datacamp

When Should I Use Python vs. R?

Python and R are both great programming languages for data science and analytics. Since they're open-source, they're free to download for everyone, unlike commercial tools like SAS and SPSS. Find out their strengths and weaknesses and figure out which is better for your specific use cases.

Purpose

Either language is suitable for almost any data science task, from data manipulation and automation to ad-hoc analysis and exploring datasets. Users may leverage both languages for different purposes, e.g., conducting early-stage data analysis and exploration in R, then switching to Python when it's time to ship some data products.

Choosing Python vs. R

It's up to the individual data scientist or data analyst to choose the language that best fits their unique needs. The following questions may help with that decision.

1. **Which language do your colleagues use?**
   The benefits of being able to share code with your colleagues and maintaining a simpler software stack outweigh any benefits of one language over another.
2. **What problems do you want to solve and what tasks do you need to accomplish?**
3. **What are the net costs of learning a language?**
   It will take time to learn a new system that is better aligned for the problem you want to solve, but staying with the system you know may not be a fit for that problem.
4. **What are the commonly used tool(s) in your field?**

Who It's Used By

Python is used by programmers that want to delve into data analysis or apply statistical techniques, and by developers and programmers that turn to data science.

Python is a production-ready language, meaning it has the capacity to be a single tool that integrates with every part of your workflow!

R has been used primarily in academics and research and is great for exploratory data analysis. In recent years, enterprise usage has rapidly expanded.

Statisticians, engineers, and scientists without computer programming skills. It's popular in academia, finance, pharmaceuticals, media, and marketing.

Usability

- People with a software engineering background may find Python comes more naturally to them than R.
- Coding and debugging is easy because of the simple syntax.
- The indentation of code affects its meaning.
- Any piece of functionality is always written the same way with Python.

- If you have no coding experience, then R may be easier to learn.
- Statistical models can be written with only a few lines.
- The same piece of functionality can be written in several ways with R.

Ecosystem

- Python has a robust ecosystem and is commonly considered one of the easier programming languages to read and learn. Its programming syntax is simple and its commands mimic the English language.
  E.g. print("Hello world!")
- Python code is syntactically clear and elegant, easily interpretable, and easy to type.
- It's great for building data science pipelines and machine learning products integrated with web frameworks at scale. But watch out for dependencies and installing Python libraries!
- It's great for building data science pipelines and machine learning products integrated with web frameworks at scale. But watch out for dependencies and installing Python libraries!
- The Python Package Index (PyPi) and Anaconda are repositories of Python software with all libraries. Users can contribute to these repositories, but it's a bit complicated in practice to do so.

- R has a rich ecosystem of cutting-edge interface packages available to communicate between open-source languages.
- This allows users to string their workflows together, which is especially useful for data analysis.
- Packages are available at:
  - Comprehensive R Archive Network (CRAN): "Task Views" page lists a wide range of tasks for which R packages are available and users can easily contribute to.
  - Bioconductor: Open source software for bioinformatics
  - GitHub: Web-based Git repository hosting service
- Search through these sources easily with Rdocumentation
- Packages are collections of R functions, data, and compiled code. They can be installed in R with one line.

Flexibility

Python is flexible for creating something that has never been done before. Developers can also use it for scripting websites or other applications.

It's easy to use complex functions in R. All kinds of statistical tests and models are readily available and easily used.

Ease of Learning

Python's focus on readability and simplicity means its learning curve is relatively linear and smooth.

Python is considered a good language for beginner programmers.

R is easier to learn when you start out, but the intricacies of advanced functionalities makes it more difficult to develop expertise.

R is not hard for experienced programmers to learn.

Advantages

- General-purpose programming languages are useful beyond just data analysis.
- Has gained popularity for its code readability, speed, and many functionalities.
- Great for mathematical computation and learning how algorithms work.
- Has high ease of deployment and reproducibility.

