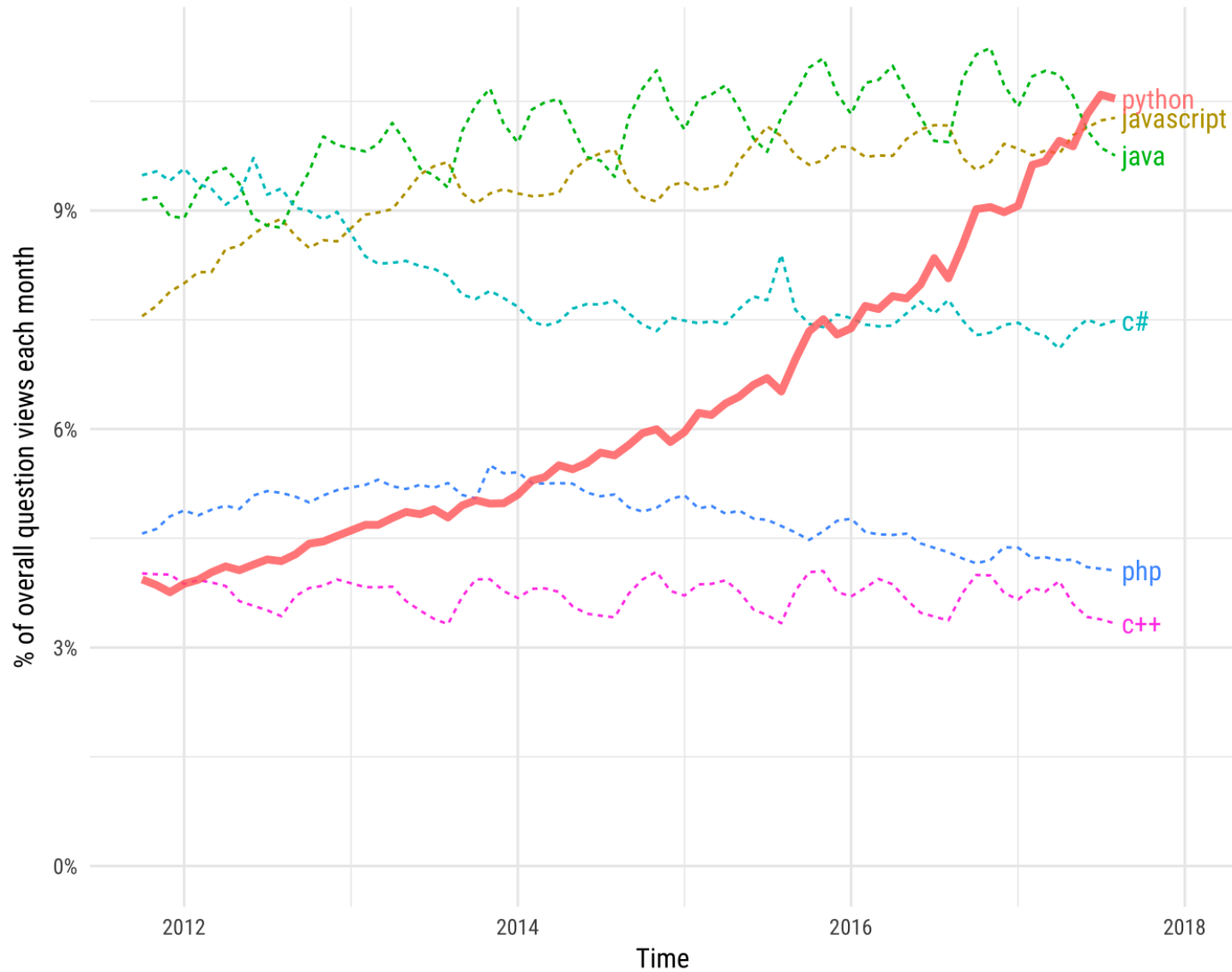


Part II: Python

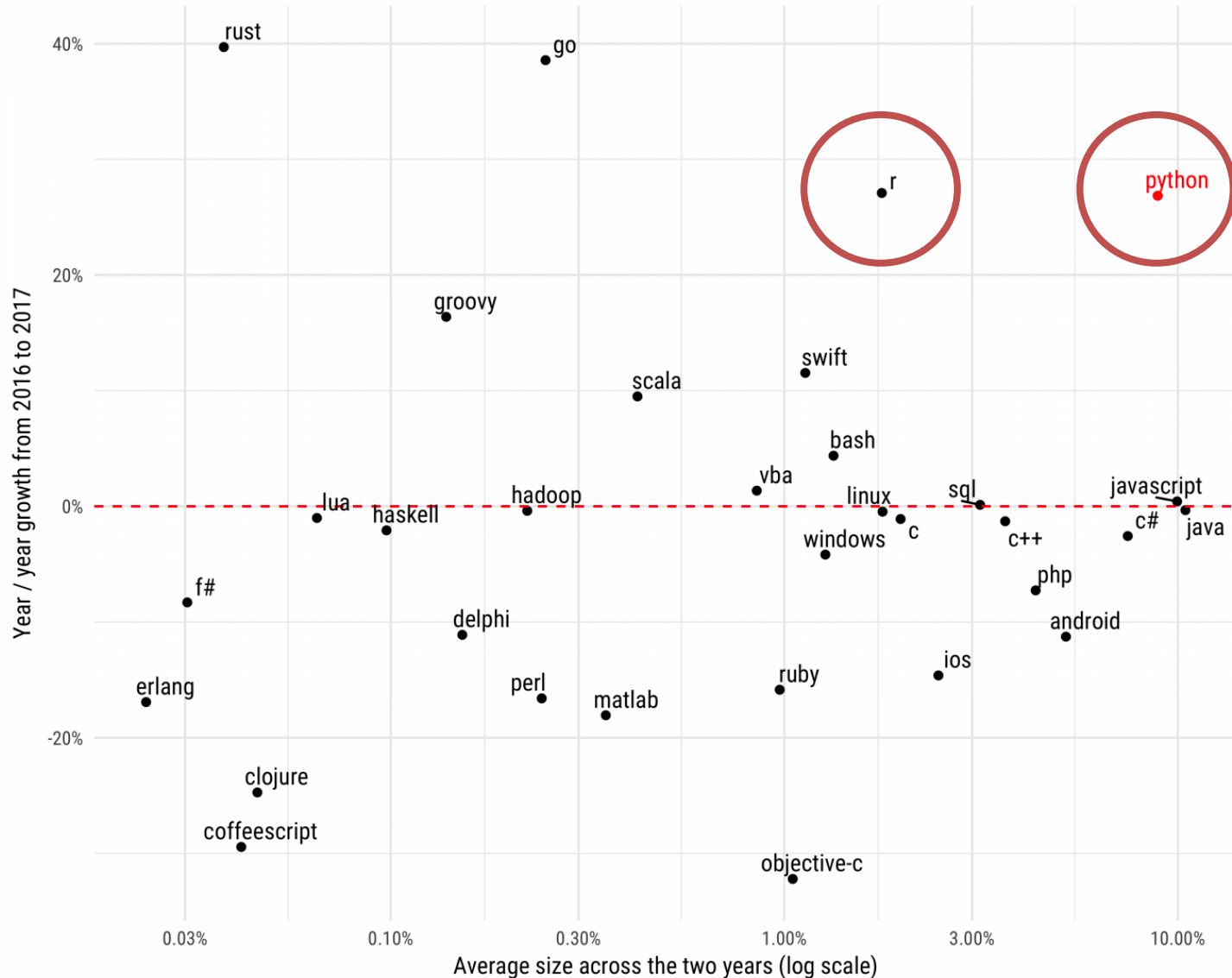
Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Year over year growth in traffic to programming languages/platforms

Comparing question views in January-August of 2016 and 2017, in World Bank high-income countries. TypeScript had a growth rate of 142% and an average size of .36%; and was omitted.



Python has many applications



- Web development
- Application development
- Computer graphics
- Scientific computing
 - Bioinformatics
 - Machine learning
 - Simulations

<https://www.python.org/about/quotes/>



Three alternatives to get Python



- Google Colaboratory (free)
<https://colab.research.google.com/>
- Anaconda (free, ~1.5GB of space required)
- CoCalc (\$14/month, can try for free)



Google Colaboratory


 Hello, Colaboratory 


File Edit View Insert Runtime Tools Help

 CODE  TEXT

 CELL  CELL

 COPY TO DRIVE  DISCARD CHANGES

 CONNECTED

 EDITING





 SHARE

Table of contents

Code snippets 


 

Welcome to Colaboratory!

Colaboratory is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.


Colaboratory notebooks are stored in [Google Drive](#) and can be shared just as you would with Google Docs or Sheets. Colaboratory is free to use.


For more information, see our [FAQ](#).

 **GPU Support (NEW!)**

Colab now supports running TensorFlow computations on a GPU. Simply select "GPU" in the Accelerator drop-down in Notebook Settings (either through the Edit menu or the command palette at cmd/ctrl-shift-P).

```
[ ] import tensorflow as tf
    tf.test.gpu_device_name()
```

 `'/device:GPU:0 '`

 **Python 3**

Colab now supports both Python2 and Python3 for code execution.

