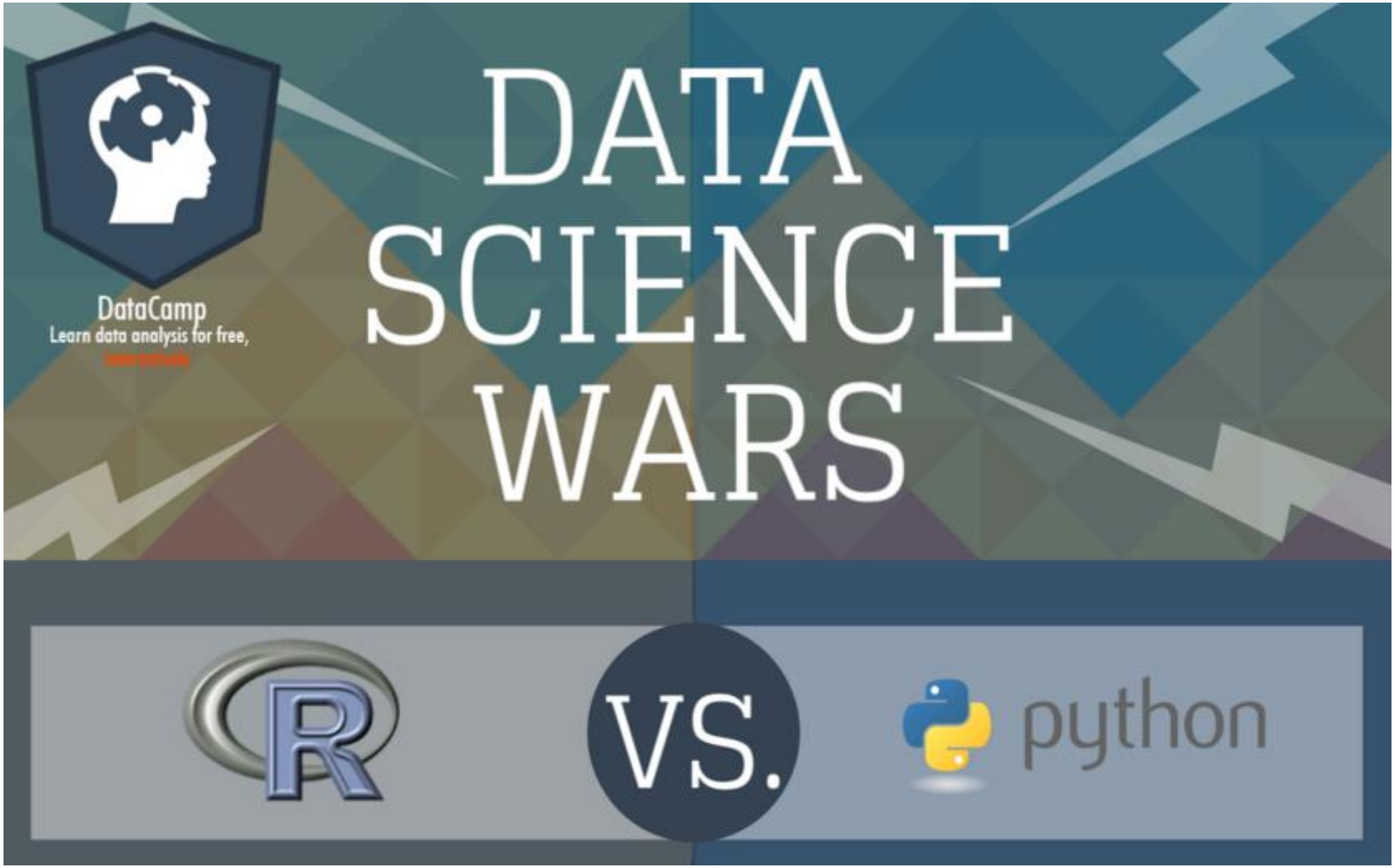
CSE 564
Visualization & Visual Analytics

