

Python and R Basic

Econ 570 Big Data Econometrics

Youngmin Ju

What is Python? - A Basic Definition

- **Python** is based on C, it is a **software development** language which is deep and huge and intuitive.
- **Python** emphasizes **productivity and code readability**.
- **Python** is used by **programmers** that want to delve into data analysis or apply statistical techniques, and by **developers** that turn to data science.
- “The closer you are to working in an **engineering** environment (**machine learning**), the more you might prefer **Python**.”

What is R? - A Basic Definition

- **R** is a **statistical and visualization** language which is deep and huge and mathematical.
- **R** focuses on better, user friendly **data analysis, statistics and graphical models**.
- **R** has been used primarily in **academics and research**. However, **R** is rapidly expanding into the enterprise market.
- “The closer you are to **statistics and research**, the more you might prefer **R**.”

Python vs R

R Co-occurring Terms



Python Co-occurring Terms



Key Differences

- **R** is mainly used for statistical analysis while **Python** provides a more general approach to data science
- The primary objective of **R** is Data analysis and Statistics whereas the primary objective of **Python** is Deployment and Production
- **R** users mainly consists of Scholars and R&D professionals while **Python** users are mostly Programmers and Developers
- **R** consists various packages and libraries like tidyverse, ggplot2, caret, zoo whereas **Python** consists packages and libraries like numpy, pandas, scipy, scikit-learn, TensorFlow, caret

Pros and Cons

	R	Python
Disadvantages	Slow High Learning curve Dependencies between library	Not as many libraries as R
Advantages	<ul style="list-style-type: none">• Graphs are made to talk. R makes it beautiful• Large catalog for data analysis• GitHub interface• RMarkdown• Shiny	<ul style="list-style-type: none">• Jupyter notebook: Notebooks help to share data with colleagues• Mathematical computation• Deployment• Code Readability• Speed• Function in Python

Why do we use Python and R?

- 1. **FREE**
- 2. Easy – beginner-friendly language

Print the text “Hello World!”

- JAVA

```
class helloworld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

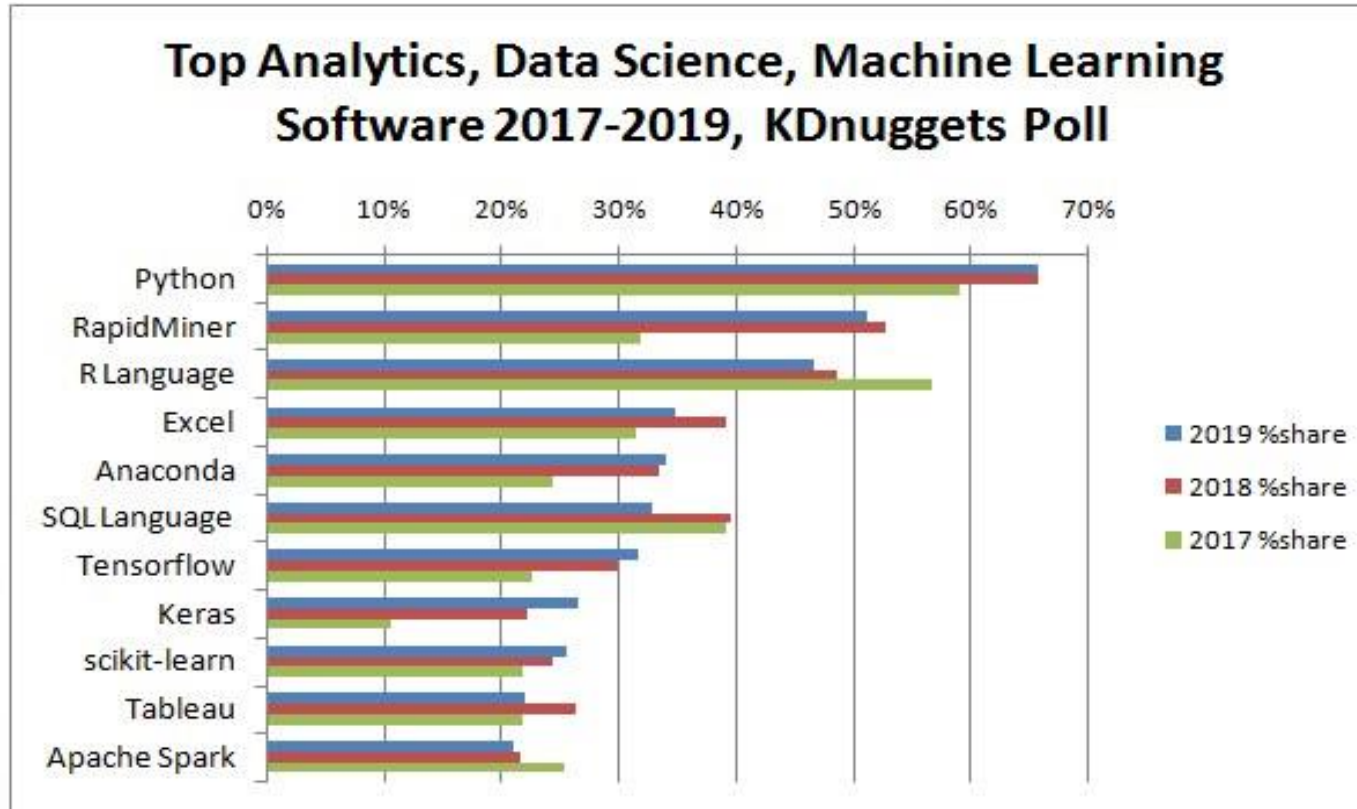
- Python and R
(easy!)

