframe with an index of type `CategoricalIndex` ([GH15487](#))
- Bug in `pd.read_msgpack()` when deserializing a `CategoricalIndex` ([GH15487](#))
- Bug in `DataFrame.to_records()` with converting a `DatetimeIndex` with a timezone ([GH13937](#))
- Bug in `DataFrame.to_records()` which failed with unicode characters in column names ([GH11879](#))
- Bug in `.to_sql()` when writing a `DataFrame` with numeric index names ([GH15404](#)).
- Bug in `DataFrame.to_html()` with `index=False` and `max_rows` raising in `IndexError` ([GH14998](#))
- Bug in `pd.read_hdf()` passing a `Timestamp` to the `where` parameter with a non date column ([GH15492](#))
- Bug in `DataFrame.to_stata()` and `StataWriter` which produces incorrectly formatted files to be produced for some locales ([GH13856](#))
- Bug in `StataReader` and `StataWriter` which allows invalid encodings ([GH15723](#))
- Bug in the `Series` repr not showing the length when the output was truncated ([GH15962](#)).

Plotting

- Bug in `DataFrame.hist` where `plt.tight_layout` caused an `AttributeError` (use `matplotlib >= 2.0.1`) ([GH9351](#))
- Bug in `DataFrame.boxplot` where `fontsize` was not applied to the tick labels on both axes ([GH15108](#))
- Bug in the date and time converters pandas registers with matplotlib not handling multiple dimensions ([GH16026](#))
- Bug in `pd.scatter_matrix()` could accept either `color` or `c`, but not both ([GH14855](#))

Groupby/resample/rolling

- Bug in `.groupby(...).resample()` when passed the `on=` kwarg. ([GH15021](#))
- Properly set `__name__` and `__qualname__` for `Groupby.*` functions ([GH14620](#))
- Bug in `GroupBy.get_group()` failing with a categorical grouper ([GH15155](#))
- Bug in `.groupby(...).rolling(...)` when `on` is specified and using a `DatetimeIndex` ([GH15130](#), [GH13966](#))
- Bug in groupby operations with `timedelta64` when passing `numeric_only=False` ([GH5724](#))
- Bug in `groupby.apply()` coercing object dtypes to numeric types, when not all values were numeric ([GH14423](#), [GH15421](#), [GH15670](#))
- Bug in `resample`, where a non-string `loffset` argument would not be applied when resampling a timeseries ([GH13218](#))

- Bug in `DataFrame.groupby().describe()` when grouping on Index containing tuples (GH14848)
- Bug in `groupby().nunique()` with a datetimelike-grouper where bins counts were incorrect (GH13453)
- Bug in `groupby.transform()` that would coerce the resultant dtypes back to the original (GH10972, GH11444)
- Bug in `groupby.agg()` incorrectly localizing timezone on datetime (GH15426, GH10668, GH13046)
- Bug in `.rolling/expanding()` functions where `count()` was not counting `np.Inf`, nor handling object dtypes (GH12541)
- Bug in `.rolling()` where `pd.Timedelta` or `datetime.timedelta` was not accepted as a window argument (GH15440)
- Bug in `Rolling.quantile` function that caused a segmentation fault when called with a quantile value outside of the range `[0, 1]` (GH15463)
- Bug in `DataFrame.resample().median()` if duplicate column names are present (GH14233)

Sparse

- Bug in `SparseSeries.reindex` on single level with list of length 1 (GH15447)
- Bug in repr-formatting a `SparseDataFrame` after a value was set on (a copy of) one of its series (GH15488)
- Bug in `SparseDataFrame` construction with lists not coercing to dtype (GH15682)
- Bug in sparse array indexing in which indices were not being validated (GH15863)

