

Intermediate Data Science

Welcome Back!

Joanna Bieri DATA201

Welcome to Intermediate Data Science!

Important Information

- Email: joanna_bieri@redlands.edu
- Office Hours take place in Duke 209 – Office Hours Schedule
- Class Website
- Syllabus

Computer Set Up

Before we can do much data science we need to make sure everyone has a stable computer set up for this semester!

We will be using Python and Jupyter Lab as our foundation for programming. You should have these already installed. They are part of The Anaconda software package.

If you need to set up your computer here is a link to the setup I use in Data 101:

How to set up your computer!.

Software

Let's start by checking what versions of the software we have:

- Python 3.13.5
- conda 24.11.3

```
# If you need to update:  
!conda update -n base conda -y  
!conda update --all -y
```

Python 3.13.5

conda 24.11.3

Version Control

We also need to make sure we have git and jupyterlab-git installed.

First make sure you have a GitHub Account if not Sign Up For GitHub Account!. Log into your GitHub. Then check that you have git installed on your computer.

- git version 2.34.1

git version 2.51.0

Version Control

We will probably use jupyterlab-git to clone/push/pull from GitHub. If you see this icon on the left side of your jupyter, then you are good to go!



Figure 1: Git Icon

Version Control

If you don't see it you might need to run the following code and restart your Jupyter Lab.

```
!conda config --add channels conda-forge
!conda install -c conda-forge -y git
!conda install -c conda-forge -y jupyterlab-git

!conda update -y pexpect
```


Set up Workflow

1. Create a Directory for your work in this class

Using the Jupyter File Browser, make a folder (directory) that will contain all of your work for this class. Make sure to name it something good so that it is easily findable!

Set up Workflow

2. Clone the Repo for our Class

Redlands-DATA201/FALL2025

Set up Workflow

In a terminal window:

```
git clone https://github.com/Redlands-DATA201/FALL25
```

In jupyter-git:

```
click on the jupyter-git icon  
select clone repo  
enter the address
```

Set up Workflow

Just clone it directly, no need to fork first.

You should see a directory created called **FALL25**. Inside this folder is are a few files and a folder named **notebooks***.

This semester we are going to learn a bit more about Git! We are going to work on our daily course materials collaboratives in individual branches.

Set up Workflow

3. Create a personal Repo for your work.

- You will want a Git repo just for your work. This is where you will store a backup of all the code you write in class. Later in the class we will learn how to collaborate using Git.
 - Make sure you are at your GitHub home screen by clicking on the cat in the upper left corner.

Set up Workflow

- Now create your own repo by clicking on the “New” button on the top left side of the screen

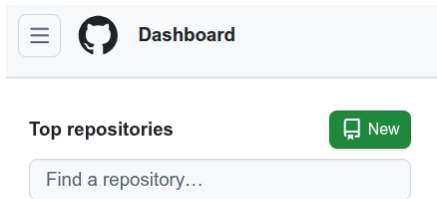


Figure 2: GitHub Home Repo

Set up Workflow

- This will send you to a page that you can fill out to create your repo.

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk ().*

1 General

Owner * **Repository name ***

 BlasTheory /

 Data101_Joanna_Blas_Code is available.

Great repository names are short and memorable. How about [ddadco-ecto-fahsick](#)?

Description

27 / 390 characters

2 Configuration

Choose visibility *

Choose who can see and commit to this repository

 Private

Add README

README's can be used as longer descriptions. [About README's](#)

On ☒

Add .gitignore

.gitignore tells git which files not to track. [About .gitignore files](#)

 Python

Add license

Licenses explain how others can use your code. [About licenses](#)

No license

[Create repository](#)

Figure 3: GitHub Create Repo

Set up Workflow

- Next you can **Clone** this repo to your JupyterLab - just like we did above with the class Repo.

Set up Workflow

4. Switch to your Working Directory and Add a README.md to your working directory.

Click on the New Launcher and choose either Markdown File or Text File. This will open a screen where you can type things in. Type something in to this file “This is my working directory!” and then save the file.

Set up Workflow

5. Commit your change and push it to Git

NOTE: You might have to redo your personal access token!

In a terminal window:

```
git config --system --unset credential.helper
```

Go to Git and generate a new personal access token. You can follow the instructions here: [How to set up your computer!](#).

Set up Workflow

Then save your changes in Jupyter

In a terminal window:

```
git status  
git add <type in your file name>  
git commit -m '<type in your commit message>'  
git branch  
git push
```

In jupyter-git

triple check that you are in your repo - you will not be able
add the untracked file using the + sign
make sure the new file has been staged
write a commit message and save it
push on the up arrow cloud to push.

Set up Workflow

When prompted copy and paste your personal access token into jupyterlab-git.

Now save your new personal access token.

In a terminal window:

```
git config --global credential.helper store
```

Daily Workflow

Each day that you are ready to work on your class materials you will do the following things:

- 1 Navigate using the File Browser to the class repo: FALL25
- 2 Pull any new changes
- 3 Copy the files you need for the daily assignment into your working directory.
- 4 Do the assignment
- 5 Push the files to your git repo (Add the file, Stage the file, Commit the change, Push the change)

Prerequisites

This class assumes that you have had a course in Introductory Data Science and Introductory Programming (Python is really helpful). This means that you are familiar with things like:

- Importing modules

```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.io as pio
pio.renderers.default = 'colab'
```

Reading data into a computer

```
DF = pd.read_csv()
```

Interacting with data in a data frame

`DF.columns`

`DF.shape`

`DF.dtypes`

`DF.describe()`

Masking data frames to focus on important columns

```
focal_column = "names"  
mask = DF[focal_column]=='Joanna_Bieri'  
DF_joanna = DF[mask].copy()
```

Finding value counts

```
DF[focal_column].value_counts()
```

Grouping and sorting data.

```
columnns = ['names', 'money']  
DF[columnns].groupby('names').sum().sort_values('money', as
```

Using operations or Applying Lambdas

```
DF['money'].mean()
```

```
DF['money'].sum()
```

```
DF['first_names'] = DF['names'].apply(lambda x: str(x).spl
```

Boolean statemets and FOR loops

```
for n in DF['first_names']:
    if n == 'Joanna':
        print(f'{n} is the best!')
    else:
        print('Still waiting for Joanna!')
```

Writing functions

```
def add_two_numbers(a,b):  
    print(f'adding {a} and {b}')
```

return a+b

Getting help:

I have lots of videos and assignments that can help you to review this material!

Data 101 - Videos Playlist

Data 101 - Archive Website 2024

Intro to Python - Videos Playlist

Intro to Python - Website