Big Data - Foundations and Applications

Lesson #2 - Data Science Platforms

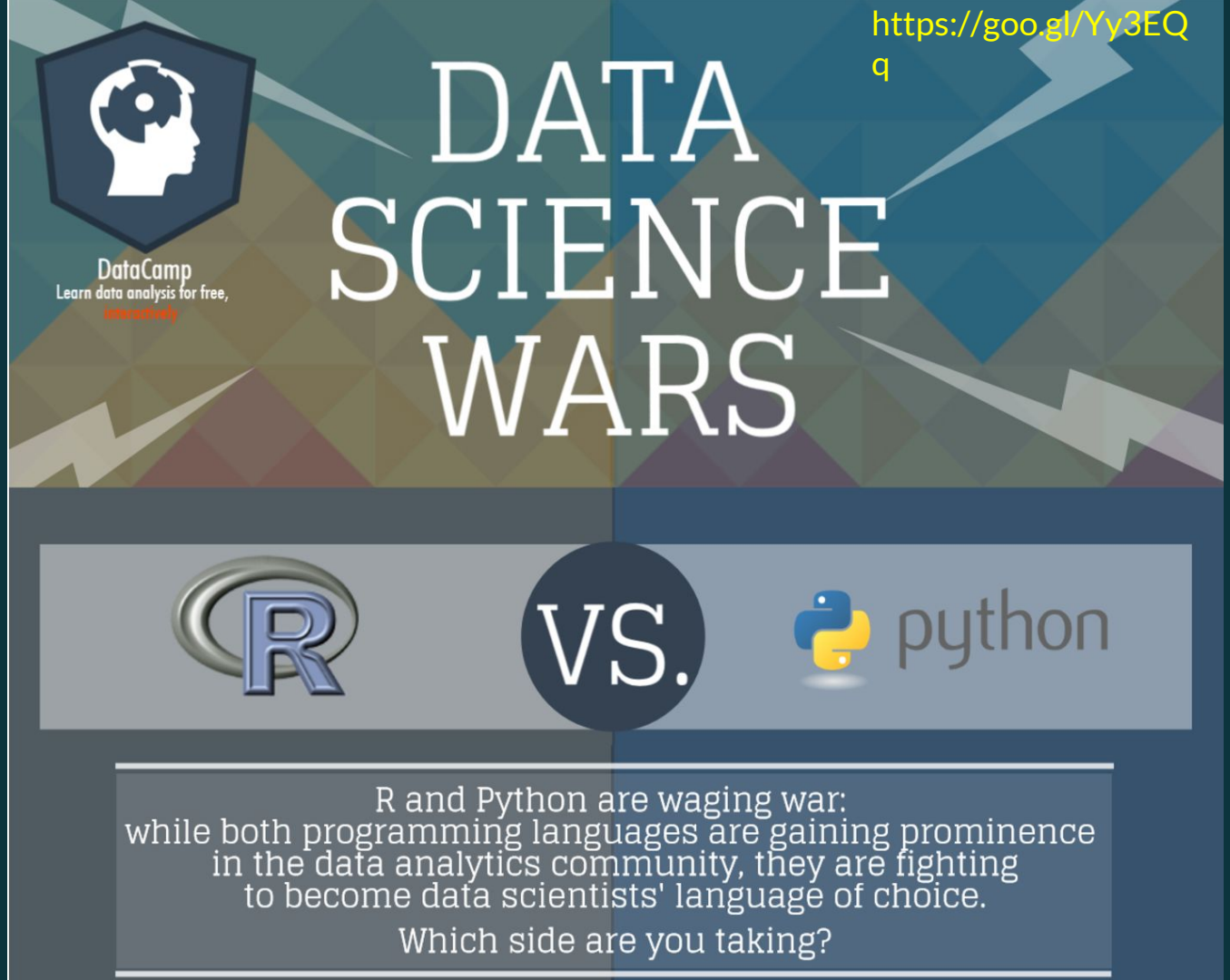
Ivanovitch Silva

July, 2017

Agenda

- Data Science War
- Anaconda
- My First Notebook
- Version Control System
- Intro to Python for Data Science

https://goo.gl/Yy3EQq



https://goo.gl/2mQm2K

If you come from a C.S./developer background, you'll probably feel more comfortable with Python. On the other hand, if you come from a statistics/analyst background, R will likely be more intuitive

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

IEEE Spectrum - Jul 2016 https://goo.gl/BqrkDI

Guido van Rossum

"I have this hope that there is a better way. Higher-level tools that actually let you see the structure of the software more clearly will be of tremendous value."