R vs. Python

Klaus Mueller

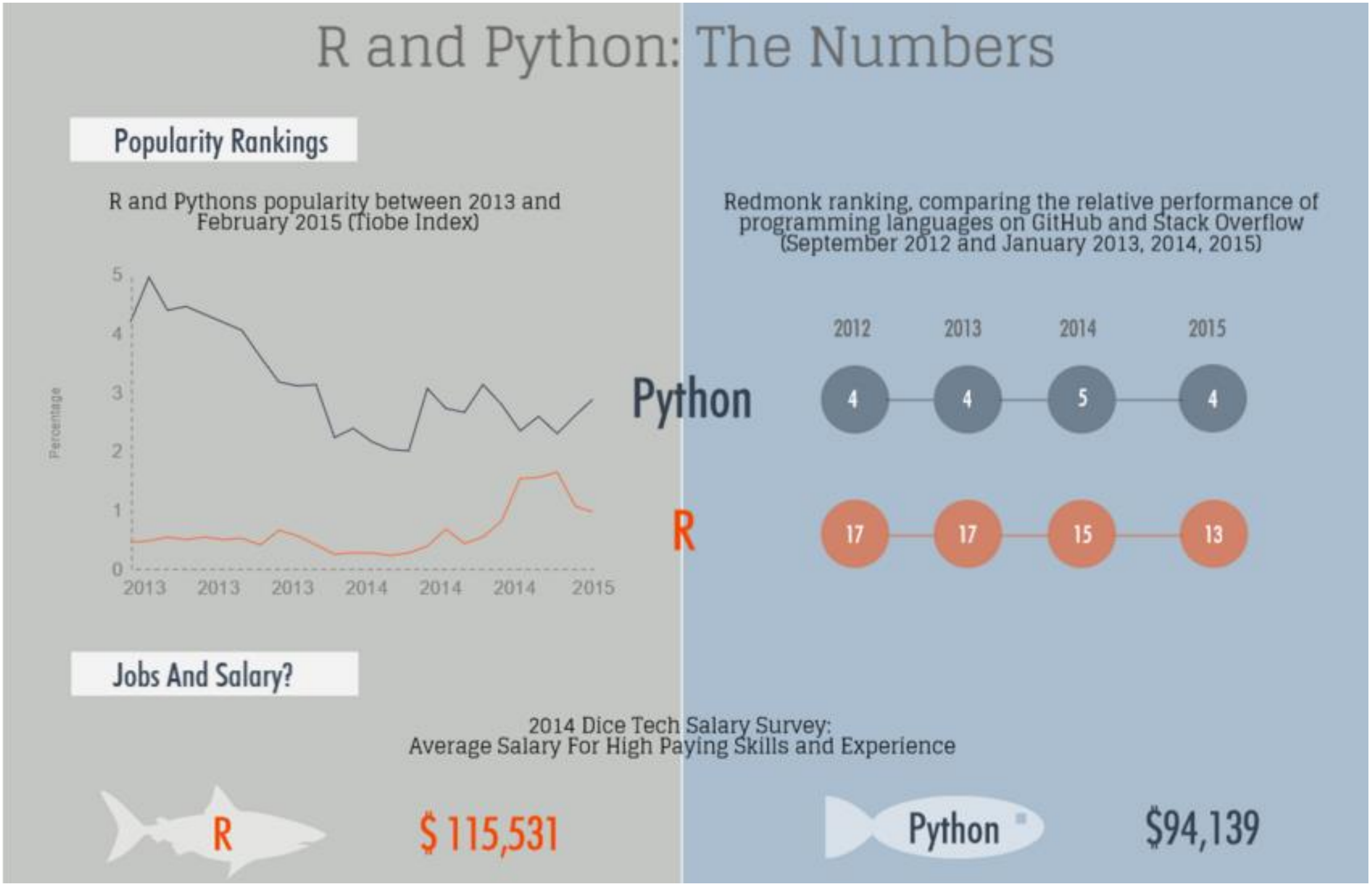
Computer Science Department
Stony Brook University

R vs. Python



https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis

R vs. Python



R and Python: The Numbers

Popularity Rankings

R and Pythons popularity between 2013 and February 2015 (Tiobe Index)

Redmonk ranking, comparing the relative performance of programming languages on GitHub and Stack Overflow (September 2012 and January 2013, 2014, 2015)

	2012	2013	2014	2015
Python	4	4	5	4
R	17	17	15	13

Jobs And Salary?

