# Class 1 - Command Line and Git Basics

INFO W18 - Python for data science
Summer 2018

# **Week 1** | agenda

Welcome to INFO W18!

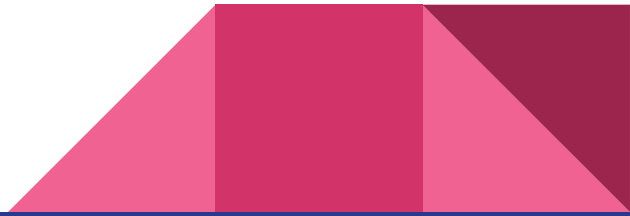
Who are you?

Coding languages

Using the command line

Version control with Git and Github

Homework preview



# Welcome to INFO W18!

The role of this course is to teach you the basic tools needed to do Data Science.

Python… but with a Data Science focus and twist!

# **Welcome** | logistics

Asynchronous, class meetings, and breakout sessions

Homeworks and assignments

https://github.com/MIDS-INFO-W18/assignments_upstream_summer18_SS

The Google group list

https://groups.google.com/forum/#!forum/w18-summer-session-2018

Using GitHub to get and submit your assignments

# **Welcome** | content

## 10 weeks

Programming flow and control structures

- Variables, conditional logic, looping, functions

Object oriented - modules, classes, OOP approach

Functional programming

Text processing

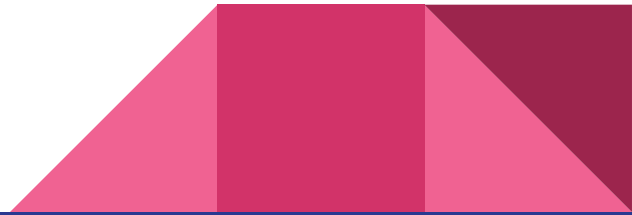
Numpy for vectorized operations

Data analytics in Jupyter (pyData)

# **Welcome** | first **5** weeks - programming

Unit 1 | Introduction, the Command Line, Source Control
Unit 2 | Starting Out with Python
Unit 3 | Sequence Types and Dictionaries
Unit 4 | More About Control and Algorithms
Unit 5 | Functions
Unit 6 | Modules and Packages
Unit 7 | Classes
Unit 8 | Object-Oriented Programming

Individual Object Oriented Project



# **Welcome** | last **5** weeks - data analysis

Unit 9 | Working With Text and Binary Data
Unit 10 | NumPy
Unit 11 | Data Analysis With Pandas
Unit 12 | More Analysis With Pandas
Unit 13 | Testing

Data Analysis Group Project

# **Welcome** | grading

Homework  (30%)

Project 1 object oriented programming, individual (20%)

Project 2 data analysis, group (20%)

Participation (10%)

Midterm (10%)

Final (10%)

# **Welcome** | schedule

https://docs.google.com/spreadsheets/d/1bzyTllTPzekNl5KpZVAVIs8Iub8Z12_hUS3XJ934WoM/edit#gid=0

# **Welcome** | the ISVC

Synchronous Session - Camera and Python access required

Module types - Live Session / Office Hours

Private and public chatting encouraged!

Breakout Rooms - Share screen

Using Github to get and submit your assignments

**How has it been using the ISVC?**

# **Week 1** | agenda

Welcome to INFO W18!

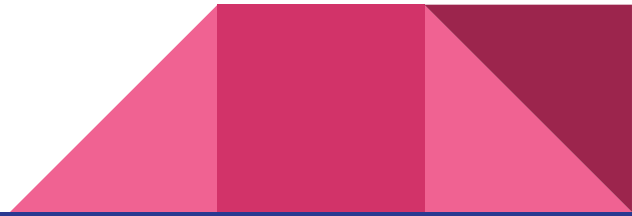
Who are you?

Coding languages

Using the command line

Version control with Git and Github

Homework preview



# **Introductions** | who are you?

1-2 minute intro

Where are you from?

