# About this Book

### Contents

#### Syllabus

- About
- · Tools and Resources
- Data Science Achievements
- Grading
- Grading Policies
- Support
- General URI Policies
- Course Communications

#### Notes

- 1. Welcome and Introduction
- 2. Syllabus and Python Review
- 3. Grading review, Pandas, and Iterables
- · 4. Pandas and Indexing
- 5. Exploratory Data Analysis (EDA)

#### Assignments

- 1. Assignment 1: Portfolio Setup, Data Science, and Python
- 2. Assignment 2: Practicing Python and Accessing Data
- 3. Assignment 3: Exploratory Data Analysis

#### Portfolio

- Portfolio
- Formatting Tips

#### FAQ

- FAQ
- · Syllabus and Grading FAQ
- Git and GitHub
- Code Errors

#### Resources

- Glossary
- References on Python
- Cheatsheet
- Data Sources
- General Tips and Resources
- How to Study in this class
- Getting Help with Programming
- Terminals and Environments
- Getting Organized for class
- Advice from FA2020 Students
- Advice from FA2021 Students
- Letters to Future students

Welcome to the course manual for CSC310 at URI with Professor Brown.

This class meets TTh 5-6:15pm in Tyler 108.

This website will contain the syllabus, class notes, and other reference material for the class.

### Course Calendar on BrightSpace



 $\underline{\text{subscribe to that calendar}} \text{ in your favorite calendar application}$ 

### Navigating the Sections

The Syllabus section has logistical operations for the course broken down into sections. You can also read straight through by starting in the first one and navigating to the next section using the arrow navigation at the end of the page.

This site is a resource for the course. We do not follow a text book for this course, but all notes from class are posted in the notes section, accessible on the left hand side menu, visible on large screens and in the menu on mobile.

The resources section has links and short posts that provide more context and explanation. Content in this section is for the most part not strictly the material that you'll be graded on, but it is often material that will help you understand and grow as a programmer and data scientist.

## Reading each page

All class notes can be downloaded in multiple formats, including as a notebook. Some pages of the syllabus and resources are also notebooks, if you want to see behind the curtain of how I manage the course information.



Notes will have exercises marked like this

#### Question from Class

Questions that are asked in class, but unanswered at that time will be answered in the notes and marked with a box like this. Long answers will be in the main notes

#### Further reading

Notes that are mostly links to background and context will be highlighted like this. These are optional, but will mostly help you understand code excerpts they relate to.

#### 9 Hint

Both notes and assignment pages will have hints from time to time. Pay attention to these on the notes, they'll typically relate to things that will appear in the assignment.

#### Think Ahead

Think ahead boxes will guide you to start thinking about what can go into your portfolio to build on the material at hand.

#### & Click here!

Special tips will be formatted like this

### Check your Comprehension

Questions to use to check your comprehension will looklike this

### About

#### About the topic

Data science exists at the intersection of computer science, statistics, and domain expertise. That means writing programs to access and manipulate data so that it becomes available for analysis using statistical and machine learning techniques is at the core of data science. Data scientists use their data and analytical ability to find and interpret rich data sources; manage large amounts of data despite hardware, software, and bandwidth constraints; merge data sources; ensure consistency of datasets; create visualizations to aid in understanding data; build mathematical models using the data; and present and communicate the data insights/findings.

### About the goals and preparation

This course provides a survey of data science. Topics include data driven programming in Python; data sets, file formats and meta-data; descriptive statistics, data visualization, and foundations of predictive data modeling and machine learning; accessing web data and databases; distributed data management. You will work on weekly programming problems such as accessing data in database and visualize it or build machine learning models of a given data set.

Basic programming skills (CSC201 or CSC211) are a prerequisite to this course. This course is a prerequisite course to machine learning, where you learn how machine learning algorithms work. In this course, we will start with a very fast review of basic programming ideas, since you've already done that before. We will learn how to use machine learning algorithms to do data science, but not how to build machine learning algorithms, we'll use packages that implement the algorithms for us.

#### About the course

This course is designed to make you a better programmer while learning data science. You may be stronger in one of those areas than the other at the beginning, but you should grow in both areas either way by the end of the semester.

### About this syllabus

This syllabus is a *living* document and accessible from BrightSpace, as a pdf for download directly online at <a href="mailto:rhotographics.org/rhub.io/BrownFall20/syllabus">rhotographics.org/rhotographics.org

Creating an issue on the repository is also a good way to ask questions about anything in the course it will prompt additions and expand the FAQ section.

### About your instructor

Name: Dr. Sarah Brown Office hours: TBA via zoom, link in BrightSpace

Dr. Brown is an Assistant Professor of Computer Science, who does research on how social context changes machine learning. Dr. Brown earned a PhD in Electrical Engineering from Northeastern University, completed a postdoctoral fellowship at University of California Berkeley, and worked as a postdoctoral research associate at Brown University before joining URI. At Brown University, Dr. Brown taught the Data and Society course for the Master's in Data Science Program.

#### Important

For assignment or notes specific issues, a comment on the corresponding repository is the best. I cannot help you with code issues from screenshots.

#### Question from class

Questions that are asked in class, but unanswered at that time will be answered in the notes and marked with a box like this. Short questions will be in the margin note

Note

Whether you use CSC or DSP does not matter.

The best way to contact me for general questions is e-mail or by dropping into my office hours. Please include [csc310] or [csc310] in the subject line of your email along with the topic of your messages. This is important, because your messages are important, but I also get a lot of e-mail. Consider these a cheat code to my inbox: I have setup a filter that will flag your e-mail if you use one of those in the subject to ensure that I see it. I rarely check e-mail between 6pm and 9am, on weekends or holidays. You might see me post or send things during these hours, but I will not reliably see emails that arrive during those hours.

### **Tools and Resources**

We will use a variety of tools to conduct class and to facilitate your programming. You will need a computer with Linux, MacOS, or Windows. It is unlikely that a tablet will be able to do all of the things required in this course. A Chromebook may work, especially with developer tools turned on. Ask Dr. Brown if you need help getting access to an adequate computer.

All of the tools and resources below are either:

- · paid for by URI OR
- · freely available online.

### **BrightSpace**

This will be the central location from which you can access all other materials. Any links that are for private discussion among those enrolled in the course will be available only from our course Brightspace site.

#### Prismia chat

Our class link for Prismia chat is available on Brightspace. We will use this for chatting and in-class understanding checks.

On Prismia, all students see the instructor's messages, but only the Instructor and TA see student responses.

#### Course website

The course manual will have content including the class policies, scheduling, class notes, assignment information, and additional resources. This will be linked from Brightspace and available publicly online at <a href="mailto:rhodyprog4ds.github.io/BrownSpring23/">rhodyprog4ds.github.io/BrownSpring23/</a>. Links to the course reference text and code documentation will also be included here in the assignments and class notes.

#### GitHub

You will need a <u>GitHub</u> Account. If you do not already have one, please <u>create one</u> by the first day of class. If you have one, but have not used it recently, you may need to update your password and login credentials as the <u>Authentication rules</u> changed over the summer. In order to use the command line with https, you will need to <u>create a Personal Access Token</u> for each device you use. In order to use the command line with SSH, set up your public key.

### **Programming Environment**

This a programming course, so you will need a programming environment. In order to complete assignments you need the items listed in the requirements list. The easiest way to meet these requirements is to follow the recommendations below. I will provide instruction assuming that you have followed the recommendations.

### Requirements:

- Python with scientific computing packages (numpy, scipy, jupyter, pandas, seaborn, sklearn)
- <u>Git</u>
- A web browser compatible with Jupyter Notebooks

#### **A** Warning

Everything in this class will be tested with the up to date (or otherwise specified) version of Jupyter Notebooks. Google Colab is similar, but not the same, and some things may not work there. It is an okay backup, but should not be your primary work environment.

### Recommendation:

- Install python via <u>Anaconda</u>
- if you use Windows, install Git with GitBash (video instructions).
- if you use MacOS, install Git with the Xcode Command Line Tools. On Mavericks (10.9) or above you can do this by trying to run git from the Terminal the very first time.git --version
- if you use Chrome OS, follow these instructions:
- 1. Find Linux (Beta) in your settings and turn that on.
- Once the download finishes a Linux terminal will open, then enter the commands: sudo apt-get update and sudo apt-get upgrade. These commands will ensure you are up to date.
- 3. Install tmux with:

```
sudo apt -t stretch-backports install tmux
```

4. Next you will install nodejs, to do this, use the following commands:

```
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash sudo apt-get install -y nodejs sudo apt-get install -y build-essential.
```

- 5. Next install Anaconda's Python from the website provided by the instructor and use the top download link under the Linux options.
- 6. You will then see a .sh file in your downloads, move this into your Linux files
- 7. Make sure you are in your home directory (something like home/YOURUSERNAME), do this by using the pwd command.
- 8. Use the bash command followed by the file name of the installer you just downloaded to start the installation.
- 9. Next you will add Anaconda to your Linux PATH, do this by using the vim .bashrc command to enter the .bashrc file, then add the export PATH=/home/YOURUSERNAME/anaconda3/bin/:\$PATH line. This can be placed at the end of the file.
- 10. Once that is inserted you may close and save the file, to do this hold escape and type :x, then press enter. After doing that you will be returned to the terminal where you will then type the source .bashrc command.
- 11. Next, use the jupyter notebook -generate-config command to generate a Jupyter Notebook
- 12. Then just type jupyter lab and a Jupyter Notebook should open up.

Optional:

#### Important

#### TL;DR [1]

- · check Brightspace
- · Log in to Prismia Chat
- Make a GitHub Account
- Install Python
- · Install Git

#### Note

Seeing the BrightSpace site requires loging in with your URI SSO and being enrolled in the course

#### Note

all Git instructions will be given as instructions for the command line interface and GitHub specific instructions via the web interface. You may choose to use GitHub desktop or built in IDE tools, but the instructional team may not be able to help.

• Text Editor: you may want a text editor outside of the Jupyter environment. Jupyter can edit markdown files (that you'll need for your portfolio), in browser, but it is more common to use a text editor like Atom or Sublime for this purpose.

Video install instructions for Anaconda:

- Windows
- Mac

On Mac, to install python via environment, this article may be helpful

. I don't have a video for linux, but it's a little more straight forward.

#### Textbook

The text for this class is a reference book and will not be a source of assignments. It will be a helpful reference and you may be directed there for answers to

Python for Data Science is available free online:

#### Zoom (backup and office hours only)

This is where we will meet if for any reason we cannot be in person. You will find the link to class zoom sessions on Brightspace.

URI provides all faculty, staff, and students with a paid Zoom account. It can run in your browser or on a mobile device, but you will be able to participate in class best if you download the Zoom client on your computer. Please log in and configure your account. Please add a photo of yourself to your account so that we can still see your likeness in some form when your camera is off. You may also wish to use a virtual background and you are welcome to do so.

Class will be interactive, so if you cannot be in a quiet place at class time, headphones with a built in microphone are strongly recommended.

For help, you can access the  $\underline{\text{instructions provided by IT}}.$ 

[1] Too long; didn't read.

### **Data Science Achievements**

In this course there are 5 learning outcomes that I expect you to achieve by the end of the semester. To get there, you'll focus on 15 smaller achievements that will be the basis of your grade. This section will describe how the topics covered, the learning outcomes, and the achievements are covered over time. In the next section, you'll see how these achievements turn into grades

### **Learning Outcomes**

By the end of the semester

- 1. (process) Describe the process of data science, define each phase, and identify standard tools
- 2. (data) Access and combine data in multiple formats for analysis
- 3. (exploratory) Perform exploratory data analyses including descriptive statistics and visualization
- 4. (modeling) Select models for data by applying and evaluating mutiple models to a single dataset
- 5. (communicate) Communicate solutions to problems with data in common industry formats

We will build your skill in the process and communicate outcomes over the whole semester. The middle three skills will correspond roughly to the content taught for each of the first three portfolio checks.

### Schedule

The course will meet TTh 5-6:15pm in Tyler 108. Every class will include participatory live coding (instructor types code while explaining, students follow along)) instruction and small exercises for you to progress toward level 1 achievements of the new skills introduced in class that day.

Each Assignment will have a deadline posted on the page. Portfolio deadlines will be announced at least 2 weeks in advance.

	topics	skills
week		
1	[admin, python review]	process
2	Loading data, Python review	[access, prepare, summarize]
3	Exploratory Data Analysis	[summarize, visualize]
4	Data Cleaning	[prepare, summarize, visualize]
5	Databases, Merging DataFrames	[access, construct, summarize]
6	Modeling, classification performance metrics, cross validation	[evaluate]
7	Naive Bayes, decision trees	[classification, evaluate]
8	Regression	[regression, evaluate]
9	Clustering	[clustering, evaluate]
10	SVM, parameter tuning	[optimize, tools]
11	KNN, Model comparison	[compare, tools]
12	Text Analysis	[unstructured]
13	Images Analysis	[unstructured, tools]
14	Deep Learning	[tools, compare]

#### **Achievement Definitions**

The table below describes how your participation, assignments, and portfolios will be assessed to earn each achievement. The keyword for each skill is a short name that will be used to refer to skills throughout the course materials; the full description of the skill is in this table.

