```
In [58]: cat.argsort()
Out[58]: array([2, 0, 1])

In [59]: cat[cat.argsort()]
Out[59]:
[a, b, NaN]
Categories (2, object): [a < b]
```

Column order is preserved when passing a list of dicts to DataFrame

Starting with Python 3.7 the key-order of `dict` is guaranteed. In practice, this has been true since Python 3.6. The *DataFrame* constructor now treats a list of dicts in the same way as it does a list of `OrderedDict`, i.e. preserving the order of the dicts. This change applies only when pandas is running on Python>=3.6 (GH27309).

```
In [60]: data = [
   ....:     {'name': 'Joe', 'state': 'NY', 'age': 18},
   ....:     {'name': 'Jane', 'state': 'KY', 'age': 19, 'hobby': 'Minecraft'},
   ....:     {'name': 'Jean', 'state': 'OK', 'age': 20, 'finances': 'good'}
   ....: ]
   ....:
```

Previous Behavior:

The columns were lexicographically sorted previously,

```
In [1]: pd.DataFrame(data)
Out[1]:
   age finances      hobby  name state
0   18      NaN        NaN   Joe    NY
1   19      NaN  Minecraft  Jane    KY
2   20     good        NaN  Jean    OK
```

New Behavior:

The column order now matches the insertion-order of the keys in the `dict`, considering all the records from top to bottom. As a consequence, the column order of the resulting DataFrame has changed compared to previous pandas versions.

```
In [61]: pd.DataFrame(data)
Out[61]:
   name state  age      hobby finances
0   Joe    NY   18        NaN      NaN
1  Jane    KY   19  Minecraft      NaN
2  Jean    OK   20        NaN     good

[3 rows x 5 columns]
```

Increased minimum versions for dependencies

Due to dropping support for Python 2.7, a number of optional dependencies have updated minimum versions (GH25725, GH24942, GH25752). Independently, some minimum supported versions of dependencies were updated (GH23519, GH25554). If installed, we now require:

Package	Minimum Version	Required
numpy	1.13.3	X
pytz	2015.4	X
python-dateutil	2.6.1	X
bottleneck	1.2.1	
numexpr	2.6.2	
pytest (dev)	4.0.2	

For optional libraries the general recommendation is to use the latest version. The following table lists the lowest version per library that is currently being tested throughout the development of pandas. Optional libraries below the lowest tested version may still work, but are not considered supported.

Package	Minimum Version
beautifulsoup4	4.6.0
fastparquet	0.2.1
gcsfs	0.2.2
lxml	3.8.0
matplotlib	2.2.2
openpyxl	2.4.8
pyarrow	0.9.0
pymysql	0.7.1
pytables	3.4.2
scipy	0.19.0
sqlalchemy	1.1.4
xarray	0.8.2
xlrd	1.1.0
xlsxwriter	0.9.8
xlwt	1.2.0

See *Dependencies* and *Optional dependencies* for more.

Other API changes

- *DatetimeTZDtype* will now standardize pytz timezones to a common timezone instance (GH24713)
- *Timestamp* and *Timedelta* scalars now implement the `to_numpy()` method as aliases to *Timestamp.to_datetime64()* and *Timedelta.to_timedelta64()*, respectively. (GH24653)
- *Timestamp.strptime()* will now rise a `NotImplementedError` (GH25016)
- Comparing *Timestamp* with unsupported objects now returns `NotImplemented` instead of raising `TypeError`. This implies that unsupported rich comparisons are delegated to the other object, and are now consistent with Python 3 behavior for `datetime` objects (GH24011)
- Bug in *DatetimeIndex.snap()* which didn't preserving the `name` of the input *Index* (GH25575)
- The `arg` argument in `pandas.core.groupby.DataFrameGroupBy.agg()` has been renamed to `func` (GH26089)

- The `arg` argument in `pandas.core.window._Window.aggregate()` has been renamed to `func` (GH26372)
- Most Pandas classes had a `__bytes__` method, which was used for getting a python2-style bytestring representation of the object. This method has been removed as a part of dropping Python2 (GH26447)
- The `.str`-accessor has been disabled for 1-level *MultiIndex*, use *MultiIndex.to_flat_index()* if necessary (GH23679)
- Removed support of gtk package for clipboards (GH26563)
- Using an unsupported version of Beautiful Soup 4 will now raise an `ImportError` instead of a `ValueError` (GH27063)
- *Series.to_excel()* and *DataFrame.to_excel()* will now raise a `ValueError` when saving timezone aware data. (GH27008, GH7056)
- `ExtensionArray.argsort()` places NA values at the end of the sorted array. (GH21801)
- *DataFrame.to_hdf()* and *Series.to_hdf()* will now raise a `NotImplementedError` when saving a *MultiIndex* with extension data types for a `fixed` format. (GH7775)
- Passing duplicate `names` in *read_csv()* will now raise a `ValueError` (GH17346)

Deprecations

Sparse subclasses

The `SparseSeries` and `SparseDataFrame` subclasses are deprecated. Their functionality is better-provided by a `Series` or `DataFrame` with sparse values.

Previous way

```
df = pd.SparseDataFrame({"A": [0, 0, 1, 2]})
df.dtypes
```

New way

```
In [62]: df = pd.DataFrame({"A": pd.SparseArray([0, 0, 1, 2])})

In [63]: df.dtypes
Out[63]:
A    Sparse[int64, 0]
Length: 1, dtype: object
```

The memory usage of the two approaches is identical. See *Migrating* for more (GH19239).

msgpack format

The msgpack format is deprecated as of 0.25 and will be removed in a future version. It is recommended to use pyarrow for on-the-wire transmission of pandas objects. (GH27084)