```
print("Hello World!")
```


Why do we use Python and R?

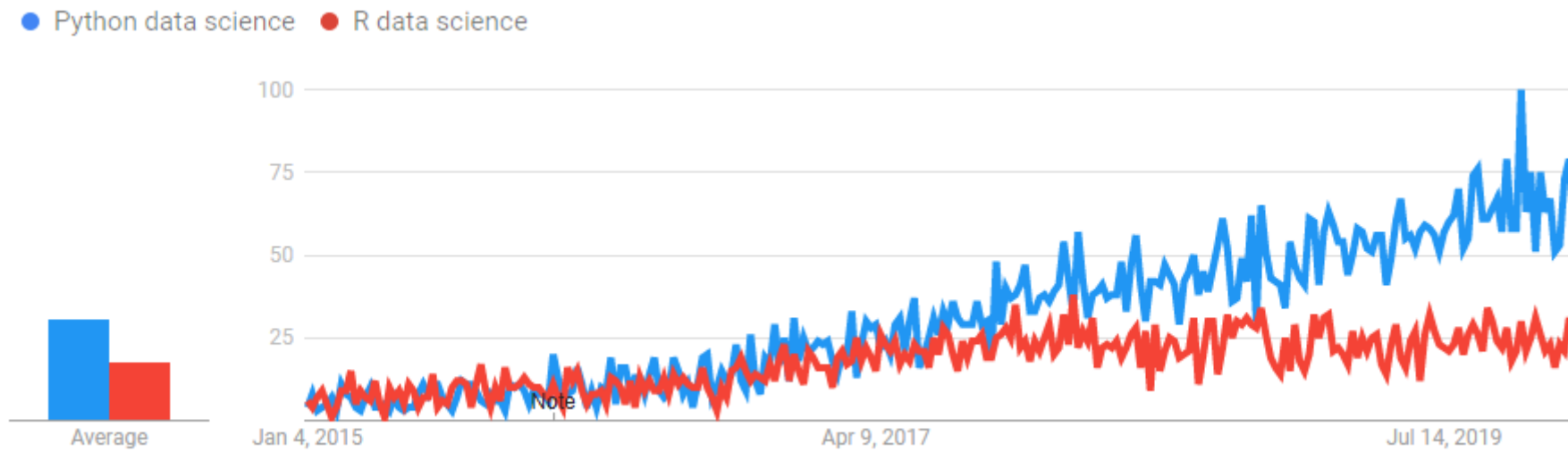
- 1. **FREE**
- 2. **Easy** – beginner-friendly language
- 3. Doesn't take long to learn
- 4. Popular

Python leads the 11 top Data Science, Machine Learning platforms: Trends and Analysis



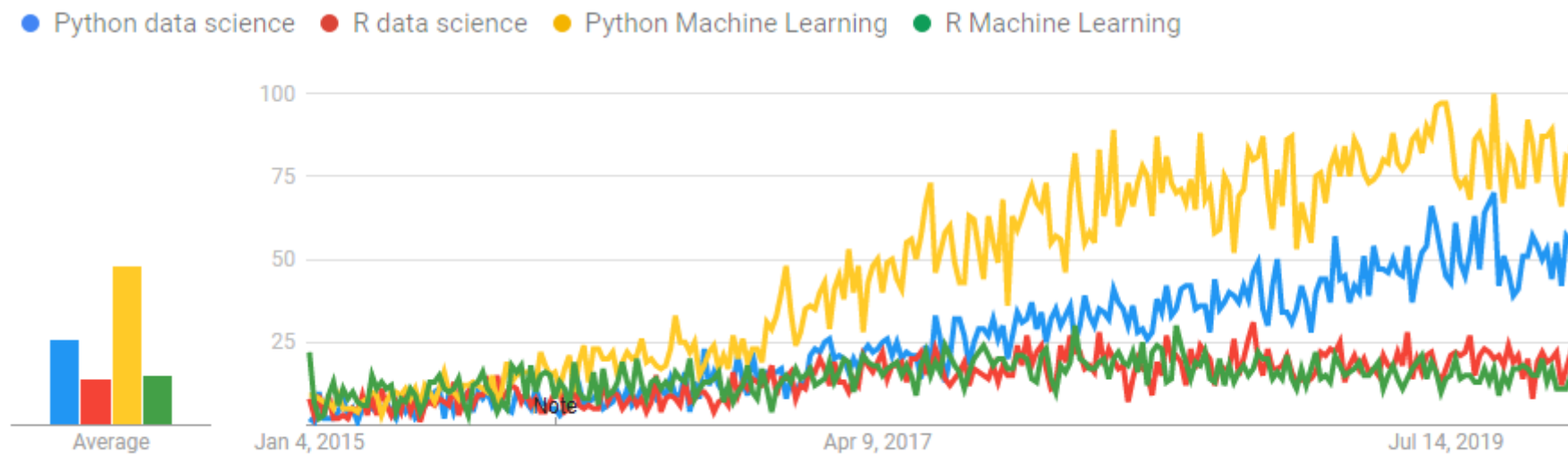
<https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html>

Google Trends (Python vs R)



<https://trends.google.com/trends/explore?date=2015-01-01%202020-01-31&q=Python%20data%20science,R%20data%20science>

Google Trends (Python vs R)



<https://trends.google.com/trends/explore?date=2015-01-01%202020-01-31&q=Python%20data%20science,R%20data%20science,Python%20Machine%20Learning,R%20Machine%20Learning>

Free Sources for learning Python and R

- Free sources for learning Python
 - <https://www.learnpython.org/>
 - <https://www.py4e.com/lessons>
 - <https://developers.google.com/edu/python> - Google