Note

On the Course Calendar on BrightSpace page you can get a feed link to add to the calendar of your choice by clicking on the subscribe (star) button on the top right of the page. Class is for 1 hour there because of Brightspace/zoom integration limitations, but that calendar includes the zoom link

	skill	Level 1	Level 2	Level 3
keyword				
python	pythonic code writing	python code that mostly runs, occasional pep8 adherance	python code that reliably runs, frequent pep8 adherance	reliable, efficient, pythonic code that consistently adheres to pep8
process	describe data science as a process	Identify basic components of data science	Describe and define each stage of the data science process	Compare different ways that data science can facilitate decision making
access	access data in multiple formats	load data from at least one format; identify the most common data formats	Load data for processing from the most common formats; Compare and constrast most common formats	access data from both common and uncommon formats and identify best practices for formats in different contexts
construct	construct datasets from multiple sources	identify what should happen to merge datasets or when they can be merged	apply basic merges	merge data that is not automatically aligned
summarize	Summarize and describe data	Describe the shape and structure of a dataset in basic terms	compute summary statndard statistics of a whole dataset and grouped data	Compute and interpret various summary statistics of subsets of data
visualize	Visualize data	identify plot types, generate basic plots from pandas	generate multiple plot types with complete labeling with pandas and seaborn	generate complex plots with pandas and plotting libraries and customize with matplotlib or additional parameters
prepare	prepare data for analysis	identify if data is or is not ready for analysis, potential problems with data	apply data reshaping, cleaning, and filtering as directed	apply data reshaping, cleaning, and filtering manipulations reliably and correctly by assessing data as received
evaluate	Evaluate model performance	Explain basic performance metrics for different data science tasks	Apply and interpret basic model evaluation metrics to a held out test set	Evaluate a model with multiple metrics and cross validation
classification	Apply classification	identify and describe what classification is, apply pre-fit classification models	fit, apply, and interpret preselected classification model to a dataset	fit and apply classification models and select appropriate classification models for different contexts
regression	Apply Regression	identify what data that can be used for regression looks like	fit and interpret linear regression models	fit and explain regrularized or nonlinear regression
clustering	Clustering	describe what clustering is	apply basic clustering	apply multiple clustering techniques, and interpret results
optimize	Optimize model parameters	Identify when model parameters need to be optimized	Optimize basic model parameters such as model order	Select optimal parameters based of mutiple quanttiateve criteria and automate parameter tuning
compare	compare models	Qualitatively compare model classes	Compare model classes in specific terms and fit models in terms of traditional model performance metrics	Evaluate tradeoffs between different model comparison types
representation	Choose representations and transform data	Identify options for representing text and categorical data in many contexts	Apply at least one representation to transform unstructured or inappropriately data for model fitting or summarizing	apply transformations in different contexts OR compare and contrast multiple representations a single type of data in terms of model performance
workflow	use industry standard data science tools and workflows to solve data science problems	Solve well strucutred fully specified problems with a single tool pipeline	Solve well- strucutred, open- ended problems, apply common structure to learn new features of standard tools	Independently scope and solve realistic data science problems OR independently learn releated tools and describe strengths and weakensses of common tools

	<b>A1</b>	A2	А3	A4	А5	A6	Α7	<b>A8</b>	Α9	A10	A11	A12	A13	# Assignments
keyword														
python	1	1	0	1	1	0	0	0	0	0	0	0	0	4
process	1	0	0	0	0	1	1	1	1	1	1	0	0	7
access	0	1	1	1	1	0	0	0	0	0	0	0	0	4
construct	0	0	0	0	1	0	1	1	0	0	0	0	0	3
summarize	0	0	1	1	1	1	1	1	1	1	1	1	1	11
visualize	0	0	1	1	0	1	1	1	1	1	1	1	1	10
prepare	0	0	0	1	1	0	0	0	0	0	0	0	0	2
evaluate	0	0	0	0	0	1	1	1	0	1	1	0	0	5
classification	0	0	0	0	0	0	1	0	0	1	0	0	0	2
regression	0	0	0	0	0	0	0	1	0	0	1	0	0	2
clustering	0	0	0	0	0	0	0	0	1	0	1	0	0	2
optimize	0	0	0	0	0	0	0	0	0	1	1	0	0	2
compare	0	0	0	0	0	0	0	0	0	0	1	0	1	2
representation	0	0	0	0	0	0	0	0	0	0	0	1	1	2
workflow	0	٥	0	0	0	0	0	Ω	Ω	1	1	1	1	4

### **▲** Warning

process achievements are accumulated a little slower. Prior to portfolio check 1, only level 1 can be earned. Portfolio check 1 is the first chance to earn level 2 for process, then level 3 can be earned on portfolio check 2 or later.

### Portfolios and Skills

The objective of your portfolio submissions is to earn Level 3 achievements. The following table shows what Level 3 looks like for each skill and identifies which portfolio submissions you can earn that Level 3 in that skill.

	Level 3	P1	P2	Р3	P4
keyword					
python	reliable, efficient, pythonic code that consistently adheres to pep8	1	1	0	1
process	Compare different ways that data science can facilitate decision making	0	1	1	1
access	access data from both common and uncommon formats and identify best practices for formats in different contexts	1	1	0	1
construct	merge data that is not automatically aligned	1	1	0	1
summarize	Compute and interpret various summary statistics of subsets of data	1	1	0	1
visualize	generate complex plots with pandas and plotting libraries and customize with matplotlib or additional parameters	1	1	0	1
prepare	apply data reshaping, cleaning, and filtering manipulations reliably and correctly by assessing data as received	1	1	0	1
evaluate	Evaluate a model with multiple metrics and cross validation	0	1	1	1
classification	fit and apply classification models and select appropriate classification models for different contexts	0	1	1	1
regression	fit and explain regrularized or nonlinear regression	0	1	1	1
clustering	apply multiple clustering techniques, and interpret results	0	1	1	1
optimize	Select optimal parameters based of mutiple quanttiateve criteria and automate parameter tuning	0	0	1	1
compare	Evaluate tradeoffs between different model comparison types	0	0	1	1
representation	apply transformations in different contexts OR compare and contrast multiple representations a single type of data in terms of model performance	0	0	1	1
workflow	Independently scope and solve realistic data science problems OR independently learn releated tools and describe strengths and weakensses of common tools	0	0	1	1

### Detailed Checklists

#### python-level1

python code that mostly runs, occasional pep8 adherance

- [] logical use of control structures
- [] callable functions
- [] correct calls to functions
- [] correct use of variables
- [] use of logical operators

### python-level2

python code that reliably runs, frequent pep8 adherance

- [] descriptive variable names
- [] pythonic loops
- [] efficient use of return vs side effects in functions
- [] correct, effective use of builtin python iterable types (lists & dictionaries)

### python-level3

reliable, efficient, pythonic code that consistently adheres to pep8

- [] pep8 adherant variable, file, class, and function names
- [] effective use of multi-paradigm abilities for efficiency gains
- . [] easy to read code that adheres to readability over other rules

#### process-level1

Identify basic components of data science

- . [] identify component disciplines OR
- . [] idenitfy phases

#### process-level2

Describe and define each stage of the data science process

- . [] correctly defines stages
- [] identifies stages in use
- [] describes general goals as well as a specific processes

#### process-level3

Compare different ways that data science can facilitate decision making

- $\bullet\,$  [ ] describes exceptions to process and iteration in process
- [] connects choices at one phase to impacts in other phases
- [] connects data science steps to real world decisions

#### access-level1

load data from at least one format; identify the most common data formats

- [] use at least one pandas read\_ function correctly
- . [] name common types
- . [] describe the structure of common types

#### access-level2

 $Load\ data\ for\ processing\ from\ the\ most\ common\ formats;\ Compare\ and\ constrast\ most\ common\ formats$ 

- [] load data from at least two of (.csv, .tsv, .dat, database, .json)
- $\bullet\,$  [ ] describe advantages and disadvantages of most commone types
- [] descive how most common types are different

#### access-level3

access data from both common and uncommon formats and identify best practices for formats in different contexts

- [] load data from at least 1 uncommon format
- [] describe when one format is better than another

#### construct-level1

identify what should happen to merge datasets or when they can be merged

- [] identify what the structure of a merged dataset should be (size, shape, columns)
- [] idenitfy when datasets can or cannot be merged

#### construct-level2

apply basic merges

- [] use 3 different types of merges
- [] choose the right type of merge for realistic scenarios

### construct-level3

merge data that is not automatically aligned

- [] manipulate data to make it mergable
- [] identify how to combine data from many sources to answer a question
- [] implement stesp to combine data from multiple sources

#### summarize-level1

Describe the shape and structure of a dataset in basic terms

- [] use attributes to produce a description of a dataset
- [] display parts of a dataset

#### summarize-level2

compute and interpret summary standard statistics of a whole dataset and grouped data

- [] compute descriptive statistics on whole datasets
- [] apply individual statistics to datasets
- $\bullet\,$  [ ] group data by a categorical variable for analysis
- $\bullet\,$  [ ] apply split-apply-combine paradigm to analyze data
- [] interprete statistics on whole datasets
- [] interpret statistics on subsets of data

### summarize-level3

Compute and interpret various summary statistics of subsets of data

- [] produce custom aggregation tables to summarize datasets
- [] compute multivariate summary statistics by grouping
- [] compute custom cacluations on datasets

### visualize-level1

identify plot types, generate basic plots from pandas

- . [] generate at least two types of plots with pandas
- [] identify plot types by name
- . [1 interpret basic information from plots

#### visualize-level2

generate multiple plot types with complete labeling with pandas and seaborn

- [] generate at least 3 types of plots
- [] use correct, complete, legible labeling on plots
- [] plot using both pandas and seaborn
- [] interpret multiple types of plots to draw conclusions

#### visualize-level3

generate complex plots with pandas and plotting libraries and customize with matplotlib or additional parameters

- [] use at least two libraries to plot
- [] generate figures with subplots
- . [] customize the display of a plot to be publication ready
- . [] interpret plot types and explain them for novices
- [] choose appopriate plot types to convey information
- . [] explain why plotting common best practices are effective

#### prepare-level1

identify if data is or is not ready for analysis, potential problems with data

- [] identify problems in a dataset
- [] anticipate how potential data setups will interfere with analysis
- [] describe the structure of tidy data
- [] label data as tidy or not

#### prepare-level2

apply data reshaping, cleaning, and filtering as directed

- [] reshape data to be analyzable as directed
- . [] filter data as directed
- . [] rename columns as directed
- [] rename values to make data more analyzable
- [] handle missing values in at least two ways
- [] transform data to tidy format

### prepare-level3

apply data reshaping, cleaning, and filtering manipulations reliably and correctly by assessing data as received

- [] identify issues in a dataset and correctly implement solutions
- [] convert varialbe representation by changing types
- [] change variable representation using one hot encoding

#### evaluate-level1

Explain basic performance metrics for different data science tasks

- [] define at least two performance metrics
- [] describe how those metrics compare or compete

#### evaluate-level2

Apply and interpret basic model evaluation metrics to a held out test set

- [] apply at least three performance metrics to models
- [] apply metrics to subsets of data
- [] apply disparity metrics
- [] interpret at least three metrics

#### evaluate-level3

Evaluate a model with multiple metrics and cross validation

- [] explain cross validation
- [] explain importance of held out test and validation data
- [] describe why cross vaidation is important
- [ ] idenitfy appropriate metrics for different types of modeling tasks
- $\bullet \ \ \hbox{[] use multiple metriccs together to create a more complete description of a model's performance}\\$

### classification-level1

identify and describe what classification is, apply pre-fit classification models

- [] describe what classification is
- [ ] describe what a dataset must look like for classifiation
- $\bullet\,$  [ ] identify appliations of classifiation in the real world
- [] describe set up for a classification problem (tes,train)

### classification-level2

fit, apply, and interpret preselected classification model to a dataset

- [] split data for training and testing
- [] fit a classification model
- [] apply a classification model to obtain predictions
- [] interpret the predictions of a classification model
- $\bullet\,$  [ ] examine parameters of at least one fit classifier to explain how the prediction is made

- [] differentiate between model fitting and generating predictions
- [] evaluate how model parameters impact model performance

#### classification-level3

fit and apply classification models and select appropriate classification models for different contexts

- [] choose appropriate classifiers based on application context
- . [] explain how at least 3 different classifiers make predictions
- [] evaluate how model parameters impact model performance and justify choices when tradeoffs are necessary

#### regression-level1

identify what data that can be used for regression looks like

- [] identify data that is/not appropriate for regression
- [] describe univariate linear regression
- [] identify appliations of regression in the real world

#### regression-level2

fit and interpret linear regression models

- . [] split data for training and testing
- [] fit univariate linear regression models
- [] interpret linear regression models
- [] fit multivariate linear regression models

#### regression-level3

fit and explain regrularized or nonlinear regression

- [] fit nonlinear or regrularized regression models
- [] interpret and explain nonlinear or regrularized regresion models

#### clustering-level1

describe what clustering is

- $\bullet$  [ ] differentiate clustering from classification and regression
- [] identify appliations of clustering in the real world

#### clustering-level2

apply basic clustering

- [] fit Kmeans
- [] interpret kmeans
- [] evaluate clustering models

#### clustering-level3

apply multiple clustering techniques, and interpret results

- [] apply at least two clustering techniques
- [] explain the differences between two clustering models

#### optimize-level1

Identify when model parameters need to be optimized

• [] identify when parameters might impact model performance

#### optimize-level2

Optimize basic model parameters such as model order

- [] automatically optimize multiple parameters
- [] evaluate potential tradeoffs
- [] interpret optimization results in context

#### optimize-level3

Select optimal parameters based of mutiple quantitateve criteria and automate parameter tuning

- [] optimize models based on multiple metrics
- [] describe when one model vs another is most appropriate

### compare-level1

Qualitatively compare model classes

 $\bullet\,$  [ ] compare models within the same task on complexity

#### compare-level2

 $Compare\ model\ classes\ in\ specific\ terms\ and\ fit\ models\ in\ terms\ of\ traditional\ model\ performance\ metrics$ 

- [] compare models in multiple terms
- [] interpret cross model comparisons in context

#### compare-level3

Evaluate tradeoffs between different model comparison types

- [] compare models on multiple criteria
- [] compare optimized models
- [] jointly interpret optimization result and compare models
- [] compare models on quanttiateve and qualitative measures

### representation-level1

Identify options for representing text and categorical data in many contexts

• [] describe the basic goals for changing the representation of data

#### representation-level2

Apply at least one representation to transform unstructured or inappropriately data for model fitting or summarizing

. [] transform text or image data for use with ML

#### representation-level3

apply transformations in different contexts OR compare and contrast multiple representations a single type of data in terms of model performance

- [] transform both text and image data for use in ml
- [] evaluate the impact of representation on model performance

#### workflow-level1

Solve well strucutred fully specified problems with a single tool pipeline

• [] pseudocode out the steps to answer basic data science questions

#### workflow-level2

Solve well-strucutred, open-ended problems, apply common structure to learn new features of standard tools

- [] plan and execute answering real questions to an open ended question
- . [] describe the necessary steps and tools

#### workflow-level3

Independently scope and solve realistic data science problems OR independently learn releated tools and describe strengths and weakensses of common tools

- [] scope and solve realistic data science problems
- [] compare different data science tool stacks

### Grading

This section of the syllabus describes the principles and mechanics of the grading for the course. This course will be graded on a basis of a set of skills (described in detail the next section of the syllabus). This is in contrast to more common grading on a basis of points earned through assignments.