Version 3.x (https://www.python.org/downloads/)

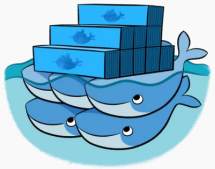
ANACONDA

Modern open source analytics platform
powered by Python

https://www.continuum.io/downloads

Why Anaconda?

- Anaconda is a distribution of packages built for data science.
- It comes with conda, a package and environment manager.
- You'll be using conda to create environments for isolating your projects that use different versions of Python and/or different packages.



ANACONDA NAVIGATOR

Upgrade Now

Sign in to Anaconda Cloud

Home

Environments

Projects (beta)

Learning

Community

Documentation

Developer Blog

Feedback

Applications on root

Channels

Refresh

glueviz

0.9.1

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

Launch

notebook

4.3.1

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

Launch

qtconsole

4.2.1

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

Launch

spyder

3.1.2

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

Launch

orange3

3.4.1

Install

rstudio

1.0.136

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

Install

An extremely brief tutorial

Introduction to Git

script.py

```
if __name__ == "__main__":
    print("Welcome to a script!")
```

you

```
import math
print(10 + 10)
if __name__ == "__main__":
    print("Welcome to a script!")
```

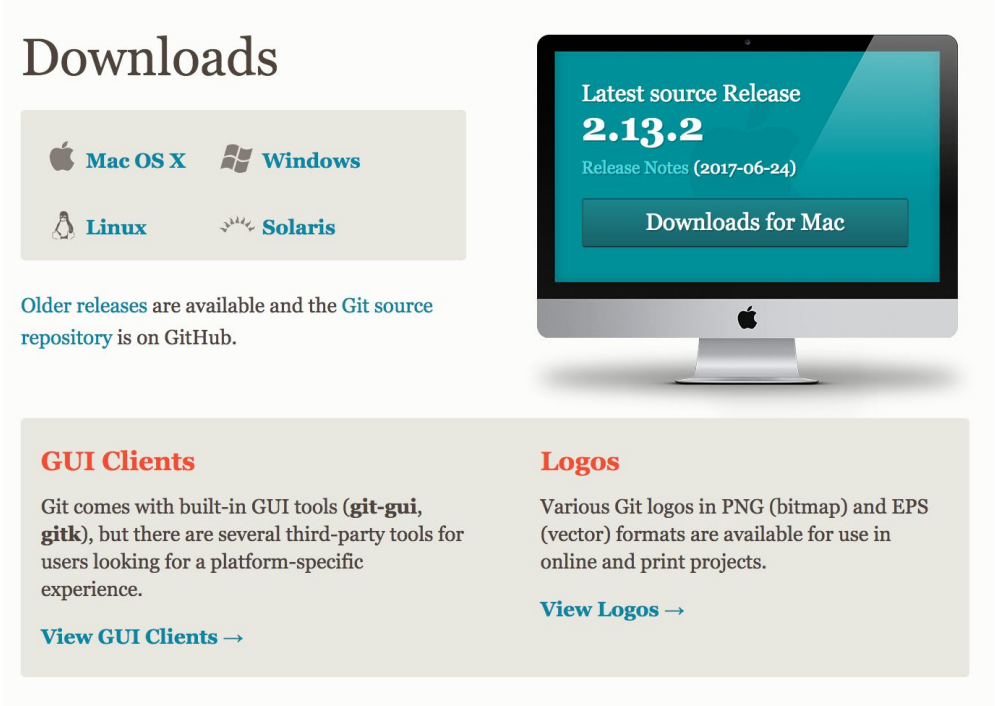
coworker

```
if __name__ == "__main__":
    print("Welcome to a script!")
    print("Here's my amazing contribution to this project!")
```

merge

```
import math
print(10 + 10)
if __name__ == "__main__":
    print("Welcome to a script!")
    print("Here's my amazing contribution to this project!")
```

Installing Git



https://git-scm.com/downloads

First step: create a repository (repo)

1. Create a folder named DataScience.
2. Navigate into this folder and initialize a Git repository (git init)
3. Run ls -la to check the contents of the DataScience folder