 - <https://www.kaggle.com/learn/overview> - Kaggle
 - <https://www.inferentialthinking.com/chapters/intro.html>
 - <https://pyecon.org/> - Econometrics
 - <https://www.kevinsheppard.com/teaching/python/notes/> - Econometrics
 - <https://www.linkedin.com/learning/search?keywords=python&u=76870426> – LinkedIn Learning
- Free sources for learning R
 - <https://r4ds.had.co.nz/index.html>
 - <https://www.econometrics-with-r.org/index.html> - Econometrics
 - <https://www.linkedin.com/learning/search?keywords=r&u=76870426> – LinkedIn Learning

* Guideline for Data Science

- **If I had to start learning Data Science again, how would I do it?**

<https://www.kdnuggets.com/2020/08/start-learning-data-science-again.html>

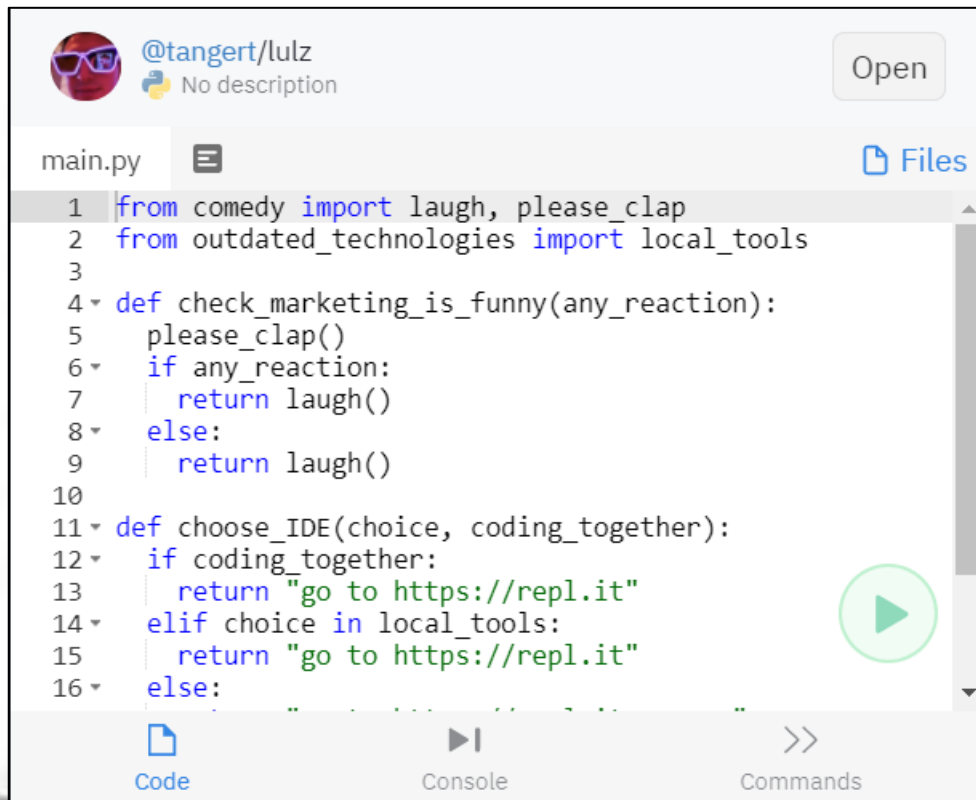
- **Machine Learning for Economists: An Introduction**

<https://antontarasenko.com/2015/12/28/machine-learning-for-economists-an-introduction/>

Without installing Python and R

- Repl.it

<https://repl.it/>