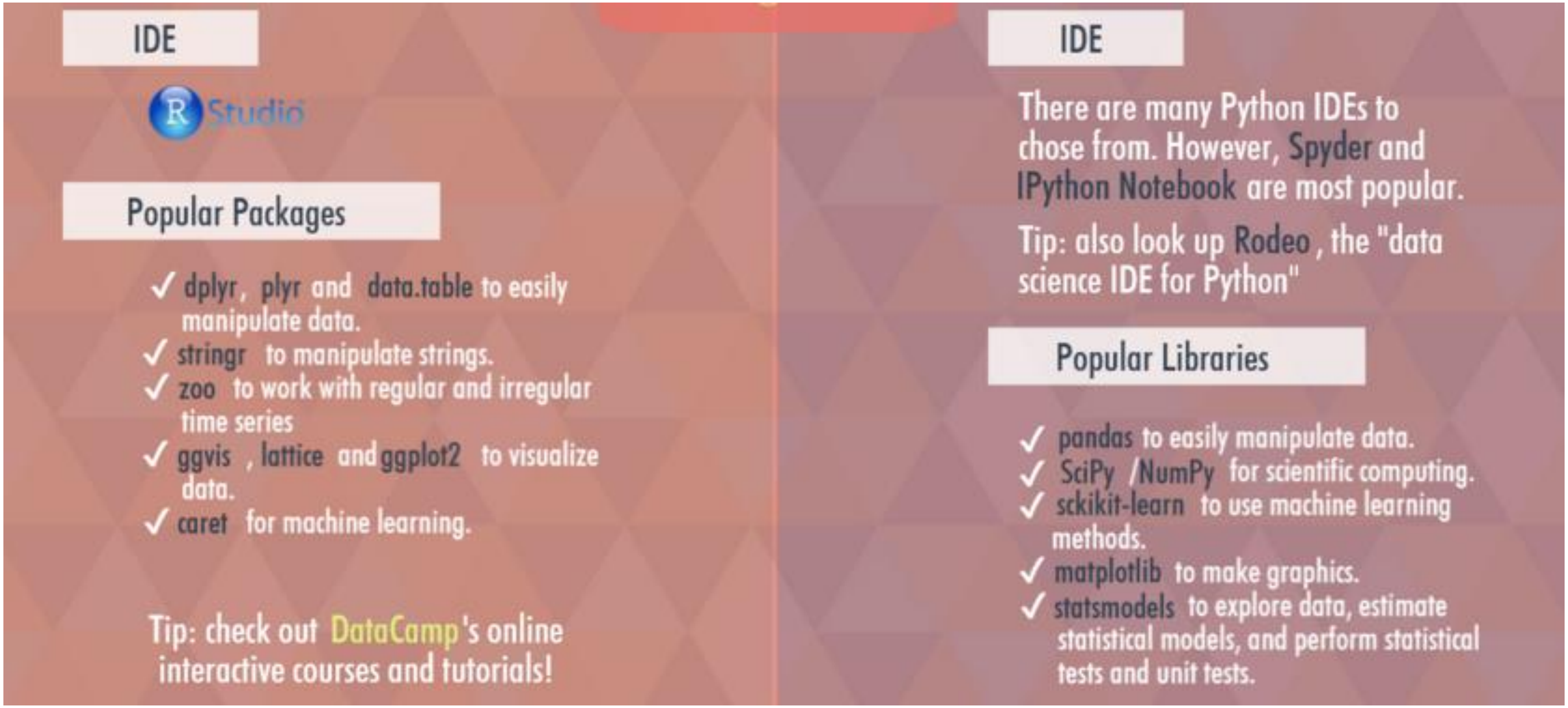
2014 Dice Tech Salary Survey:
Average Salary For High Paying Skills and Experience

R $ 115,531

Python $94,139

https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis

R vs. Python



IDE

R Studio

Popular Packages

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

Tip: check out DataCamp's online interactive courses and tutorials!

IDE

There are many Python IDEs to chose from. However, Spyder and IPython Notebook are most popular.