Reshaping

- Bug in `pd.merge_asof()` where `left_index` or `right_index` caused a failure when multiple `by` was specified (GH15676)
- Bug in `pd.merge_asof()` where `left_index/right_index` together caused a failure when `tolerance` was specified (GH15135)
- Bug in `DataFrame.pivot_table()` where `dropna=True` would not drop all-NaN columns when the columns was a category dtype (GH15193)
- Bug in `pd.melt()` where passing a tuple value for `value_vars` caused a `TypeError` (GH15348)
- Bug in `pd.pivot_table()` where no error was raised when `values` argument was not in the columns (GH14938)
- Bug in `pd.concat()` in which concatenating with an empty dataframe with `join='inner'` was being improperly handled (GH15328)
- Bug with `sort=True` in `DataFrame.join` and `pd.merge` when joining on indexes (GH15582)
- Bug in `DataFrame.nsmallest` and `DataFrame.nlargest` where identical values resulted in duplicated rows (GH15297)
- Bug in `pandas.pivot_table()` incorrectly raising `UnicodeError` when passing unicode input for `margins` keyword (GH13292)

Numeric

- Bug in `.rank()` which incorrectly ranks ordered categories (GH15420)
- Bug in `.corr()` and `.cov()` where the column and index were the same object (GH14617)
- Bug in `.mode()` where mode was not returned if was only a single value (GH15714)
- Bug in `pd.cut()` with a single bin on an all 0s array (GH15428)
- Bug in `pd.qcut()` with a single quantile and an array with identical values (GH15431)
- Bug in `pandas.tools.utils.cartesian_product()` with large input can cause overflow on windows (GH15265)
- Bug in `.eval()` which caused multi-line evals to fail with local variables not on the first line (GH15342)

Other

- Compat with SciPy 0.19.0 for testing on `.interpolate()` (GH15662)
- Compat for 32-bit platforms for `.qcut/cut`; bins will now be `int64` dtype (GH14866)
- Bug in interactions with Qt when a `QtApplication` already exists (GH14372)
- Avoid use of `np.finfo()` during `import pandas` removed to mitigate deadlock on Python GIL misuse (GH14641)

Contributors

A total of 204 people contributed patches to this release. People with a “+” by their names contributed a patch for the first time.

- Adam J. Stewart +
- Adrian +
- Ajay Saxena
- Akash Tandon +
- Albert Villanova del Moral +
- Aleksey Bilogur +
- Alexis Mignon +
- Amol Kahat +
- Andreas Winkler +
- Andrew Kittredge +
- Anthonios Partheniou
- Arco Bast +
- Ashish Singal +
- Baurzhan Muftakhidinov +
- Ben Kandel
- Ben Thayer +

- Ben Welsh +
- Bill Chambers +
- Brandon M. Burroughs
- Brian +
- Brian McFee +
- Carlos Souza +
- Chris
- Chris Ham
- Chris Warth
- Christoph Gohlke
- Christoph Paulik +
- Christopher C. Aycock
- Clemens Brunner +
- D.S. McNeil +
- DaanVanHauwermeiren +
- Daniel Himmelstein
- Dave Willmer
- David Cook +
- David Gwynne +
- David Hoffman +
- David Krych
- Diego Fernandez +
- Dimitris Spathis +
- Dmitry L +
- Dody Suria Wijaya +
- Dominik Stanczak +
- Dr-Irv
- Dr. Irv +
- Elliott Sales de Andrade +
- Ennemoser Christoph +
- Francesc Altet +
- Fumito Hamamura +
- Giacomo Ferroni
- Graham R. Jeffries +
- Greg Williams +
- Guilherme Beltramini +

- Guilherme Samora +
- Hao Wu +
- Harshit Patni +
- Ilya V. Schurov +
- Iván Vallés Pérez
- Jackie Leng +
- Jaehoon Hwang +
- James Draper +
- James Goppert +
- James McBride +
- James Santucci +
- Jan Schulz
- Jeff Carey
- Jeff Reback
- JennaVergeynst +
- Jim +
- Jim Crist
- Joe Jevnik
- Joel Nothman +
- John +
- John Tucker +
- John W. O'Brien
- John Zwinck
- Jon M. Mease
- Jon Mease
- Jonathan Whitmore +
- Jonathan de Bruin +
- Joost Kranendonk +
- Joris Van den Bossche
- Joshua Bradt +
- Julian Santander
- Julien Marrec +
- Jun Kim +
- Justin Solinsky +
- Kacawi +
- Kamal Kamalaldin +

