



Python Introduction

MODERN DATA ANALYTICS
[G0Z39A]

PROF. DR. IR. JAN DE SPIEGELEER

Contents

- Python
- Writing Code
- Jupyter vs. Google Colab
- Github
- Kaggle
- Environments

Python



















Python vs. R ?

1. Which language do your colleagues use?
2. How steep & long is the learning curve?
3. What problems do you want to solve and what tasks do you need to accomplish?
4. What are the commonly used tool(s) in your field?

Python vs. R

Python is generally used when the data analysis tasks need to be integrated with web apps or if statistics code needs to be incorporated into a production database. Since it's a full-fledged programming language, Python is a good tool to implement algorithms for use in production.

R is mainly used when the data analysis tasks require standalone computing or analysis on individual servers. For exploratory work, R is easier for beginners. Statistical models can be written with a few lines of code.

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6

Source: <http://bit.ly/3cT6Ep4>



Bryan C. · a year ago



I'm am so disappointed with the IEEE and this list. I mean, atleast you posted your methodology which confirms the dubious value this list provides. What is more disheartening is the number of ill-informed readers who will report this in their news feeds as some kind of factual data based on the author. Shame on you guys, frequency of "X Programming", and not including atleast "X Scripting". This makes the data woefully incomplete.

6 ^ | v 2 · Reply · Share ›



Ronald Mutegeki ➔ Bryan C. · a year ago

However, I'm disappointed that JavaScript isn't being considered in the Mobile app development category

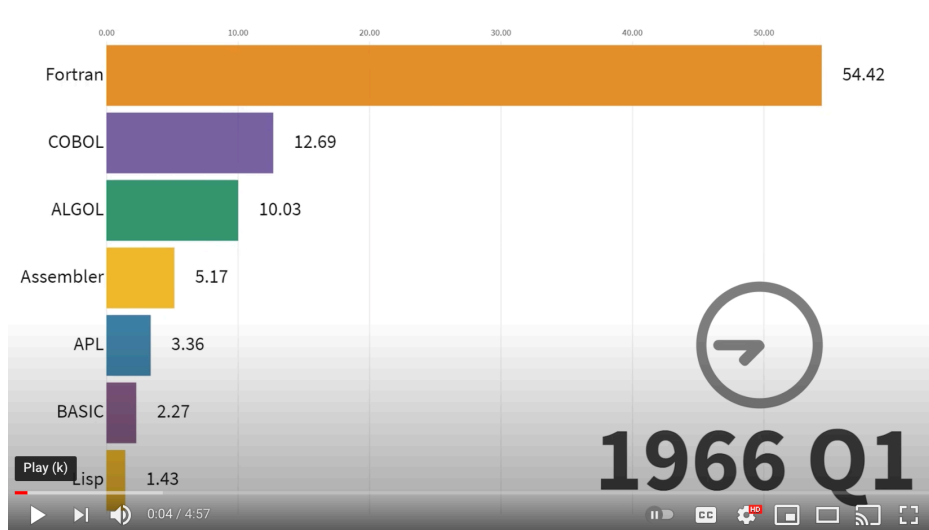
1 ^ | v · Reply · Share ›



Ronald Mutegeki ➔ Bryan C. · a year ago

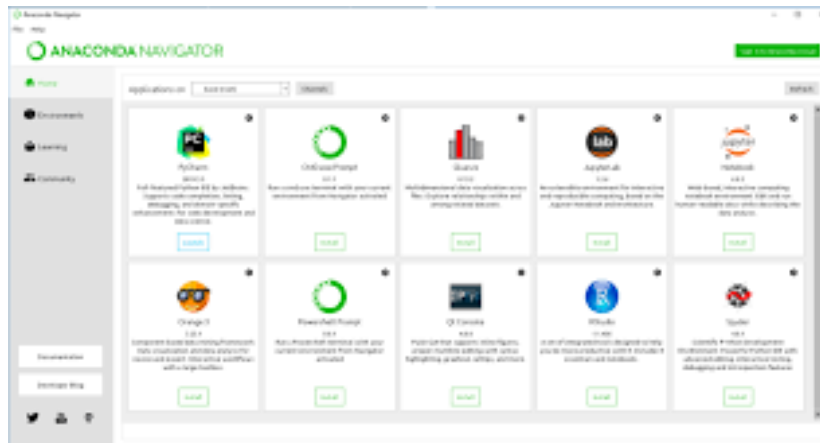
What would you say about Cuda? Would you consider it just another platform or... I think they also considered things what people (engineers for

History of Python



Source: <https://www.youtube.com/watch?v=Og847HVwRSI>

Download & Start



The quickest way to get started is to use Anaconda which can be found at **www.continuum.io**.

