Installation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Package** | **Version** | **Note** | **Python Version** | **Install Command** |
| pandas |  |  |  |  |
| NumPy |  |  |  |  |
| setuptools |  |  |  |  |
| python-dateutil |  |  |  |  |
| pytz |  |  |  |  |
| numexpr |  |  |  |  |
| PyMySQL |  |  |  |  |
| SQLAlchemy |  |  |  |  |
| argparse |  |  |  |  |
| loguru |  |  |  |  |
| matplotlib |  |  |  |  |
| plotly |  |  |  |  |
| orca |  |  |  |  |
| requests |  |  |  |  |
| seaborn |  |  |  |  |
| scipy |  |  |  |  |
| Scikit-learn |  |  |  |  |
| Unittest2 |  |  |  |  |
| pip |  |  |  |  |

The easiest way to install pandas is to install it as part of the [Anaconda](https://docs.continuum.io/anaconda/) distribution, a cross platform distribution for data analysis and scientific computing. This is the recommended installation method for most users.

Instructions for installing from source, [PyPI](https://pypi.org/project/pandas), [ActivePython](https://www.activestate.com/activepython/downloads), various Linux distributions, or a [development version](https://github.com/pandas-dev/pandas) are also provided.

Python version support

Officially Python 3.6.1 and above, 3.7, and 3.8.

Installing pandas

Installing with Anaconda

Installing pandas and the rest of the [NumPy](https://www.numpy.org/) and [SciPy](https://www.scipy.org/) stack can be a little difficult for inexperienced users.

The simplest way to install not only pandas, but Python and the most popular packages that make up the [SciPy](https://www.scipy.org/) stack ([IPython](https://ipython.org/), [NumPy](https://www.numpy.org/), [Matplotlib](https://matplotlib.org/), …) is with [Anaconda](https://docs.continuum.io/anaconda/), a cross-platform (Linux, Mac OS X, Windows) Python distribution for data analytics and scientific computing.

After running the installer, the user will have access to pandas and the rest of the [SciPy](https://www.scipy.org/) stack without needing to install anything else, and without needing to wait for any software to be compiled.

Installation instructions for [Anaconda](https://docs.continuum.io/anaconda/) [can be found here](https://docs.continuum.io/anaconda/install.html).

A full list of the packages available as part of the [Anaconda](https://docs.continuum.io/anaconda/) distribution [can be found here](https://docs.continuum.io/anaconda/packages/pkg-docs/).

Another advantage to installing Anaconda is that you don’t need admin rights to install it. Anaconda can install in the user’s home directory, which makes it trivial to delete Anaconda if you decide (just delete that folder).

Installing with Miniconda

The previous section outlined how to get pandas installed as part of the [Anaconda](https://docs.continuum.io/anaconda/) distribution. However this approach means you will install well over one hundred packages and involves downloading the installer which is a few hundred megabytes in size.

If you want to have more control on which packages, or have a limited internet bandwidth, then installing pandas with [Miniconda](https://conda.pydata.org/miniconda.html) may be a better solution.

[Conda](https://conda.pydata.org/docs/) is the package manager that the [Anaconda](https://docs.continuum.io/anaconda/) distribution is built upon. It is a package manager that is both cross-platform and language agnostic (it can play a similar role to a pip and virtualenv combination).

[Miniconda](https://conda.pydata.org/miniconda.html) allows you to create a minimal self contained Python installation, and then use the [Conda](https://conda.pydata.org/docs/) command to install additional packages.

First you will need [Conda](https://conda.pydata.org/docs/) to be installed and downloading and running the [Miniconda](https://conda.pydata.org/miniconda.html) will do this for you. The installer [can be found here](https://conda.pydata.org/miniconda.html)

The next step is to create a new conda environment. A conda environment is like a virtualenv that allows you to specify a specific version of Python and set of libraries. Run the following commands from a terminal window:

conda create -n name\_of\_my\_env python

This will create a minimal environment with only Python installed in it. To put your self inside this environment run:

source activate name\_of\_my\_env

On Windows the command is:

activate name\_of\_my\_env

The final step required is to install pandas. This can be done with the following command:

conda install pandas

To install a specific pandas version:

conda install pandas=0.20.3

To install other packages, IPython for example:

conda install ipython

To install the full [Anaconda](https://docs.continuum.io/anaconda/) distribution:

conda install anaconda

If you need packages that are available to pip but not conda, then install pip, and then use pip to install those packages:

conda install pip pip install django

Installing from PyPI

pandas can be installed via pip from [PyPI](https://pypi.org/project/pandas).

pip install pandas

Installing with ActivePython

Installation instructions for [ActivePython](https://www.activestate.com/activepython) can be found [here](https://www.activestate.com/activepython/downloads). Versions 2.7, 3.5 and 3.6 include pandas.