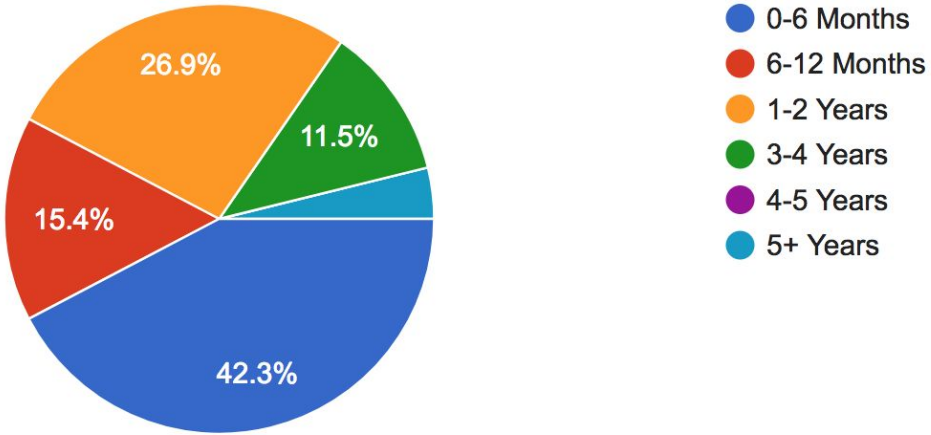
Are you a student? What do/did you study?

What are you excited for in learning Python?

# **Introductions** | programming experience

How much programming experience do you have?

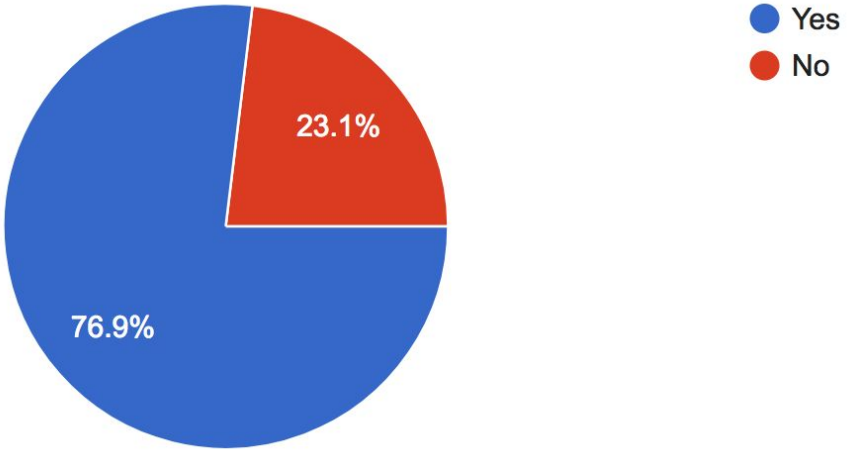
26 responses



# **Introductions** | technical experience

Have you ever used the command line/terminal on a computer?

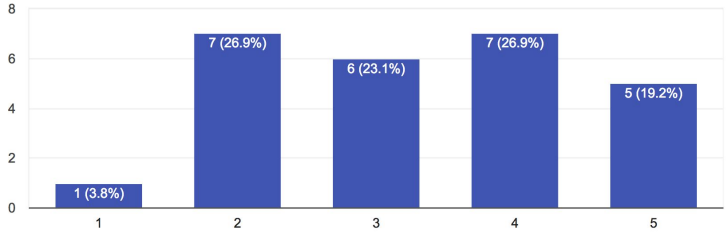
26 responses



# **Introductions** | misc

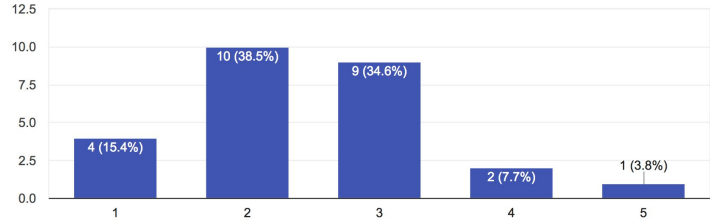
How would you rate your experience with excel?