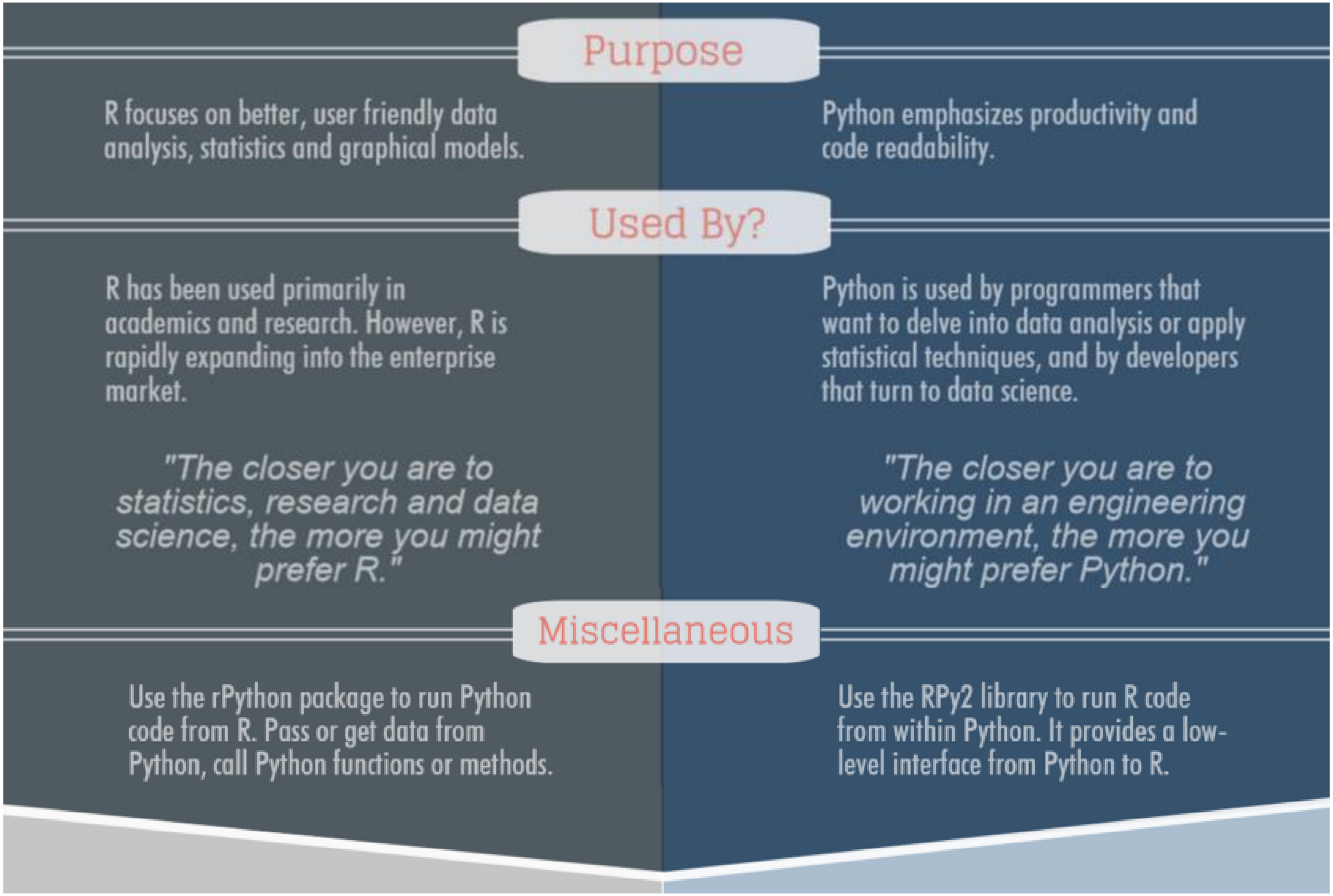
Tip: also look up Rodeo, the "data science IDE for Python"

Popular Libraries

- ✓ pandas to easily manipulate data.
- ✓ SciPy /NumPy for scientific computing.
- ✓ sckikit-learn to use machine learning methods.
- ✓ matplotlib to make graphics.
- ✓ statsmodels to explore data, estimate statistical models, and perform statistical tests and unit tests.

https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis

R vs. Python



Purpose

R focuses on better, user friendly data analysis, statistics and graphical models.

Python emphasizes productivity and code readability.

Used By?

R has been used primarily in academics and research. However, R is rapidly expanding into the enterprise market.

"The closer you are to statistics, research and data science, the more you might prefer R."

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, the more you might prefer Python."

Miscellaneous

Use the rPython package to run Python code from R. Pass or get data from Python, call Python functions or methods.

Use the RPy2 library to run R code from within Python. It provides a low-level interface from Python to R.

https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis

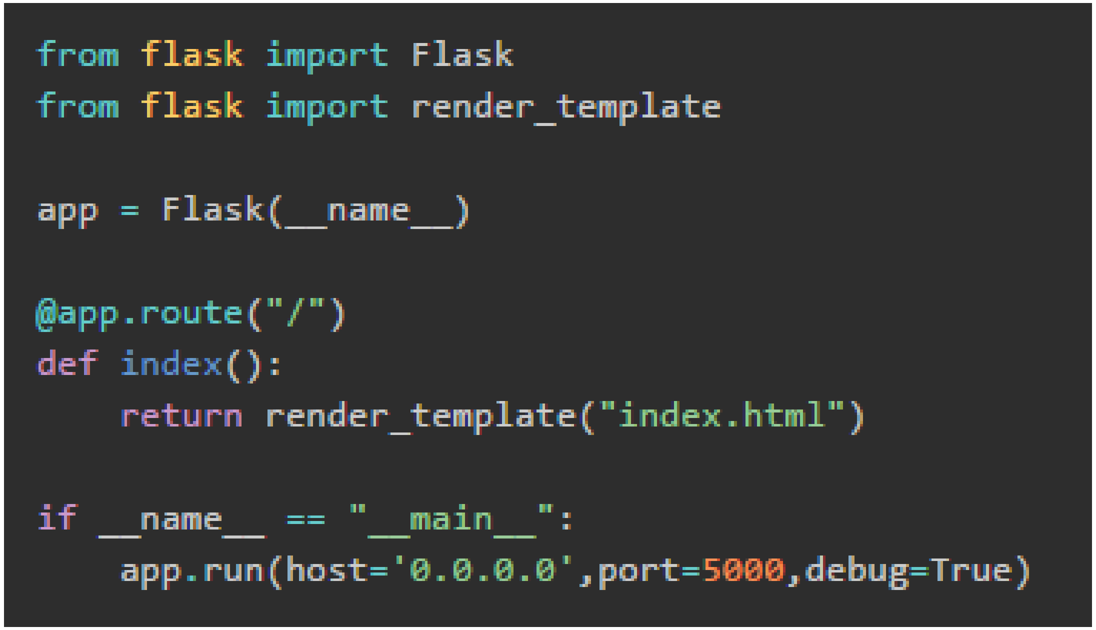
Coupling D3 with Python

See this excellent page for more detail

- uses MongoDB as a NoSQL database (non-relational SQL)