```
1 from comedy import laugh, please_clap
2 from outdated_technologies import local_tools
3
4 def check_marketing_is_funny(any_reaction):
5     please_clap()
6     if any_reaction:
7         return laugh()
8     else:
9         return laugh()
10
11 def choose_IDE(choice, coding_together):
12     if coding_together:
13         return "go to https://repl.it"
14     elif choice in local_tools:
15         return "go to https://repl.it"
16     else:
```

Instant IDE

Code right in your browser.

Embeddable, 0 setup, and collaborative. Repl.it is the best tool for quickly starting, sharing, and developing projects in any programming language, right from your browser.

[Learn more about the IDE >](#)

Without installing Python and R

- **Repl.it**

Pros

- No need to install Python and R

Cons

- Need internet connection

How to install Python

1. Install **Python** (Latest version is 3.9.1)

www.python.org


- choose Python 3


2. Install **Anaconda** (Python distribution)


<https://www.anaconda.com/>

- a free and open-source distribution of the Python
- aims to simplify **package management** and deployment
- Install guide for Windows (<https://docs.anaconda.com/anaconda/install/windows/>)
- Install guide for macOS (<https://docs.anaconda.com/anaconda/install/mac-os/>)

 Home

 Environments

 Learning

 Community

Documentation

Developer Blog



Applications on

base (root)

Channels

Refresh



CMD.exe Prompt

0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

Launch



JupyterLab

2.1.5

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch



Notebook

6.1.1

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch



Powershell Prompt

0.0.1