### Principles of Grading

Learning happens through practice and feedback. My goal as a teacher is for you to learn. The grading in this course is based on your learning of the material, rather than your completion of the activities that are assigned.

This course is designed to encourage you to work steadily at learning the material and demonstrating your new knowledge. There are no single points of failure, where you lose points that cannot be recovered. Also, you cannot cram anything one time and then forget it. The material will build and you have to demonstrate that you retained things.

- Earning a C in this class means you have a general understanding of Data Science and could participate in a basic conversation about all of the topics we cover. I expect everyone to reach this level.
- Earning a B means that you could solve simple data science problems on your own and complete parts of more complex problems as instructed by, for
  example, a supervisor in an internship or entry level job. This is a very accessible goal, it does not require you to get anything on the first try or to
  explore topics on your own. I expect most students to reach this level.
- Earning an A means that you could solve moderately complex problems independently and discus the quality of others' data science solutions. This class will be challenging, it requires you to explore topics a little deeper than we cover them in class, but unlike typical grading it does not require all of your assignments to be near perfect.

Grading this way also is more amenable to the fact that there are correct and incorrect ways to do things, but there is not always a single correct answer to a realistic data science problem. Your work will be assessed on whether or not it demonstrates your learning of the targeted skills. You will also receive feedback on how to improve.

#### How it works

There are 15 skills that you will be graded on in this course. While learning these skills, you will work through a progression of learning. Your grade will be based on earning 45 achievements that are organized into 15 skill groups with 3 levels for each.

These map onto letter grades roughly as follows:

- If you achieve level 1 in all of the skills, you will earn at least a C in the course.
- To earn a B, you must earn all of the level 1 and level 2 achievements.
- To earn an A, you must earn all of the achievements.

You will have at least three opportunities to earn every level 2 achievement. You will have at least two opportunities to earn every level 3 achievement. You will have three types of opportunities to demonstrate your current skill level: participation, assignments, and a portfolio.

Each level of achievement corresponds to a phase in your learning of the skill:

- To earn level 1 achievements, you will need to demonstrate basic awareness of the required concepts and know approximately what to do, but you
  may need specific instructions of which things to do or to look up examples to modify every step of the way. You can earn level 1 achievements in
  class, assignments, or portfolio submissions.
- To earn level 2 achievements you will need to demonstrate understanding of the concepts and the ability to apply them with instruction after earning the level 1 achievement for that skill. You can earn level 2 achievements in assignments or portfolio submissions.
- To earn level 3 achievements you will be required to consistently execute each skill and demonstrate deep understanding of the course material, after achieving level 2 in that skill. You can earn level 3 achievements only through your portfolio submissions.

For each skill these are defined in the Achievement Definition Table

#### Participation

While attending synchronous class sessions, there will be understanding checks and in class exercises. Completing in class exercises and correctly answering questions in class can earn level 1 achievements. In class questions will be administered through the classroom chat platform Prismia.chat; these records will be used to update your skill progression. You can also earn level 1 achievements from adding annotation to a section of the class notes.

#### Assignments

For your learning to progress and earn level 2 achievements, you must practice with the skills outside of class time.

Assignments will each evaluate certain skills. After your assignment is reviewed, you will get qualitative feedback on your work, and an assessment of your demonstration of the targeted skills.

#### Portfolio Checks

To earn level 3 achievements, you will build a portfolio consisting of reflections, challenge problems, and longer analyses over the course of the semester. You will submit your portfolio for review 4 times. The first two will cover the skills taught up until 1 week before the submission deadline.

The third and fourth portfolio checks will cover all of the skills. The fourth will be due during finals. This means that, if you have achieved mastery of all of the skills by the 3rd portfolio check, you do not need to submit the fourth one.

Portfolio prompts will be given throughout the class, some will be strucutred questions, others may be questions that arise in class, for which there is not time to answer.

#### TI DR

You could earn a C through in class participation alone, if you make nearly zero mistakes. To earn a B, you must complete assignments and participate in class. To earn an A you must participate, complete assignments, and build a portfolio.

#### Detailed mechanics

On Brightspace there are 45 Grade items that you will get a 0 or a 1 grade for. These will be revealed, so that you can view them as you have an opportunity to demonstrate each one. The table below shows the minimum number of skills at each level to earn each letter grade.

	Level 3	Level 2	Level 1
letter grade			
А	15	15	15
A-	10	15	15
B+	5	15	15
В	0	15	15
B-	0	10	15
C+	0	5	15
С	0	0	15
C-	0	0	10
D+	0	0	5
D	0	0	3

For example, if you achieve level 2 on all of the skills and level 3 on 7 skills, that will be a B+.

If you achieve level 3 on 14 of the skills, but only level 1 on one of the skills, that will be a B-, because the minimum number of level 2 achievements for a B is 15. In this scenario the total number of achievements is 14 at level 3, 14 at level 2 and 15 at level 3, because you have to earn achievements within a skill in sequence.

The letter grade can be computed as follows



this will be revealed after assignment 1

For example you can run the code like this in a cell to see the output



Or use assert to test it formally

```
assert compute_grade(14,14,14) == 'C-'

[assert compute_grade(15,15,15) == 'A'

[assert compute_grade(15,15,11) == 'A-'
```

#### Late work

Late assignments will not be graded. Every skill will be assessed through more than one assignment, so missing assignments occasionally not necessarily hurt your grade. If you do not submit any assignments that cover a given skill, you may earn the level 2 achievement in that skill through a portfolio check, but you will not be able to earn the level 3 achievement in that skill. If you submit work that is not complete, however, it will be assessed and receive feedback. Submitting pseudocode or code with errors and comments about what you have tried could earn a level 1 achievement. Additionally, most assignments cover multiple skills, so partially completing the assignment may earn level 2 for one, but not all. Submitting something even if it is not perfect is important to keeping conversation open and getting feedback and help continuously.

Building your Data Science Portfolio should be an ongoing process, where you commit work to your portfolio frequently. If something comes up and you cannot finish all that you would like assessed by the deadline, open an Extension Request issue on your repository.

In this issue, include:

- 1. A new deadline proposal
- 2. What additional work you plan to add

### ▲ Warning

If you will skip an assignment, please accept the GitHub assignment and then close the Feedback pull request with a comment. This way we can make sure that you have support you need.

### Note

In this example, you will have also achieved level 1 on all of the skills, because it is a prerequisite to level 2.

#### Note

You may visit office hours to discuss assignments that you did not complete on time to get feedback and check your own understanding, but they will not count toward skill demonstration.

- 3. Why the extension is important to your learning
- 4. Why the extension will not hinder your ability to complete the next assignment on time.

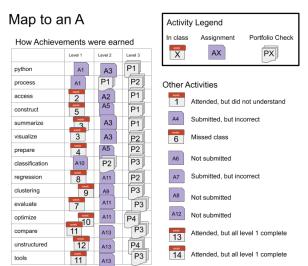
This request should be no more than 7 sentences.

Portfolio due dates will be announced well in advance and prompts for it will be released weekly. You should spend some time working on it each week, applying what you've learned so far, from the feedback on previous assignments.

#### **Grading Examples**

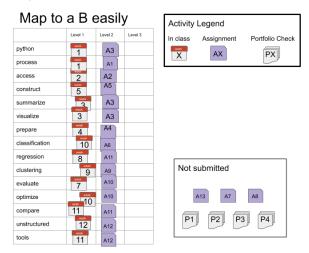
If you always attend and get everything correct, you will earn and A and you won't need to submit the 4th portfolio check.

Getting an A Without Perfection



In this example the student made several mistakes, but still earned an A. This is the advantage to this grading scheme. For the python, process, and classification skills, the level 1 achievements were earned on assignments, not in class. For the process and classification skills, the level 2 achievements were not earned on assignments, only on portfolio checks, but they were earned on the first portfolio of those skills, so the level 3 achievements were earned on the second portfolio checks, but they were earned on the first portfolio of those skills; optimize and unstructured. It included only 1 analysis, a text analysis with optimizing the parameters of the model. Assignments 4 and 7 were both submitted, but didn't earn any achievements, the student got feedback though, that they were able to apply in later assignments to earn the achievements. The student missed class week 6 and chose to not submit assignment 6 and use week 7 to catch up. The student had too much work in another class and chose to skip assignment 8. The student tried assignment 12, but didn't finish it on time, so it was not graded, but the student visited office hours to understand and be sure to earn the level 2 unstructured achievement on assignment 13.

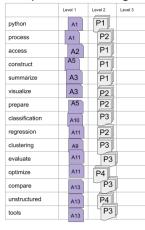
#### Getting a B with minimal work

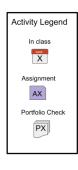


In this example, the student earned all level 1 achievements in class and all level 2 on assignments. This student was content with getting a B and chose to

Getting a B while having trouble

### Map to a B, having trouble





In this example, the student struggled to understand in class and on assignments. Assignments were submitted that showed some understanding, but all had some serious mistakes, so only level 1 achievements were earned from assignments. The student wanted to get a B and worked hard to get the level 2 achievements on the portfolio checks.

### **Grading Policies**

#### Late Work

Late assignments will not be graded. Every skill will be assessed through more than one assignment, so missing assignments occasionally not necessarily hurt your grade. If you do not submit any assignments that cover a given skill, you may earn the level 2 achievement in that skill through a portfolio check, but you will not be able to earn the level 3 achievement in that skill. If you submit work that is not complete, however, it will be assessed and receive feedback. Submitting pseudocode or code with errors and comments about what you have tried could earn a level 1 achievement. Additionally, most assignments cover multiple skills, so partially completing the assignment may earn level 2 for one, but not all. Submitting something even if it is not perfect is important to keeping conversation open and getting feedback and help continuously.

Building your Data Science Portfolio should be an ongoing process, where you commit work to your portfolio frequently. If something comes up and you cannot finish all that you would like assessed by the deadline, open an Extension Request issue on your repository.

In this issue, include:

- 1. A new deadline proposal
- 2. What additional work you plan to add
- 3. Why the extension is important to your learning
- 4. Why the extension will not hinder your ability to complete the next assignment on time.

This request should be no more than 7 sentences.

Portfolio due dates will be announced well in advance and prompts for it will be released weekly. You should spend some time working on it each week, applying what you've learned so far, from the feedback on previous assignments.

### Regrading

Re-request a review on your Feedback Pull request.

For general questions, post on the conversation tab of your Feedback PR with your request.

For specific questions, reply to a specifc comment.

If you think we missed where you did something, add a comment on that line (on the code tab of the PR, click the plus (+) next to the line) and then post on the conversation tab with an overview of what you're requestion and tag @brownsarahm

### Support



URI changed some links and this page is not yet up to date

### Academic Enhancement Center

Academic Enhancement Center (for undergraduate courses): Located in Roosevelt Hall, the AEC offers free face-to-face and web-based services to undergraduate students seeking academic support. Peer tutoring is available for STEM-related courses by appointment online and in-person. The Writing Center offers peer tutoring focused on supporting undergraduate writers at any stage of a writing assignment. The UCS160 course and academic skills consultations offer students strategies and activities aimed at improving their studying and test-taking skills. Complete details about each of these programs, up-to-date schedules, contact information and self-service study resources are all available on the AEC website.