- When creating a new notebook, you'll have the choice between Python 2 and Python 3.

CoCalc

[Policies](#)[Software](#)[Pricing](#)[API](#)[Sign In](#)

Collaborative Calculation in the Cloud

Create Your Free CoCalc Account!

or [sign in](#) with your account

Online computing environment

Anaconda

PYTHON
THE FASTEST
GROWING
OPEN DATA
SCIENCE
PLATFORM



Modern open source analytics platform
powered by Python

[DOWNLOAD FOR FREE](#)

<http://www.continuum.io>

[Home](#)[Environments](#)[Projects \(beta\)](#)[Learning](#)[Community](#)[Documentation](#)[Developer Blog](#)[Feedback](#)Applications on base (root)[Channels](#)[Refresh](#)

jupyterlab

0.31.5

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

[Launch](#)

notebook

5.4.0

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

[Launch](#)

qtconsole

4.3.1

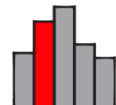
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

[Launch](#)

spyder

3.2.6

Scientific PYTHON Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing



glueviz

0.12.0

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



orange3

3.4.1

Component based data mining framework. Data visualization and data analysis for guided and unsupervised interactive workflows.

[Home](#)[Environments](#)[Projects \(beta\)](#)[Learning](#)[Community](#)[Documentation](#)[Developer Blog](#)[Feedback](#)

Appl

Mix text and python code,
similar to R Studio/ R Markdown

[Refresh](#)

jupyterlab

0.31.5

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

[Launch](#)jupyter
notebook

5.4.0

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

[Launch](#)

qtconsole

4.3.1

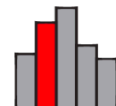
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

[Launch](#)

spyder

3.2.6

Scientific PYTHON Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing



glueviz

0.12.0

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



orange3

3.4.1

Component based data mining framework. Data visualization and data analysis for guided and unsupervised interactive workflows.

Jupyter Lab

Jupyter Lab runs in a
web page in your browser

The screenshot shows the Jupyter Lab web interface in a browser window. The browser's address bar displays `localhost:8889/lab`. The Jupyter Lab interface includes a top menu bar with `File`, `Edit`, `View`, `Run`, `Kernel`, `Tabs`, `Settings`, and `Help`. On the left, a sidebar contains a `Files` panel showing the `AnacondaProjects` directory with a table of files:

Name	Last Modified
Untitled.ipynb	a minute ago

Below the `Files` panel are sections for `Running`, `Commands`, `Cell Tools`, and `Tabs`. The main workspace displays a notebook titled `Untitled.ipynb` with a `Code` cell type. The first code cell contains the input `In [1]: 1+2` and the output `Out[1]: 3`. A second, empty code cell is visible below it, starting with `In []:`. The browser window's title bar shows the name `Claus`.

Counting like a computer scientist

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...

Indexing in Python

P	y	t	h	o	n
0	1	2	3	4	5

Indexing in Python

P	y	t	h	o	n
0	1	2	3	4	5

```
In [1]: x="Python"
```

```
In [2]: x[0]
```

```
Out[2]: 'P'
```

Indexing in Python

P	y	t	h	o	n
0	1	2	3	4	5

```
In [1]: x="Python"
```

```
In [2]: x[1:4] ← We index from the first element to  
Out[2]: 'yth'    one past the last element
```

Indexing in Python

P	y	t	h	o	n
0	1	2	3	4	5

```
In [1]: x="Python"
```

```
In [2]: x[3:] ← Missing number means "to the end"
```

```
Out[2]: 'hon'
```


We can also index in reverse

P	y	t	h	o	n
-6	-5	-4	-3	-2	-1

We can also index in reverse

P	y	t	h	o	n
-6	-5	-4	-3	-2	-1

```
In [1]: x="Python"
```

```
In [2]: x[-6]
```

```
Out[2]: 'P'
```

We can also index in reverse

P	y	t	h	o	n
-6	-5	-4	-3	-2	-1

```
In [1]: x="Python"
```

```
In [2]: x[-5:-2] ← Again, we index one  
Out[2]: 'yth'      past the last element
```

We can also index in reverse

P	y	t	h	o	n
-6	-5	-4	-3	-2	-1

```
In [1]: x="Python"
```

```
In [2]: x[-3:] ← This captures the last 3 characters
```

```
Out[2]: 'hon'
```