- Widely considered the best tool for making beautiful graphs and visualizations.
- Has many functionalities for data analysis.
- Great for statistical analysis.
- Built around a command line, but the majority of R users work inside of RStudio, an environment that includes a data editor, debugging support, and a window to hold graphics as well.

Disadvantages

- Python doesn't have as many libraries for data science as R.
- Python requires rigorous testing as errors show up in runtime.
- Visualizations are more convoluted in Python than in R, and results are not as eye-pleasing or informative.

Python packages for data visualization:

- **seaborn**: Library based on Matplotlib
- **Bokeh**: Interactive visualization library
- **Pygal**: Create dynamic dynamic svg charts

- For people with no software engineering experience, base R can be more difficult to learn because it was developed by statisticians, not to make coding easier. But R has a set of packages known as the Tidyverse, which provides powerful yet easy-to-learn tools for importing, manipulating, visualizing, and reporting on data.
- Finding the right packages to use in R may be time consuming.
- There are many dependencies between R libraries.
- R can be considered slow if code is written poorly.
- Not as popular as Python for deep learning and NLP.

Use Case: Data Analysis

Usage

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.

Data Handling Capabilities

Python requires users to install packages for data analysis, and these packages have greatly improved in recent years.

NumPy and **pandas**, among others, are popular for data analysis.

R is great for data analysis because of its huge number of packages, readily usable tests, and the advantage of using formulas.

It can handle basic data analysis without needing to install packages. Big datasets require the use of packages such as **data.table** and **dplyr**.

Getting Started

IDE

There are many Python IDEs to choose from which drastically reduce the overhead of organizing code, output, and notes files. **Jupyter Notebooks** and **Spyder** are popular, and **Jupyter Lab** is gaining traction.
Tip: Also try **Rodeo**, the "data science IDE for Python."

RStudio is the most popular R IDE. It's available in two formats: **RStudio Desktop** for running locally as a regular desktop application and **RStudio Server** for access via web browser while running on a remote Linux server.

Popular Libraries and Packages

- pandas to easily manipulate data
- **SciPy** and **NumPy** for scientific computing
- **Scikit-learn** for machine learning
- **Matplotlib and seaborn** to make graphics
- **statsmodels** to explore data, estimate statistical models, and perform statistical tests and unit tests

- **dplyr, tidyr** and **data.table** to easily manipulate data
- **stringr** to manipulate strings
- **zoo** to work with regular and irregular time series
- **ggplot2** to visualize data
- **caret** for machine learning

> "R is currently head-and-shoulders above Python for data analysis, but I remain convinced that Python can catch up, easily and quickly."
>
> JAN GALKOWSKI, COMPUTATIONAL ENGINEER

Support and Communities

- DataCamp Slack Community
- Stack Overflow
- Reddit Python
- PyLadies
- pydata
- pystatsmodels
- numpy-discussion and sci-py-user

- DataCamp Slack Community
- Stack Overflow
- Reddit rstats
- Rdocumentation
- R-help
- ROpenSci
- Jumping Rivers list of local R User Groups