- Kerby Shedden
- Kernc
- Keshav Ramaswamy
- Kevin Sheppard
- Kyle Kelley
- Larry Ren
- Leon Yin +
- Line Pedersen +
- Lorenzo Cestaro +
- Luca Scarabello
- Lukasz +
- Mahmoud Lababidi
- Mark Mandel +
- Matt Roeschke
- Matthew Brett
- Matthew Roeschke +
- Matti Picus
- Maximilian Roos
- Michael Charlton +
- Michael Felt
- Michael Lamparski +
- Michiel Stock +
- Mikolaj Chwalisz +
- Min RK
- Miroslav Šedivý +
- Mykola Golubyev
- Nate Yoder
- Nathalie Rud +
- Nicholas Ver Halen
- Nick Chmura +
- Nolan Nichols +
- Pankaj Pandey +
- Pawel Kordek
- Pete Huang +
- Peter +
- Peter Csizsek +

- Petio Petrov +
- Phil Ruffwind +
- Pietro Battiston
- Piotr Chromiec
- Prasanjit Prakash +
- Rob Forgione +
- Robert Bradshaw
- Robin +
- Rodolfo Fernandez
- Roger Thomas
- Rouz Azari +
- Sahil Dua
- Sam Foo +
- Sami Salonen +
- Sarah Bird +
- Sarma Tangirala +
- Scott Sanderson
- Sebastian Bank
- Sebastian Gsänger +
- Shawn Heide
- Shyam Saladi +
- Sinhrks
- Stephen Rauch +
- Sébastien de Menten +
- Tara Adiseshan
- Thiago Serafim
- Thoralf Gutierrez +
- Thrasibule +
- Tobias Gustafsson +
- Tom Augspurger
- Tong SHEN +
- Tong Shen +
- TrigonaMinima +
- Uwe +
- Wes Turner
- Wiktor Tomczak +

- WillAyd
- Yaroslav Halchenko
- Yimeng Zhang +
- abaldenko +
- adrian-stepien +
- alexandercbooth +
- atbd +
- bastewart +
- bmagnusson +
- carlosdanielcsantos +
- chaimdemulder +
- chris-b1
- dickreuter +
- discort +
- dr-leo +
- dubourg
- dwkenefick +
- funnycrab +
- gfyong
- goldenbull +
- hesham.shabana@hotmail.com
- jojomdt +
- linebp +
- manu +
- manuels +
- mattip +
- maxalbert +
- mcocdawc +
- nuffe +
- paul-mannino
- pbreach +
- sakkemo +
- scls19fr
- sinhrks
- stijnvandhoeve +
- the-nose-knows +

- themrmax +
- tomrod +
- tzinckgraf
- wandersoncferreira
- watercrossing +
- wcwagner
- xgdgsc +
- yui-knk

5.8 Version 0.19

5.8.1 v0.19.2 (December 24, 2016)

This is a minor bug-fix release in the 0.19.x series and includes some small regression fixes, bug fixes and performance improvements. We recommend that all users upgrade to this version.

Highlights include:

- Compatibility with Python 3.6
- Added a [Pandas Cheat Sheet](#). (GH13202).

What's new in v0.19.2

- *Enhancements*
- *Performance improvements*
- *Bug fixes*
- *Contributors*

Enhancements

The `pd.merge_asof()`, added in 0.19.0, gained some improvements:

- `pd.merge_asof()` gained `left_index/right_index` and `left_by/right_by` arguments (GH14253)
- `pd.merge_asof()` can take multiple columns in `by` parameter and has specialized dtypes for better performance (GH13936)

Performance improvements

- Performance regression with `PeriodIndex` (GH14822)
- Performance regression in indexing with `getitem` (GH14930)
- Improved performance of `.replace()` (GH12745)
- Improved performance `Series` creation with a datetime index and dictionary data (GH14894)