Step 1: Build a python server, say app.py

- use Flask as the web framework

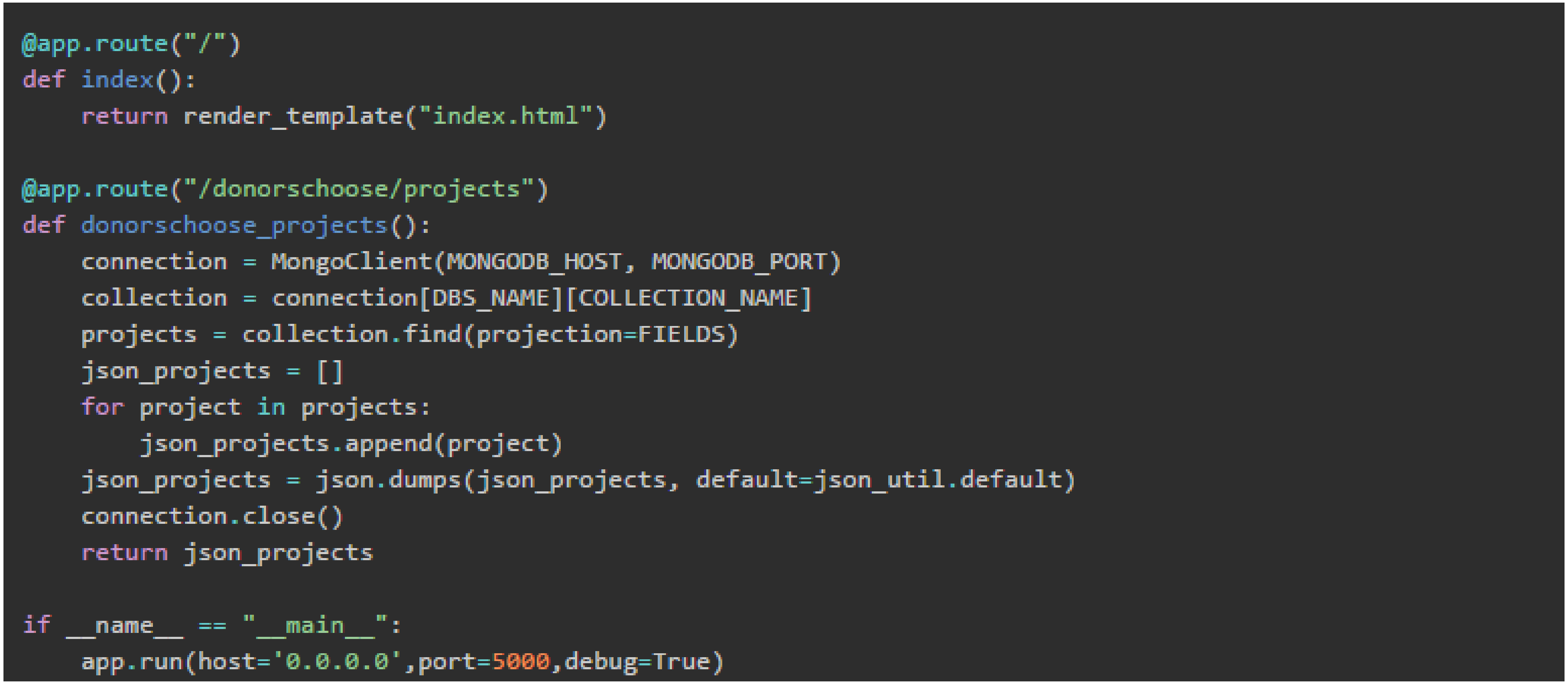


```
from flask import Flask
from flask import render_template

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("index.html")

if __name__ == "__main__":
    app.run(host='0.0.0.0',port=5000,debug=True)
```

