

# Introduction to Data Science

## Data Science Essentials

---





# Slack

## Channels

- use for teamwork and class communication
- public (anyone can see)


## Tags

- notify a person by typing the person's user id (starting with @)
- notify an entire channel by typing **@channel** notify only the people who are online in a particular channel with **@here**
- 
- **Direct Messages**
- communicate privately (no @ needed)

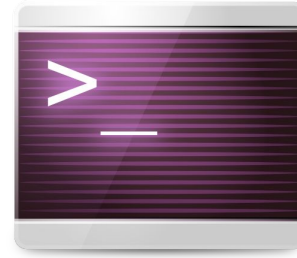
The screenshot shows the Slack interface for a workspace named 'Nashville Softw...'. The left sidebar lists channels: # alumni, # data-science-01, # data-science-02 (selected), # general, # instructors, # instructors-data-01, # instructors-data-02, # nss-staff, # random, and # til. Below channels are direct messages with slackbot, Mary van Valkenburg, Alex, Dereje, Jeremiah Vasquez, Kristin McKinney, Mandy Arola, Jarrod Jo..., Massih Forootan, and Mittal Mukul K.

The main channel view for #data-science-02 shows messages from Tuesday, August 7th to Friday, August 10th. Messages include:

- A post about the Analytics Summit on 8/20 and 8/21, mentioning a student code **STUDENT** for a highly discounted rate.
- John Wark joined #data-science-02.
- Mary van Valkenburg mentioned @Davis Thraikill & @Michael Holloway.
- Davis Thraikill joined #data-science-02 by invitation from Mary van Valkenburg, along with Michael Holloway.
- Mary van Valkenburg mentioned @UrLeaka Newsome.
- UrLeaka Newsome joined #data-science-02 by invitation from Mary van Valkenburg.
- Mary van Valkenburg mentioned @Alicia.
- Alicia Ortiz joined #data-science-02 by invitation from Mary van Valkenburg.
- Taylor Perkins mentioned DS2!! and said 'Good morning DS2!! I am looking forward to finally meeting you all tonight!'.

- It's polite to acknowledge messages with **:+1:** 
- Save important shared items (free slack has limitations)

# Command Line



`pwd`

`ls`

`cd`

`cd ..`

`mkdir`

`rmdir`

`mv`

`touch`

`cat`

`conda list`

`conda install <package name>`

**Linux/Mac:** Terminal

**Windows:** Git Bash (for using Git); Anaconda Prompt (for working with conda)

## Resources:

<https://www.learnenough.com/command-line-tutorial>

<https://learnpythonthehardway.org/python3/appendixa.html>

<https://www.datacamp.com/courses/introduction-to-shell-for-data-science>



# git and GitHub



You may already have git installed: from the terminal (or Anaconda Prompt on Windows), try typing

```
$ git --version
```

If this doesn't work, browse to <https://git-scm.com/downloads> and download the appropriate version. If on Windows, download from <https://gitforwindows.org/>, which will also download Git Bash.

# Today's tasks:

- Create an account on GitHub (unless you already have one)
- Fork the repo on GitHub to put a copy in your account
- Clone your fork to your laptop (local repository)
- Launch the Jupyter notebook from your local repository
- Get a tour of Jupyter
- Walkthrough the python review notebook
- Push changes back to GitHub

# Command Line Introduction

**pwd**

“print working directory” - tells you where you are located

**ls**

lists the contents of the current directory

**cd <directory name>**

changes the directory to the one specified

**cd ..**

Takes you up one level in the directory structure

Use these commands to navigate to the directory where you want to create the repo for this class.

# Repository Setup

First, fork the repository here:

<https://github.com/Vanderbilt-Aspire-Data-Science/data-science-essentials-4>

Then, ensure that you are on the forked version on your personal GitHub

Clone or download ▼

Finally, clone your remote repository to create a local repo by copying the clone url from github.com and then running from the terminal:

**git clone <url to repository that you copied>**

# Create a PAT

Before pushing changes, we must create a personal access token on GitHub.

1. Open *settings* by clicking on your profile picture in the upper right corner.
2. In the menu on the left, choose *Developer Settings*, then choose *Personal access tokens*
3. Choose *Generate new token*, give it a name, and ensure that you have selected the **repo** scope
4. Take the resulting token and save it somewhere (or email to yourself).



# Pushing work to GitHub

1. Work on files locally. When done, run

```
git add <files>
```

2. Commit changes:

```
git commit -m '<message>'
```

3. Push to GitHub:

```
git push origin main
```