26 responses

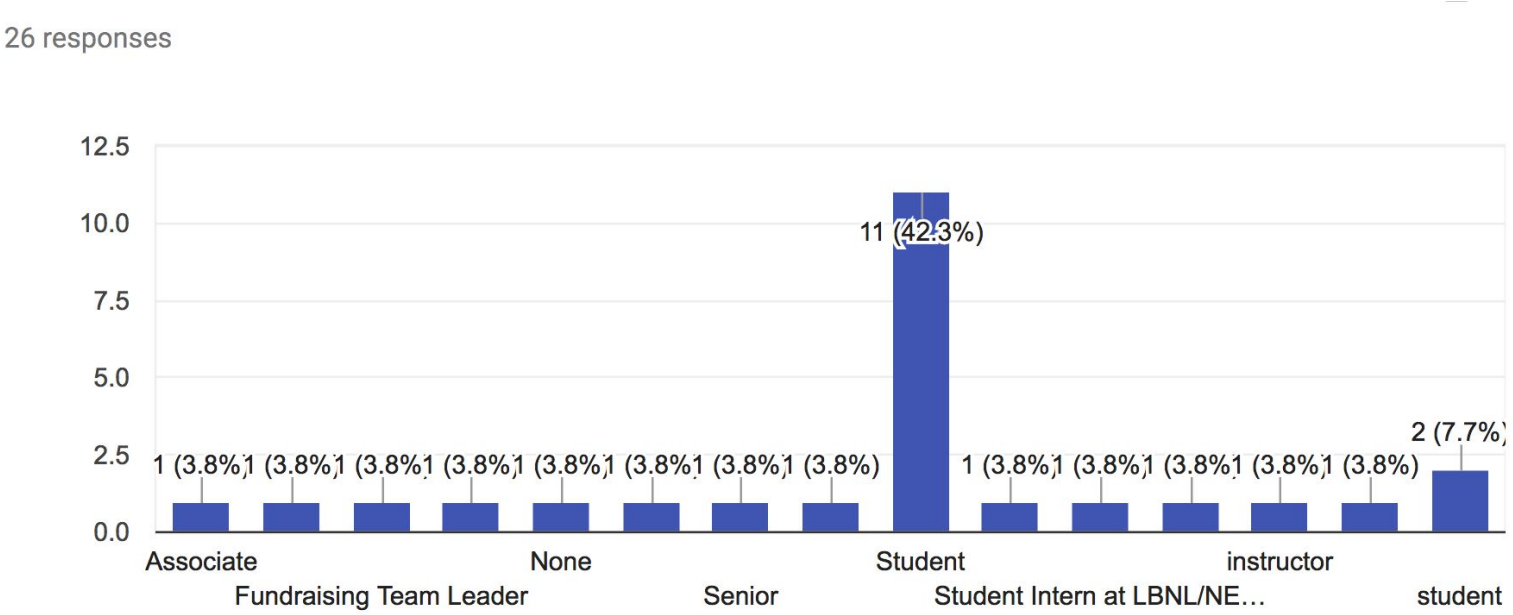



How would you rate your experience with programming?

26 responses



# **Introductions** | misc



# **Introductions** | survey and access

FIll out intake survey

Obtain access to Google Group

Have access to a terminal prompt

Set up GitHub/homework REPO and playground

Push first activity to repo

Able to open Jupyter notebook with Python 3 Kernel

Can share screen on ISVC

# **Week 1** | agenda

Welcome to INFO W18!

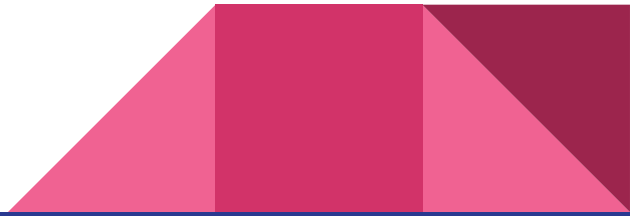
Who are you?

Coding languages

Using the command line

Version control with Git and Github

Homework preview



# **Coding Languages** | compared to alternatives

Python is a high level interpreted language

- C/C++
- Java
- C#
- Scala
- Perl

# **Coding Languages** | compared to alternatives

Python can be used for all sorts of data analysis

- R
- SAS
- Stata
- Excel/VBA

# **Week 1** | agenda

Welcome to INFO W18

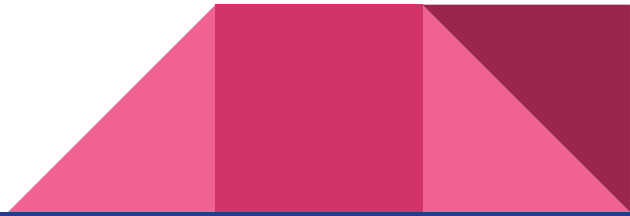
Who are you?