Coupling D3 with Python

Step 2: Add all your processing code to app.py

- in this case it mainly involves storing data into the database

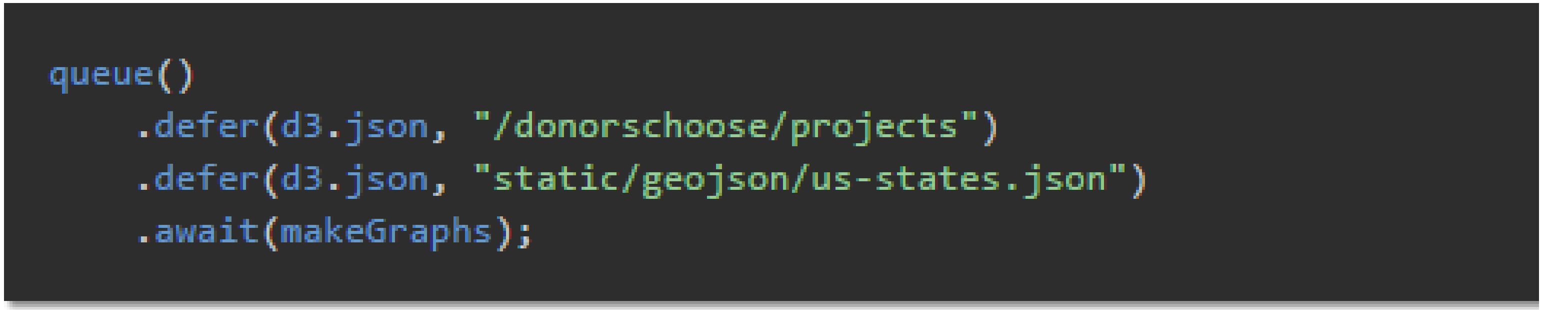


```
@app.route("/")
def index():
    return render_template("index.html")

@app.route("/donorschoose/projects")
def donorschoose_projects():
    connection = MongoClient(MONGODB_HOST, MONGODB_PORT)
    collection = connection[DBS_NAME][COLLECTION_NAME]
    projects = collection.find(projection=FIELDS)
    json_projects = []
    for project in projects:
        json_projects.append(project)
    json_projects = json.dumps(json_projects, default=json_util.default)
    connection.close()
    return json_projects

if __name__ == "__main__":
    app.run(host='0.0.0.0',port=5000,debug=True)
```

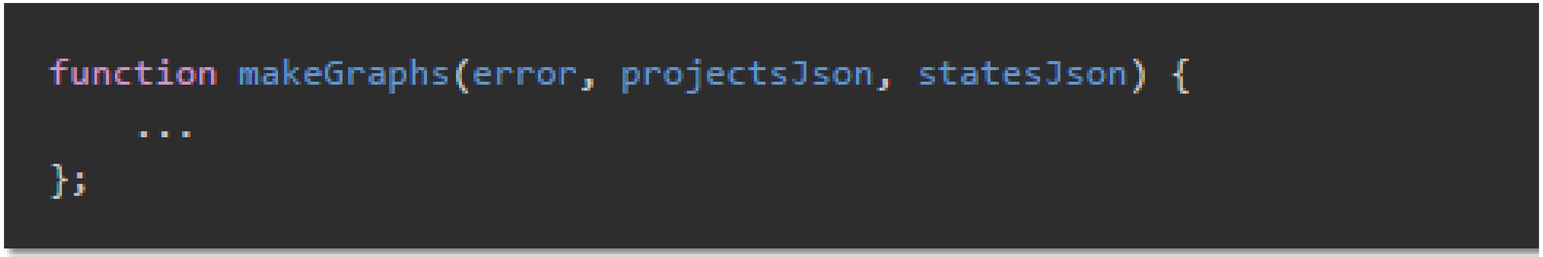
Coupling D3 with Python

Step 3: Build the charts

- create a JavaScript file, say, graphs.js
- gets the data from the python URL and other provided JSON files
- calls function, here makeGraphs(), to do the d3 rendering



```
queue()
    .defer(d3.json, "/donorschoose/projects")
    .defer(d3.json, "static/geojson/us-states.json")
    .await(makeGraphs);
```



```
function makeGraphs(error, projectsJson, statesJson) {
    ...
};
```

Initiating a New Computation

Use Ajax

- part of the javaScript jQuery library
- update a web page without reloading the page
- request data from a server - after the page has loaded
- receive data from a server - after the page has loaded
- send data to a server - in the background

Create a new route in the python server program

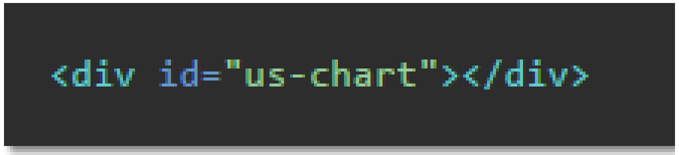
- serves an Ajax GET request
- possibly performs some computation on the data
- sends the results back as JSON in the usual way
- example in js "/newComp?category=2&city=Boston"
- couple with GET