- STEM Tutoring helps students navigate 100 and 200 level math, chemistry, physics, biology, and other select STEM courses. The STEM Tutoring
  program offers free online and limited in-person peer-tutoring this fall. Undergraduates in introductory STEM courses have a variety of small group
  times to choose from and can select occasional or weekly appointments. Appointments and locations will be visible in the TutorTrac system on
  September 14th, 2020. The TutorTrac application is available through URI Microsoft 365 single sign-on and by visiting aec.uri.edu. More detailed
  information and instructions can be found on the AEC tutoring page.
- Academic Skills Development resources helps students plan work, manage time, and study more effectively. In Fall 2020, all Academic Skills and Strategies programming are offered both online and in-person. UCS160: Success in Higher Education is a one-credit course on developing a more effective approach to studying. Academic Consultations are 30-minute, 1 to 1 appointments that students can schedule on Starfish with Dr. David

#### Not

You may visit office hours to discuss assignments that you did not complete on time to get feedback and check your own understanding, but they will not count toward skill demonstration.

Hayes to address individual academic issues. Study Your Way to Success is a self-guided web portal connecting students to tips and strategies on studying and time management related topics. For more information on these programs, visit the <a href="Academic Skills Page">Academic Skills Page</a> or contact Dr. Hayes directly at <a href="davidhayes@uri.edu">davidhayes@uri.edu</a>.

• The Undergraduate Writing Center provides free writing support to students in any class, at any stage of the writing process: from understanding an assignment and brainstorming ideas, to developing, organizing, and revising a draft. Fall 2020 services are offered through two online options: 1) real-time synchronous appointments with a peer consultant (25- and 50-minute slots, available Sunday - Friday), and 2) written asynchronous consultations with a 24-hour turn-around response time (available Monday - Friday). Synchronous appointments are video-based, with audio, chat, document-sharing, and live captioning capabilities, to meet a range of accessibility needs. View the synchronous and asynchronous schedules and book online, visit uri mywconline.com.

### General URI Policies



URI changed some links and this page is not yet up to date

### Anti-Bias Statement:

We respect the rights and dignity of each individual and group. We reject prejudice and intolerance, and we work to understand differences. We believe that equity and inclusion are critical components for campus community members to thrive. If you are a target or a witness of a bias incident, you are encouraged to submit a report to the URI Bias Response Team at <a href="https://www.uri.edu/brt">www.uri.edu/brt</a>. There you will also find people and resources to help.

#### Disability Services for Students Statement:

Your access in this course is important. Please send me your Disability Services for Students (DSS) accommodation letter early in the semester so that we have adequate time to discuss and arrange your approved academic accommodations. If you have not yet established services through DSS, please contact them to engage in a confidential conversation about the process for requesting reasonable accommodations in the classroom. DSS can be reached by calling: 401-874-2098, visiting: web.uri.edu/disability, or emailing: dss@etal.uri.edu. We are available to meet with students enrolled in Kingston as well as Providence courses.

#### **Academic Honesty**

Students are expected to be honest in all academic work. A student's name on any written work, quiz or exam shall be regarded as assurance that the work is the result of the student's own independent thought and study. Work should be stated in the student's own words, properly attributed to its source.

Students have an obligation to know how to quote, paraphrase, summarize, cite and reference the work of others with integrity. The following are examples of academic dishonesty.

- · Using material, directly or paraphrasing, from published sources (print or electronic) without appropriate citation
- · Claiming disproportionate credit for work not done independently
- · Unauthorized possession or access to exams
- · Unauthorized communication during exams
- · Unauthorized use of another's work or preparing work for another student
- Taking an exam for another student
- Altering or attempting to alter grades
- The use of notes or electronic devices to gain an unauthorized advantage during exams
- Fabricating or falsifying facts, data or references
- · Facilitating or aiding another's academic dishonesty
- Submitting the same paper for more than one course without prior approval from the instructors

### URI COVID-19 Statement

The University is committed to delivering its educational mission while protecting the health and safety of our community. While the university has worked to create a healthy learning environment for all, it is up to all of us to ensure our campus stays that way.

As members of the URI community, students are required to comply with standards of conduct and take precautions to keep themselves and others safe. Visit web.uri.edu/coronavirus/ for the latest information about the URI COVID-19 response.

- Universal indoor masking is required by all community members, on all campuses, regardless of vaccination status. If the universal mask mandate is
  discontinued during the semester, students who have an approved exemption and are not fully vaccinated will need to continue to wear a mask
  indoors and maintain physical distance.
- Students who are experiencing symptoms of illness should not come to class. Please stay in your home/room and notify URI Health Services via
  phone at 401-874-2246.
- If you are already on campus and start to feel ill, go home/back to your room and self-isolate. Notify URI Health Services via phone immediately at 401-874-2246.

If you are unable to attend class, please notify me at <a href="mailto:brownsarahm@uri.edu">brownsarahm@uri.edu</a>. We will work together to ensure that course instruction and work is completed for the semester.

### **Course Communications**

#### Announcements

Announcements will be made via GitHub Release. You can view them online in the releases page or you can get notifications by watching the repository, choosing "Releases" under custom see GitHub docs for instructions with screenshots. You can choose GitHub only or e-mail notification from the notification settings page.

#### Help Hours

```
AttributeError Traceback (most recent call last)
/tmp/ipykernel_1909/2404517823.py in <module>
----> 1 help_df.style.hide(axis="index")
AttributeError: 'Styler' object has no attribute 'hide'
```

## To reach out, By usage

note	area	platform	usage
outside of class time this is not monitored closely	chat	prismia	in class
use after class to get preliminary notes eg if you miss a class	download transcript	prismia	any time
eg bugs in your code"	issue on assignment repo	github	private questions to your assignment
eg what the instructions of an assignment mean or questions about the syllabus	issue on course website	github	for general questions that can help others
include links in your portfolio	discussion on community repo	github	to share resources or ask general questions in a semi-private forum
remember to include `[CSC310]` or `[DSP310]` (note `verbatim` no space)	to brownsarahm@uri.edu	e-mail	matters that don't fit into another category

O Note

e-mail is last because it's not collaborative; other platforms allow us (Proessor + TA) to collaborate on who responds to things more easily.

### By Platform

#### Use e-mail for

usage	area	note	!
matters that don't fit into another category	to brownsarahm@uri.edu	remember to include `[CSC310]` or `[DSP310]` (note `verbatim` no space)	

### Use github for

note	area	usage
eg bugs in your code"	issue on assignment repo	private questions to your assignment
eg what the instructions of an assignment mean or questions about the syllabus	issue on course website	for general questions that can help others
include links in your portfolio	discussion on community repo	to share resources or ask general questions in a semi-private forum

### Use prismia for

usage	area	note
in class	chat	outside of class time this is not monitored closely
anv time	download transcript	use after class to get preliminary notes eg if you miss a class

### Tips

### For assignment help

• send in advance, leave time for a response I check e-mail/github a small number of times per day, during work hours, almost exclusively. You might see me post to this site, post to BrightSpace, or comment on your assignments outside of my normal working hours, but I will not reliably see emails that arrive during those hours. This means that it is important to start assignments early.

### Using issues

- use issues for content directly related to assignments. If you push your code to the repository and then open an issue, I can see your code and your question at the same time and download it to run it if I need to debug it
- use issues for questions about this syllabus or class notes. At the top right there's a GitHub logo (7) that allows you to open a issue (for a question) or suggest an edit (eg if you think there's a typo or you find an additional helpful resource related to something)

#### For E-email

- use e-mail for general inquiries or notifications
- Please include [csc310] or [DSP310] in the subject line of your email along with the topic of your message. This is important, because your messages are important, but I also get a lot of e-mail. Consider these a cheat code to my inbox: I have setup a filter that will flag your e-mail if you use one of those in the subject to ensure that I see it.

## 1. Welcome and Introduction

### 1.1. Prismia Chat

We will use these to monitor your participation in class and to gather information. Features:

- · instructor only
- · reply to you directly
- share responses for all

### 1.2. How this class will work

#### Participatory Live Coding

What is a topic you want to use data to learn about?

Note

Whether you use CSC or DSP does not matter.

### 1.3. Programming for Data Science vs other Programming

The audience is different, so the form is different.

In Data Science our product is more often a report than a program.



Also, in data science we are *using code* to interact with data, instead of having a plan in advance

So programming for data science is more like writing it has a narrative flow and is made to be seen more than some other programming thaat you may have

## one.

# 1.4. Jupyter Notebooks

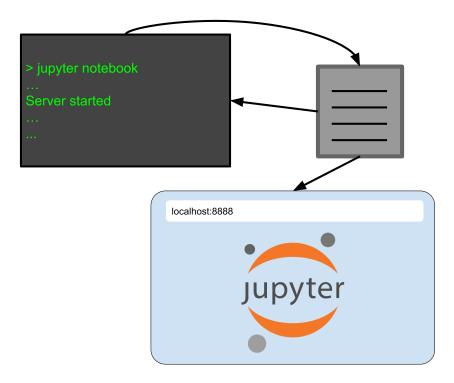
Launch a jupyter notebook server:

- · on Windows, use anaconda terminal
- on Mac/Linux, use terminal

cd path/to/where/you/save/notes jupyter notebook

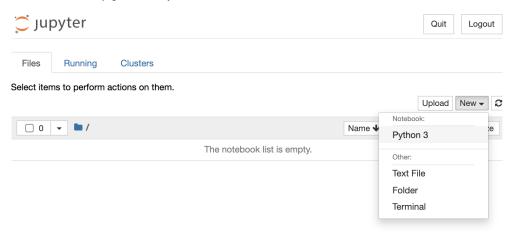
### 1.4.1. What just happened?

- · launched a local web server
- opened a new browser tab pointed to it



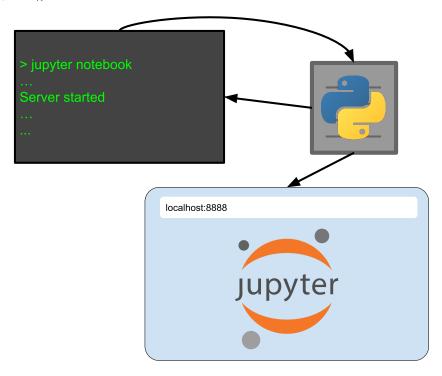
### 1.4.2. Start a Notebook

Go to the new menu in the top right and choose Python 3



### ▲ Warning

Sometimes there will be points in the notes that were not made in class due to time or in response questions that came at the end of class.

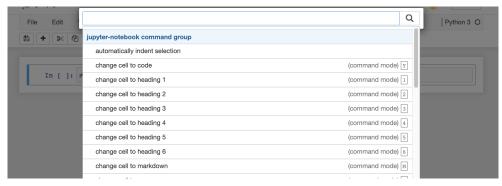


### 1.4.3. A jupyter notebook tour

A Jupyter notebook has two modes. When you first open, it is in command mode. The border is blue in command mode.



When you press a key in command mode it works like a shortcut. For example  $\ensuremath{\text{p}}$  shows the command search menu.



If you press enter (or return) or click on the highlighted cell, which is the boxes we can type in, it changes to edit mode. The border is green in edit mode

```
In [ ]: # This is a comment in a code cell
```

There are two type of cells that we will used: code and markdown. You can change that in command mode with y for code and m for markdown or on the cell type menu at the top of the notebook.



This is a markdown cell

- we can make
- · itemized lists of
- bullet points
- 1. and we can make nubmered
- 2. lists, and not have to worry
- 3. about renumbering them
- 4. if we add a step in the middle later

```
# this is a comment in a code cell , we can run this to see output
3+4
```

the output here is the value returned by the python interpretter for the last line of the cell

We set variables

```
name = 'sarah'
```

The notebook displays nothing when we do an assignment, bcause it returns nothing

```
name
'sarah'
```

we can put a variable there to ee it

```
course = 'csc310'
course
name

'sarah'
```

Note that this version doesn't show use the value for course

```
name = 'Sarah'
```

Important

In class, we ran these cells out of order and noticed how the value does not update unless we run the new version

```
name

'Sarah'

course

'csc310'
```

### 1.4.4. Notebook Reminders

Blue border is command mode, green border is edit mode

use Escape to get to command mode

Common command mode actions:

- m: switch cell to markdown
- y: switch cell to code
- a: add a cell above
- b: add a cell below
- c: copy cel
- v: paste the cell
- 0 + 0: restart kernel
- p: command menu

use enter/return to get to edit mode

In code cells, we can use a python interpreter, for example as a calculator.

```
10
```

It prints out the last line of code that it ran, even though it executes all of them

```
name = 'sarah'
4+5
name *3

'sarahsarahsarah'
```

Getting help is important in programming

When your cursor is inside the () of a function if you hold the shift key and press tab it will open a popup with information. If you press tab twice, it gets bigger and three times will make a popup window.

Python has a print function and we can use the help in jupyter to learn about how to use it in different ways.

We can print the docstring out, as a whole instead of using the shfit + tab to view it.

```
help(print)

Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file: a file-like object (stream); defaults to the current sys.stdout.
    sep: string inserted between values, default a space.
    end: string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

The first line says that it can take multiple values, because it says value, ..., sep

It also has a keyword argument (must be used like argument=value and has a default) described as sep=' '. This means that by default it adds a space as above.

```
print(name)
sarah
```

How do you use the print function to output: Sarah\_csc310?