Anaconda is a package manager. In stead of going to the Python.org website to download the required version of Python and then installing the different packages needed, Anaconda is a quicker introduction.

Anaconda offers the most useful packages for mathematics, science and engineering

Anaconda installer(s) : <https://www.anaconda.com/products/individual>

Download & Start

But of course Anaconda does not deliver every package out there. In this case you will have to run the pip install command in the command window in order to fetch a particular package and install it on your computer. If one wants to download the `airflow` package for example, the following pip-command will do the job

```
pip install airflow
```

```
>>> import airflow

Traceback (most recent call last):

  File "<stdin>", line 1, in <module>

ModuleNotFoundError: No module named
airflow'
```

Dependencies : `$pipdeptree`

```
jandespiegeleer — -bash — 92x32
- parso [required: >=0.7.0,<0.8.0, installed: 0.7.0]
- pluggy [required: Any, installed: 0.13.1]
- python-jsonrpc-server [required: >=0.4.0, installed: 0.4.0]
  - ujson [required: >=3.0.0, installed: 4.0.1]
- ujson [required: >=3.0.0, installed: 4.0.1]
- pyzmq [required: >=17, installed: 19.0.2]
- qdarkstyle [required: >=2.8, installed: 2.8.1]
- helpdev [required: >=0.6.10, installed: ?]
- qtpy [required: >=1.9, installed: 1.9.0]
- qtawesome [required: >=0.5.7, installed: 1.0.1]
- qtpy [required: Any, installed: 1.9.0]
- qtconsole [required: >=4.6.0, installed: 4.7.7]
- ipykernel [required: >=4.1, installed: 5.3.4]
  - appnope [required: Any, installed: 0.1.0]
  - ipython [required: >=5.0.0, installed: 7.19.0]
    - appnope [required: Any, installed: 0.1.0]
    - backcall [required: Any, installed: 0.2.0]
    - decorator [required: Any, installed: 4.4.2]
    - jedi [required: >=0.10, installed: 0.17.1]
      - parso [required: >=0.7.0,<0.8.0, installed: 0.7.0]
  - pexpect [required: >4.3, installed: 4.8.0]
  - ptyprocess [required: >=0.5, installed: 0.6.0]
  - pickleshare [required: Any, installed: 0.7.5]
  - prompt-toolkit [required: >=2.0.0,<3.1.0,!3.0.1,!3.0.0, installed: 3.0.8]
    - wcwidth [required: Any, installed: 0.2.5]
  - pygments [required: Any, installed: 2.7.2]
  - setuptools [required: >=18.5, installed: 50.3.1.post20201107]
  - traitlets [required: >=4.2, installed: 5.0.5]
    - ipython-genutils [required: Any, installed: 0.2.0]
- jupyter-client [required: Any, installed: 6.1.7]
- jupyter-core [required: >=4.6.0, installed: 4.6.3]
  - traitlets [required: Any, installed: 5.0.5]
```

Download & Start

Over 250 packages are automatically installed with Anaconda.

Important Packages

- **SciPy and Numpy** for numerical computing. NumPy provides tools to help build multi-dimensional arrays and perform calculations on the data stored in them.
- **Pandas** for easy data processing, cleaning, ...
- **Matplotlib** and **Seaborn** to produce graphs
- **Scikit-Learn** for Machine Learning
- **Statsmodels** for statistical models and unit tests.
- **PIL**: For basic image importing, manipulation, and exporting.
- **MoviePy** is to videos what Pillow is to images. It provides a range of functionality for common tasks associated with importing, modifying, and exporting video files.
- **Requests** if your application sends any data over HTTP
- **Plotly** : The plotly is an interactive, open-source plotting library that supports over 40 unique chart types
- **openpyxl** is a Python library to read/write Excel 2010 xlsx/xlsm/xltx/xltm files.