Bug fixes

- Compat with python 3.6 for pickling of some offsets (GH14685)
- Compat with python 3.6 for some indexing exception types (GH14684, GH14689)
- Compat with python 3.6 for deprecation warnings in the test suite (GH14681)
- Compat with python 3.6 for Timestamp pickles (GH14689)
- Compat with `dateutil==2.6.0`; segfault reported in the testing suite (GH14621)
- Allow nanoseconds in `Timestamp.replace` as a kwarg (GH14621)
- Bug in `pd.read_csv` in which aliasing was being done for `na_values` when passed in as a dictionary (GH14203)
- Bug in `pd.read_csv` in which column indices for a dict-like `na_values` were not being respected (GH14203)
- Bug in `pd.read_csv` where reading files fails, if the number of headers is equal to the number of lines in the file (GH14515)
- Bug in `pd.read_csv` for the Python engine in which an unhelpful error message was being raised when multi-char delimiters were not being respected with quotes (GH14582)
- Fix bugs (GH14734, GH13654) in `pd.read_sas` and `pandas.io.sas.sas7bdat.SAS7BDATReader` that caused problems when reading a SAS file incrementally.
- Bug in `pd.read_csv` for the Python engine in which an unhelpful error message was being raised when `skipfooter` was not being respected by Python's CSV library (GH13879)
- Bug in `.fillna()` in which timezone aware `datetime64` values were incorrectly rounded (GH14872)
- Bug in `.groupby(..., sort=True)` of a non-lexsorted `MultiIndex` when grouping with multiple levels (GH14776)
- Bug in `pd.cut` with negative values and a single bin (GH14652)
- Bug in `pd.to_numeric` where a 0 was not unsigned on a `downcast='unsigned'` argument (GH14401)
- Bug in plotting regular and irregular timeseries using shared axes (`sharex=True` or `ax.twinx()`) (GH13341, GH14322).
- Bug in not propagating exceptions in parsing invalid datetimes, noted in python 3.6 (GH14561)
- Bug in resampling a `DatetimeIndex` in local TZ, covering a DST change, which would raise `AmbiguousTimeError` (GH14682)
- Bug in indexing that transformed `RecursionError` into `KeyError` or `IndexingError` (GH14554)
- Bug in `HDFStore` when writing a `MultiIndex` when using `data_columns=True` (GH14435)
- Bug in `HDFStore.append()` when writing a `Series` and passing a `min_itemsize` argument containing a value for the `index` (GH11412)

- Bug when writing to a `HDFStore` in table format with a `min_itemsize` value for the index and without asking to append ([GH10381](#))
- Bug in `Series.groupby().nunique()` raising an `IndexError` for an empty `Series` ([GH12553](#))
- Bug in `DataFrame.nlargest` and `DataFrame.nsmallest` when the index had duplicate values ([GH13412](#))
- Bug in clipboard functions on linux with python2 with unicode and separators ([GH13747](#))
- Bug in clipboard functions on Windows 10 and python 3 ([GH14362](#), [GH12807](#))
- Bug in `.to_clipboard()` and Excel compat ([GH12529](#))
- Bug in `DataFrame.combine_first()` for integer columns ([GH14687](#)).
- Bug in `pd.read_csv()` in which the `dtype` parameter was not being respected for empty data ([GH14712](#))
- Bug in `pd.read_csv()` in which the `nrows` parameter was not being respected for large input when using the C engine for parsing ([GH7626](#))
- Bug in `pd.merge_asof()` could not handle timezone-aware `DatetimeIndex` when a tolerance was specified ([GH14844](#))
- Explicit check in `to_stata` and `StataWriter` for out-of-range values when writing doubles ([GH14618](#))
- Bug in `.plot(kind='kde')` which did not drop missing values to generate the KDE Plot, instead generating an empty plot. ([GH14821](#))
- Bug in `unstack()` if called with a list of column(s) as an argument, regardless of the dtypes of all columns, they get coerced to `object` ([GH11847](#))

Contributors

A total of 33 people contributed patches to this release. People with a “+” by their names contributed a patch for the first time.