Creating files in the repo

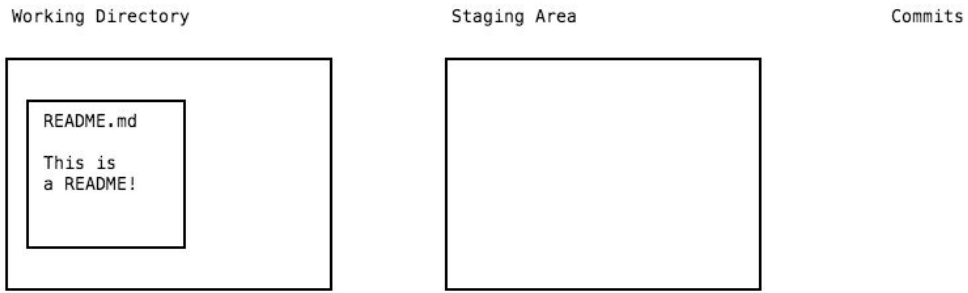
1. Create a file named README.md with the following content:

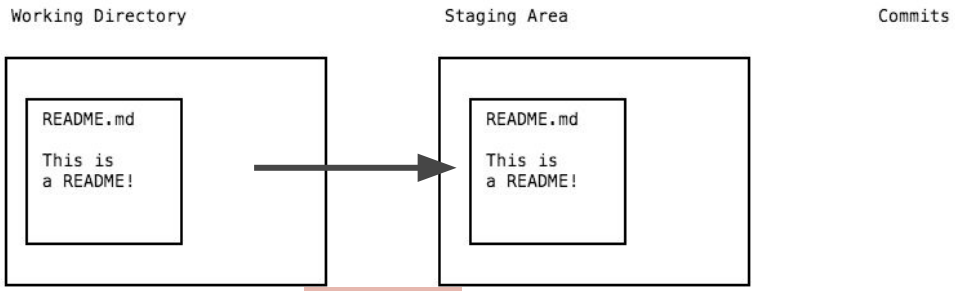
> **My first git project**

2. Create a file named script.py with this content:

```
if __name__ == "__main__":
    print("10")
```

Checking file status





git add

Verify the status of files: git status

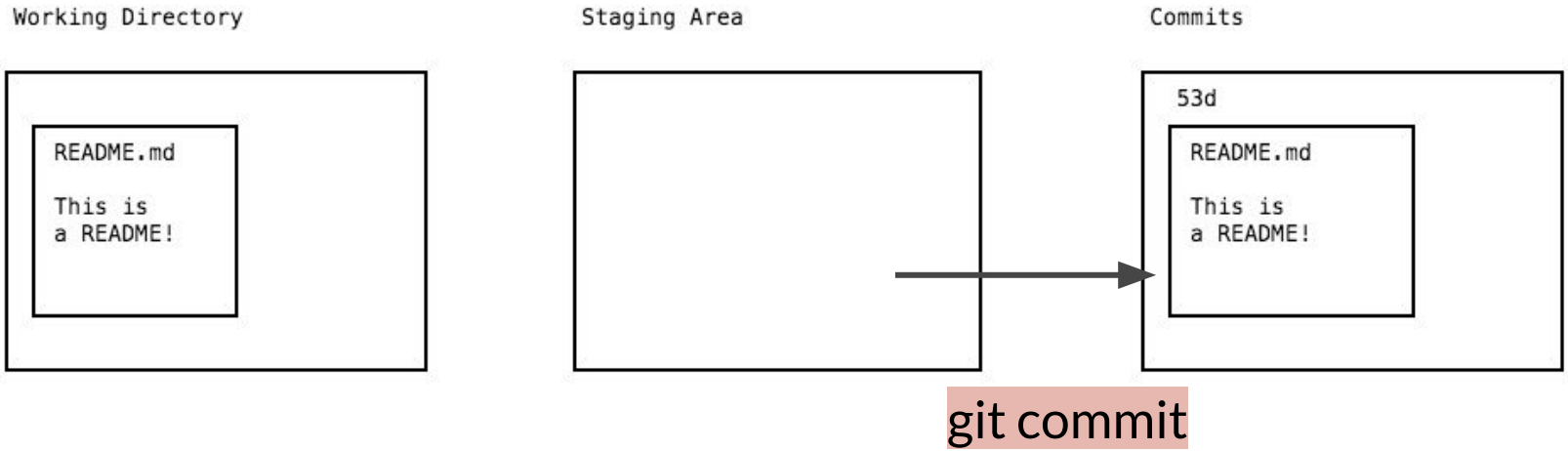
1. Check the status of the repo.
2. Add script.py to the staging area.
3. Add README.md to the staging area.