```
print(name,course,sep='_')

sarah_csc310

help(print)

Help on built-in function print in module builtins:
print(...)
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
```

We can put as many values as we want there. Thats what the . . . in the function signature means

```
print(name,course,'hello','bye',sep='_')

sarah_csc310_hello_bye

print(name,course,'hello','bye',sep='\n')

sarah
csc310
hello
bye
```

Important

Basic programming is a prereq and we will go faster soon, but the goal of this review was to understand notebooks, getting help, and reading docstrings

### 1.6. What is Data Science?



statistics is the type of math we use to make sense of data. Formally, a statistic is just a function of data.

computer science is so that we can manipulate visualize and automate the inferences we make.

domain expertise helps us have the intuition to know if what we did worked right. A statistic must be interpreted in context; the relevant context determines what they mean and which are valid. The context will say whether automating something is safe or not, it can help us tell whether our code actually worked right or not.

### 1.6.1. In this class,



We'll focus on the programming as our main means of studying data science, but we will use bits of the other parts. In particular, you're encouraged to choose datasets that you have domain expertise about, or that you want to learn about.

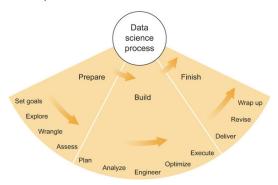
But there are many definitions. We'll use this one, but you may come across others.

### 1.6.2. How does data science happen?

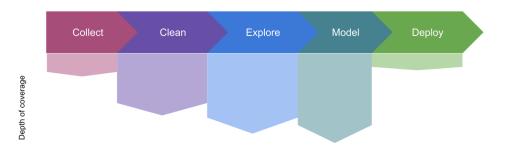
The most common way to think about what doing data science means is to think of this pipeline. It is in the perspective of the data, these are all of the things that happen to the data.



Another way to think about it

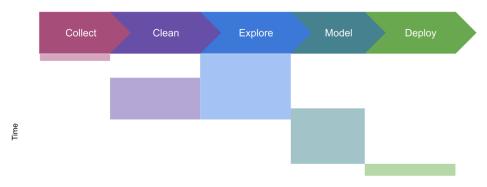


### 1.6.3. how we'll cover Data Science, in depth



- collect: Discuss only a little; Minimal programming involved
- $\bullet \ \textit{clean} : \textbf{Cover the main programming techniques}; \textbf{Some requires domain knowledge beyond scope of course}\\$
- $\bullet \ \ \text{\it explore} : \text{\it Cover the main programming techniques}; \text{\it Some requires domain knowledge beyond scope of course}$
- $\bullet \ \textit{model}{:} \textbf{Cover the main programming, basic idea of models; How to use models, not how learning algorithms work} \\$
- deploy: A little bit at the end, but a lot of preparation for decision making around deployment

### 1.6.4. how we'll cover it in, time



We'll cover exploratory data analysis before cleaning because those tools will help us check how we've cleaned the data.

### 1.7. Prepare for the next class

- Read carefully the syllabus section of the course website
- skim the rest of the course website
- Bring questions about how the class will work to class on Thursday.
- Review <u>Git & GitHub Fundamentals</u>
- Bring git/github questions on Thursday.
- Begin reading chapter 1 of think like a data scientist (finish in time for it to help you with the assignment due Monday night)

On Thursday we will start with a review of the syllabus. You will answer an ungraded quiz to confirm that you understand and I'll answer all of your questions. Then we will do a little bit with Git/GitHub and start your first assignment in class.

Think like a data scientist is written for practitioners; not as a text book for a class. It does not have a lot of prerequisite background, but the sections of it that I assign will help you build a better mental picture of what doing Data Science about.

Only the first assignment will be due this fast, it's a short review and setup assignment. It's due quickly so that we know that you have everything set up and the prerequisite material before we start new material next week.

## 2. Syllabus and Python Review

### 2.1. Course Logistics

Class is designed to avoid this:





Read more about how I'm designing this course to help you learn on the how to learn page.

#### 2.2. Check your understanding of the syllabus

It's easy when reading something long to lose track of it. Your eyes can go over each word, without actually retaining the information, but it's important to understand the syllabus for the course.

You can find the answers to the following questions on the syllabus. If you've already read it, try answering them to check your understanding. If you haven't read it yet, use these to guide you to get familiar with finding key facts about the course on the syllabus.

- 1. What do you need to bring to class each day?
- 2. What is the basis of grading for this course?
- 3. How do you reference the course text?
- 4. What is the penalty for missing an assignment?

More information about the course is available throughout the site, the next few questions will help you self-check that you've found the important things. Remember, the goal is not necessarily to memorize all of this, but to be able to find it.

- 1. When & what are you expected to read for this class?
- [] read the text book before class
- [] review notes & documentation after class
- [] preview the notes & documentation before class
- [] read documentation and text book after class
- 1. Your assignment says to find a dataset that has variables of a specific type, which website can you use?
- 2. Your assignment says to find a dataset of any type about something you're interested in, which resource would you use?

### 2.3. Python Review

Official source on python:

- pep8 official style
- documentation note that you can change which version you are using

We will go quickly through these focusing on pythonic style, because the prerequisite is a programming course.

### 2.3.1. Functions

Syntax of a function in python:

A few things to note:

- the def keywords starts a function
- then the name of the function
- parameters in ( ) then :
- · the body is indented

- the first thing in the body should be a docstring, denoted in ''' which is a multiline comment
- · returning is more reliable than printing in a function

← Tip

In python, PEP 257 says how to write a docstring, but it is very broad.

In Data Science, numpydoc style docstrings are popular.

- Pandas follows numpydoc
- [Numpy uses it]
- · Scipy follows numpydoc

Once the cell with the function definition is run, we can use the function

```
greeting('sarah')

'hi sarah'
```

With a return this works to check that it does the right thing

```
assert greeting('sarah') == "hi sarah"
```

#### 2.3.2. Conditionals

key points in this function:

- an if also has the conditional part indented
- for a bool variable we can just use the variable
- · we can set a default value

```
greeting2('sarah', True)

'hello sarah'

greeting2('sarah', False)

'hi sarah'
```

because of the default value we do not have to pass the second variable:

```
greeting2('sarah')

'hi sarah'

help(greeting)

Help on function greeting in module __main__:
greeting(name)
    say hi to a person
    Parameters
    .......
name : string
    the name of who to greet
```

### 2.4. Questions After Class

### 2.4.1. Why is indentation important in python but not other languages like C++?

Python is a newer language than C and C++. Older languages had to contend with the fact that a space character uses the same amount of memory as any other character, so they were not used. However, whitespace is easy to read.

Python was started in 1989, compared to C in 1972, C++ was started in 1985ish, but stuck with a lot of things from C, so tht 1972 is strongly operative.

Python is designed to be easy. It is designed to make complex tasks easier to do. C is designed to be efficient and to compeile well, even if it is hard to learn and do.

#### 2.4.2. Why is python so much slower as well?

Python is slower because it is an interpreted language. That means another program called the Python interpreter (which mostly are written in C) is actually running on your computer, that program parses the text of your source code and then executes the code. The interpreter cannot look ahead and change things in how you wrote your code while it runs.

In contrast, C++ (like C) is a compiled language. This means that a program called a compiler parses your code and translates it into assembly then to machine code. During this process it can optimize your code to make sure that it is fast.

2.4.3. Are portfolios simply whatever we submit, such as assignments, to Github or are there other things that need to be submitted to the portfolios for level 3?

Assignments are separate from the portfolio checks. It will become more clear what to do in your portfolio after you get feedback on assignment 2, and start working on assignment 3.

2.4.4. Will we know the specific criteria to fulfill a level 3 achievement when doing the portfolio?

The evaluation criteria are already listed on the Detailed Checklists.

### 2.4.5. when is the Assignment 1 due?

Monday, end of day. See the details: Assignment 1: Portfolio Setup, Data Science, and Python

2.4.6. how many large scale programs are we going to write in jupyter?

We won't actually write large scale programs, per se, but we will write some long analyses.

## 3. Grading review, Pandas, and Iterables

### 3.1. Grading Calculation

here is my solution

Note that we can verify it works using assert

```
| assert compute_grade(15,15,15) =='A' | | assert compute_grade(15,15,9) =='B+'
```

this also means we can assign the value out

```
my_grade = compute_grade(15,11,5)

my_grade

'B-'
```

Alternatively if we use a side effect instead, printing the value instead of returning it.

Look this way it looks similar:

```
compute_grade_sideeffect(15,15,15)
```

^

```
compute_grade(15,15,15)
```

'A'

and python lets us assign something

```
my_grade = compute_grade_sideeffect(15,15,15)
```

but the output is nothing

```
type(my_grade)
```

NoneType

3.2. Loading data with Pandas
First we learned about the dataset then we can load it.

```
{\tt coffee\_data\_url = 'https://raw.githubusercontent.com/jldbc/coffee-quality-database/master/data/robusta\_data\_cleaned.csv'}
```

We will use pandas.

```
import pandas as pd
```

load the data

```
coffee_df = pd.read_csv(coffee_data_url,index_col=0)
```

type(coffee\_df)

```
pandas.core.frame.DataFrame
```

coffee\_df.columns

```
coffee_df.columns[0]
```

'Species'	
len(coffee_df.columns)	
43	
coffee_df.shape	
(28, 43)	
coffee_df.head()	

	Species	Owner	Country.of.Origin	Farm.Name	Lot.Number	Mill	ICO.Number	Company	Altitude	Region	 Color	Category.Two.Defects	Е	ı
1	Robusta	ankole coffee producers coop	Uganda	kyangundu cooperative society	NaN	ankole coffee producers	0	ankole coffee producers coop	1488	sheema south western	 Green	2		
2	Robusta	nishant gurjer	India	sethuraman estate kaapi royale	25	sethuraman estate	14/1148/2017/21	kaapi royale	3170	chikmagalur karnataka indua	 NaN	2	! ;	:
3	Robusta	andrew hetzel	India	sethuraman estate	NaN	NaN	0000	sethuraman estate	1000m	chikmagalur	 Green	0		
4	Robusta	ugacof	Uganda	ugacof project area	NaN	ugacof	0	ugacof ltd	1212	central	 Green	7		
5	Robusta	katuka development trust ltd	Uganda	katikamu capca farmers association	NaN	katuka development trust	0	katuka development trust ltd	1200- 1300	luwero central region	 Green	3		
5 r	ows × 43 co	olumns												
[	offee_df.I	nead												

```
Species
                                                                                                                                                          Owner
                                        ankole coffee producers coop
nishant gurjer
andrew hetzel
                                                                                                                             Uganda
India
India
                                        ugacof
katuka development trust ltd
                                                                                                                               Uganda
                                                                                                                              Uganda
India
India
India
India
                                                                     andrew hetzel
andrew hetzel
nishant gurjer
nishant gurjer
                                                                                       ugacof
                                                                                                                               Uganda
                                                                                       ugacof
                                                                                                                              Uganda
India
                            ugacof
nishant gurjer
nadrew hetzel
kasozi coffee farmers association
ankole coffee producers coop
andrew hetzel
andrew hetzel
                                                                                                                              India
Uganda
Uganda
India
                                                                                                                                India
                                                              kawacom uganda ltd
nitubaasa ltd
                                                                                                                               Uganda
                                                                                                                               Uganda
                                                       mannya coffee project
andrew hetzel
andrew hetzel
andrew hetzel
luis robles
luis robles
                                                                                                               Uganda
India
India
United States
                                                                                                                            Ecuador
                                                                                                                            Ecuador
                                                                        james moore
cafe politico
cafe politico
                                                                                                               United States
                                                                                                                                India
                                                                                                                           Vietnam
         Farm.Name
kyangundu cooperative society
sethuraman estate kaapi royale
sethuraman estate
ugacof project area
katikamu capca farmers association
Nath
sethuraman estates
sethuraman estate
sethuraman estate
ishaka
                                                                                                    NaN
  25
                                                                                                    Nan
Nan
Nan
Nan
Nan
                                                                                                    7
RKR
                                                                     ishaka
                                                                                                Nan
Nan
RC AB
Nan
Nan
Nan
Nan
Nan
Nan
Nan
Nan
Nan
                  ugacof project area
sethuraman estate kaapi royale
sethuraman estates
kasozi coffee farmers
                                    kyangundu coop society
                                               sethuraman estate
                                            sethuraman estates
            sethuraman estates
bushenyi
kigezi coffee farmers association
mannya coffee project
sethuraman estates
sethuraman estates
                                            sethuraman estates
                                                                                                    NaN
                                                  robustasa
robustasa
robustasa
fazenda cazengo
NaN
NaN
                                                                                         Nan
Lavado 1
Lavado 3
Nan
Nan
                                                                                   Mill
                                                                                                               ICO.Number
                                          ankole coffee producers
sethuraman estate
 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
                                                                                                    14/1148/2017/21
                                                                              NaN
ugacof
                                         katuka development trust
                                                                              (self)
                                                                                                                               NaN
                                                                                     Nań
                                                                                                                              NaN
                                                       NaN
sethuraman estate
sethuraman estate
nsubuga umar
ugacof
sethuraman estate
                                                                                                    14/1148/2017/18
14/1148/2016/17
                                                                                                    14/1148/2016/12
                                                                                                                             NaN
0
0
                                                                                      NaN
          ankole coffee producers coop union ltd
                                                                                                                            0000
                                                     NAN
sethuraman estates
kawacom
nitubaasa
                                                                                                                              NaN
0
```