Run a Powershell terminal with your current environment from Navigator activated

Launch



PyCharm

2020.2.1

Full-featured Python IDE by JetBrains. Supports code completion, linting, debugging, and domain-specific enhancements for web development and data science.

Launch



Qt Console

4.7.5

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch



Spyder

4.1.4

Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Launch



VS Code

1.48.2

Streamlined code editor with support for development operations like debugging, task running and version control.

Launch



Glueviz

0.15.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install



Orange 3

3.26.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install



RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Install

How to install R

1. Install **R** (Latest version is 4.0.3)

<https://cran.rstudio.com/>

2. Install **R studio** in Anaconda (or install R studio separately)

<https://www.anaconda.com/>

<https://rstudio.com/products/rstudio/>

Text/Code Editor vs IDE

- **Text/Code Editor:**

Code editors are the lightweight tool that allows you to write and edit the code with some features such as syntax highlighting and code formatting. It provided fewer features than IDE.

- **Integrated Development Environment (IDE):**

IDEs are full-fledged environment which provide all the essential tools needed for software development. It just doesn't handle the code (for example, write, edit, syntax highlighting and auto-completion) but also provides other features such as debugging, execution, testing, and code formatting that helps programmers.

Visual Studio Code

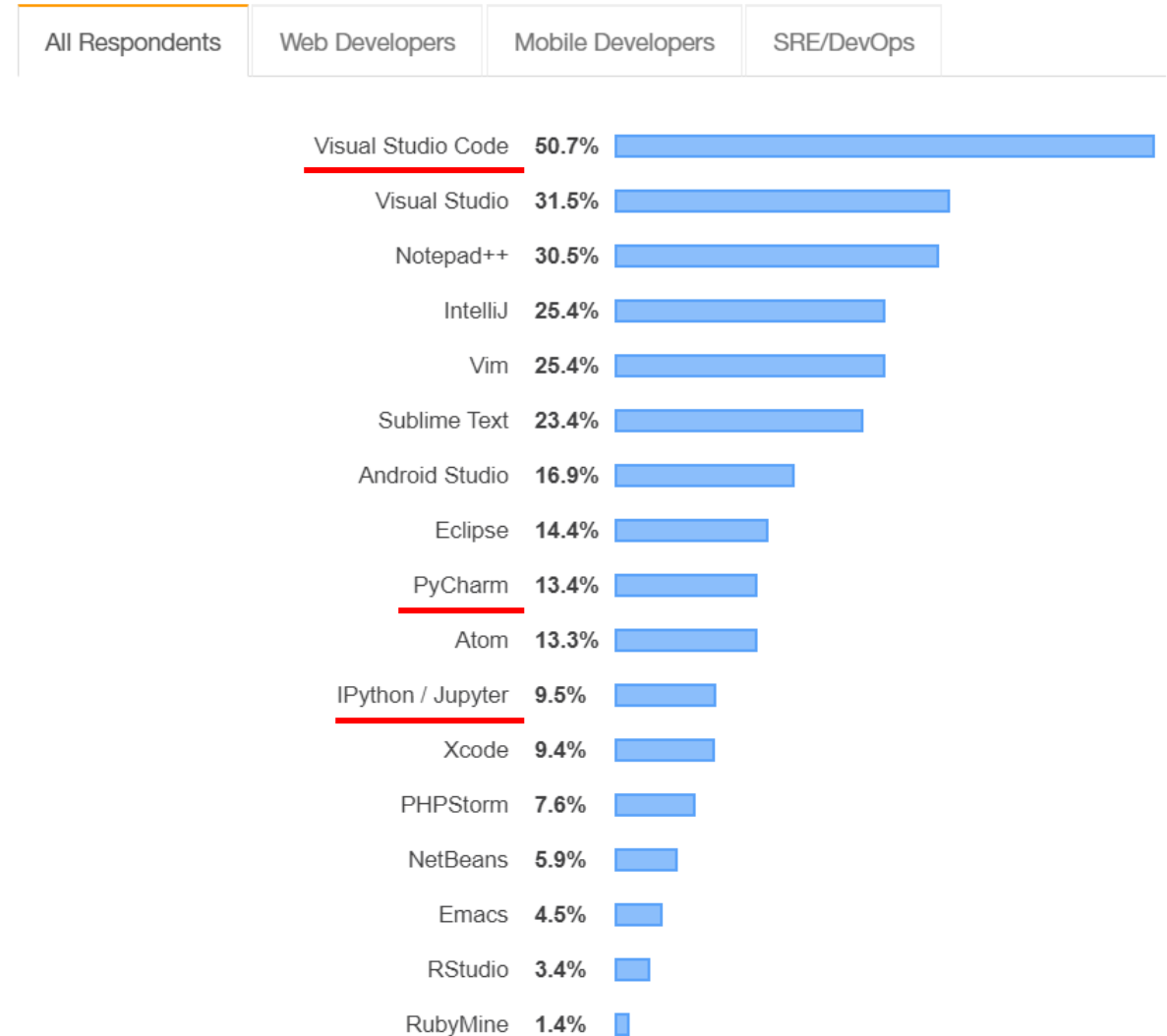