Configuring identity in Git

```
git config --global user.email "your.email@domain.com"
```

```
git config --global user.name "Your name"
```

Committing changes



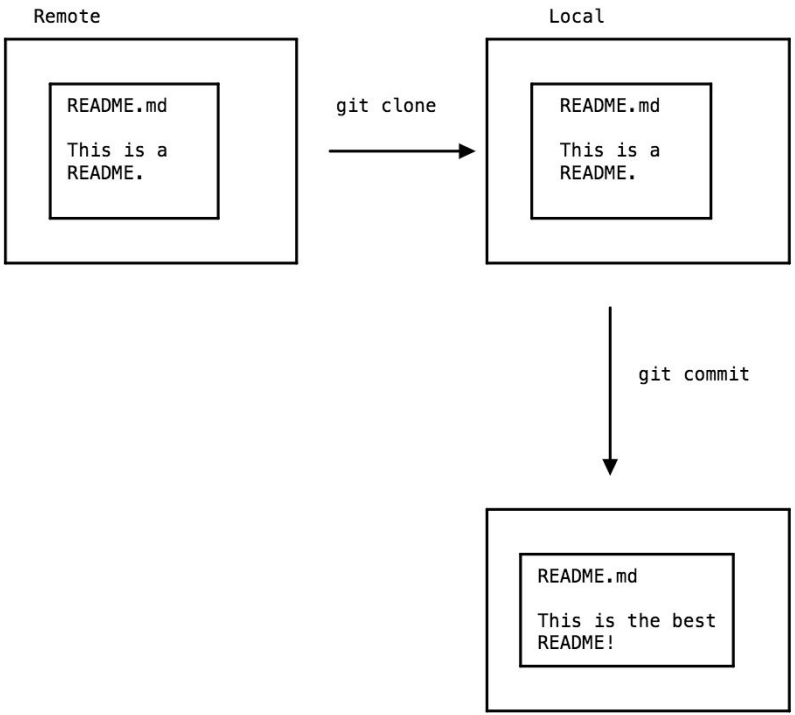
Type `git commit -m "Initial commit. Added script.py and README.md"` to make the first commit to the repository with an informative message.

Reviewing the commit history

Description:

1. Run git log to explore the commit history of the repository.

Remote repositories



- Share our code with others and build a portfolio
- Collaborate with others on a project and build code together.
- Download and use code others have created

Remote repositories

Here's how we'd typically clone the Amazon Deep Learning repo from GitHub:

- git clone https://github.com/amznlabs/amazon-dsstne.git

Remote repositories [exercise]

1. Clone the "fast style transfer" project from Github to your local repository.
2. https://github.com/lengstrom/fast-style-transfer
3. Show history from git lot
4. Clone the material of course

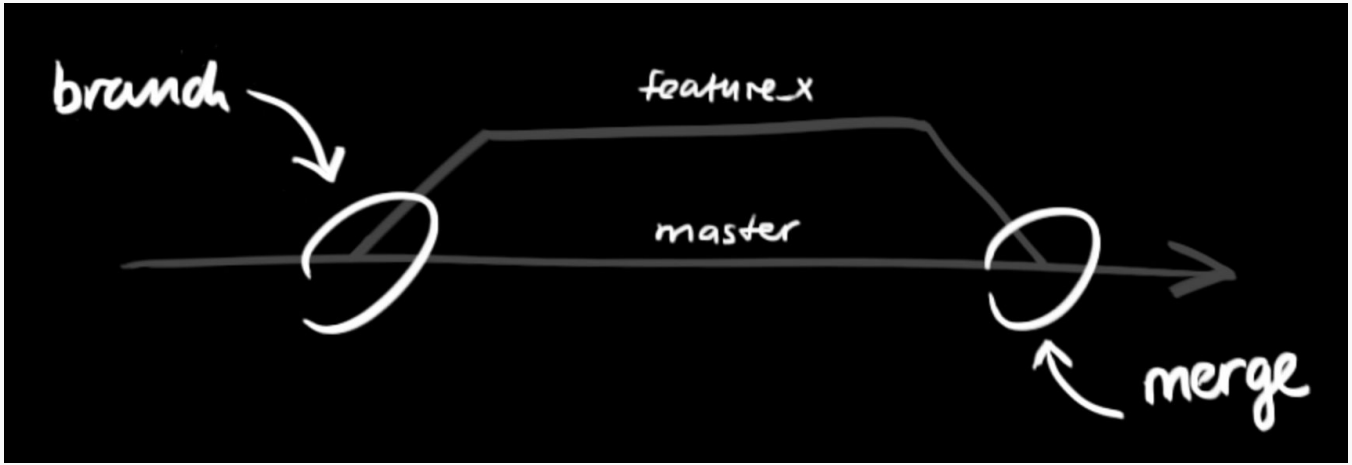
… go back to "DataScience" repo!!!