Writing Code

IDE VS. NOTEBOOKS

IDE

An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:

- An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
- Build, execution, and debugging tools
- Source control

IDEs for Python

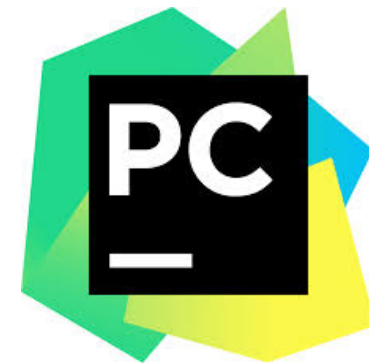
- Atom
- Spyder (shipped with Anaconda)
- **Visual Studio**
- **Pycharm**

IDE : Pycharm

PyCharm: <https://www.jetbrains.com/pycharm/>

One of the best (and only) full-featured, dedicated IDEs for Python is PyCharm. Available in both paid (Professional) and free open-source (Community) editions, PyCharm installs quickly and easily on Windows, Mac OS X, and Linux platforms.

Out of the box, PyCharm supports Python development directly. You can just open a new file and start writing code. You can run and debug Python directly inside PyCharm, and it has support for source control and projects.



References

IDEs :<https://realpython.com/python-ides-code-editors-guide/>

Jupyter Notebooks

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code (in Python and other packages) with supporting equations, visualizations and narrative text.

Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.

Notebooks can be shared with others using email, Dropbox, GitHub

With nbviewer Jupyter notebooks can be turned into blogs, or even books
(<http://bit.ly/3tC4Pm1>)

Our choice for this course: Notebooks

BOTH JUPYTER AND GOOGLE COLLABORATE

Components of the notebook

- **Menu bar**
The menu bar presents different options that may be used to manipulate the way the notebook functions.
- **Toolbar**
The tool bar gives a quick way of performing the most-used operations within the notebook, by clicking on an icon.
- **Cell**
A cell can be either “markup” or “code”.
 - Code : here you write your Python scrips
 - Markup: here goes the markup to turn the notebook into a real research document.

Keyboard Shortcuts

All actions in the notebook can be performed with the mouse, but keyboard shortcuts are also available for the most common ones. The essential shortcuts to remember are the following:

Shift-Enter

Execute the current cell, show any output, and jump to the next cell below. If Shift-Enter is invoked on the last cell, it makes a new cell below. This is equivalent to clicking the Cell, Run menu item, or the Play button in the toolbar.

Ctrl-Enter

Execute the cell, show the output and stay in the cell

Notebook's Kernel

The image shows a Jupyter Notebook terminal window and the JupyterLab interface. The terminal window, titled "jandespiegeleer — jupyter-notebook — 92x32", displays the following output:

```
File "/Users/jandespiegeleer/opt/anaconda3/lib/python3.8/site-packages/notebook/notebookapp.py", line 1942, in init_server_extensions
    mod = importlib.import_module(modulename)
File "/Users/jandespiegeleer/opt/anaconda3/lib/python3.8/importlib/__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
File "<frozen importlib._bootstrap>", line 1014, in _gcd_import
File "<frozen importlib._bootstrap>", line 991, in _find_and_load
File "<frozen importlib._bootstrap>", line 973, in _find_and_load
ModuleNotFoundError: No module named 'jupyter_nbextensions_configurator'
[I 13:12:03.993 NotebookApp] JupyterLab extension loaded from /Users/jandespiegeleer/opt/anaconda3/lib/python3.8/site-packages/jupyterlab
[I 13:12:03.993 NotebookApp] JupyterLab application directory is /Users/jandespiegeleer/opt/anaconda3/share/jupyter/lab
[I 13:12:03.996 NotebookApp] Serving notebooks from local directory: /Users/jandespiegeleer/opt/anaconda3/share/jupyter
[I 13:12:03.996 NotebookApp] Jupyter Notebook 6.1.4 is running at:
```

The JupyterLab interface shows the URL "localhost:8888/tree/FilesJDS/Ris..." in the address bar. A blue arrow points from the terminal output to the address bar. The Jupyter logo is visible, and the "Running" tab is selected. Below the tabs, it says "Select items to perform actions on them."