Other deprecations

- The deprecated `.ix[]` indexer now raises a more visible `FutureWarning` instead of `DeprecationWarning` (GH26438).
- Deprecated the `units=M` (months) and `units=Y` (year) parameters for `units` of *pandas.to_timedelta()*, *pandas.Timedelta()* and *pandas.TimedeltaIndex()* (GH16344)
- *pandas.concat()* has deprecated the `join_axes`-keyword. Instead, use *DataFrame.reindex()* or *DataFrame.reindex_like()* on the result or on the inputs (GH21951)
- The `SparseArray.values` attribute is deprecated. You can use `np.asarray(...)` or the `SparseArray.to_dense()` method instead (GH26421).
- The functions *pandas.to_datetime()* and *pandas.to_timedelta()* have deprecated the `box` keyword. Instead, use `to_numpy()` or *Timestamp.to_datetime64()* or *Timedelta.to_timedelta64()*. (GH24416)
- The `DataFrame.compound()` and `Series.compound()` methods are deprecated and will be removed in a future version (GH26405).
- The internal attributes `_start`, `_stop` and `_step` attributes of *RangeIndex* have been deprecated. Use the public attributes *start*, *stop* and *step* instead (GH26581).
- The `Series.ftype()`, `Series.ftypes()` and `DataFrame.ftypes()` methods are deprecated and will be removed in a future version. Instead, use *Series.dtype()* and *DataFrame.dtypes()* (GH26705).
- The `Series.get_values()`, `DataFrame.get_values()`, `Index.get_values()`, `SparseArray.get_values()` and `Categorical.get_values()` methods are deprecated. One of `np.asarray(..)` or *to_numpy()* can be used instead (GH19617).
- The 'outer' method on NumPy ufuncs, e.g. `np.subtract.outer` has been deprecated on *Series* objects. Convert the input to an array with *Series.array* first (GH27186)
- *Timedelta.resolution()* is deprecated and replaced with *Timedelta.resolution_string()*. In a future version, *Timedelta.resolution()* will be changed to behave like the standard library `datetime.timedelta.resolution` (GH21344)
- *read_table()* has been undeprecated. (GH25220)
- `Index.dtype_str` is deprecated. (GH18262)
- `Series.imag` and `Series.real` are deprecated. (GH18262)
- `Series.put()` is deprecated. (GH18262)
- *Index.item()* and *Series.item()* is deprecated. (GH18262)
- The default value `ordered=None` in `CategoricalDtype` has been deprecated in favor of `ordered=False`. When converting between categorical types `ordered=True` must be explicitly passed in order to be preserved. (GH26336)
- `Index.contains()` is deprecated. Use `key in index` (`__contains__`) instead (GH17753).
- `DataFrame.get_dtype_counts()` is deprecated. (GH18262)
- `Categorical.ravel()` will return a *Categorical* instead of a `np.ndarray` (GH27199)

Removal of prior version deprecations/changes

- Removed `Panel` (GH25047, GH25191, GH25231)
- Removed the previously deprecated `sheetname` keyword in *`read_excel()`* (GH16442, GH20938)
- Removed the previously deprecated `TimeGrouper` (GH16942)
- Removed the previously deprecated `parse_cols` keyword in *`read_excel()`* (GH16488)
- Removed the previously deprecated `pd.options.html.border` (GH16970)
- Removed the previously deprecated `convert_objects` (GH11221)
- Removed the previously deprecated `select` method of `DataFrame` and `Series` (GH17633)
- Removed the previously deprecated behavior of *`Series`* treated as list-like in *`rename_categories()`* (GH17982)
- Removed the previously deprecated `DataFrame.reindex_axis` and `Series.reindex_axis` (GH17842)
- Removed the previously deprecated behavior of altering column or index labels with *`Series.rename_axis()`* or *`DataFrame.rename_axis()`* (GH17842)
- Removed the previously deprecated `tupleize_cols` keyword argument in *`read_html()`*, *`read_csv()`*, and *`DataFrame.to_csv()`* (GH17877, GH17820)
- Removed the previously deprecated `DataFrame.from.csv` and `Series.from_csv` (GH17812)
- Removed the previously deprecated `raise_on_error` keyword argument in *`DataFrame.where()`* and *`DataFrame.mask()`* (GH17744)
- Removed the previously deprecated `ordered` and `categories` keyword arguments in `astype` (GH17742)
- Removed the previously deprecated `cdate_range` (GH17691)
- Removed the previously deprecated `True` option for the `dropna` keyword argument in `SeriesGroupBy.nth()` (GH17493)
- Removed the previously deprecated `convert` keyword argument in *`Series.take()`* and *`DataFrame.take()`* (GH17352)
- Removed the previously deprecated behavior of arithmetic operations with `datetime.date` objects (GH21152)

Performance improvements

- Significant speedup in `SparseArray` initialization that benefits most operations, fixing performance regression introduced in v0.20.0 (GH24985)
- *`DataFrame.to_stata()`* is now faster when outputting data with any string or non-native endian columns (GH25045)
- Improved performance of *`Series.searchsorted()`*. The speedup is especially large when the dtype is int8/int16/int32 and the searched key is within the integer bounds for the dtype (GH22034)
- Improved performance of `pandas.core.groupby.GroupBy.quantile()` (GH20405)
- Improved performance of slicing and other selected operation on a *`RangeIndex`* (GH26565, GH26617, GH26722)
- *`RangeIndex`* now performs standard lookup without instantiating an actual hashtable, hence saving memory (GH16685)