- Ajay Saxena +
- Ben Kandel
- Chris
- Chris Ham +
- Christopher C. Aycock
- Daniel Himmelstein +
- Dave Willmer +
- Dr-Irv
- Jeff Carey +
- Jeff Reback
- Joe Jevnik
- Joris Van den Bossche
- Julian Santander +
- Kerby Shedden
- Keshav Ramaswamy

- Kevin Sheppard
- Luca Scarabello +
- Matt Roeschke +
- Matti Picus +
- Maximilian Roos
- Mykola Golubyev +
- Nate Yoder +
- Nicholas Ver Halen +
- Pawel Kordek
- Pietro Battiston
- Rodolfo Fernandez +
- Tara Adiseshan +
- Tom Augspurger
- Yaroslav Halchenko
- gfyoun
- hesham.shabana@hotmail.com +
- sinhrks
- wandersoncferreira +

5.8.2 v0.19.1 (November 3, 2016)

This is a minor bug-fix release from 0.19.0 and includes some small regression fixes, bug fixes and performance improvements. We recommend that all users upgrade to this version.

What's new in v0.19.1

- *Performance improvements*
- *Bug fixes*
- *Contributors*

Performance improvements

- Fixed performance regression in factorization of Period data ([GH14338](#))
- Fixed performance regression in `Series.asof(where)` when where is a scalar ([GH14461](#))
- Improved performance in `DataFrame.asof(where)` when where is a scalar ([GH14461](#))
- Improved performance in `.to_json()` when `lines=True` ([GH14408](#))
- Improved performance in certain types of *loc* indexing with a MultiIndex ([GH14551](#)).

Bug fixes

- Source installs from PyPI will now again work without `cython` installed, as in previous versions ([GH14204](#))
- Compat with Cython 0.25 for building ([GH14496](#))
- Fixed regression where user-provided file handles were closed in `read_csv` (c engine) ([GH14418](#)).
- Fixed regression in `DataFrame.quantile` when missing values were present in some columns ([GH14357](#)).
- Fixed regression in `Index.difference` where the `freq` of a `DatetimeIndex` was incorrectly set ([GH14323](#))
- Added back `pandas.core.common.array_equivalent` with a deprecation warning ([GH14555](#)).
- Bug in `pd.read_csv` for the C engine in which quotation marks were improperly parsed in skipped rows ([GH14459](#))
- Bug in `pd.read_csv` for Python 2.x in which Unicode quote characters were no longer being respected ([GH14477](#))
- Fixed regression in `Index.append` when categorical indices were appended ([GH14545](#)).
- Fixed regression in `pd.DataFrame` where constructor fails when given dict with `None` value ([GH14381](#))
- Fixed regression in `DatetimeIndex._maybe_cast_slice_bound` when index is empty ([GH14354](#)).
- Bug in localizing an ambiguous timezone when a boolean is passed ([GH14402](#))
- Bug in `TimedeltaIndex` addition with a `Datetime`-like object where addition overflow in the negative direction was not being caught ([GH14068](#), [GH14453](#))
- Bug in string indexing against data with object `Index` may raise `AttributeError` ([GH14424](#))
- Correctly raise `ValueError` on empty input to `pd.eval()` and `df.query()` ([GH13139](#))
- Bug in `RangeIndex.intersection` when result is a empty set ([GH14364](#)).
- Bug in groupby-transform broadcasting that could cause incorrect dtype coercion ([GH14457](#))
- Bug in `Series.__setitem__` which allowed mutating read-only arrays ([GH14359](#)).
- Bug in `DataFrame.insert` where multiple calls with duplicate columns can fail ([GH14291](#))
- `pd.merge()` will raise `ValueError` with non-boolean parameters in passed boolean type arguments ([GH14434](#))
- Bug in `Timestamp` where dates very near the minimum (1677-09) could underflow on creation ([GH14415](#))
- Bug in `pd.concat` where names of the keys were not propagated to the resulting `MultiIndex` ([GH14252](#))
- Bug in `pd.concat` where axis cannot take string parameters 'rows' or 'columns' ([GH14369](#))
- Bug in `pd.concat` with dataframes heterogeneous in length and tuple keys ([GH14438](#))
- Bug in `MultiIndex.set_levels` where illegal level values were still set after raising an error ([GH13754](#))
- Bug in `DataFrame.to_json` where `lines=True` and a value contained a `}` character ([GH14391](#))
- Bug in `df.groupby` causing an `AttributeError` when grouping a single index frame by a column and the index level ([GH14327](#))
- Bug in `df.groupby` where `TypeError` raised when `pd.Grouper(key=...)` is passed in a list ([GH14334](#))
- Bug in `pd.pivot_table` may raise `TypeError` or `ValueError` when index or columns is not scalar and values is not specified ([GH14380](#))

Contributors

A total of 30 people contributed patches to this release. People with a “+” by their names contributed a patch for the first time.