Jupyter Notebook : Tutorials

- <https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>
- <https://realpython.com/jupyter-notebook-introduction/>
- <https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook>
- <https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/>
- <https://www.tutorialspoint.com/jupyter/index.htm>
- ...<many more>

Extending Your Notebook

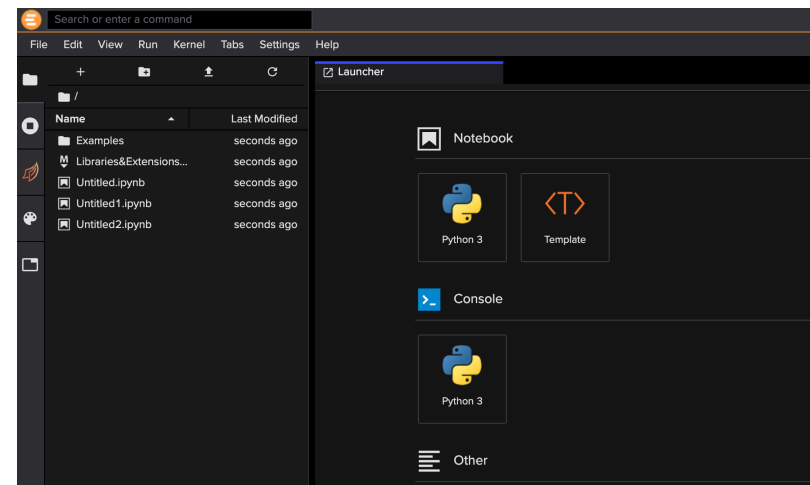
Info : <https://towardsdatascience.com/jupyter-notebook-extensions-517fa69d2231>

- Run the following lines in your command line on your PC/Mac
 - `pip install jupyter_contrib_nbextensions && jupyter contrib nbextension install`
- Extensions that can be added
 - Hide input cells
 - Outline
 - Spelling checker
 -

Jupyter vs. Google Colab

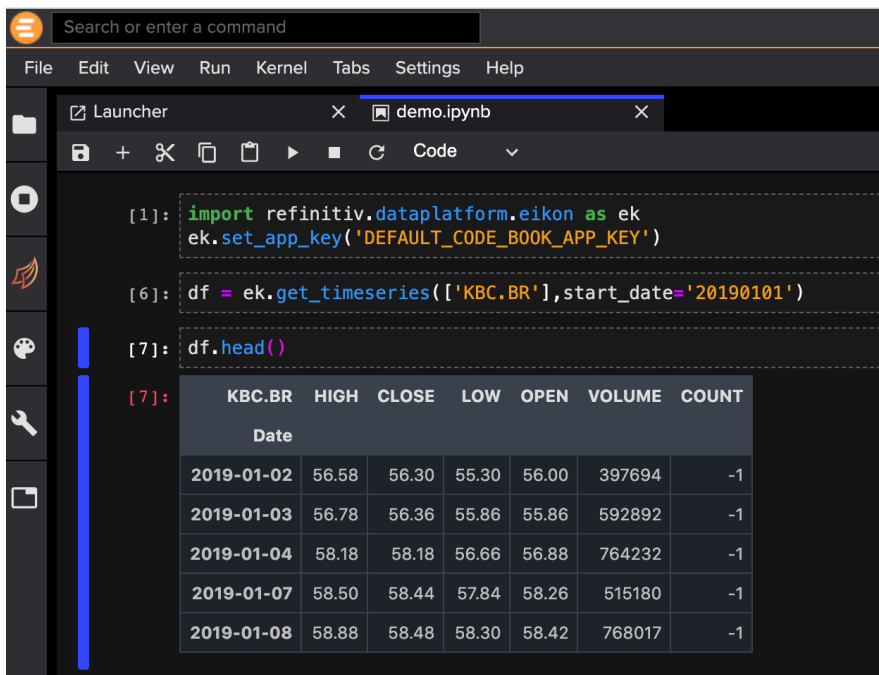
Other Notebook-like environments

- **Bloomberg B-Quant (BQNT)**
A JupyterLab inside Bloomberg for quants who need to do a lot of testings on trading ideas, filtering of securities, etc., this integrated environment is absolutely a good place (but expensive way) to sort everything out.
- **Reuters Codebook**
Available in Refinitiv Workspace and Eikon, CodeBook provides access to Reuters's data.
- **Google Collaborate** ("Colab")
Jupyter notebook on Google Cloud



Reuters Codebook

Other Notebook-like environments



The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar includes a search bar and a menu with File, Edit, View, Run, Kernel, Tabs, Settings, and Help. Below the menu, there are tabs for 'Launcher' and 'demo.ipynb'. The code area shows three cells:

```
[1]: import reinitiv.dataplatform.eikon as ek
     ek.set_app_key('DEFAULT_CODE_BOOK_APP_KEY')

[6]: df = ek.get_timeseries(['KBC.BR'], start_date='20190101')

[7]: df.head()
```

The output of the third cell is a table with 7 columns: KBC.BR, HIGH, CLOSE, LOW, OPEN, VOLUME, and COUNT. The first column is labeled 'Date'.