Visual Studio Code is a dominant player among **developer environment** tools this year.

There are differences in tool choices by developer type and role, but Visual Studio Code was a top choice across the board.

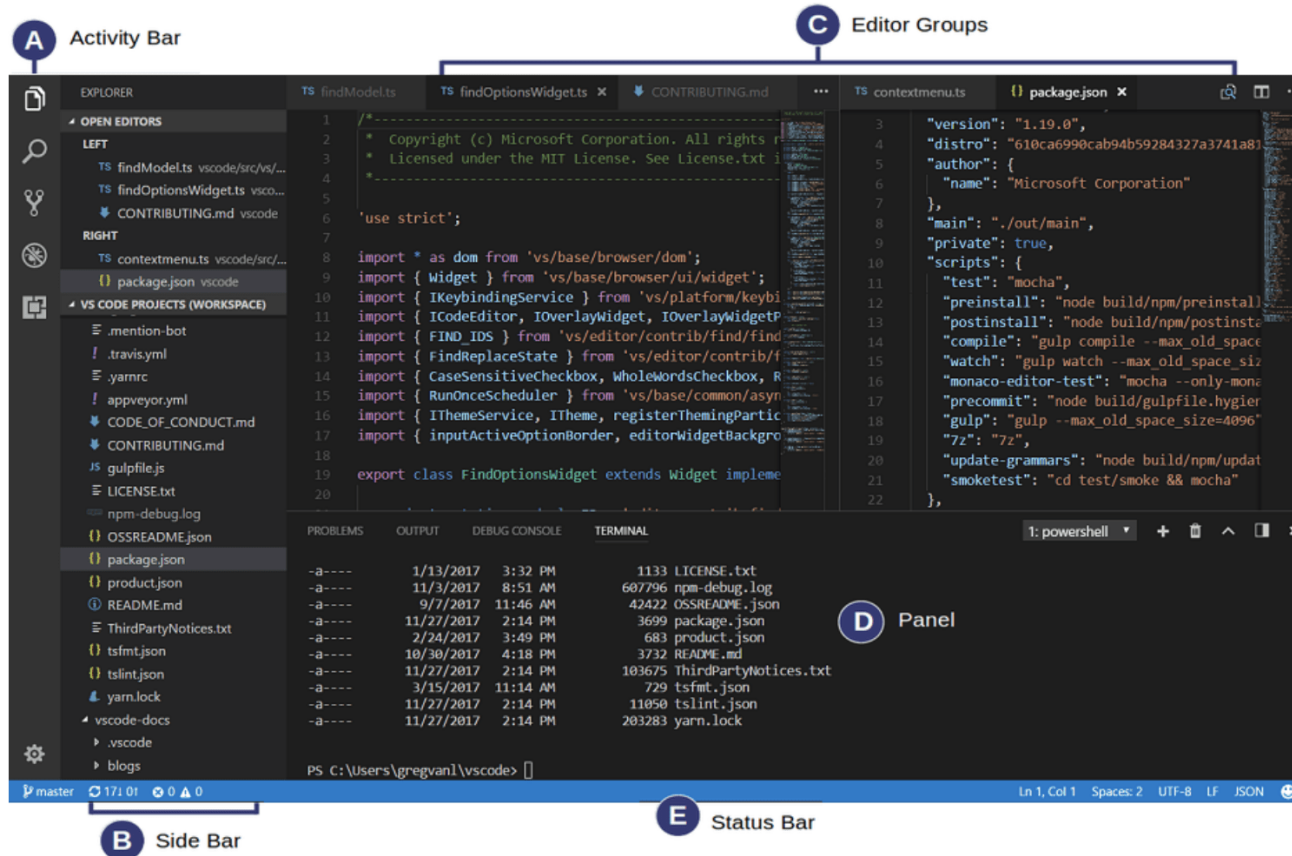
Developers who write code for mobile apps are more likely to choose Android Studio and Xcode.

A popular choice for DevOps and SREs is Vim, and **data scientists** are more likely to work in **IPython/Jupyter, PyCharm, and RStudio**.

Most Popular Development Environments



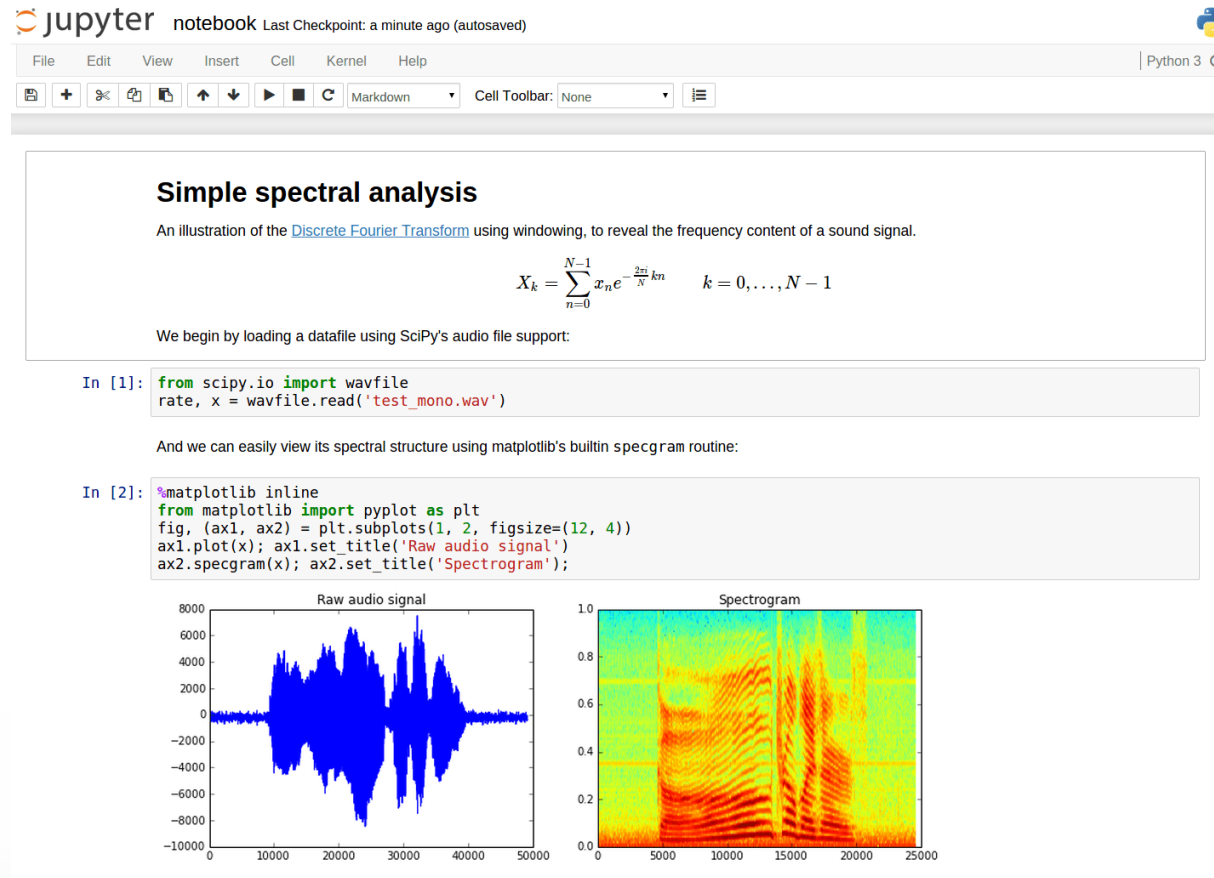
Visual Studio Code



Visual Studio Code

- Visual Studio Code (VS Code) is a free and open-source code editor created by Microsoft that can be used for Python development.
- You can add the extension to create a Python development environment.
- It provides support for debugging, embedded Git control, syntax highlighting, IntelliSense code completion, snippets, and code refactoring.
- Some of its best features are given below.
 - Thousands of plugins/extensions available through the VS Code Marketplace.
 - Powerful debugger by which the user can debug code from the editor itself.
 - Easily customizable.
 - Multi-platform, multi-language support, multi-split window feature and vertical orientation.

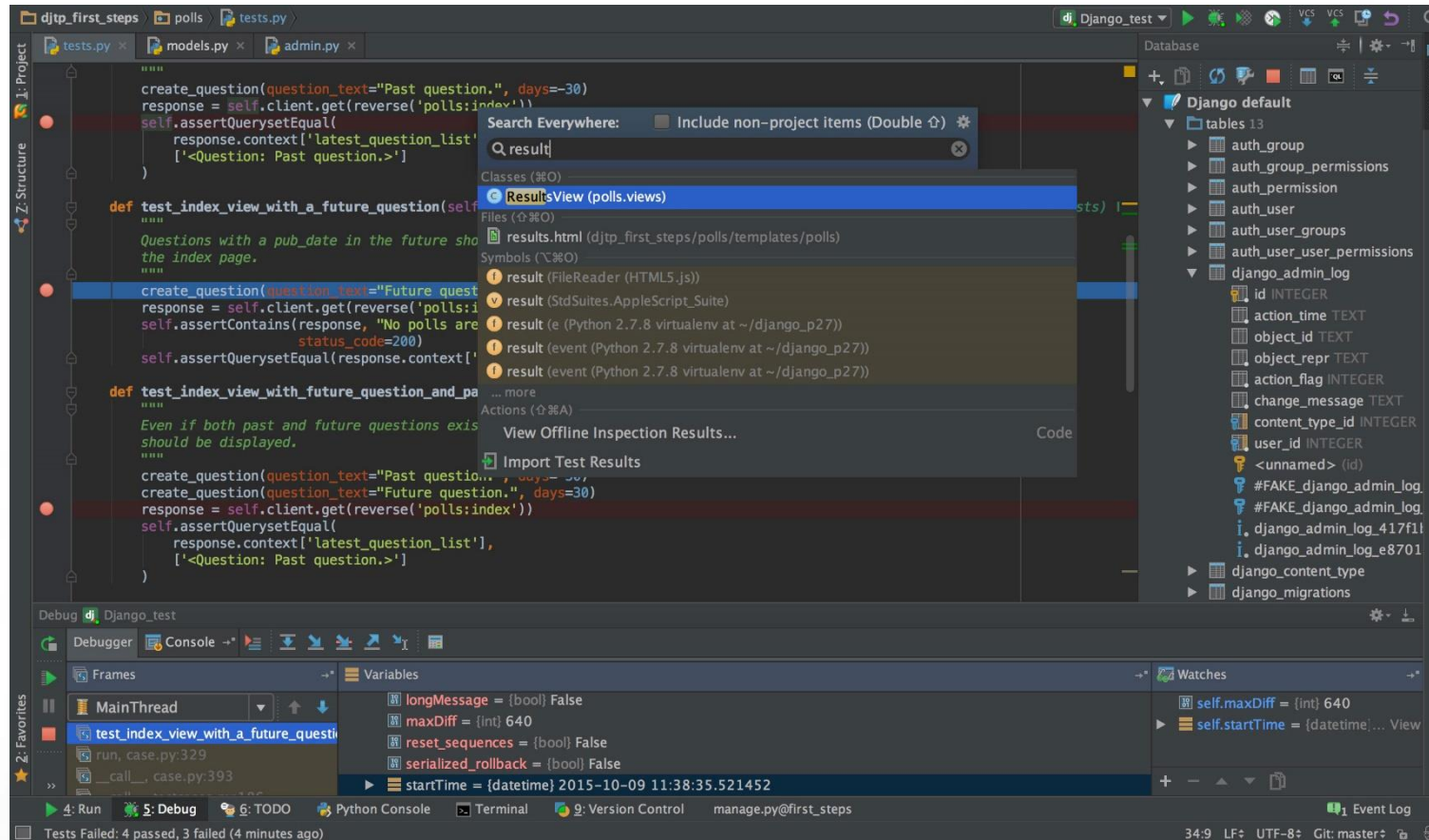
Jupyter Notebook



Jupyter Notebook

- Jupyter Notebook is a **web-based** interactive development environment
- It's well known in the data science community for analyzing, sharing and presenting the information.
- It is easy to use, open-source software that allows you to create and share live code, visualizations, etc.
- Some of its good features are given below...