- Adam Chainz +
- Anthonios Partheniou
- Arash Rouhani +
- Ben Kandel
- Brandon M. Burroughs +
- Chris
- Chris Warth
- David Krych +
- Iván Vallés Pérez +
- Jeff Reback
- Joe Jevnik
- Jon M. Mease +
- Jon Mease +
- Joris Van den Bossche
- Josh Owen +
- Keshav Ramaswamy +
- Larry Ren +
- Michael Felt +
- Piotr Chromiec +
- Robert Bradshaw +
- Sinhrks
- Thiago Serafim +
- Tom Bird
- bkandel +
- chris-b1
- dubourg +
- gfyoun
- matrijk +
- paul-mannino +
- sinhrks

5.8.3 v0.19.0 (October 2, 2016)

This is a major release from 0.18.1 and includes number of API changes, several new features, enhancements, and performance improvements along with a large number of bug fixes. We recommend that all users upgrade to this version.

Highlights include:

- `merge_asof()` for asof-style time-series joining, see [here](#)
- `.rolling()` is now time-series aware, see [here](#)
- `read_csv()` now supports parsing Categorical data, see [here](#)
- A function `union_categorical()` has been added for combining categoricals, see [here](#)
- `PeriodIndex` now has its own `period` dtype, and changed to be more consistent with other `Index` classes. See [here](#)
- Sparse data structures gained enhanced support of `int` and `bool` dtypes, see [here](#)
- Comparison operations with `Series` no longer ignores the index, see [here](#) for an overview of the API changes.
- Introduction of a pandas development API for utility functions, see [here](#).
- Deprecation of `Panel4D` and `PanelND`. We recommend to represent these types of n-dimensional data with the `xarray` package.
- Removal of the previously deprecated modules `pandas.io.data`, `pandas.io.wb`, `pandas.tools.rplot`.

Warning: pandas >= 0.19.0 will no longer silence numpy ufunc warnings upon import, see [here](#).

What's new in v0.19.0

- *New features*
 - *merge_asof for asof-style time-series joining*
 - *.rolling() is now time-series aware*
 - *read_csv has improved support for duplicate column names*
 - *read_csv supports parsing Categorical directly*
 - *Categorical concatenation*
 - *Semi-month offsets*
 - *New Index methods*
 - *Google BigQuery Enhancements*
 - *Fine-grained numpy errstate*
 - *get_dummies now returns integer dtypes*
 - *Downcast values to smallest possible dtype in to_numeric*
 - *pandas development API*
 - *Other enhancements*
- *API changes*

- *Series.tolist()* will now return Python types
- *Series operators for different indexes*
 - * *Arithmetic operators*
 - * *Comparison operators*
 - * *Logical operators*
 - * *Flexible comparison methods*
- *Series type promotion on assignment*
- *.to_datetime()* changes
- *Merging changes*
- *.describe()* changes
- *Period changes*
 - * *PeriodIndex now has period dtype*
 - * *Period('NaT') now returns pd.NaT*
 - * *PeriodIndex.values now returns array of Period object*
- *Index +/ - no longer used for set operations*
- *Index.difference and .symmetric_difference changes*
- *Index.unique consistently returns Index*
- *MultiIndex constructors, groupby and set_index preserve categorical dtypes*
- *read_csv will progressively enumerate chunks*
- *Sparse Changes*
 - * *int64 and bool support enhancements*
 - * *Operators now preserve dtypes*
 - * *Other sparse fixes*
- *Indexer dtype changes*
- *Other API changes*
- *Deprecations*
- *Removal of prior version deprecations/changes*
- *Performance improvements*
- *Bug fixes*
- *Contributors*

New features

`merge_asof` for asof-style time-series joining

A long-time requested feature has been added through the `merge_asof()` function, to support asof style joining of time-series ([GH1870](#), [GH13695](#), [GH13709](#), [GH13902](#)). Full documentation is [here](#).