- Improved performance of *read_csv()* by faster tokenizing and faster parsing of small float numbers (GH25784)
- Improved performance of *read_csv()* by faster parsing of N/A and boolean values (GH25804)
- Improved performance of `IntervalIndex.is_monotonic`, `IntervalIndex.is_monotonic_increasing` and `IntervalIndex.is_monotonic_decreasing` by removing conversion to *MultiIndex* (GH24813)
- Improved performance of *DataFrame.to_csv()* when writing datetime dtypes (GH25708)
- Improved performance of *read_csv()* by much faster parsing of `MM/YYYY` and `DD/MM/YYYY` datetime formats (GH25922)
- Improved performance of nanops for dtypes that cannot store NaNs. Speedup is particularly prominent for *Series.all()* and *Series.any()* (GH25070)
- Improved performance of *Series.map()* for dictionary mappers on categorical series by mapping the categories instead of mapping all values (GH23785)
- Improved performance of `IntervalIndex.intersection()` (GH24813)
- Improved performance of *read_csv()* by faster concatenating date columns without extra conversion to string for integer/float zero and float `NaN`; by faster checking the string for the possibility of being a date (GH25754)
- Improved performance of `IntervalIndex.is_unique` by removing conversion to `MultiIndex` (GH24813)
- Restored performance of `DatetimeIndex.__iter__()` by re-enabling specialized code path (GH26702)
- Improved performance when building *MultiIndex* with at least one *CategoricalIndex* level (GH22044)
- Improved performance by removing the need for a garbage collect when checking for `SettingWithCopyWarning` (GH27031)
- For *to_datetime()* changed default value of cache parameter to `True` (GH26043)
- Improved performance of *DatetimeIndex* and *PeriodIndex* slicing given non-unique, monotonic data (GH27136).
- Improved performance of `pd.read_json()` for index-oriented data. (GH26773)
- Improved performance of `MultiIndex.shape()` (GH27384).

Bug fixes

Categorical

- Bug in *DataFrame.at()* and *Series.at()* that would raise exception if the index was a *CategoricalIndex* (GH20629)
- Fixed bug in comparison of ordered *Categorical* that contained missing values with a scalar which sometimes incorrectly resulted in `True` (GH26504)
- Bug in *DataFrame.dropna()* when the *DataFrame* has a *CategoricalIndex* containing *Interval* objects incorrectly raised a `TypeError` (GH25087)

Datetimelike

- Bug in *to_datetime()* which would raise an (incorrect) ValueError when called with a date far into the future and the format argument specified instead of raising OutOfBoundsDatetime (GH23830)
- Bug in *to_datetime()* which would raise InvalidIndexError: Reindexing only valid with uniquely valued Index objects when called with cache=True, with arg including at least two different elements from the set {None, numpy.nan, pandas.NaT} (GH22305)
- Bug in *DataFrame* and *Series* where timezone aware data with dtype='datetime64[ns] was not cast to naive (GH25843)
- Improved *Timestamp* type checking in various datetime functions to prevent exceptions when using a subclassed datetime (GH25851)
- Bug in *Series* and *DataFrame* repr where np.datetime64('NaT') and np.timedelta64('NaT') with dtype=object would be represented as NaN (GH25445)
- Bug in *to_datetime()* which does not replace the invalid argument with NaT when error is set to coerce (GH26122)
- Bug in adding DateOffset with nonzero month to *DatetimeIndex* would raise ValueError (GH26258)
- Bug in *to_datetime()* which raises unhandled OverflowError when called with mix of invalid dates and NaN values with format='%Y%m%d' and error='coerce' (GH25512)
- Bug in isin() for datetimelike indexes; *DatetimeIndex*, *TimedeltaIndex* and *PeriodIndex* where the levels parameter was ignored. (GH26675)
- Bug in *to_datetime()* which raises TypeError for format='%Y%m%d' when called for invalid integer dates with length >= 6 digits with errors='ignore'
- Bug when comparing a *PeriodIndex* against a zero-dimensional numpy array (GH26689)
- Bug in constructing a Series or DataFrame from a numpy datetime64 array with a non-ns unit and out-of-bound timestamps generating rubbish data, which will now correctly raise an OutOfBoundsDatetime error (GH26206).
- Bug in *date_range()* with unnecessary OverflowError being raised for very large or very small dates (GH26651)
- Bug where adding *Timestamp* to a np.timedelta64 object would raise instead of returning a *Timestamp* (GH24775)
- Bug where comparing a zero-dimensional numpy array containing a np.datetime64 object to a *Timestamp* would incorrect raise TypeError (GH26916)
- Bug in *to_datetime()* which would raise ValueError: Tz-aware datetime.datetime cannot be converted to datetime64 unless utc=True when called with cache=True, with arg including datetime strings with different offset (GH26097)
-

Timedelta

- Bug in `TimedeltaIndex.intersection()` where for non-monotonic indices in some cases an empty `Index` was returned when in fact an intersection existed (GH25913)
- Bug with comparisons between *`Timedelta`* and `NaT` raising `TypeError` (GH26039)
- Bug when adding or subtracting a `BusinessHour` to a *`Timestamp`* with the resulting time landing in a following or prior day respectively (GH26381)
- Bug when comparing a *`TimedeltaIndex`* against a zero-dimensional numpy array (GH26689)

Timezones

- Bug in *`DatetimeIndex.to_frame()`* where timezone aware data would be converted to timezone naive data (GH25809)
- Bug in *`to_datetime()`* with `utc=True` and datetime strings that would apply previously parsed UTC offsets to subsequent arguments (GH24992)
- Bug in *`Timestamp.tz_localize()`* and *`Timestamp.tz_convert()`* does not propagate `freq` (GH25241)
- Bug in *`Series.at()`* where setting *`Timestamp`* with timezone raises `TypeError` (GH25506)
- Bug in *`DataFrame.update()`* when updating with timezone aware data would return timezone naive data (GH25807)
- Bug in *`to_datetime()`* where an uninformative `RuntimeError` was raised when passing a naive *`Timestamp`* with datetime strings with mixed UTC offsets (GH25978)
- Bug in *`to_datetime()`* with `unit='ns'` would drop timezone information from the parsed argument (GH26168)
- Bug in *`DataFrame.join()`* where joining a timezone aware index with a timezone aware column would result in a column of `NaN` (GH26335)
- Bug in *`date_range()`* where ambiguous or nonexistent start or end times were not handled by the `ambiguous` or `nonexistent` keywords respectively (GH27088)
- Bug in `DatetimeIndex.union()` when combining a timezone aware and timezone unaware *`DatetimeIndex`* (GH21671)
- Bug when applying a numpy reduction function (e.g. `numpy.minimum()`) to a timezone aware *`Series`* (GH15552)