```
mannya coffee project
20
21
22
23
24
25
26
27
28
                                                                                                                                                                                                             Nan
Nan
Nan
Nan
Nan
                                                                                     sethuraman estates
sethuraman estates
our own lab
own laboratory
                                                                                                        cafe cazengo
                                                                                                                                                                                                             NaN
                                                                                                                                                          14-1118-2014-0087
                              Company ankole coffee producers coop
                                                                                                                                                              1488
                                                                     kaapi royale
sethuraman estate
                                                                                                                                                              3170
                                                                                                                                                            1000m
                              sethuraman estate
ugacof Itd
katuka development trust Itd
cafemakers, Ilc
cafemakers
kaapi royale
                                                                                                                                                              1212
                                                                                                                                            1200-1300
                                                                                                                                                            3000 '
750m
3140
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
              kaapi royale
kaapi royale
ugacof ltd
ugacof ltd
cafemakers
kasozi coffee farmers association
ankole coffee producers cop
                                                                                                                                                               1000
                                                                                                                                                900-1300
                                                                                                                                                              1095
                                                                                                                                                               1488
                                                                     sethuraman estate
                                                                                                                                                            1000m
                                                    kawacom uganda ltd
nitubaasa ltd
mannya coffee project
cafemakers, llc
cafemakers, llc
                                                                                                                                                              750m
                                                                                                                                                              1600
                                                                                                                                                               1745
                                                                                                                                                              1200
750m
750m
                                                                                                                                                           3000
                                                                                     robustasa
robustasa
                                                                                                                                                                NaN
                                                                                                                                                                      40
                                                global opportunity fund
cafe politico
cafe politico
                                                                                                                                         795 meters
                                                                                                                                                                 NaN
NaN
                                                                                              Region
                                                                                                                                                                     Color
                                                                                                                                                                                             Category. Two. Defects
                                              sheema south western
                                                                                                                                                                     Green
                    chikmagalur karnataka indua
                                                                                                                                                                            NaN
                                          chikmagalur
central
luwero central region
                                                                                                                                                                     Green
Green
Green
                                                                            chikmagalur
chikmagalur
                                                                                                                                                                      Green
                                                                                                                                                                       Green
                    chikmagalur karnataka india
                                                                                                                          ... Bluish-Green
... Green
... Green
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
                                                                                                                                                                     Green
Green
Green
Green
                                          chikmagalur karnataka
                                chikmagalur karnataka
western
iganga namadrope eastern
chikmagalur karnataka
chikmagalur
eastern
south western
                                                                                                                                                                       Green
                                                                                                                                                                      Green
                                                                                                                                                                     Green
                                                                           chikmagalur
chikmagalur
western
western
southern
                                                                                                                                                                      Green
                                                                                                                                                   Blue
                                                                                                                                                                       Green
                                                                             chikmagalur
                                                                                                                           ... Bluish-Green
                                                                             chikmagalur
chikmagalur
                                                                                                                                                                    Green
                                                                                                                                                                     Green
              san juan, playas
san juan, playas
kwanza norte province, angola
NaN
                                                                                                        NaN
                                                                                                                                                                        None
                                             Expiration
                                                                                                                                                   Certification.Body
                                                                                     Certification.Body
Uganda Coffee Development Authority
Specialty Coffee Association
Specialty Coffee Association
Uganda Coffee Development Authority
Uganda Coffee Development Authority
Specialty Coffee Association
Specialty Coffee Association
Specialty Coffee Association
Specialty Coffee Association
Uganda Coffee Development Authority
Uganda Coffee Development Authority
Uganda Coffee Development Authority
Uganda Coffee Development Authority
                Expiration
June 26th, 2015
October 31st, 2018
April 29th, 2016
July 14th, 2015
June 26th, 2016
              February 28th, 2013
                 May 15th, 2015
October 25th, 2018
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
                    August 17th, 2017
August 5th, 2015
June 26th, 2015
August 23rd, 2017
                                                                                                                 Specialty Coffee Association
Specialty Coffee Association
                                May 19th, 2015
                        May 19th, 2015
July 14th, 2015
July 14th, 2015
April 29th, 2016
June 3rd, 2014
June 27th, 2015
June 27th, 2015
                           May 19th, 2015
July 14th, 2015
July 14th, 2015
June 3rd, 2014
June 27th, 2015

                                                                                                              Coffee Development Authority
Specialty Coffee Association
                               May 19th, 2015
             May 19th, 2015
June 20th, 2014
February 28th, 2013
January 18th, 2017
January 18th, 2017
December 23rd, 2015
                    August 25th, 2015
28
                     August 25th, 2015
                                                                                Certification.Address
               Certification.Address
e36d0270932c3b657e96b7b0278dfd85d0ef6743
ff7c18ad303d4b603ac3f8cff7e611ffc735e720
ff7c18ad303d4b603ac3f8cff7e611ffc735e720
                e36d0270932c3b657e96b7b0278dfd85dc0fe743
                e36d0270932c3b657e96b7b0278dfd85dc0fe743
              e36d0270932c3b657e96b7b0278df485dc0fe743
ff7c18ad303d4b693ac3f8cff7e61ff6735e720
ff7c18ad303d4b693ac3f8cff7e611ffc735e720
ff7c18ad303d4b693ac3f8cff7e611ffc735e720
ff7c18ad303d4b693ac3f8cff7e611ffc735e720
e36d270932c3b657e96b7b0278dfd85dc0fe743
8
9
10
11
12
13
14
15
16
17
18
                e36d0270932c3b657e96b7b0278dfd85dc0fe743
                ff7c18ad303d4b603ac3f8cff7e611ffc735e720
                ff7c18ad303d4h603ac3f8cff7e611ffc735e720
               F17(188d39304b0938C318C117e611T1C735e720
e36d0270932C3b657e96b7b0278dfd85dc0fe743
e36d0270932C3b657e96b7b0278dfd85dc0fe743
ff7c18ad303d4b603ac3f8cff7e611ffc735e720
               ff7c18ad303d4b603ac3f8cff7e611ffc735e720
               e36d0270932c3b657e96b7b0278dfd85dc0fe743
               e36d0270932c3b657e96b7b0278dfd85dc0fe743
19
20
21
22
23
24
25
26
27
28
              e36dd270932c3b657e96b7b0278dfd85dc0fe743
e56d0270932c3b657e96b7b0278dfd85dc0fe743
ff7c18ad303d4b603ac3f8cff7e611ffc735e720
ff7c18ad303dd4b603ac3f8cff7e61ff7c135e720
ff7c18ad303dd4b603ac3f8cff7e61ff7735e720
ff7c18ad303ddb603ac3f8cff7e61ff7735e720
                ff7c18ad303d4b603ac3f8cff7e611ffc735e720
                ff7c18ad303d4b603ac3f8cff7e611ffc735e720
                ff7c18ad303d4b603ac3f8cff7e611ffc735e726
               ff7c18ad303d4b603ac3f8cff7e611ffc735e720
```

Certification.Contact unit\_of\_measurement

2	03077a1c6bac60e6f514691634a7f6eb5c8 352d0cf7f3e9be14dad7df644ad65efc276		m	
3	352d0cf7f3e9be14dad7df644ad65efc276		m m	
4	03077a1c6bac60e6f514691634a7f6eb5c8		m	
5	03077a1c6bac60e6f514691634a7f6eb5c8		m	
6	352d0cf7f3e9be14dad7df644ad65efc276		m	
7	352d0cf7f3e9be14dad7df644ad65efc276		m	
8	352d0cf7f3e9be14dad7df644ad65efc276		m	
9	352d0cf7f3e9be14dad7df644ad65efc276		m	
-	03077a1c6bac60e6f514691634a7f6eb5c8		m	
11	03077a1c6bac60e6f514691634a7f6eb5c8		m	
12	352d0cf7f3e9be14dad7df644ad65efc276		m	
	352d0cf7f3e9be14dad7df644ad65efc276		m	
	03077a1c6bac60e6f514691634a7f6eb5c8		m	
	03077a1c6bac60e6f514691634a7f6eb5c8		m m	
	352d0cf7f3e9he14dad7df644ad65efc276		m m	
17	352d0cf7f3e9be14dad7df644ad65efc276		m m	
	03077a1c6bac60e6f514691634a7f6eb5c8		m m	
	03077a1c6bac60e6f514691634a7f6eb5c8		m m	
	03077a1c6bac60e6f514691634a7f6eb5c8		m	
21	352d0cf7f3e9be14dad7df644ad65efc276		m	
	352d0cf7f3e9be14dad7df644ad65efc276		m	
23	352d0cf7f3e9be14dad7df644ad65efc276		m	
	352d0cf7f3e9be14dad7df644ad65efc276		m	
25	352d0cf7f3e9be14dad7df644ad65efc276		m	
26	352d0cf7f3e9be14dad7df644ad65efc276		m	
27	352d0cf7f3e9be14dad7df644ad65efc276		m	
28	352d0cf7f3e9be14dad7df644ad65efc276	J5ae2	m	
	altitude_low_meters altitude_high_me			
1		38.0	1488.0	
2		70.0	3170.0	
3		90.0	1000.0	
4		12.0	1212.0	
5		90.0	1250.0	
6		90.0	3000.0	
7		50.0	750.0	
8		40.0	3140.0	
9		90.0	1000.0	
10		90.0	1100.0	
11		95.0	1095.0	
12		90.0	1000.0	
13		50.0	750.0	
14		67.0	1367.0	
15		38.0	1488.0	
16		90.0	1000.0	
17		50.0	750.0	
18		90.0	1600.0	
19	1745.0 17	45.0	1745.0	
20	1200.0 12	90.0	1200.0	
21	750.0 7	50.0	750.0	
22	750.0 7	50.0	750.0	
23	3000.0 30	90.0	3000.0	
24	NaN	NaN	NaN	
25	40.0	40.0	40.0	
26	795.0 79	95.0	795.0	
27	NaN	NaN	NaN	
		NaN	NaN	
28	NaN			
	Nan	IVCLIV		
28	rows x 43 columns]>	IVALIV		

coffee\_df.head(2)

	Species	Owner	Country.of.Origin	Farm.Name	Lot.Number	Mill	ICO.Number	Company	Altitude	Region	•••	Color	Category.Two.Defects	Expirat	
1	Robusta	ankole coffee producers coop	Uganda	kyangundu cooperative society	NaN	ankole coffee producers	0	ankole coffee producers coop	1488	sheema south western		Green	2	June 2	
2	Robusta	nishant gurjer	India	sethuraman estate kaapi royale	25	sethuraman estate	14/1148/2017/21	kaapi royale	3170	chikmagalur karnataka indua		NaN	2	Octo 31st, 2	

2 rows × 43 columns

coffee\_df.tail()

	Species	Owner	Country.of.Origin	Farm.Name	Lot.Number	Mill	ICO.Number	Company	Altitude	Region	 Color	Category.Two.Defects	Expiration	Cer
24	Robusta	luis robles	Ecuador	robustasa	Lavado 1	our own lab	NaN	robustasa	NaN	san juan, playas	 Blue- Green	1	January 18th, 2017	5
25	Robusta	luis robles	Ecuador	robustasa	Lavado 3	own laboratory	NaN	robustasa	40	san juan, playas	 Blue- Green	0	January 18th, 2017	5
26	Robusta	james moore	United States	fazenda cazengo	NaN	cafe cazengo	NaN	global opportunity fund	795 meters	kwanza norte province, angola	 NaN	6	December 23rd, 2015	٤
27	Robusta	cafe politico	India	NaN	NaN	NaN	14-1118- 2014-0087	cafe politico	NaN	NaN	 Green	1	August 25th, 2015	\$
28	Robusta	cafe politico	Vietnam	NaN	NaN	NaN	NaN	cafe politico	NaN	NaN	 None	9	August 25th, 2015	\$

5 rows × 43 columns

coffee\_df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 28 entries, 1 to 28
Data columns (total 43 columns):
# Column Non-Null (
                                                                                                     Non-Null Count Dtype
                   Species
Owner
Country.of.Origin
Farm.Name
Lot.Number
Mill
                                                                                                    28 non-null
28 non-null
28 non-null
25 non-null
                                                                                                                                                             object
int64
                                                                                                    25 non-null
6 non-null
20 non-null
17 non-null
28 non-null
26 non-null
26 non-null
28 non-null
28 non-null
                   Mill
ICO.Number
Company
Altitude
Region
Producer
Number.of.Bags
       Number.of.Bags
Bag.Weight
In.Country.Partner
Harvest.Year
Grading.Date
Owner.1
Variety
                                                                                                    28 non-null
28 non-null
28 non-null
28 non-null
28 non-null
28 non-null
3 non-null
                                                                                                                                                             object
object
int64
object
object
                 Variety
Processing.Method
Fragrance...Aroma
Flavor
Aftertaste
Salt...Acid
Bitter...Sweet
Mouthfeel
Uniform.Cup
                                                                                                    3 non-null

10 non-null

28 non-null

28 non-null

28 non-null

28 non-null

28 non-null

28 non-null
                                                                                                                                                             object
float64
float64
float64
float64
float64
                                                                                                   28 non-null
                                                                                                                                                              float64
float64
                    Clean.Cup
                    Balance
Cupper.Points
Total.Cup.Points
Moisture
                                                                                                                                                              float64
float64
float64
float64
                  Category.One.Defects
Quakers
Color
                                                                                                                                                              int64
                                                                                                      28 non-null
26 non-null
                                                                                                                                                             int64
                                                                                                                                                            object
int64
object
object
object
object
 33 Color 26 non-null
34 Category.Two.Defects 28 non-null
35 Expiration 28 non-null
36 Certification.Body 28 non-null
37 Certification.Address 28 non-null
38 Certification.Contact 28 non-null
39 unit_of_measurement 28 non-null
40 altitude_low_meters 5 non-null
41 altitude_mean_meters 25 non-null
42 altitude_mean_meters 25 non-null
dtypes: float64(15), int64(5), object(23)
memory usage: 9.6+ KB
                                                                                                                                                              float64
                                                                                                                                                              float64
coffee_df.columns[42]
     'altitude_mean_meters'
  col_types = coffee_df.dtypes
  col_types[:5]
   Species
                                                                             object
   0wner
                                                                             object
   Country.of.Origin
Farm.Name
Lot.Number
dtype: object
                                                                             object
object
object
type(col_types[0])
   numpy.dtype[object_]
   for col_name in coffee_df.columns:
    print(col_name[:3])
  Spe
Own
Cou
Far
Lot
Mil
ICO
  Com
Alt
Reg
Pro
Num
Bag
In.
Har
Gra
Own
Var
Pro
Fra
Aft
Sal
Bit
Uni
Cle
Bal
  Cup
Tot
Moi
Cat
Qua
Col
Cat
Exp
Cer
Cer
uni
alt
alt
```

```
my_list = ['honda','ford','nissan']
type(my_list)
             list
my_list[-1]
short_names = [col_name[:3] for col_name in coffee_df.columns]
type(short_names)
             list
short_names
         ['Spe',
'Own',
'Cou',
'Far',
'Lot',
'Mil',
'ICO',
'Com',
'Alt',
'Reg',
'Pro',
'Num',
'Bag',
'In.',
'Own',
'Yar',
'Pro',
'Sal',
'Sal',
'Gra',
'Bal',
'Cul',
'Bal',
'Cul',
'Cat',
'
```

### 3.3. Questions After Class

3.3.1. • will we be gathering our own data or will it all be provided for the course?