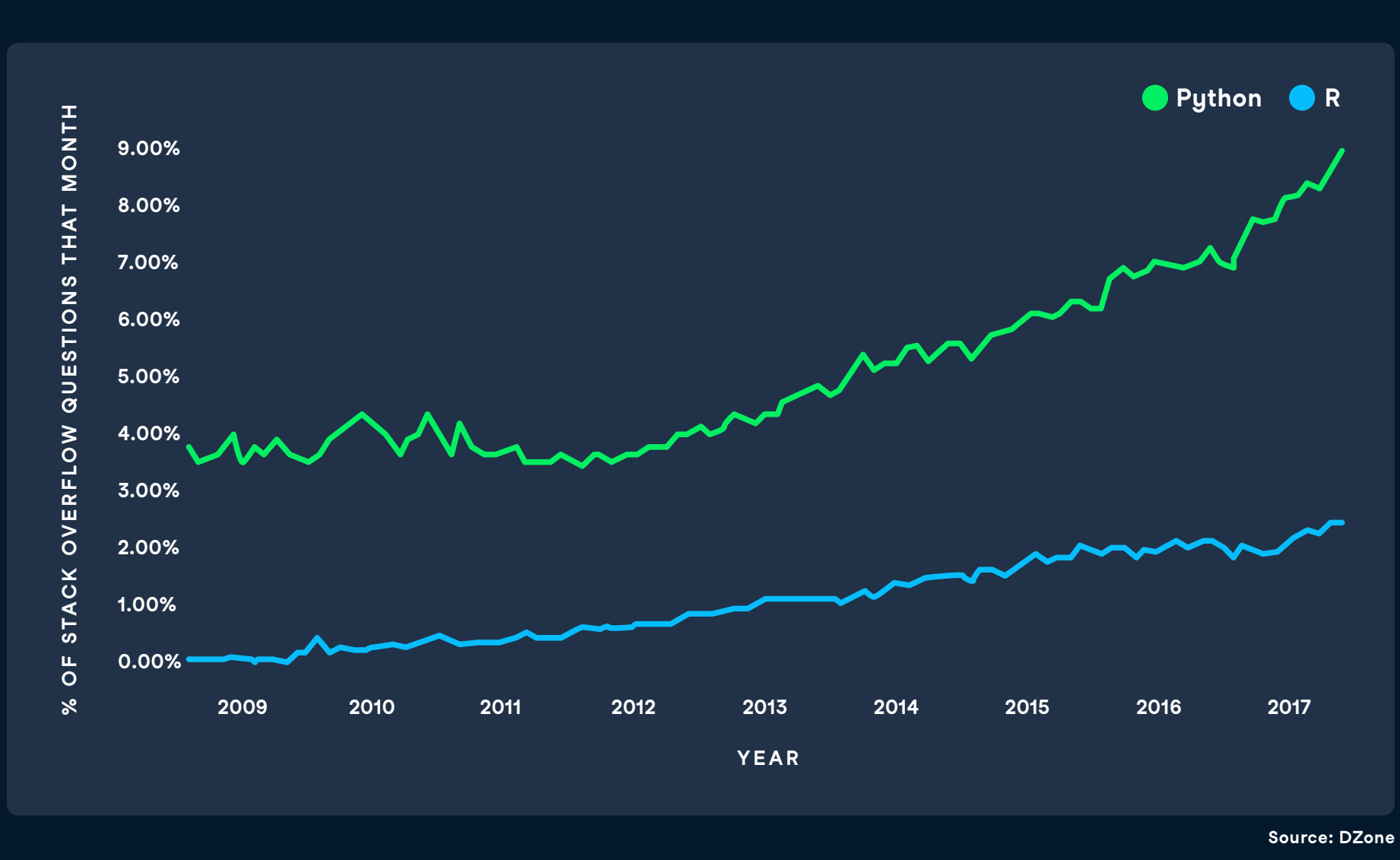
Trends and Highlights

Popularity Rankings, Python vs. R

	Python	R
2016	#3	#5
2017	#1	#6
2018	#1	#7
2019	#1	#5

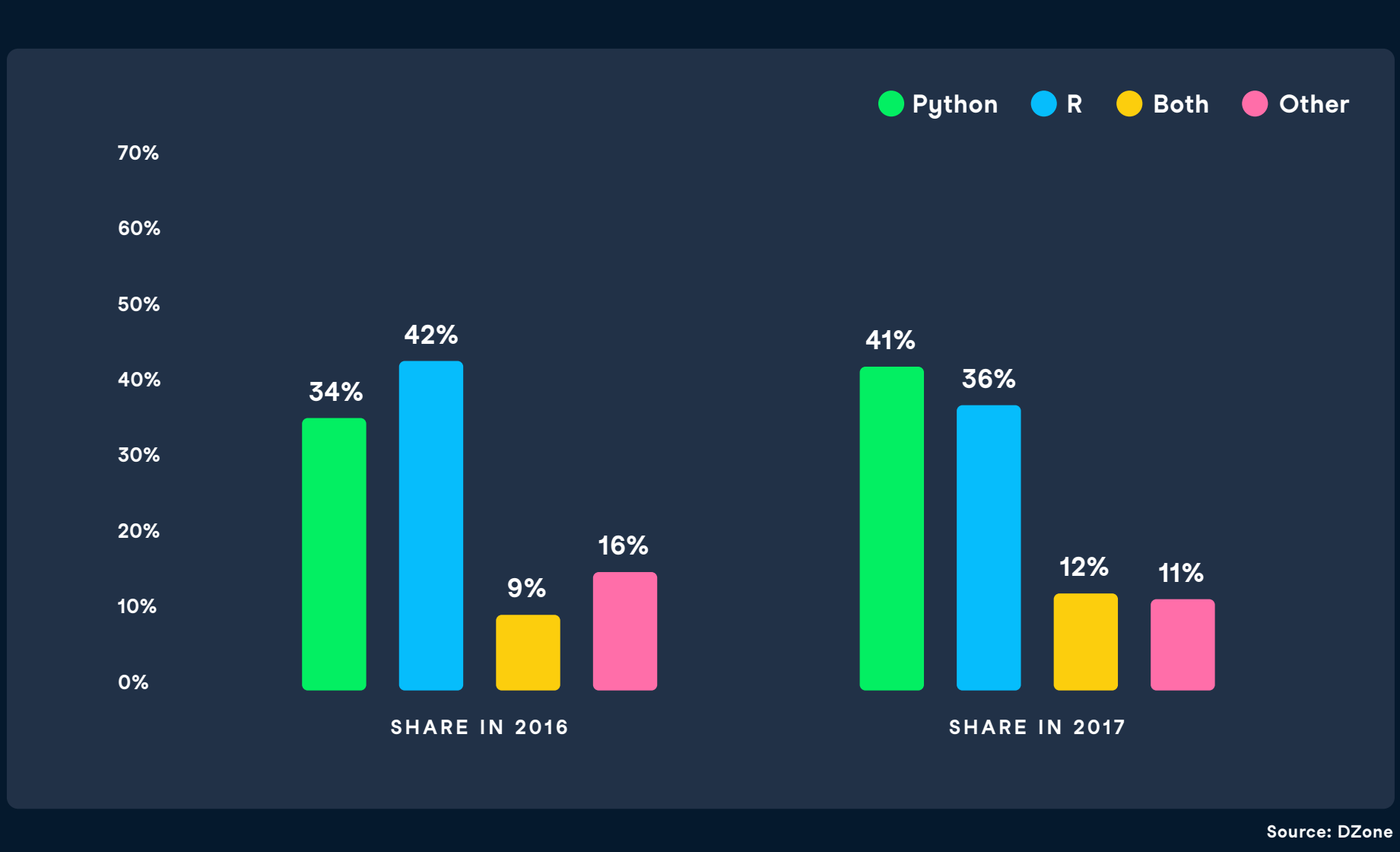
Source: IEEE Spectrum, Sep 2019

Popularity on Stack Overflow, Python vs. R