Numeric

- Bug in *`to_numeric()`* in which large negative numbers were being improperly handled (GH24910)
- Bug in *`to_numeric()`* in which numbers were being coerced to float, even though `errors` was not `coerce` (GH24910)
- Bug in *`to_numeric()`* in which invalid values for `errors` were being allowed (GH26466)
- Bug in `format` in which floating point complex numbers were not being formatted to proper display precision and trimming (GH25514)
- Bug in error messages in *`DataFrame.corr()`* and *`Series.corr()`*. Added the possibility of using a callable. (GH25729)

- Bug in *Series.divmod()* and *Series.rdivmod()* which would raise an (incorrect) ValueError rather than return a pair of *Series* objects as result (GH25557)
- Raises a helpful exception when a non-numeric index is sent to interpolate() with methods which require numeric index. (GH21662)
- Bug in *eval()* when comparing floats with scalar operators, for example: x < -0.1 (GH25928)
- Fixed bug where casting all-boolean array to integer extension array failed (GH25211)
- Bug in divmod with a *Series* object containing zeros incorrectly raising AttributeError (GH26987)
- Inconsistency in *Series* floor-division (//) and divmod filling positive//zero with NaN instead of Inf (GH27321)
-

Conversion

- Bug in *DataFrame.astype()* when passing a dict of columns and types the errors parameter was ignored. (GH25905)
-
-

Strings

- Bug in the __name__ attribute of several methods of *Series.str*, which were set incorrectly (GH23551)
- Improved error message when passing *Series* of wrong dtype to *Series.str.cat()* (GH22722)
-

Interval

- Construction of *Interval* is restricted to numeric, *Timestamp* and *Timedelta* endpoints (GH23013)
- Fixed bug in *Series*/*DataFrame* not displaying NaN in *IntervalIndex* with missing values (GH25984)
- Bug in *IntervalIndex.get_loc()* where a KeyError would be incorrectly raised for a decreasing *IntervalIndex* (GH25860)
- Bug in *Index* constructor where passing mixed closed *Interval* objects would result in a ValueError instead of an object dtype Index (GH27172)

Indexing

- Improved exception message when calling *DataFrame.iloc()* with a list of non-numeric objects (GH25753).
- Improved exception message when calling .iloc or .loc with a boolean indexer with different length (GH26658).
- Bug in KeyError exception message when indexing a *MultiIndex* with a non-existent key not displaying the original key (GH27250).

- Bug in `.iloc` and `.loc` with a boolean indexer not raising an `IndexError` when too few items are passed (GH26658).
- Bug in *DataFrame.loc()* and *Series.loc()* where `KeyError` was not raised for a `MultiIndex` when the key was less than or equal to the number of levels in the *MultiIndex* (GH14885).
- Bug in which *DataFrame.append()* produced an erroneous warning indicating that a `KeyError` will be thrown in the future when the data to be appended contains new columns (GH22252).
- Bug in which *DataFrame.to_csv()* caused a segfault for a reindexed data frame, when the indices were single-level *MultiIndex* (GH26303).
- Fixed bug where assigning a *arrays.PandasArray* to a `pandas.core.frame.DataFrame` would raise error (GH26390)
- Allow keyword arguments for callable local reference used in the *DataFrame.query()* string (GH26426)
- Fixed a `KeyError` when indexing a `MultiIndex`` level with a list containing exactly one label, which is missing (GH27148)
- Bug which produced `AttributeError` on partial matching *Timestamp* in a *MultiIndex* (GH26944)
- Bug in *Categorical* and *CategoricalIndex* with *Interval* values when using the `in` operator (`__contains`) with objects that are not comparable to the values in the `Interval` (GH23705)
- Bug in *DataFrame.loc()* and *DataFrame.iloc()* on a *DataFrame* with a single timezone-aware datetime64[ns] column incorrectly returning a scalar instead of a *Series* (GH27110)
- Bug in *CategoricalIndex* and *Categorical* incorrectly raising `ValueError` instead of `TypeError` when a list is passed using the `in` operator (`__contains__`) (GH21729)
- Bug in setting a new value in a *Series* with a *Timedelta* object incorrectly casting the value to an integer (GH22717)
- Bug in *Series* setting a new key (`__setitem__`) with a timezone-aware datetime incorrectly raising `ValueError` (GH12862)
- Bug in *DataFrame.iloc()* when indexing with a read-only indexer (GH17192)
- Bug in *Series* setting an existing tuple key (`__setitem__`) with timezone-aware datetime values incorrectly raising `TypeError` (GH20441)

Missing

- Fixed misleading exception message in *Series.interpolate()* if argument `order` is required, but omitted (GH10633, GH24014).
- Fixed class type displayed in exception message in *DataFrame.dropna()* if invalid `axis` parameter passed (GH25555)
- A `ValueError` will now be thrown by *DataFrame.fillna()* when `limit` is not a positive integer (GH27042)
-

MultiIndex

- Bug in which incorrect exception raised by *Timedelta* when testing the membership of *MultiIndex* (GH24570)
-