### 3.3.2. Will you always give us an answer key for assignments?

No, but we will always give personalized feedback.

### 3.3.3. will we be cleaning data?

Yes, see the notes from the first class.

### 3.3.4. Will we do correlations later?

Yes

### 3.3.5. will we go over more imports like pandas?

Yes, we will use other libraries. In A2, you'll even import your own code.

### 3.3.6. How will datasets be used in conjunction with one another?

We will combine data in a few weeks.

### 3.3.7. How similar of an alternative to modeling in R exists in python?

They are both compelte languages so at some level, they can both do all the same things. Some libraries are easier/better for specific things in one language or the other.

### 3.3.8. Will we be working with databases at all in this class?

A little. We'll pull from a database into python.

### 3.3.9. Are we going going to be using pandas a lot in this class?

Yes. We will use pandas for almost every single remaining class session this semester.

### 3.3.10. how do you short-cut fill variable names in jupyter

Press tab to autocomplete

3.3.11. what is the most efficient way to get help on the homework  $\ensuremath{\textbf{?}}$ 

Accept the assignment and make an issue or go to office hours for general questions. For clarifying questions, you can post an issue on the course website.

If you are stuck with some progress made, upload (or push) your code first and then ask for help so we can see where you are.

### 3.3.12. is there a best or most common way to organize data for use

CSV

### 3.3.13. how would you laod a csv file with python if i was using visual studio instead of jupyter?

All of the code we are writing will work in any python environment that has the libraries. the Editor you use (jupyter vs VSCode vs PyCharm) does not impact what you can do with python. Which interpreter you use can impact and jupyter does default to ipython instead of the core python kernel, but that does not change how to load data.

Remember that jupyter is not on a cloud service. YOu can use any file on your computer with a relative path.

## 4. Pandas and Indexing

### 4.1. Iterable types

```
a = [char for char in 'abcde']
b = {char:i for i, char in enumerate('abcde')}
c = '(a', 'b', 'c', 'd', 'e')
d = 'a b c d e'.split()

a

['a', 'b', 'c', 'd', 'e']

type(a)

list

b

{'a': 0, 'b': 1, 'c': 2, 'd': 3, 'e': 4}

type(b)

dict

c

('a', 'b', 'c', 'd', 'e')

type(c)

tuple

d

['a', 'b', 'c', 'd', 'e']

type(d)

list
```

### 4.2. Reading data other ways

```
import pandas as pd

course_comms_url =
  'https://rhodyprog4ds.github.io/BrownSpring23/syllabus/communication.html'
```

THis reads in from the html directly.

```
pd.read_html(course_comms_url)
```

```
usage platform
                                                                          usage platform
in class prismia
any time
private questions to your assignment
for general questions that can help others
to share resources or ask general questions in... github
matters that don't fit into another category
e-mail
                                                                    1
                                                                                                                                            area \
                                                                                                                                             chat
                                                                                                    download transcript
                                                                          issue on assignment repo
issue on course website
discussion on community repo
to brownsarahm@uri.edu
                                                                   outside of class time this is not monitored cl...
use after class to get preliminary notes eg if...
pugs eg bugs in your code"
eg bugs in your code"
eg what the instructions of an assignment mean...
friedde links in your portfolio
remember to include [CSC310] or [DSP310] (...,
                                                                                                                                                                                      usage
                                                                    0 matters that don't fit into another category to brownsarahm@uri.edu
                                                                    % note of the contraction of th
                                                                    e private questions to your assignment
for general questions that can help others
to share resources or ask general questions in...
                                                                    area \
0 issue on assignment repo
1 issue on course website
2 discussion on community repo
                                                                        eg bugs in your code"
eg what the instructions of an assignment mean...
include links in your portfolio ,
usage area \
in class chat
any time double.
                                                                   0 in class chat
1 any time download transcript
                                                                    0 outside of class time this is not monitored cl...
1 use after class to get preliminary notes eg if... ]
                                                               html_list = pd.read_html(course_comms_url)
                                                               type(html_list)
                                                               type(html_list[0])
                                                                  pandas.core.frame.DataFrame
                                                                [type(h) for h in html_list]
                                                                  [pandas.core.frame.DataFrame,
                                                                     pandas.core.frame.DataFrame, pandas.core.frame.DataFrame,
                                                                      pandas.core.frame.DataFrame]
                                                                  achievements_ur1 =
'https://rhodyprog4ds.github.io/BrownSpring23/syllabus/achievements.html'
                                                               achievements_df_list = pd.read_html(achievements_url)
make a list means use a list comprehension
                                                               [ach.shape for ach in achievements_df_list]
                                                                  [(14, 3), (15, 5), (15, 15), (15, 6)]
                                                                achievements_df_list.shape
                                                                  AttributeError Traceback (most recent call last)
/tmp/ipykernel_1798/4120816115.py in <module>
----> 1 achievements_df_list.shape
                                                                  AttributeError: 'list' object has no attribute 'shape'
                                                                 \label{local_confer_data_url} {\tt coffee\_data\_url} = {\tt 'https://raw.githubusercontent.com/jldbc/coffee-quality-database/master/data/robusta\_data\_cleaned.csv'}
                                                               coffee_df = pd.read_csv(coffee_data_url,index_col=0)
                                                               coffee_df.head(1)
                                                                         Species
                                                                                                         Owner Country.of.Origin Farm.Name Lot.Number
                                                                                                                                                                                                                                                                Mill ICO.Number Company Altitude Region ... Color Category.Two.Defects Expiration Cer
                                                                                                                                                                                  kyangundu
                                                                                                                                                                                                                                                         ankole
                                                                                                                                                                                                                                                                                                                                                                       sheema
                                                                                                                                                                                                                                                                                                                                                                                                                                                                               2 June 26th,
                                                                1 Robusta producers
                                                                                                                                                                                                                                                                                                                            coffee
                                                                                                                                                      Uganda cooperative society
                                                                                                                                                                                                                                   NaN
                                                                                                                                                                                                                                                           coffee
                                                                                                                                                                                                                                                                                                          0
                                                                                                                                                                                                                                                                                                                                                       1488
                                                                                                                                                                                                                                                                                                                                                                         south ... Green
                                                                                                                                                                                                                                                                                                                 producers
                                                                                                                                                                                                                                                                                                                                                                       western
                                                                                                                                                                                                                                                  producers
                                                              1 rows x 43 columns
                                                               coffee_df['Species'].head()
```

get the tables

```
Robusta
Robusta
                   Robusta
                    Robusta
                    Robusta
    Name: Species, dtype: object
type(coffee_df['Species'])
    pandas.core.series.Series
coffee_df.columns
  Index(['Species', 'Owner', 'Country.of.Origin', 'Farm.Name', 'Lot.Number',
    'Mill', 'ICO.Number', 'Company', 'Altitude', 'Region', 'Producer',
    'Number.of.Bags', 'Bag.Weight', 'In.Country.Partner', 'Harvest.Year',
    'Grading.Date', 'Owner.1', 'Variety', 'Processing.Method',
    'Fragrance...Aroma', 'Flavor', 'Aftertaste', 'Salt...Acid',
    'Bitter...Sweet', 'Mouthfeel', 'Uniform.Cup', 'Clean.Cup', 'Balance',
    'Cupper.Points', 'Total.Cup.Points', 'Moisture', 'Category.One.Defects',
    'Quakers', 'Color', 'Category.Two.Defects', 'Expiration',
    'Certification.Body', 'Certification.Address', 'Certification.Contact',
    'unit_of_measurement', 'altitude_low_meters', 'altitude_high_meters',
    'attitude_mean_meters'],
    dtype='object')
                       dtype='object')
coffee_df['Number.of.Bags']
                      300
320
300
320
   1
200
320
320
320
320
320
320
100
1
                      320
300
140
1
20
6
100
250
100
    28 1
Name: Number.of.Bags, dtype: int64
new_values = {0:'<100',1:'100-199',2:'200-299',3:'300+'}
[new_values[int(num/100)] for num in coffee_df['Number.of.Bags']]
   ['300+',
'300+',
'300+',
'300+',
'200-299',
       '200-299',
'300+',
'300+',
'300+',
'300+',
'300+',
'100-199',
       '100-199',
'<100',
'300+',
'300+',
'100-199',
'<100',
'<100',
'200-299',
'100-199',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
```

```
bags_bin = lambda num: int(num/100)
[new_values[bags_bin(num)] for num in coffee_df['Number.of.Bags']]
```

'<100', '<100']

```
['300+',
'300+',
'300+',
'300+',
'<100',
'200-299',
'300+',
'300+',
'300+',
'300+',
'300+',
'300+',
'300+',
'300+',
'100-199',
'<100',
'300+',
'100-199',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<100',
'<10
```

### 4.3. Importing locally

If I make a file in the same folder as my notebook called <code>example.py</code> and then put

```
%%bash
cat example.py
name = 'sarah'
```

in the file, we can use that file like:

from example import name

name

'sarah'

import example

example.name

'sarah'

### 4.4. Questions After Class

4.4.1. why does casting the int over the (num/100) give you the right number? Is it because of floor division?

First let's look at an interim value, lets pick a value for num

```
num = 307
```

Then do the calculation without casting to int

```
1.07 num/100
```

Remember that int type is an integer or whole number, no fraction. So, casting drops the decimal part.

### 4.4.2. How would adding 2 DataFrametogether of separate types affect the type command?

It depends what "add" means. If addition it might error, but if it worked, then it would still be a DataFrame. If stacking with pd.concat it would also be a DatFrame.

If you make them into a list, then the would be a list.

#### 4.4.3. what keys to use in the dictionaries?

In the assignment the instruction say

4.4.4. how to save as a local csv file?

pandas.DataFrame.to\_csv

### 4.4.5. how to create a Dataframe?

Use the constructor

4.4.6. how to read using relative path?

#### 4.4.7. I would like to know about other common forms of data files.

The pandas documentation's **I/O** page is where I recommend starting

#### 4.5. What other libraries do we end up using?

Next week we will use seaborn for plotting. Later in the semester we will use sklearn for machine learning. We will use a few other libaries for a few features, but these three are the main ones.

### 5. Exploratory Data Analysis (EDA)

Again, we import pandas as usual

```
import pandas as pd
```

and loaded the data in again

```
coffee_data_url = 'https://raw.githubusercontent.com/jldbc/coffee-quality-
database/master/data/robusta_data_cleaned.csv'
coffee_df = pd.read_csv(coffee_data_url,index_col=0)
```

### 5.1. Summarizing and Visualizing Data are very important

- · People cannot interpret high dimensional or large samples quickly
- Important in EDA to help you make decisions about the rest of your analysis
- · Important in how you report your results
- . Summaries are similar calculations to performance metrics we will see later
- · visualizations are often essential in debugging models

#### THEREFORE

- You have a lot of chances to earn summarize and visualize
- · we will be picky when we assess if you earned them or not

### 5.2. Describing a Dataset

So far, we've loaded data in a few different ways and then we've examined DataFrames as a data structure, looking at what different attributes they have and what some of the methods are, and how to get data into them.