The `merge_asof()` performs an asof merge, which is similar to a left-join except that we match on nearest key rather than equal keys.

```
In [1]: left = pd.DataFrame({'a': [1, 5, 10],
...:                        'left_val': ['a', 'b', 'c']})
...:
...:

In [2]: right = pd.DataFrame({'a': [1, 2, 3, 6, 7],
...:                          'right_val': [1, 2, 3, 6, 7]})
...:
...:

In [3]: left
Out[3]:
   a left_val
0  1         a
1  5         b
2 10         c

[3 rows x 2 columns]

In [4]: right
Out[4]:
   a right_val
0  1          1
1  2          2
2  3          3
3  6          6
4  7          7

[5 rows x 2 columns]
```

We typically want to match exactly when possible, and use the most recent value otherwise.

```
In [5]: pd.merge_asof(left, right, on='a')
Out[5]:
   a left_val right_val
0  1         a         1
1  5         b         3
2 10         c         7

[3 rows x 3 columns]
```

We can also match rows ONLY with prior data, and not an exact match.

```
In [6]: pd.merge_asof(left, right, on='a', allow_exact_matches=False)
Out[6]:
   a left_val right_val
0  1         a      NaN
1  5         b       3.0
2 10         c       7.0
```

(continues on next page)

(continued from previous page)

[3 rows x 3 columns]

In a typical time-series example, we have trades and quotes and we want to asof-join them. This also illustrates using the `by` parameter to group data before merging.

```
In [7]: trades = pd.DataFrame({
...:     'time': pd.to_datetime(['20160525 13:30:00.023',
...:                             '20160525 13:30:00.038',
...:                             '20160525 13:30:00.048',
...:                             '20160525 13:30:00.048',
...:                             '20160525 13:30:00.048']),
...:     'ticker': ['MSFT', 'MSFT',
...:                'GOOG', 'GOOG', 'AAPL'],
...:     'price': [51.95, 51.95,
...:               720.77, 720.92, 98.00],
...:     'quantity': [75, 155,
...:                  100, 100, 100]},
...:     columns=['time', 'ticker', 'price', 'quantity'])
...:
```

```
In [8]: quotes = pd.DataFrame({
...:     'time': pd.to_datetime(['20160525 13:30:00.023',
...:                             '20160525 13:30:00.023',
...:                             '20160525 13:30:00.030',
...:                             '20160525 13:30:00.041',
...:                             '20160525 13:30:00.048',
...:                             '20160525 13:30:00.049',
...:                             '20160525 13:30:00.072',
...:                             '20160525 13:30:00.075']),
...:     'ticker': ['GOOG', 'MSFT', 'MSFT', 'MSFT',
...:                'GOOG', 'AAPL', 'GOOG', 'MSFT'],
...:     'bid': [720.50, 51.95, 51.97, 51.99,
...:             720.50, 97.99, 720.50, 52.01],
...:     'ask': [720.93, 51.96, 51.98, 52.00,
...:             720.93, 98.01, 720.88, 52.03]},
...:     columns=['time', 'ticker', 'bid', 'ask'])
...:
```

In [9]: trades

Out [9]:

	time	ticker	price	quantity
0	2016-05-25 13:30:00.023	MSFT	51.95	75
1	2016-05-25 13:30:00.038	MSFT	51.95	155
2	2016-05-25 13:30:00.048	GOOG	720.77	100
3	2016-05-25 13:30:00.048	GOOG	720.92	100
4	2016-05-25 13:30:00.048	AAPL	98.00	100

[5 rows x 4 columns]

In [10]: quotes

Out [10]:

	time	ticker	bid	ask
0	2016-05-25 13:30:00.023	GOOG	720.50	720.93
1	2016-05-25 13:30:00.023	MSFT	51.95	51.96
2	2016-05-25 13:30:00.030	MSFT	51.97	51.98

(continues on next page)