I/O

- Bug in *DataFrame.to_html()* where values were truncated using display options instead of outputting the full content (GH17004)
- Fixed bug in missing text when using `to_clipboard()` if copying utf-16 characters in Python 3 on Windows (GH25040)
- Bug in *read_json()* for `orient='table'` when it tries to infer dtypes by default, which is not applicable as dtypes are already defined in the JSON schema (GH21345)
- Bug in *read_json()* for `orient='table'` and float index, as it infers index dtype by default, which is not applicable because index dtype is already defined in the JSON schema (GH25433)
- Bug in *read_json()* for `orient='table'` and string of float column names, as it makes a column name type conversion to *Timestamp*, which is not applicable because column names are already defined in the JSON schema (GH25435)
- Bug in *json_normalize()* for `errors='ignore'` where missing values in the input data, were filled in resulting `DataFrame` with the string `"nan"` instead of `numpy.nan` (GH25468)
- *DataFrame.to_html()* now raises `TypeError` when using an invalid type for the `classes` parameter instead of `AssertionError` (GH25608)
- Bug in *DataFrame.to_string()* and *DataFrame.to_latex()* that would lead to incorrect output when the `header` keyword is used (GH16718)
- Bug in *read_csv()* not properly interpreting the UTF8 encoded filenames on Windows on Python 3.6+ (GH15086)
- Improved performance in *pandas.read_stata()* and `pandas.io.stata.StataReader` when converting columns that have missing values (GH25772)
- Bug in *DataFrame.to_html()* where header numbers would ignore display options when rounding (GH17280)
- Bug in *read_hdf()* where reading a table from an HDF5 file written directly with PyTables fails with a `ValueError` when using a sub-selection via the `start` or `stop` arguments (GH11188)
- Bug in *read_hdf()* not properly closing store after a `KeyError` is raised (GH25766)
- Improved the explanation for the failure when value labels are repeated in Stata dta files and suggested workarounds (GH25772)
- Improved *pandas.read_stata()* and `pandas.io.stata.StataReader` to read incorrectly formatted 118 format files saved by Stata (GH25960)
- Improved the `col_space` parameter in *DataFrame.to_html()* to accept a string so CSS length values can be set correctly (GH25941)
- Fixed bug in loading objects from S3 that contain `#` characters in the URL (GH25945)
- Adds `use_bqstorage_api` parameter to *read_gbq()* to speed up downloads of large data frames. This feature requires version 0.10.0 of the `pandas-gbq` library as well as the `google-cloud-bigquery-storage` and `fastavro` libraries. (GH26104)

- Fixed memory leak in `DataFrame.to_json()` when dealing with numeric data (GH24889)
- Bug in `read_json()` where date strings with `Z` were not converted to a UTC timezone (GH26168)
- Added `cache_dates=True` parameter to `read_csv()`, which allows to cache unique dates when they are parsed (GH25990)
- `DataFrame.to_excel()` now raises a `ValueError` when the caller's dimensions exceed the limitations of Excel (GH26051)
- Fixed bug in `pandas.read_csv()` where a BOM would result in incorrect parsing using engine='python' (GH26545)
- `read_excel()` now raises a `ValueError` when input is of type `pandas.io.excel.ExcelFile` and `engine` param is passed since `pandas.io.excel.ExcelFile` has an engine defined (GH26566)
- Bug while selecting from `HDFStore` with `where=''` specified (GH26610).
- Fixed bug in `DataFrame.to_excel()` where custom objects (i.e. *PeriodIndex*) inside merged cells were not being converted into types safe for the Excel writer (GH27006)
- Bug in `read_hdf()` where reading a timezone aware `DatetimeIndex` would raise a `TypeError` (GH11926)
- Bug in `to_msgpack()` and `read_msgpack()` which would raise a `ValueError` rather than a `FileNotFoundError` for an invalid path (GH27160)
- Fixed bug in `DataFrame.to_parquet()` which would raise a `ValueError` when the dataframe had no columns (GH27339)
- Allow parsing of `PeriodDtype` columns when using `read_csv()` (GH26934)

Plotting

- Fixed bug where `api.extensions.ExtensionArray` could not be used in matplotlib plotting (GH25587)
- Bug in an error message in `DataFrame.plot()`. Improved the error message if non-numerics are passed to `DataFrame.plot()` (GH25481)
- Bug in incorrect ticklabel positions when plotting an index that are non-numeric / non-datetime (GH7612, GH15912, GH22334)
- Fixed bug causing plots of `PeriodIndex` timeseries to fail if the frequency is a multiple of the frequency rule code (GH14763)
- Fixed bug when plotting a `DatetimeIndex` with `datetime.timezone.utc` timezone (GH17173)
-
-

Groupby/resample/rolling

- Bug in `pandas.core.resample.Resampler.agg()` with a timezone aware index where `OverflowError` would raise when passing a list of functions (GH22660)
- Bug in *`pandas.core.groupby.DataFrameGroupBy.nunique()`* in which the names of column levels were lost (GH23222)
- Bug in *`pandas.core.groupby.GroupBy.agg()`* when applying an aggregation function to timezone aware data (GH23683)
- Bug in *`pandas.core.groupby.GroupBy.first()`* and *`pandas.core.groupby.GroupBy.last()`* where timezone information would be dropped (GH21603)
- Bug in *`pandas.core.groupby.GroupBy.size()`* when grouping only NA values (GH23050)
- Bug in *`Series.groupby()`* where `observed` kwarg was previously ignored (GH24880)
- Bug in *`Series.groupby()`* where using `groupby` with a *`MultiIndex`* Series with a list of labels equal to the length of the series caused incorrect grouping (GH25704)
- Ensured that ordering of outputs in `groupby` aggregation functions is consistent across all versions of Python (GH25692)
- Ensured that result group order is correct when grouping on an ordered `Categorical` and specifying `observed=True` (GH25871, GH25167)
- Bug in `pandas.core.window.Rolling.min()` and `pandas.core.window.Rolling.max()` that caused a memory leak (GH25893)
- Bug in `pandas.core.window.Rolling.count()` and `pandas.core.window.Expanding.count` was previously ignoring the `axis` keyword (GH13503)
- Bug in `pandas.core.groupby.GroupBy.idxmax()` and `pandas.core.groupby.GroupBy.idxmin()` with datetime column would return incorrect dtype (GH25444, GH15306)
- Bug in *`pandas.core.groupby.GroupBy.cumsum()`*, *`pandas.core.groupby.GroupBy.cumprod()`*, *`pandas.core.groupby.GroupBy.cummin()`* and *`pandas.core.groupby.GroupBy.cummax()`* with categorical column having absent categories, would return incorrect result or segfault (GH16771)
- Bug in *`pandas.core.groupby.GroupBy.nth()`* where NA values in the grouping would return incorrect results (GH26011)
- Bug in `pandas.core.groupby.SeriesGroupBy.transform()` where transforming an empty group would raise a `ValueError` (GH26208)
- Bug in `pandas.core.frame.DataFrame.groupby()` where passing a `pandas.core.groupby.grouper.Grouper` would return incorrect groups when using the `.groups` accessor (GH26326)
- Bug in *`pandas.core.groupby.GroupBy.agg()`* where incorrect results are returned for uint64 columns. (GH26310)
- Bug in `pandas.core.window.Rolling.median()` and `pandas.core.window.Rolling.quantile()` where MemoryError is raised with empty window (GH26005)
- Bug in `pandas.core.window.Rolling.median()` and `pandas.core.window.Rolling.quantile()` where incorrect results are returned with `closed='left'` and `closed='neither'` (GH26005)
- Improved `pandas.core.window.Rolling`, `pandas.core.window.Window` and `pandas.core.window.EWM` functions to exclude nuisance columns from results instead of raising errors and raise a `DataError` only if all columns are nuisance (GH12537)