Date	KBC.BR	HIGH	CLOSE	LOW	OPEN	VOLUME	COUNT
2019-01-02		56.58	56.30	55.30	56.00	397694	-1
2019-01-03		56.78	56.36	55.86	55.86	592892	-1
2019-01-04		58.18	58.18	56.66	56.88	764232	-1
2019-01-07		58.50	58.44	57.84	58.26	515180	-1
2019-01-08		58.88	58.48	58.30	58.42	768017	-1

Reuters Codebook :
Retrieving TimeSeries

Google Collaborate

Colab is a service that provides GPU-powered Notebooks for free. It's based on, but slightly different to, regular Jupyter Notebooks, so be sure to read the Colab docs to learn how it works.















Documentation: <https://colab.research.google.com/notebooks/welcome.ipynb>

Why/When do you need this ?

- If you want to create a machine learning model but you don't have a computer that can take the workload
- Even if you have a GPU or a good computer creating a local environment with anaconda and installing packages and resolving installation issues can be difficult.
- Colaboratory is a free Jupyter notebook environment provided by Google where you can use free GPUs and TPUs which can solve all these issues.

ExamplesRecentGoogle DriveGitHubUpload

Filter notebooks

Title	First opened	Last opened	
 Welcome To Colaboratory	Feb 3, 2020	0 minutes ago	
 Bond Pricer.ipynb	Jan 20, 2021	Jan 20, 2021	 
 widgets list.ipynb	Jan 20, 2021	Jan 20, 2021	 
 <u>GE Ford Example.ipynb</u>	Jan 19, 2021	Jan 19, 2021	 
 Untitled5.ipynb	Jan 19, 2021	Jan 19, 2021	 

[NEW NOTEBOOK](#) CANCEL

Google Collaborate

Google Collaborate

- To get started
 - make sure you have a google account (gmail)
 - then go to this link <https://colab.research.google.com>.
- When a new notebook is created, collab will create **Untitled0.ipynb** saves it to you Google Drive in a folder named Colab Notebooks.
- It is actually a Jupyter notebook, hence all commands of the Jupyter notebook will work here.
- Choose your runtime type (CPU, GPU, TPU)

Google Collaborate

- **Installing Python packages**

Use can use pip to install any package. For example : `!pip install papermill`

Google Collaborate

If you plan to store our project related files onto your Google, you have to “mount” your google drive in to the runtime of your notebook.

In order to do that, There is a **3-step** procedure to follow:

```
from google.colab import drive  
drive.mount('/content/mydrive')
```

The lines above will prompt a URL with an authentication code. After you insert that authentication code in the provided space, your google drive will be mounted. You can check the contents of the current folder in the runtime by typing the following and running the cell.

Google Collaborate

- **Mounting Drives (3-step procedure)**

Users must grant access to the runtime of the notebook to read / write to the G-Drive

Git and Github

STORING AND COLLABORATING VIA ON-LINE REPOSITORIES

Using Git : distributed development

Imagine **you** are working with a **colleague** on a **project**. You have decided to split the task. Each one of you is working on a separate item: visualisation, prediction, storage,

You want to avoid sending files (jupyter notebooks) by email, ftp, etc... You have by no means control over what version of a file you are working on. Does it contain the changes of my colleague or not ?

This is where a version control system becomes mandatory.

In many circles, Git has come to be the expected version control system for new projects. Companies expect their new hires on to be familiar with this workflow.

Using Git : distributed development

Compared to other version control systems, Git is

- Responsive
- easy to use
- Inexpensive

One thing that really sets Git apart from other version control systems, is its **branching model**. Branching allows you to create independent local branches in your code. This means you can try out new ideas, set aside branches for production work, jump back to earlier branches, and easily delete, merge, and recall branches at the click of a button.

