

# What is GitHub anyway? Use GitHub for Reproducible Research

Vivek Katial

11/11/2019

- 1 Introduction
- 2 Getting Started
- 3 Basics
- 4 Branches
- 5 Final Notes and Best Practices

# Section 1

## Introduction

# About me!

- Vivek Katial (PhD Candidate + Data Scientist for the last 3 years)
- I love travel and food
- (Image of me jumping over the pyramids)

# Agenda

- Getting Started
- Basics
- Branches
- Best Practice

## Section 2

### Getting Started

# Install git and create GitHub Account

- Linux (Debian)
  - Command: `sudo apt-get install git`
- Linux (Fedora)
  - Command: `sudo yum install git`
- Mac OSX
  - <https://git-scm.com/download/mac>
- Windows
  - <https://git-scm.com/download/windows>

# What is version control? Why is it important?

- A system designed to keep track of all your changes
- Designed for collaborative development
- Enables you to know **who** made **what** changes **when**
- Enables you to **revert any changes and go back to a previous state**



# Other Version Control Systems

- Subversion (SVN)
- Mercurial
- Dropbox
- Google Drive

# Why Git?

- Git is **local**:
  - Except pulling and pushing to remote servers
- Git is **fast**
- Git versions the *project* not files
- In Git each version is deliberately tied to a meaningful change

# Short History of Github

# Installing Github

## Section 3

### Basics

# Quick Summary of UNIX/Terminal Commands

- `ls`
- `mkdir`
- `touch`
- `cd`
- `rm`
- `mv`
- `cp`

# Creating your own repository!

- First repository
- (Online Demo)

# Basic summary of git commands

- `git status`
- `git clone`
- `git log`
- `git pull`
- `git commit`
- `git add`
- `git push`
- Cheat Sheet attached



# Recording Changes

- Demo

# Viewing your commit history

- `git log`

# Working with Remotes

- Let's write a travel blog!

# Git Basics Exercise

- Basic exercise

## Section 4

# Branches

# Why Branching?

# Working/Collab with other people

# Branching/Merging



# Pull Requests

# Resolving Merge Conflicts

# Git Branching Exercise

## Section 5

### Final Notes and Best Practices

# Using a .gitignore

- Passwords
- Parameter files

# Commit Messages

- Commit History

# README

- Nice README File

# Branches



# Group Exercise - All of Github