- Bug in `pandas.core.window.Rolling.max()` and `pandas.core.window.Rolling.min()` where incorrect results are returned with an empty variable window (GH26005)
- Raise a helpful exception when an unsupported weighted window function is used as an argument of `pandas.core.window.Window.aggregate()` (GH26597)

Reshaping

- Bug in *pandas.merge()* adds a string of `None`, if `None` is assigned in suffixes instead of remain the column name as-is (GH24782).
- Bug in *merge()* when merging by index name would sometimes result in an incorrectly numbered index (missing index values are now assigned NA) (GH24212, GH25009)
- `to_records()` now accepts dtypes to its `column_dtypes` parameter (GH24895)
- Bug in *concat()* where order of `OrderedDict` (and `dict` in Python 3.6+) is not respected, when passed in as `objs` argument (GH21510)
- Bug in *pivot_table()* where columns with `NaN` values are dropped even if `dropna` argument is `False`, when the `aggfunc` argument contains a `list` (GH22159)
- Bug in *concat()* where the resulting `freq` of two *DatetimeIndex* with the same `freq` would be dropped (GH3232).
- Bug in *merge()* where merging with equivalent Categorical dtypes was raising an error (GH22501)
- bug in *DataFrame* instantiating with a dict of iterators or generators (e.g. `pd.DataFrame({'A': reversed(range(3))})`) raised an error (GH26349).
- Bug in *DataFrame* instantiating with a `range` (e.g. `pd.DataFrame(range(3))`) raised an error (GH26342).
- Bug in *DataFrame* constructor when passing non-empty tuples would cause a segmentation fault (GH25691)
- Bug in *Series.apply()* failed when the series is a timezone aware *DatetimeIndex* (GH25959)
- Bug in *pandas.cut()* where large bins could incorrectly raise an error due to an integer overflow (GH26045)
- Bug in *DataFrame.sort_index()* where an error is thrown when a multi-indexed `DataFrame` is sorted on all levels with the initial level sorted last (GH26053)
- Bug in *Series.nlargest()* treats `True` as smaller than `False` (GH26154)
- Bug in *DataFrame.pivot_table()* with a *IntervalIndex* as pivot index would raise `TypeError` (GH25814)
- Bug in which *DataFrame.from_dict()* ignored order of `OrderedDict` when `orient='index'` (GH8425).
- Bug in *DataFrame.transpose()* where transposing a DataFrame with a timezone-aware datetime column would incorrectly raise `ValueError` (GH26825)
- Bug in *pivot_table()* when pivoting a timezone aware column as the `values` would remove timezone information (GH14948)
- Bug in *merge_asof()* when specifying multiple `by` columns where one is `datetime64[ns, tz]` dtype (GH26649)

Sparse

- Significant speedup in `SparseArray` initialization that benefits most operations, fixing performance regression introduced in v0.20.0 (GH24985)
- Bug in `SparseFrame` constructor where passing `None` as the data would cause `default_fill_value` to be ignored (GH16807)
- Bug in `SparseDataFrame` when adding a column in which the length of values does not match length of index, `AssertionError` is raised instead of raising `ValueError` (GH25484)
- Introduce a better error message in *`Series.sparse.from_coo()`* so it returns a `TypeError` for inputs that are not coo matrices (GH26554)
- Bug in `numpy.modf()` on a `SparseArray`. Now a tuple of `SparseArray` is returned (GH26946).

Build Changes

- Fix install error with PyPy on macOS (GH26536)

ExtensionArray

- Bug in *`factorize()`* when passing an `ExtensionArray` with a custom `na_sentinel` (GH25696).
- *`Series.count()`* miscounts NA values in ExtensionArrays (GH26835)
- Added `Series.__array_ufunc__` to better handle NumPy ufuncs applied to Series backed by extension arrays (GH23293).
- Keyword argument `deep` has been removed from `ExtensionArray.copy()` (GH27083)

Other

- Removed unused C functions from vendored UltraJSON implementation (GH26198)
- Allow *`Index`* and *`RangeIndex`* to be passed to numpy `min` and `max` functions (GH26125)
- Use actual class name in repr of empty objects of a `Series` subclass (GH27001).
- Bug in *`DataFrame`* where passing an object array of timezone-aware *datetime* objects would incorrectly raise `ValueError` (GH13287)

Contributors

A total of 231 people contributed patches to this release. People with a "+" by their names contributed a patch for the first time.