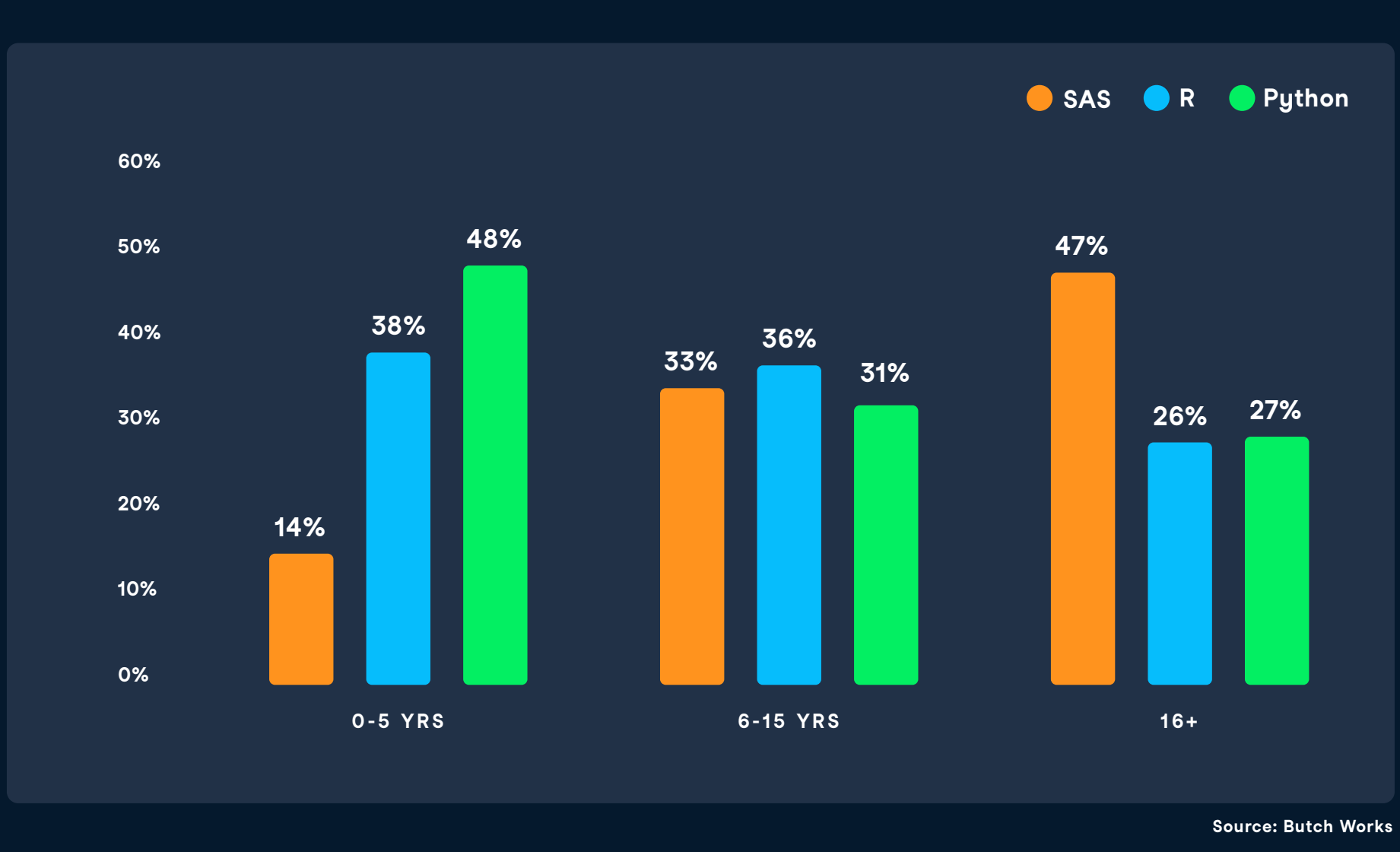
Source: DZone

Python, R or both or other platforms for data science



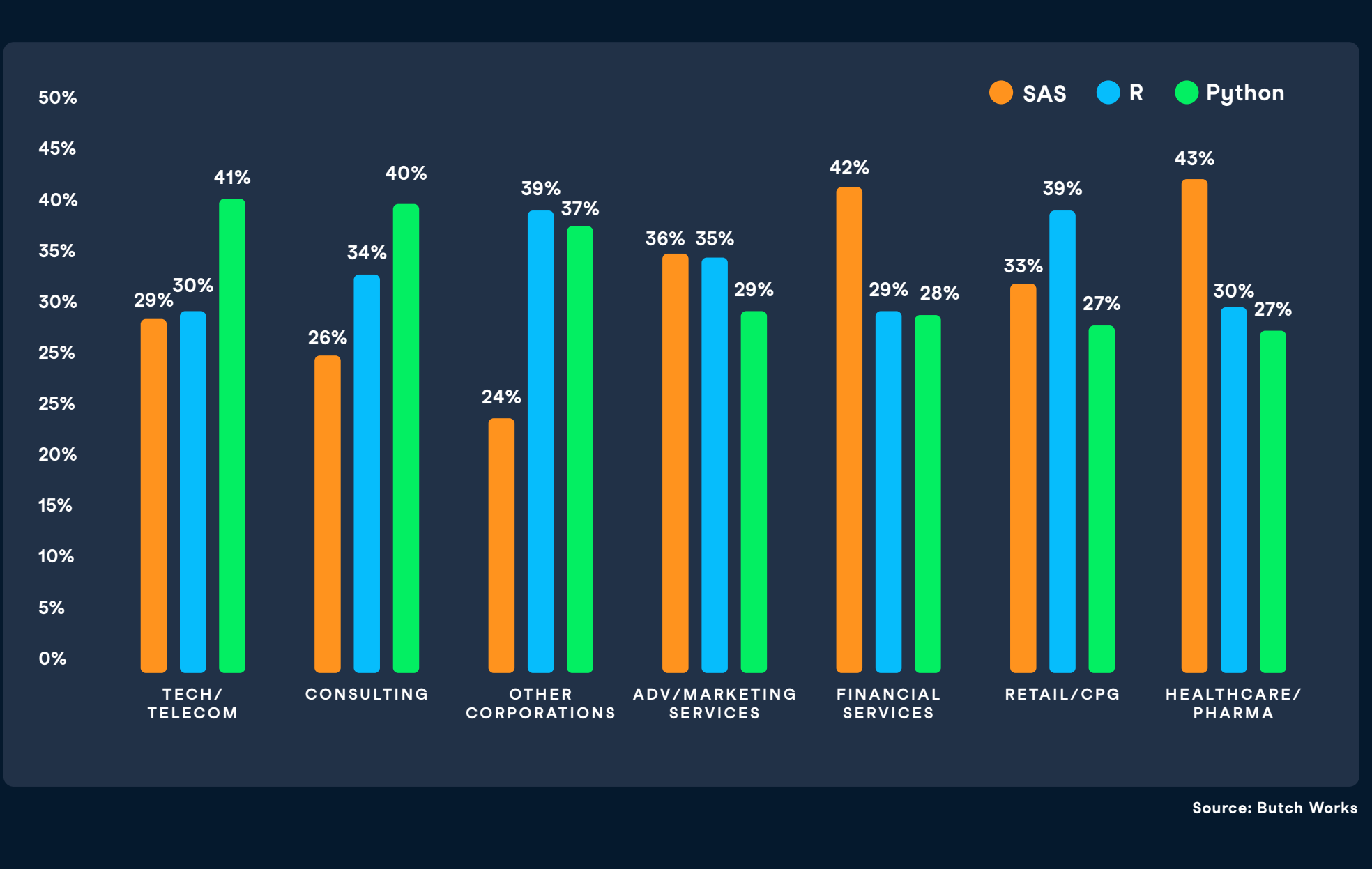
Source: DZone

SAS, R or Python preference by years of experience



Source: Butch Works

SAS, R or Python preference by industry



Source: Butch Works

User loyalty, Python vs. R

91% OF PYTHON USERS REMAIN LOYAL TO PYTHON

10% switch from R to Python

5% switch from Python to R

74% OF R USERS REMAIN LOYAL TO R

Source: KDnuggets polls 2016

datacamp

Learn Python and R online with DataCamp

The skills people and businesses need to succeed are changing. No matter where you are in your career or what field you work in, you will need to understand the language of data. With DataCamp, you learn data science today and apply it tomorrow.

Get Started