Using GitHub



python package for weather forecasting



github.com › zspatter › weather-forecast ▾

zspatter/weather-forecast: A Python script that ... - GitHub

A **Python** script that displays the 5-day **weather forecast** for a given location at 3 hour ...
Furthermore, you need the PyOWM wrapper **library** for OpenWeatherMap ...

People also ask

- How do you forecast weather in Python? ▾
- Which algorithm is best for weather prediction? ▾
- What is a common scientific tool for forecasting weather? ▾
- What tools are used for weather forecasting? ▾

Feedback

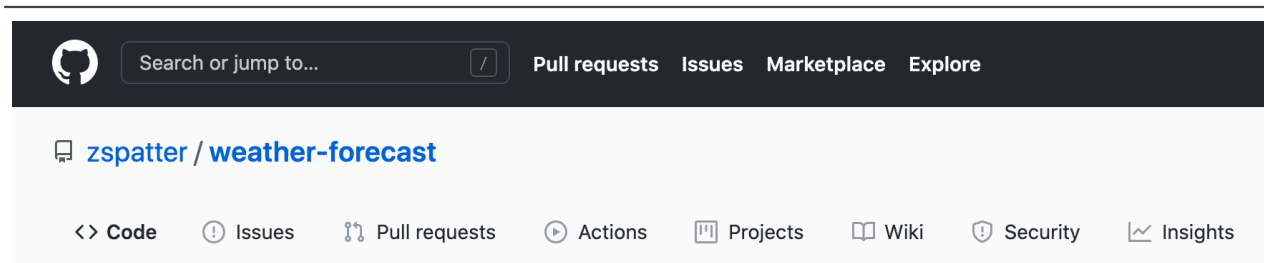
Using GitHub

The screenshot shows the GitHub interface for the repository `zspatter / weather-forecast`. The top navigation bar includes the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. The 'Code' tab is selected, showing the master branch with 1 branch and 0 tags. A 'Go to file' button, an 'Add file' button, and a 'Code' button are visible. The commit history table shows the following entries:

Commit Hash	Author	Message	Date	Commits
4daac15	zspatter	simplifies condition (replaces `is not None`)	on 24 Jun 2019	39
		.gitignore		
		Adds .csv to .gitignore		
		2 years ago		
		LICENSE		
		Adds LICENSE		
		2 years ago		
		README.md		
		Converts sample output reference to relative link		
		2 years ago		
		sample_output.png		
		Crops the console output		
		2 years ago		
		weather_forecasts.py		
		simplifies condition (replaces `is not None`)		
		2 years ago		


<http://bit.ly/3aFsae2>


Using GitHub




Don't just copy and run notebooks !

- who wrote the notebook
- How long ago ?
- Does it specify what modules are required ? (eg requirements.txt file)
- readme.md – file ?
- does it provide data (xlsx, csv, ...)


 master ▾


 1 branch

 0 tags


Go to file


Add file ▾

 Code ▾

 **zspatter** simplifies condition (replaces `is not None`)

4daac15 on 24 Jun 2019

 39 commits

 .gitignore


Adds .csv to .gitignore

2 years ago

 LICENSE


Adds LICENSE

2 years ago

 README.md


Converts sample output reference to relative link

2 years ago

 sample_output.png

Crops the console output

2 years ago

 weather_forecasts.py

simplifies condition (replaces `is not None`)

2 years ago

Using GitHub

```
(base) $ git clone https://github.com/zspatter/weather-forecast.git
Cloning into 'weather-forecast'...
remote: Enumerating objects: 113, done.
remote: Total 113 (delta 0), reused 0 (delta 0), pack-reused 113
Receiving objects: 100% (113/113), 1.08 MiB | 699.00 KiB/s, done.
Resolving deltas: 100% (56/56), done.
Checking connectivity... done.
```

Using GitHub

GitHub is a cloud-based database that allows you to keep track of and share your Git version control projects outside of your local computer/server.

Has an intuitive, graphically represented user interface, and provides programmers with built-in control and task-management tools.

Because GitHub is cloud-based, an individual's Git repositories can be remotely accessed by any authorized person, from any computer, anywhere in the world.

Through GitHub, you can share your code with others, giving them the power to make revisions or edits on your various Git branches. This makes it possible for entire teams to coordinate together on single projects in real-time.