- 1_x7 +
- Abdullah İhsan Seçer +
- Adam Bull +
- Adam Hooper
- Albert Villanova del Moral
- Alex Watt +

- AlexTereshenkov +
- Alexander Buchkovsky
- Alexander Hendorf +
- Alexander Nordin +
- Alexander Ponomaroff
- Alexandre Batisse +
- Alexandre Decan +
- Allen Downey +
- Alyssa Fu Ward +
- Andrew Gaspari +
- Andrew Wood +
- Antoine Viscardi +
- Antonio Gutierrez +
- Arno Veenstra +
- ArtinSarraf
- Batalex +
- Baurzhan Muftakhidinov
- Benjamin Rowell
- Bharat Raghunathan +
- Bhavani Ravi +
- Big Head +
- Brett Randall +
- Bryan Cutler +
- C John Klehm +
- Caleb Braun +
- Cecilia +
- Chris Bertinato +
- Chris Stadler +
- Christian Haege +
- Christian Hudon
- Christopher Whelan
- Chuanzhu Xu +
- Clemens Brunner
- Damian Kula +
- Daniel Hrisca +
- Daniel Luis Costa +

- Daniel Saxton
- DanielFEvans +
- David Liu +
- Deepyaman Datta +
- Denis Belavin +
- Devin Petersohn +
- Diane Trout +
- EdAbati +
- Enrico Rotundo +
- EternalLearner42 +
- Evan +
- Evan Livelo +
- Fabian Rost +
- Flavien Lambert +
- Florian Rathgeber +
- Frank Hoang +
- Gaibo Zhang +
- Gioia Ballin
- Giuseppe Romagnuolo +
- Gordon Blackadder +
- Gregory Rome +
- Guillaume Gay
- HHest +
- Hielke Walinga +
- How Si Wei +
- Hubert
- Huize Wang +
- Hyukjin Kwon +
- Ian Dunn +
- Inevitable-Marzipan +
- Irv Lustig
- JElfner +
- Jacob Bundgaard +
- James Cobon-Kerr +
- Jan-Philip Gehrcke +
- Jarrod Millman +

- Jayanth Katuri +
- Jeff Reback
- Jeremy Schendel
- Jiang Yue +
- Joel Ostblom
- Johan von Forstner +
- Johnny Chiu +
- Jonas +
- Jonathon Vandezande +
- Jop Vermeer +
- Joris Van den Bossche
- Josh
- Josh Friedlander +
- Justin Zheng
- Kaiqi Dong
- Kane +
- Kapil Patel +
- Kara de la Marck +
- Katherine Surta +
- Katrin Leinweber +
- Kendall Masse
- Kevin Sheppard
- Kyle Kosic +
- Lorenzo Stella +
- Maarten Rietbergen +
- Mak Sze Chun
- Marc Garcia
- Mateusz Woś
- Matias Heikkilä
- Mats Maiwald +
- Matthew Roeschke
- Max Bolingbroke +
- Max Kovalovs +
- Max van Deursen +
- Michael
- Michael Davis +

- Michael P. Moran +
- Mike Cramblett +
- Min ho Kim +
- Misha Veldhoen +
- Mukul Ashwath Ram +
- MusTheDataGuy +
- Nanda H Krishna +
- Nicholas Musolino
- Noam Hershtig +
- Noora Husseini +
- Paul
- Paul Reidy
- Pauli Virtanen
- Pav A +
- Peter Leimbigler +
- Philippe Ombredanne +
- Pietro Battiston
- Richard Eames +
- Roman Yurchak
- Ruijing Li
- Ryan
- Ryan Joyce +
- Ryan Nazareth
- Ryan Rehman +
- Sakar Panta +
- Samuel Sinayoko
- Sandeep Pathak +
- Sangwoong Yoon
- Saurav Chakravorty
- Scott Talbert +
- Sergey Kopylov +
- Shantanu Gontia +
- Shivam Rana +
- Shorokhov Sergey +
- Simon Hawkins
- Soyoun(Rose) Kim

- Stephan Hoyer
- Stephen Cowley +
- Stephen Rauch
- Sterling Paramore +
- Steven +
- Stijn Van Hoey
- Sumanau Sareen +
- Takuya N +
- Tan Tran +
- Tao He +
- Tarbo Fukazawa
- Terji Petersen +
- Thein Oo
- ThibTrip +
- Thijs Damsma +
- Thiviyan Thanapalasingam
- Thomas A Caswell
- Thomas Kluiters +
- Tilen Kusterle +
- Tim Gates +
- Tim Hoffmann
- Tim Swast
- Tom Augspurger
- Tom Neep +
- Tomáš Chvátal +
- Tyler Reddy
- Vaibhav Vishal +
- Vasily Litvinov +
- Vibhu Agarwal +
- Vikramjeet Das +
- Vladislav +
- Víctor Moron Tejero +
- Wenhuan
- Will Ayd +
- William Ayd
- Wouter De Coster +

- Yoann Goular +
- Zach Angell +
- alimcmaster1
- anmyachev +
- chris-b1
- danielplawrence +
- endenis +
- enisnazif +
- ezcitron +
- fjetter
- froessler
- gfyoung
- gwrome +
- h-vetinari
- haison +
- hannah-c +
- heckeop +
- iamshwin +
- jamesoliverh +
- jbrockmendel
- jkovacevic +
- killerontherun1 +
- knuu +
- kpapdac +
- kpflugshaupt +
- krsnik93 +
- leerssej +
- lrjball +
- mazayo +
- nathalier +
- nrebena +
- nullptr +
- pilkibun +
- pmaxey83 +
- rbenes +
- robbuckley

- shawnbrown +
- sudhir mohanraj +
- tadeja +
- tamuhey +
- thatneat
- topper-123
- willweil +
- yehia67 +
- yhaque1213 +

5.3 Version 0.24

5.3.1 Whats new in 0.24.2 (March 12, 2019)

Warning: The 0.24.x series of releases will be the last to support Python 2. Future feature releases will support Python 3 only. See Dropping Python 2.7 for more.

These are the changes in pandas 0.24.2. See *Release Notes* for a full changelog including other versions of pandas.