Coding languages

Using the command line

Version control with Git and Github

Homework preview



# **The Command Line** | overview

“Shell”, “Terminal”, “Command Prompt”, “Git Bash”

In this class, Mac and Unix users will use “Terminal”

Windows users will use “Git Shell”

# **The Command Line** | the basics

Demo using the following commands:

- cd - change directory
- pwd - return working directory
- man - help on commands
- mkdir - create new directory
- echo - write to terminal screen
- mv - rename and/or move file or directory
- cp - copy file and/or directory
- rm - delete file and/or directory
- ls - list the files in a directory

# **The Command Line** | drills

Please answer the following questions in the polls:

What is the command to switch into the directory ~/Desktop?

What is the command to create the directory ~/Desktop?

What is the command to display the contents of ~/Desktop?

# **The Command Line** | bash script

Demo changing commands into bash script

# **Week 1** | agenda

Welcome to MIDS! (and ISVC)

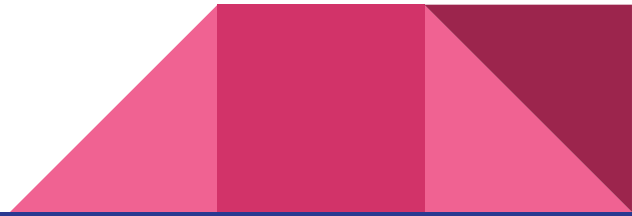
Who are you?

Coding languages

Using the command line

Version control with Git and Github

Homework preview



# **Git and GitHub** | mini-agenda

Git vs. GitHub

Git

GitHub

Pushing to GitHub

Merge Conflicts

# **Git and GitHub** | big picture

What is the difference between Git and GitHub?

# **Git and GitHub** | big picture

What is the difference between Git and GitHub?

Git is *local* version control. You can use Git by yourself.

GitHub is an *online way* to sync Git version control across machines. Certain “git” commands communicate with GitHub.

Some companies use their own internal GitHub like software.