Installing using your Linux distribution’s package manager.

The commands in this table will install pandas for Python 3 from your distribution.

| **Distribution** | **Status** | **Download / Repository Link** | **Install method** |
| --- | --- | --- | --- |
| Debian | stable | [official Debian repository](https://packages.debian.org/search?keywords=pandas&searchon=names&suite=all&section=all) | sudo apt-get install python3-pandas |
| Debian & Ubuntu | unstable (latest packages) | [NeuroDebian](http://neuro.debian.net/index.html#how-to-use-this-repository) | sudo apt-get install python3-pandas |
| Ubuntu | stable | [official Ubuntu repository](https://packages.ubuntu.com/search?keywords=pandas&searchon=names&suite=all&section=all) | sudo apt-get install python3-pandas |
| OpenSuse | stable | [OpenSuse Repository](https://software.opensuse.org/package/python-pandas?search_term=pandas) | zypper in python3-pandas |
| Fedora | stable | [official Fedora repository](https://admin.fedoraproject.org/pkgdb/package/rpms/python-pandas/) | dnf install python3-pandas |
| Centos/RHEL | stable | [EPEL repository](https://admin.fedoraproject.org/pkgdb/package/rpms/python-pandas/) | yum install python3-pandas |

**However**, the packages in the linux package managers are often a few versions behind, so to get the newest version of pandas, it’s recommended to install using the pip or conda methods described above.

Handling ImportErrors

If you encounter an ImportError, it usually means that Python couldn’t find pandas in the list of available libraries. Python internally has a list of directories it searches through, to find packages. You can obtain these directories with:

**import** **sys** sys.path

One way you could be encountering this error is if you have multiple Python installations on your system and you don’t have pandas installed in the Python installation you’re currently using. In Linux/Mac you can run which python on your terminal and it will tell you which Python installation you’re using. If it’s something like “/usr/bin/python”, you’re using the Python from the system, which is not recommended.

It is highly recommended to use conda, for quick installation and for package and dependency updates. You can find simple installation instructions for pandas in this document: *installation instructions </getting\_started.html>*.

Installing from source

See the [contributing guide](https://pandas.pydata.org/pandas-docs/stable/development/contributing.html#contributing) for complete instructions on building from the git source tree. Further, see [creating a development environment](https://pandas.pydata.org/pandas-docs/stable/development/contributing.html#contributing-dev-env) if you wish to create a *pandas* development environment.

Running the test suite

pandas is equipped with an exhaustive set of unit tests, covering about 97% of the code base as of this writing. To run it on your machine to verify that everything is working (and that you have all of the dependencies, soft and hard, installed), make sure you have [pytest](https://docs.pytest.org/en/latest/) >= 5.0.1 and [Hypothesis](https://hypothesis.readthedocs.io/) >= 3.58, then run:

**>>>** pd.test() running: pytest --skip-slow --skip-network C:\Users\TP\Anaconda3\envs\py36\lib\site-packages\pandas ============================= test session starts ============================= platform win32 -- Python 3.6.2, pytest-3.6.0, py-1.4.34, pluggy-0.4.0 rootdir: C:\Users\TP\Documents\Python\pandasdev\pandas, inifile: setup.cfg collected 12145 items / 3 skipped ..................................................................S...... ........S................................................................ ......................................................................... ==================== 12130 passed, 12 skipped in 368.339 seconds =====================

Dependencies

| **Package** | **Minimum supported version** |
| --- | --- |
| [setuptools](https://setuptools.readthedocs.io/en/latest/) | 24.2.0 |
| [NumPy](https://www.numpy.org/) | 1.15.4 |
| [python-dateutil](https://dateutil.readthedocs.io/en/stable/) | 2.7.3 |
| [pytz](https://pypi.org/project/pytz/) | 2017.2 |

Recommended dependencies

* [numexpr](https://github.com/pydata/numexpr): for accelerating certain numerical operations. numexpr uses multiple cores as well as smart chunking and caching to achieve large speedups. If installed, must be Version 2.6.2 or higher.
* [bottleneck](https://github.com/pydata/bottleneck): for accelerating certain types of nan evaluations. bottleneck uses specialized cython routines to achieve large speedups. If installed, must be Version 1.2.1 or higher.