Fixed regressions

- Fixed regression in *DataFrame.all()* and *DataFrame.any()* where `bool_only=True` was ignored (GH25101)
- Fixed issue in `DataFrame` construction with passing a mixed list of mixed types could segfault. (GH25075)
- Fixed regression in *DataFrame.apply()* causing `RecursionError` when `dict`-like classes were passed as argument. (GH25196)
- Fixed regression in *DataFrame.replace()* where `regex=True` was only replacing patterns matching the start of the string (GH25259)
- Fixed regression in *DataFrame.duplicated()*, where empty dataframe was not returning a boolean dtyped Series. (GH25184)
- Fixed regression in *Series.min()* and *Series.max()* where `numeric_only=True` was ignored when the `Series` contained `Categorical` data (GH25299)
- Fixed regression in subtraction between *Series* objects with `datetime64[ns]` dtype incorrectly raising `OverflowError` when the `Series` on the right contains null values (GH25317)
- Fixed regression in *TimedeltaIndex* where `np.sum(index)` incorrectly returned a zero-dimensional object instead of a scalar (GH25282)
- Fixed regression in `IntervalDtype` construction where passing an incorrect string with 'Interval' as a prefix could result in a `RecursionError`. (GH25338)
- Fixed regression in creating a period-dtype array from a read-only NumPy array of period objects. (GH25403)

- Fixed regression in *Categorical*, where constructing it from a categorical Series and an explicit categories= that differed from that in the Series created an invalid object which could trigger segfaults. (GH25318)
- Fixed regression in *to_timedelta()* losing precision when converting floating data to Timedelta data (GH25077).
- Fixed pip installing from source into an environment without NumPy (GH25193)
- Fixed regression in *DataFrame.replace()* where large strings of numbers would be coerced into int64, causing an OverflowError (GH25616)
- Fixed regression in *factorize()* when passing a custom na_sentinel value with sort=True (GH25409).
- Fixed regression in *DataFrame.to_csv()* writing duplicate line endings with gzip compress (GH25311)

Bug fixes

I/O

- Better handling of terminal printing when the terminal dimensions are not known (GH25080)
- Bug in reading a HDF5 table-format DataFrame created in Python 2, in Python 3 (GH24925)
- Bug in reading a JSON with orient='table' generated by *DataFrame.to_json()* with index=False (GH25170)
- Bug where float indexes could have misaligned values when printing (GH25061)

Categorical

- Bug where calling *Series.replace()* on categorical data could return a Series with incorrect dimensions (GH24971)
-
-

Reshaping

- Bug in *transform()* where applying a function to a timezone aware column would return a timezone naive result (GH24198)
- Bug in *DataFrame.join()* when joining on a timezone aware *DatetimeIndex* (GH23931)

Visualization

- Bug in *Series.plot()* where a secondary y axis could not be set to log scale (GH25545)

Other

- Bug in *Series.is_unique()* where single occurrences of NaN were not considered unique (GH25180)
- Bug in *merge()* when merging an empty DataFrame with an Int64 column or a non-empty DataFrame with an Int64 column that is all NaN (GH25183)
- Bug in IntervalTree where a RecursionError occurs upon construction due to an overflow when adding endpoints, which also causes *IntervalIndex* to crash during indexing operations (GH25485)
- Bug in *Series.size* raising for some extension-array-backed Series, rather than returning the size (GH25580)
- Bug in resampling raising for nullable integer-dtype columns (GH25580)

Contributors

A total of 25 people contributed patches to this release. People with a "+" by their names contributed a patch for the first time.

- Albert Villanova del Moral
- Arno Veenstra +
- chris-b1
- Devin Petersohn +
- EternalLearner42 +
- Flavien Lambert +
- gfyoung
- Gioia Ballin
- jbrockmendel
- Jeff Reback
- Jeremy Schendel
- Johan von Forstner +
- Joris Van den Bossche
- Josh
- Justin Zheng
- Kendall Masse
- Matthew Roeschke
- Max Bolingbroke +
- rbenes +
- Sterling Paramore +
- Tao He +
- Thomas A Caswell
- Tom Augspurger
- Vibhu Agarwal +
- William Ayd
- Zach Angell

5.3.2 Whats new in 0.24.1 (February 3, 2019)

Warning: The 0.24.x series of releases will be the last to support Python 2. Future feature releases will support Python 3 only. See Dropping Python 2.7 for more.

These are the changes in pandas 0.24.1. See *Release Notes* for a full changelog including other versions of pandas. See *What's new in 0.24.0 (January 25, 2019)* for the 0.24.0 changelog.

API changes

Changing the `sort` parameter for `Index` set operations

The default `sort` value for `Index.union()` has changed from `True` to `None` (GH24959). The default *behavior*, however, remains the same: the result is sorted, unless

1. `self` and `other` are identical
2. `self` or `other` is empty
3. `self` or `other` contain values that can not be compared (a `RuntimeWarning` is raised).

This change will allow `sort=True` to mean "always sort" in a future release.

The same change applies to `Index.difference()` and `Index.symmetric_difference()`, which would not sort the result when the values could not be compared.

The *sort* option for `Index.intersection()` has changed in three ways.

1. The default has changed from `True` to `False`, to restore the pandas 0.23.4 and earlier behavior of not sorting by default.
2. The behavior of `sort=True` can now be obtained with `sort=None`. This will sort the result only if the values in `self` and `other` are not identical.
3. The value `sort=True` is no longer allowed. A future version of pandas will properly support `sort=True` meaning "always sort".

Fixed regressions

- Fixed regression in `DataFrame.to_dict()` with `records` orient raising an `AttributeError` when the `DataFrame` contained more than 255 columns, or wrongly converting column names that were not valid python identifiers (GH24939, GH24940).
- Fixed regression in `read_sql()` when passing certain queries with MySQL/pymysql (GH24988).
- Fixed regression in `Index.intersection` incorrectly sorting the values by default (GH24959).
- Fixed regression in `merge()` when merging an empty `DataFrame` with multiple timezone-aware columns on one of the timezone-aware columns (GH25014).
- Fixed regression in `Series.rename_axis()` and `DataFrame.rename_axis()` where passing `None` failed to remove the axis name (GH25034)
- Fixed regression in `to_timedelta()` with *box=False* incorrectly returning a `datetime64` object instead of a `timedelta64` object (GH24961)
- Fixed regression where custom hashable types could not be used as column keys in `DataFrame.set_index()` (GH24969)