(continued from previous page)

```

3 2016-05-25 13:30:00.041  MSFT  51.99  52.00
4 2016-05-25 13:30:00.048  GOOG  720.50  720.93
5 2016-05-25 13:30:00.049  AAPL   97.99  98.01
6 2016-05-25 13:30:00.072  GOOG  720.50  720.88
7 2016-05-25 13:30:00.075  MSFT   52.01  52.03

```

[8 rows x 4 columns]

An asof merge joins on the `on`, typically a datetimelike field, which is ordered, and in this case we are using a grouper in the `by` field. This is like a left-outer join, except that forward filling happens automatically taking the most recent non-NaN value.

```

In [11]: pd.merge_asof(trades, quotes,
.....:                 on='time',
.....:                 by='ticker')
.....:

```

Out [11]:

	time	ticker	price	quantity	bid	ask
0	2016-05-25 13:30:00.023	MSFT	51.95	75	51.95	51.96
1	2016-05-25 13:30:00.038	MSFT	51.95	155	51.97	51.98
2	2016-05-25 13:30:00.048	GOOG	720.77	100	720.50	720.93
3	2016-05-25 13:30:00.048	GOOG	720.92	100	720.50	720.93
4	2016-05-25 13:30:00.048	AAPL	98.00	100	NaN	NaN

[5 rows x 6 columns]

This returns a merged DataFrame with the entries in the same order as the original left passed DataFrame (`trades` in this case), with the fields of the quotes merged.

`.rolling()` is now time-series aware

`.rolling()` objects are now time-series aware and can accept a time-series offset (or convertible) for the window argument ([GH13327](#), [GH12995](#)). See the full documentation [here](#).

```

In [12]: dft = pd.DataFrame({'B': [0, 1, 2, np.nan, 4]},
.....:                      index=pd.date_range('20130101 09:00:00',
.....:                      periods=5, freq='s'))
.....:

```

In [13]: dft

Out [13]:

	B
2013-01-01 09:00:00	0.0
2013-01-01 09:00:01	1.0
2013-01-01 09:00:02	2.0
2013-01-01 09:00:03	NaN
2013-01-01 09:00:04	4.0

[5 rows x 1 columns]

This is a regular frequency index. Using an integer window parameter works to roll along the window frequency.

In [14]: dft.rolling(2).sum()

Out [14]:

B

(continues on next page)

(continued from previous page)

```

2013-01-01 09:00:00 NaN
2013-01-01 09:00:01 1.0
2013-01-01 09:00:02 3.0
2013-01-01 09:00:03 NaN
2013-01-01 09:00:04 NaN

[5 rows x 1 columns]

In [15]: dft.rolling(2, min_periods=1).sum()
Out[15]:
           B
2013-01-01 09:00:00  0.0
2013-01-01 09:00:01  1.0
2013-01-01 09:00:02  3.0
2013-01-01 09:00:03  2.0
2013-01-01 09:00:04  4.0

[5 rows x 1 columns]

```

Specifying an offset allows a more intuitive specification of the rolling frequency.

```

In [16]: dft.rolling('2s').sum()
Out[16]:
           B
2013-01-01 09:00:00  0.0
2013-01-01 09:00:01  1.0
2013-01-01 09:00:02  3.0
2013-01-01 09:00:03  2.0
2013-01-01 09:00:04  4.0

[5 rows x 1 columns]

```

Using a non-regular, but still monotonic index, rolling with an integer window does not impart any special calculation.

```

In [17]: dft = pd.DataFrame({'B': [0, 1, 2, np.nan, 4]},
.....:                      index=pd.Index([pd.Timestamp('20130101 09:00:00'),
.....:                                     pd.Timestamp('20130101 09:00:02'),
.....:                                     pd.Timestamp('20130101 09:00:03'),
.....:                                     pd.Timestamp('20130101 09:00:05'),
.....:                                     pd.Timestamp('20130101 09:00:06')],
.....:                                     name='foo'))

In [18]: dft
Out[18]:
           B
foo
2013-01-01 09:00:00  0.0
2013-01-01 09:00:02  1.0
2013-01-01 09:00:03  2.0
2013-01-01 09:00:05  NaN
2013-01-01 09:00:06  4.0

[5 rows x 1 columns]

In [19]: dft.rolling(2).sum()
Out[19]:

```

(continues on next page)