**Note**

You are highly encouraged to install these libraries, as they provide speed improvements, especially when working with large data sets.

Optional dependencies

Pandas has many optional dependencies that are only used for specific methods. For example, [**pandas.read\_hdf()**](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_hdf.html#pandas.read_hdf) requires the pytables package, while [**DataFrame.to\_markdown()**](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.to_markdown.html#pandas.DataFrame.to_markdown) requires the tabulate package. If the optional dependency is not installed, pandas will raise an ImportError when the method requiring that dependency is called.

| **Dependency** | **Minimum Version** | **Notes** |
| --- | --- | --- |
| BeautifulSoup4 | 4.6.0 | HTML parser for read\_html (see [note](https://pandas.pydata.org/pandas-docs/stable/getting_started/install.html#optional-html)) |
| Jinja2 |  | Conditional formatting with DataFrame.style |
| PyQt4 |  | Clipboard I/O |
| PyQt5 |  | Clipboard I/O |
| PyTables | 3.4.3 | HDF5-based reading / writing |
| SQLAlchemy | 1.1.4 | SQL support for databases other than sqlite |
| SciPy | 0.19.0 | Miscellaneous statistical functions |
| XLsxWriter | 0.9.8 | Excel writing |
| blosc |  | Compression for HDF5 |
| fsspec | 0.7.4 | Handling files aside from local and HTTP |
| fastparquet | 0.3.2 | Parquet reading / writing |
| gcsfs | 0.6.0 | Google Cloud Storage access |
| html5lib |  | HTML parser for read\_html (see [note](https://pandas.pydata.org/pandas-docs/stable/getting_started/install.html#optional-html)) |
| lxml | 3.8.0 | HTML parser for read\_html (see [note](https://pandas.pydata.org/pandas-docs/stable/getting_started/install.html#optional-html)) |
| matplotlib | 2.2.2 | Visualization |
| numba | 0.46.0 | Alternative execution engine for rolling operations |
| openpyxl | 2.5.7 | Reading / writing for xlsx files |
| pandas-gbq | 0.12.0 | Google Big Query access |
| psycopg2 |  | PostgreSQL engine for sqlalchemy |
| pyarrow | 0.12.0 | Parquet, ORC (requires 0.13.0), and feather reading / writing |
| pymysql | 0.7.11 | MySQL engine for sqlalchemy |
| pyreadstat |  | SPSS files (.sav) reading |
| pytables | 3.4.3 | HDF5 reading / writing |
| pyxlsb | 1.0.6 | Reading for xlsb files |
| qtpy |  | Clipboard I/O |
| s3fs | 0.4.0 | Amazon S3 access |
| tabulate | 0.8.3 | Printing in Markdown-friendly format (see [tabulate](https://github.com/astanin/python-tabulate)) |
| xarray | 0.8.2 | pandas-like API for N-dimensional data |
| xclip |  | Clipboard I/O on linux |
| xlrd | 1.1.0 | Excel reading |
| xlwt | 1.2.0 | Excel writing |
| xsel |  | Clipboard I/O on linux |
| zlib |  | Compression for HDF5 |

Optional dependencies for parsing HTML

One of the following combinations of libraries is needed to use the top-level [**read\_html()**](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_html.html#pandas.read_html) function:

*Changed in version 0.23.0.*

* [BeautifulSoup4](https://www.crummy.com/software/BeautifulSoup) and [html5lib](https://github.com/html5lib/html5lib-python)
* [BeautifulSoup4](https://www.crummy.com/software/BeautifulSoup) and [lxml](https://lxml.de/)
* [BeautifulSoup4](https://www.crummy.com/software/BeautifulSoup) and [html5lib](https://github.com/html5lib/html5lib-python) and [lxml](https://lxml.de/)
* Only [lxml](https://lxml.de/), although see [HTML Table Parsing](https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html#io-html-gotchas) for reasons as to why you should probably **not** take this approach.

**Warning**

* if you install [BeautifulSoup4](https://www.crummy.com/software/BeautifulSoup) you must install either [lxml](https://lxml.de/) or [html5lib](https://github.com/html5lib/html5lib-python) or both. [**read\_html()**](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_html.html#pandas.read_html) will **not** work with *only* [BeautifulSoup4](https://www.crummy.com/software/BeautifulSoup) installed.
* You are highly encouraged to read [HTML Table Parsing gotchas](https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html#io-html-gotchas). It explains issues surrounding the installation and usage of the above three libraries.