Coupling D3 with Python

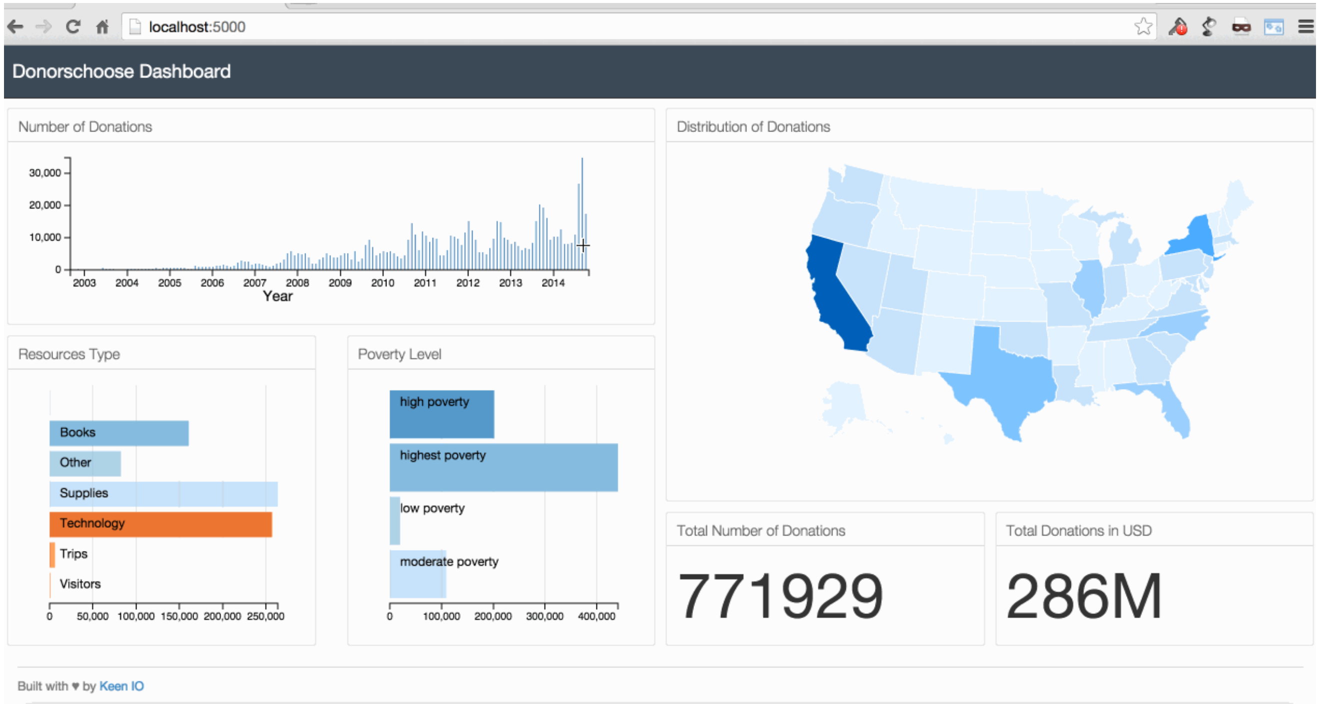
Start app.py web server

Add query to the index,html file

Call http://localhost:5000/ to see the dashboard



```
<div id="us-chart"></div>
```