Github integration

Create a Github account

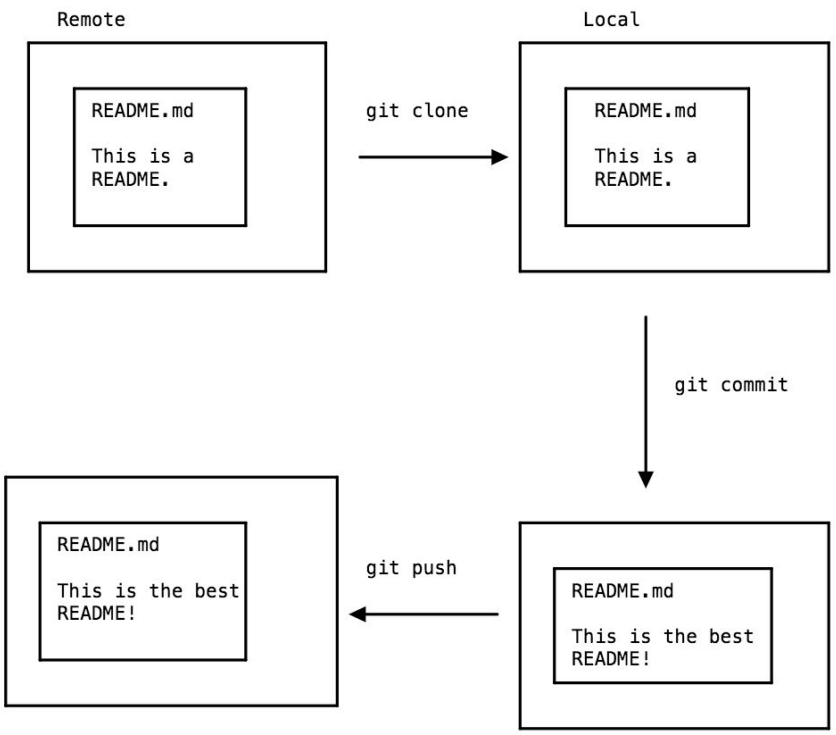
- Create a personal account. Select a unique username and password and enter your email.
- Choose a plan. If you select the free plan, all of your code (which is organized in repositories) will be public. Select the free plan for now. You can always upgrade to a paid plan later on, which would allow you to have private repositories.
- Read the GitHub Hello World guide.

Branch on repository



- Every Git repository consists of one or more branches.
- The main branch of a Git repo is typically called master.
- Use the git branch command to visualize the current branch of project

Pushing repo to Github



Pushing repo to Github [Exercise]

- Use the git remote command to visualize information about the repo.
- Create a repository in Github

```
git remote add origin https://github.com/<your_github_user>/hello-world.git
git push -u origin master
```

See the following notebooks for additional info

Git and a Version Control - Introduction to Git.ipynb
Git and a Version Control - Git Remotes.ipynb
Git and a Version Control - Git Branches.ipynb


MacBook Pro

Introduction to Python for Data Science

- Python versions
- Basic data types
- List
- Files and Loops
- If statements
- Dictionaries
- Functions and Packages

Notebook: "Intro Python for Data Science.ipynb"

References

- http://rogerdudler.github.io/git-guide/
- http://product.hubspot.com/blog/git-and-github-tutorial-for-beginners
- Dataquest.io
- Datacamp.com

Lesson #2