# **Git and GitHub** | distributed version control

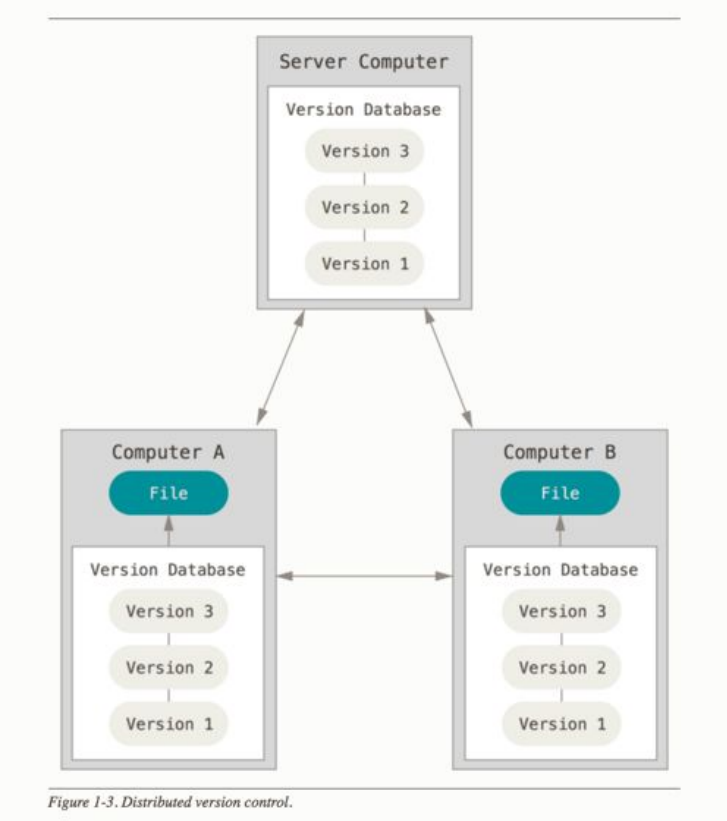



*Figure 1-3. Distributed version control.*

# **Git and GitHub** | local version control

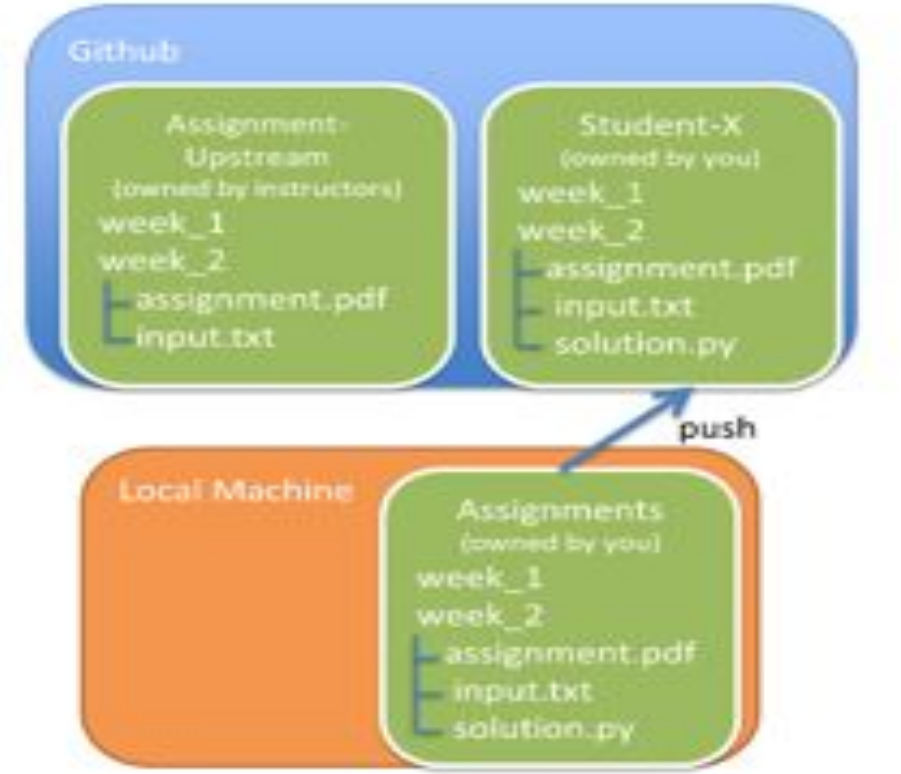
Demo creating a local git repository

# **Git and GitHub** | distributed version control

Walkthrough setting up github playground

https://github.com/MIDS-INFO-W18/github-playground

# **Git and GitHub** | GitHub vs local machine



# **Git and GitHub** | pushing to GitHub

Three steps to fully commit and store changes online:

0. Always “**git pull**” before you make changes !

1. “git add”
2. “git commit -m”
3. “git push”

Use “git status” for help throughout!

# **Git and GitHub** | breakout rooms

Set up homework repo in class:

https://github.com/MIDS-INFO-W18/Installation/blob/master/Completing%20Your%20First%20Exercise%20With%20Git%20Reading%20Card.md

# **Git and GitHub** | merge conflicts

What if multiple people edit the same file?

Sometimes, there is no conflict.

> *You edit lines 1-30.*
>
> *Your friend edits lines 70-100.*

**But what if you both edit the same lines?**

Merge conflicts must be resolved manually.

# **Git and GitHub** | drills

Please answer the following questions on the polls:

Which command tells you which files have changed?

Which command sends changes to GitHub?

Which command creates a new local repository?

# **Week 1** | agenda

Welcome to INFO W18!

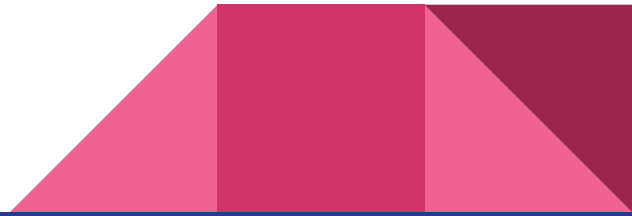
Who are you?

Coding languages

Using the command line

Version control with Git and Github

Homework preview



# **Homework Preview** | overview

Create folders and files via the command line.

Connect to a new GitHub repository (github playground) and edit a file.

# **Homework Preview** | tricky spots

You will be asked to create a “bash script”

Other people will be editing the same file - which will lead to “merge conflicts”

# **Homework Preview** | bash script

A bash script is just a small program that can be run from the command line. It is made up of command line commands (same as in terminal).

On a Mac “.sh”, On Windows “.bat”

You will get to play with and figure out the details from here.

# **Homework Preview** | merge conflicts

We want you to work on a file that others are working on as well.

You will likely to run into merge conflicts especially if you wait until the last minute

There is information in the lecture and installation files which show you how to resolve ***some*** conflicts.