localhost:5000

Donorschoose Dashboard

Number of Donations

Distribution of Donations

Resources Type

Poverty Level

Total Number of Donations

771929

Total Donations in USD

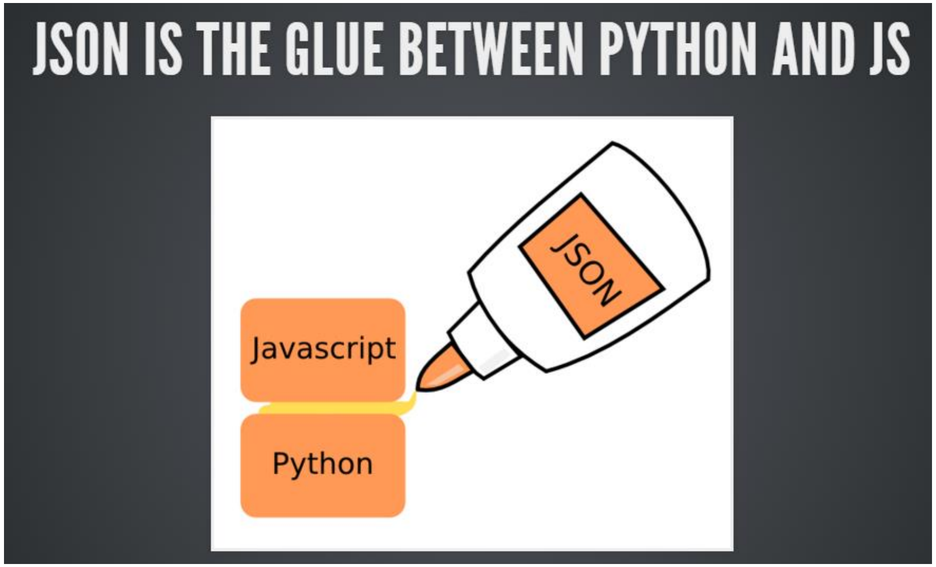
286M

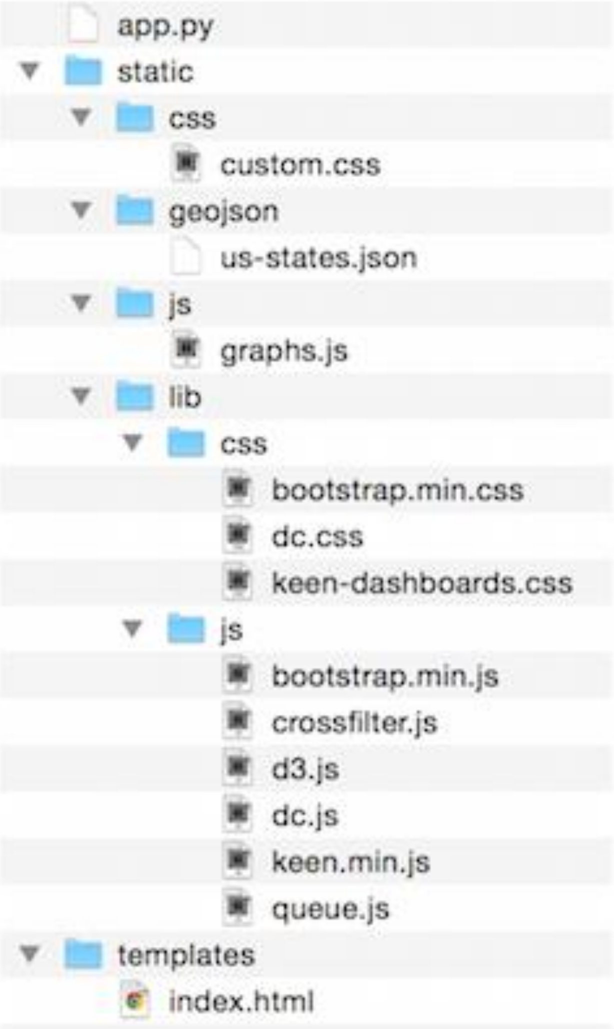
Built with ♥ by Keen IO

Folder Structure

All files are available in a dedicated github repository

One more thing:





```
    app.py
▼   static
  ▼   css
        custom.css
  ▼   geojson
        us-states.json
  ▼   js
        graphs.js
  ▼   lib
    ▼   css
          bootstrap.min.css
          dc.css
          keen-dashboards.css
    ▼   js
          bootstrap.min.js
          crossfilter.js
          d3.js
          dc.js
          keen.min.js
          queue.js
▼   templates
      index.html
```

Some Useful pages

http://adilmoujahid.com/posts/2015/01/interactive-data-visualization-d3-dc-python-mongodb/

- csv data gets stored in MongoDB (4th most popular database)

https://realpython.com/blog/python/web-development-with-flask-fetching-data-with-requests/

- data stays in a csv file

http://kyrandale.com/static/talks/pydata-to-the-web/index.html#/

There are other pages … use google

Coupling D3 with R

Solution 1:

- call R from within Python using the rpy2 package

Solution 2:

- use Shiny by RStudio
- uses two files: ui.R and server.R (both written in R)
- using d3 requires a third file, render.js (written in JavaScript)
- ui.R has the UI code
- server.R has the server code
- render.js receives rendering requests from ui.R
- without render.js:
  - server.R passes an image to ui.R (not interactive, nor scalable)
  - server.R uses an R graphing package, such as plotly or ggplot2

Coupling D3 with R

Shiny is a commercial package

- you will run your Shiny apps within the Shiny by RStudio IDE
- hides all the setup, easier than Python
- can focus on data processing and d3-visualization

Deploy your Shiny app to a URL

- required for the mini and final project
- you will need Shiny Server
- however, only a version with limited capabilities is free
- single user without authentication
- might be enough for mini and final project

What is the best solution – Python, R, or a combination?

- up to you

MINI PROJECT #2

Practice some of the initial steps of data analytics

Data sources of mini project #1 or find new, data should be large!

Use R or python for processing, D3 for visualization

- note: shiny only works on windows currently, python might be better
- deliver a read-me report and URL that allows selection of all options

Data decimation (20 points)

- compare random sampling with adaptive sampling

Dimension reduction (use decimated data) (10 points)

- find the intrinsic dimensionality of the data using PCA

Visualization (use dimension reduced data) (40 points)

- compare PCA, MDS (Euclidian, cosine, correlation), and isomap

due Thursday 3/10
start early!!!

Extra credit: text visualization (30 points)

- stem, remove stop words, perform tf-idf, then LSA, and visualize