 - Support for Numerical simulation, data cleaning machine learning data visualization, and statistical modeling.
 - Markdown and HTML integration.
 - Integrated data science libraries (matplotlib, NumPy, Pandas).
 - It offers you to see and edit your code to create powerful presentations.
 - You can also convert your complete work into PDF and HTML files, or you can just export it as a .py file.
 - Starting and stopping servers, opening folders and files.

PyCharm



PyCharm

- In industries most of the professional developers use PyCharm and it has been considered the best IDE for python developers.
- It was developed by the Czech company JetBrains and it's a cross-platform IDE.
- It gives daily tips to improve your knowledge of how you can use it more efficiently which is a very good feature.
- It comes in two versions community version and a professional version where community version is free but the professional version is paid.
- Below are some other features of this IDE.
 - It is considered as an intelligent code editor, fast and safe refactoring, and smart code.
 - Features for debugging, profiling, remote development, testing the code, auto code completion, quick fixing, error detection and tools of the database.
 - Support for Popular web technologies, web frameworks, scientific libraries and version control.

Spyder

The screenshot displays the Spyder Python IDE interface. The main window is divided into several panels:

- Project Explorer:** Shows a tree view of the project structure, including folders like 'Data', 'scripts', and 'app'.
- Code Editor:** Contains a Python script named 'interpolation.py'. The script generates data for analysis, performs calculations, and plots results. Key lines include:

```
7 import pylab
8 from numpy import cos, linspace, pi, sin, random
9 from scipy.interpolate import splprep, splev
10
11 # XX Generate data for analysis
12
13 # Make ascending spiral in 3-space
14 t = linspace(0, 1.75 * 2 * pi, 100)
15
16 x = sin(t)
17 y = cos(t)
18 z = t
19
20 # Add noise
21 x += random.normal(scale=0.1, size=x.shape)
22 y += random.normal(scale=0.1, size=y.shape)
23 z += random.normal(scale=0.1, size=z.shape)
24
25
26 # XX Perform calculations
27
28 # Spline parameters
29 smoothness = 3.0 # Smoothness parameter
30 k_param = 2 # Spline order
31 nests = -1 # Estimate of number of knots needed (-1 = maximal)
32
33 # Find the knot points
34 knot_points, u = splprep([x, y, z], s=smoothness, k=k_param, nests=-1)
35
36 # Evaluate spline, including interpolated points
37 xnew, ynew, znew = splev(linspace(0, 1, 400), knot_points)
38
39
40 # XX Plot results
41
42 # TODO: Rewrite to avoid code smell
43 pylab.subplot(2, 2, 1)
44 data = pylab.plot(x, y, 'bo-', label='Data with X-Y Cross Section')
45 fit = pylab.plot(xnew, ynew, 'r-', label='Fit with X-Y Cross Section')
46 pylab.legend()
47 pylab.xlabel('x')
48 pylab.ylabel('y')
49
50 pylab.subplot(2, 2, 2)
51 data = pylab.plot(x, z, 'bo-', label='Data with X-Z Cross Section')
52 fit = pylab.plot(xnew, znew, 'r-', label='Fit with X-Z Cross Section')
53 pylab.legend()
54 pylab.xlabel('x')
```
- Variable Explorer:** Displays a table of variables and their values. The table has columns for Name, Type, Size, and Value. Variables include 'array_int8', 'array_uint32', 'bars', 'df', 'filename', 'list_test', 'nrows', 'r', 'radii', 'region', 'rgb', 'series', 'test_none', and 'test_none'.
- Python Console:** Shows the execution of the code, including the generation of a 3D surface plot and a 2D polar plot. The console output includes:

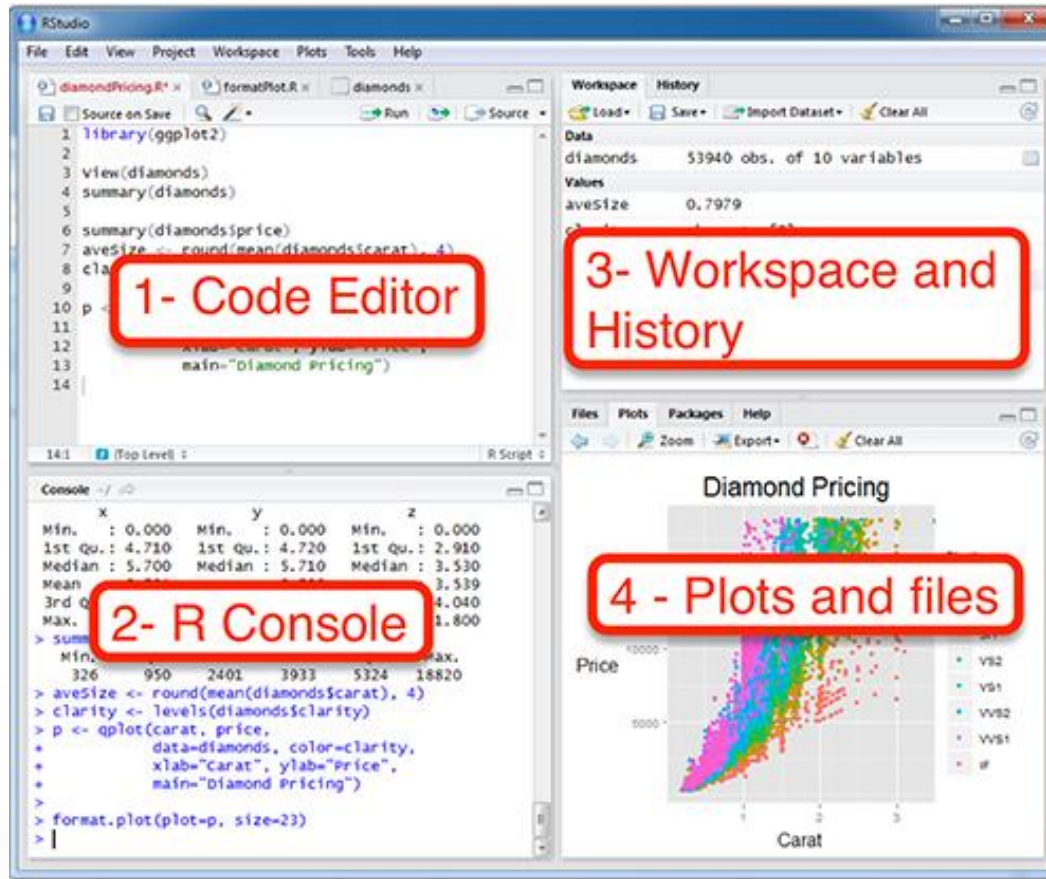
```
....:
....: ls = LightSource(270, 45)
....: # To use a custom hillshading mode, override the built-in shading
....: # in the rgb colors of the shaded surface calculated from "shade".
....: rgb = ls.shade(z, cmap=cm.gist_earth, vert_exag=0.1, blend_mode='soft')
....: surf = ax.plot_surface(x, y, z, rstride=1, cstride=1, facecolors=rgb,
....:                      linewidth=0, antialiased=False, shade=False)
....: plt.show()
```

The bottom status bar shows the current file's permissions (RW), end-of-lines (LF), encoding (UTF-8), line number (26), column number (4), memory usage (49%), and CPU usage (15%).

Spyder

- Spyder is another good open-source and cross-platform IDE written in Python.
- It is also called Scientific Python Development IDE and it is the most lightweight IDE for Python.
- It is mainly used by data scientists who can integrate with Matplotlib, SciPy, NumPy, Pandas, Cython, IPython, SymPy, and other open-source software.
- It comes with the Anaconda package manager distribution and it has some good advanced features such as edit, debug, and data exploration.
- Below are some other features of this IDE.
 - Auto code completion and syntax highlighting.
 - Ability to search and edit the variables from the graphical user interface itself.
 - Static code analysis
 - It is very efficient in tracing each step of the script execution by a powerful debugger.

R studio



Learn the Basics of Python

- <https://www.learnpython.org/>
- [Hello, World!](#)
- [Variables and Types](#)
- [Lists](#)
- [Basic Operators](#)
- [String Formatting](#)
- [Basic String Operations](#)
- [Conditions](#)
- [Loops](#)
- [Functions](#)
- [Classes and Objects](#)
- [Dictionaries](#)
- [Modules and Packages](#)

Learn the Basics of R

- <https://www.guru99.com/r-tutorial.html>
- [R Data Types, Arithmetic & Logical Operators with Example](#)
- [R Matrix Tutorial: Create, Print, add Column, Slice](#)
- [Factor in R: Categorical & Continuous Variables](#)