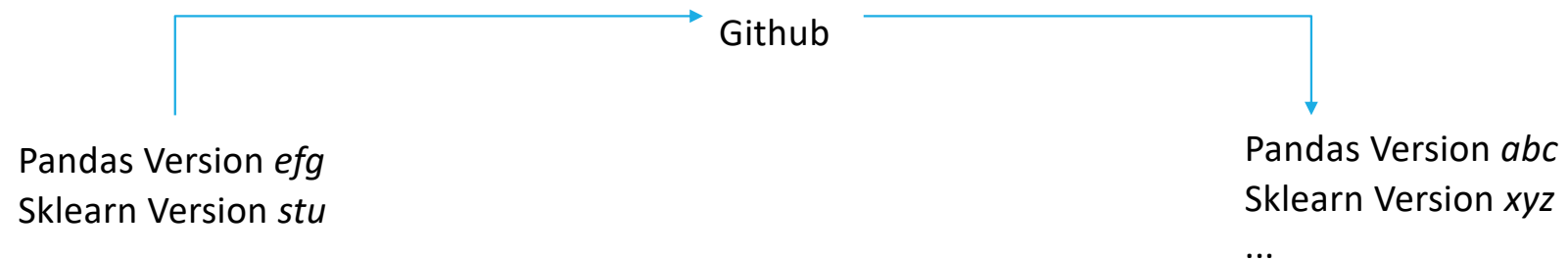
As changes are introduced, new branches are created, allowing the team to continue to revise the code without overwriting each other's work. These branches are like copies, and changes made on them do not reflect in the main directories on other users' machines unless users choose to push/pull the changes to incorporate them.

Virtual Environments

Self Study: <http://bit.ly/3h4f8ty> or <http://bit.ly/3r5KUL4>

Why using virtual environments ?

Main purpose of Python virtual environments is to create an isolated environment for Python projects. This means that each project can have its own dependencies, regardless of what dependencies every other project has.



Why using virtual environments ?

It is good practice to add a text file to your project (eg requirements.txt) that contains all the packages used in your project, and the python version used.



Create a file with a list of all the packages in **YOUR** environment

```
pip freeze > requirements.txt
```

This file has to be part of your projects (and corresponding git-repository)

1. Create a new environment “course_kul”

```
$ pyenv virtualenv 3.8.0 course_kul
```

2. Activate the environment

```
$ pyenv activate course_kul
```

3. Install all the modules in the file `requirement.txt`

Kaggle

History of Kaggle

Kaggle is the most well known **competition** platform for predictive modeling and analytics. The company was founded in 2010 in Melbourne, Australia, and a year later, it moved to San Francisco after receiving funding from Silicon Valley. In 2017, it was acquired by Google.

It still is a competition platform but is offering different products.

1. **Kaggle Kernels:** a cloud-based workbench for data science and machine learning. Allows data scientists to share code and analysis in Python and R. Over 150K "kernels" (code snippets) have been shared on Kaggle covering everything from sentiment analysis to object detection.
2. **Public datasets platform:** community members share datasets with each other.

Kaggle Rankings


Competitions


Datasets


Notebooks


Discussion


Learn more about rankings >


























 203
Grandmasters

 1,525
Masters

 6,255
Experts

 57,819
Contributors

 86,338
Novices

Rank	Tier	User		Medals	Points
1		 Guanshuo Xu	joined 5 years ago	 17  15  2	234,478
2		 bestfitting	joined 4 years ago	 29  9  1	209,773
3		 Psi	joined 9 years ago	 15  6  0	203,387
4		 Dieter	joined 3 years ago	 15  8  3	177,175
5		 Μαριος Μιχαηλιδης KazAnova	joined 8 years ago	 39  54  38	155,613

Kaggle Datasets

Create Public Datasets

Open a dialogue, accept contributions, and get insights: improve your dataset by publishing it on Kaggle.

Create Public Dataset



Search 71,213 datasets

Filters

Public Your Datasets Favorites

Hottest



Reddit WallStreetBets Posts

Gabriel Preda · Updated 7 hours ago
Usability 10.0 · 1 File (CSV) · 6 MB · 2 Tasks

83

Silver



Restaurant Business Rankings 2020

Michal Bogacz · Updated 9 days ago

13

Object Oriented Programming

List of tutorials

- <https://bit.ly/38AE7Rj>
- <https://bit.ly/3ryOqOa>
- <https://bit.ly/3poTrHg>
- ...