We can also get more structural information with the info method.

```
coffee_df.info()
  <class 'pandas.core.frame.DataFrame'>
Int64Index: 28 entries, 1 to 28
Data columns (total 43 columns):
# Column Non-Null Count Dtype
                                                                                28 non-null
                Species
                                                                                                                             object
                                                                                28 non-null
                                                                                                                             object
                 Country.of.Origin
                                                                                 28 non-null
                                                                                                                              object
                                                                               28 non-null
25 non-null
6 non-null
20 non-null
17 non-null
28 non-null
               Farm.Name
Lot.Number
Mill
ICO.Number
                                                                                                                             object
object
object
object
                 Company
                                                                                                                             object
                                                                                                                             object
object
object
int64
object
object
                 Altitude
                                                                                 25 non-null
26 non-null
              Region
Producer
Number.of.Bags
Bag.Weight
In.Country.Partner
Harvest.Year
                                                                                26 non-null
28 non-null
28 non-null
28 non-null
                                                                                 28 non-null
28 non-null
                                                                                                                             int64
               Grading.Date
                                                                                                                             object
                                                                               28 non-null
28 non-null
3 non-null
10 non-null
28 non-null
28 non-null
28 non-null
                                                                                                                            object
object
object
float64
float64
               Owner.1
     16
17
18
19
20
21
22
               Variety
Processing.Method
Fragrance...Aroma
Flavor
               Aftertaste
                                                                                                                              float64
              Artertaste
Salt...Acid
Bitter...Sweet
Mouthfeel
Uniform.Cup
Clean.Cup
Balance
Cupper.Points
                                                                                                                              float64
                                                                               28 non-null
                                                                                                                             float64
float64
float64
float64
float64
float64
                Total.Cup.Points
                                                                                 28 non-null
                                                                                                                              float64
              Total.Cup.Points
Moisture
Category.One.Defects
Quakers
Color
Category.Two.Defects
Expiration
Certification.Body
                                                                               28 non-null
28 non-null
28 non-null
28 non-null
26 non-null
28 non-null
28 non-null
                                                                                                                             float64
float64
int64
int64
object
int64
                                                                                                                             object
                                                                                 28 non-null
                                                                                                                             object
object
              Certification.Body
Certification.Address
Certification.Contact
unit_of_measurement
altitude_low_meters
altitude_high_meters
altitude_mean_meters
                                                                                28 non-null
                                                                               28 non-null
28 non-null
28 non-null
25 non-null
25 non-null
25 non-null
                                                                                                                              float64
                                                                                                                             float64
  dtypes: float64(15), int64(5), object(23) memory usage: 9.6+ KB
```

Now, we can actually start to analyze the data itself.

The <code>describe</code> method provides us with a set of summary statistics that broadly describe the data overall.

```
coffee_df.describe()
```

	Number.of.Bags	Harvest.Year	FragranceAroma	Flavor	Aftertaste	SaltAcid	BitterSweet	Mouthfeel	Uniform.Cup	Clean.Cup	Balance	Cupper.Points
count	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000
mean	168.000000	2013.964286	7.702500	7.630714	7.559643	7.657143	7.675714	7.506786	9.904286	9.928214	7.541786	7.761429
std	143.226317	1.346660	0.296156	0.303656	0.342469	0.261773	0.317063	0.725152	0.238753	0.211030	0.526076	0.330507
min	1.000000	2012.000000	6.750000	6.670000	6.500000	6.830000	6.670000	5.080000	9.330000	9.330000	5.250000	6.920000
25%	1.000000	2013.000000	7.580000	7.560000	7.397500	7.560000	7.580000	7.500000	10.000000	10.000000	7.500000	7.580000
50%	170.000000	2014.000000	7.670000	7.710000	7.670000	7.710000	7.750000	7.670000	10.000000	10.000000	7.670000	7.830000
75%	320.000000	2015.000000	7.920000	7.830000	7.770000	7.830000	7.830000	7.830000	10.000000	10.000000	7.830000	7.920000
max	320.000000	2017.000000	8.330000	8.080000	7.920000	8.000000	8.420000	8.250000	10.000000	10.000000	8.000000	8.580000

From this, we can draw several conclusions. FOr example straightforward ones like:

- the smallest number of bags rated is 1 and at least 25% of the coffees rates only had 1 bag
- the first ratings included were 2012 and last in 2017 (min & max)
- the mean Mouthfeel was 7.5
- Category One defects are not very common ( the 75th% is 0)

Or more nuanced ones that compare across variables like

- the raters scored coffee higher on Uniformity.Cup and Clean.Cup than other scores (mean score; only on the ones that seem to have a scale of up to 8/10)
- the coffee varied more in Mouthfeel and Balance that most other scores (the std; only on the ones that seem to have a scale of of up to 8/10)
- there are 3 ratings with no altitude (count of other variables is 28; alt is 25

And these all give us a sense of the values and the distribution or spread fo the data in each column.

We can use the descriptive statistics on individual columns as well.

## 5.2.1. Understanding Quantiles

The 50% has another more common name: the median. It means 50% of the data are lower (and higher) than this value.

We can use the descriptive statistics on individual columns as well.

```
coffee_df['Uniform.Cup'].describe()

count 28.000000
mean 9.904286
std 0.238753
min 9.330000
25% 10.000000
50% 10.000000
10.000000
max 10.000000
Mame: Uniform.Cup, dtype: float64

coffee_df[['Uniform.Cup','Mouthfeel']].describe()
```

	Uniform.Cup	Mouthfeel
count	28.000000	28.000000
mean	9.904286	7.506786
std	0.238753	0.725152
min	9.330000	5.080000
25%	10.000000	7.500000
50%	10.000000	7.670000
75%	10.000000	7.830000
max	10.000000	8.250000

coffee\_df['Aftertaste'].mean()

### 5.3. Individual statistics

We can also extract each of the statistics that the describe method calculates individually, by name. The quantiles are tricky, we cannot just .25%() to get the 25% percentile, we have to use the quantile method and pass it a value between 0 and 1.

```
coffee_df.mean(numeric_only=True)
 Number.of.Bags
Harvest.Year
Fragrance...Aroma
Flavor
                                                168.000000
2013.964286
7.702500
7.630714
 Aftertaste
                                                       7.559643
Salt...Acid
Bitter...Sweet
Mouthfeel
Uniform.Cup
Clean.Cup
                                                      7.657143
                                                      7.675714
7.506786
9.904286
9.928214
 Balance
                                                       7.541786
 Cupper.Points
Total.Cup.Points
                                                       7.761429
                                                    80.868929
Total.Cup.Points
Moisture
Category.One.Defects
Quakers
Category.Two.Defects
altitude_low_meters
                                               80.868929
0.065714
2.964286
0.000000
1.892857
1367.600000
 altitude_high_meters
altitude_mean_meters
                                                1387.600000
                                                1377.600000
 dtype: float64
coffee_df['Flavor'].quantile(.8)
 7.83
```

#### 🔓 further reading

On the <u>documentation page for describe</u> the "① See Also" shows the links to the documentation of most of the individual functions. This is a good way to learn about other things, or find something when you are not quite sure what it would be named. Go to a function thats similar to what you want and then look at the related functions.

7.559642857142856		
coffee_df.head(2)		

	Species	Owner	Country.of.Origin	Farm.Name	Lot.Number	Mill	ICO.Number	Company	Altitude	Region	 Color	Category.Two.Defects	Expirat	
1	Robusta	ankole coffee producers coop	Uganda	kyangundu cooperative society	NaN	ankole coffee producers	0	ankole coffee producers coop	1488	sheema south western	 Green	2	June 2	
2	Robusta	nishant gurjer	India	sethuraman estate kaapi royale	25	sethuraman estate	14/1148/2017/21	kaapi royale	3170	chikmagalur karnataka indua	 NaN	2	Octo 31st, 2	

2 rows × 43 columns

## 5.4. Working with categorical data

There are different columns in the describe than the the whole dataset:

We can get the prevalence of each one with value\_counts

```
Coffee_df['Color'].value_counts()

Green 20
Blue-Green 3
Bluish-Green 2
None 1
Name: Color, dtype: int64
```

## </>Try it Yourself

Note value\_counts does not count the NaN values, but count counts all of the not missing values and the shape of the DataFrame is the total number of rows. How can you get the number of missing Colors?

Describe only operates on the numerical columns, but we might want to know about the others. We can get the number of each value with value\_counts

```
India 13
Uganda 10
United States 2
Ecuador 2
Vietnam 1
Name: Country.of.Origin, dtype: int64
```

Value counts returns a pandas Series that has two parts: values and index

```
coffee_df['Country.of.Origin'].value_counts().values

array([13, 10, 2, 2, 1])

coffee_df['Country.of.Origin'].value_counts().index

Index(['India', 'Uganda', 'United States', 'Ecuador', 'Vietnam'], dtype='object')
```

The max takes the max of the values.

```
coffee_df['Country.of.Origin'].value_counts().max()
13
```

We can get the name of the most common country out of this Series using idmax

```
type(coffee_df['Country.of.Origin'].value_counts())
pandas.core.series.Series
```

Or see only how many different values with the related:

```
coffee_df['Country.of.Origin'].nunique()
```

5

# 5.5. Split-Apply-Combine

So, we can summarize data now, but the summaries we have done so far have treated each variable one at a time. The most interesting patterns are in often in how multiple variables interact. We'll do some modeling that looks at multivariate functions of data in a few weeks, but for now, we do a little more with summary statistics.

For example, how does the flavor ratings relate to the country?

coffee\_df.groupby('Country.of.Origin')['Flavor'].describe()

	count	mean	std	min	25%	50%	75%	max
Country.of.Origin								
Ecuador	2.0	7.625000	0.063640	7.58	7.6025	7.625	7.6475	7.67
India	13.0	7.640769	0.279835	6.83	7.5800	7.750	7.7500	7.92
Uganda	10.0	7.758000	0.197754	7.42	7.6025	7.790	7.8975	8.08
United States	2.0	7.415000	0.120208	7.33	7.3725	7.415	7.4575	7.50
Vietnam	1.0	6.670000	NaN	6.67	6.6700	6.670	6.6700	6.67

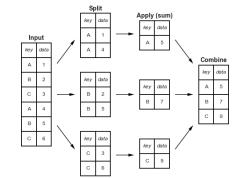
Above we saw which country had the most ratings (remember one row is one rating), but what if we wanted to see the mean number of bags per country?

```
coffee_df.groupby('Country.of.Origin')['Number.of.Bags'].mean()

Country.of.Origin
Ecuador 1.000000
India 230.076923
Uganda 160.900000
United States 50.500000
Vietnam 1.000000
Name: Number.of.Bags, dtype: float64
```

#### Important

This data is only about coffee that was <u>rated by a particular agency</u> it is not economic data, so we cannot, for example conclude which country produces the amount of data. If we had economic dataset, a Number of Bags columns's mean would tell us exactly that, but the context of the dataset defines what a row means and therefore how we can interpret the **every single statistic** we calculate.



What just happened?

Groupby splits the whole dataframe into parts where each part has the same value for <code>country.of.origin</code> and then after that, we extracted the <code>Number.of.Bags</code> column, took the sum (within each separate group) and then put it all back together in one table (in this case, a <code>Series</code> becuase we picked one variable out)

## 5.5.1. How does Groupby Work?

### Important

This is more details with code examples on how the groupby works. If you want to run this code for yourself, use the download icon at the top right to download these notes as a notebook.

We can view this by saving the groupby object as a variable and exploring it.

Trying to look at it without applying additional functions, just tells us the type. But, it's iterable, so we can loop over.

```
for country,df in country_grouped:
    print(type(country), type(df))

<class 'str'> <class 'pandas.core.frame.DataFrame'>
    <class 'str'> <class 'pandas.core.frame.DataFrame'>
    <class 'str'> <class 'pandas.core.frame.DataFrame'>
    <class 'str'> <class 'pandas.core.frame.DataFrame'>
    <class 'str'> <class 'pandas.core.frame.DataFrame'>
```

We could manually compute things using the data structure, if needed, though using pandas functionality will usually do what we want. For example:

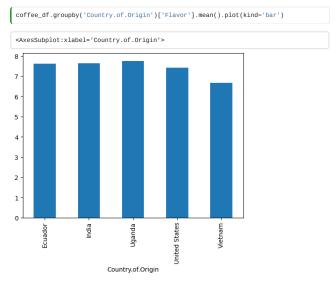
	Number.of.Bags.Sur				
Ecuador	2				
India	2991				
Uganda	1609				
United States	101				
Vietnam	1				

is the same as what we did before

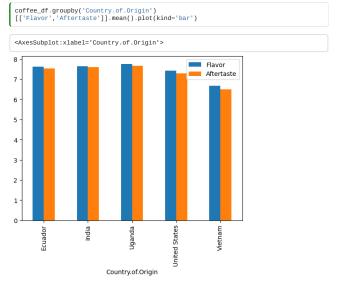
## 5.6. Plotting with Pandas

Pandas allows us to do basic plots on a DataFrame or Series with the plot method.

We want bars so we will use the kind parameter to switch it.



It can also be done on a dataframe like this



Note that it adds a legend for us and uses two colors.

 $\ensuremath{\mbox{\$}}$  What is the default plot type

Try removing the kind=bar and see what it does

# 5.7. Questions after Class

# 5.7.1. Are there any calls for examples such as plot or .describe that you do not want us to use?

Everything in pandas is welcome, unless it is deprecated or not recommended by pandas. Pandas will tell, like the FutureWarning that we saw today. Work will be accepted with warnings, most of the time but it is not best practice and some that I specifically tell you to avoid as we encounter them will not be accepted.

#### Note

I used this feature to build the separate view of the communication channels on this website. You can view that source using the github icon on that page.

#### Not

I tried putting this dictionary into the dataframe for display purposes using the regular constructor and got an error, so I googled about making one from a dictionary to get the docs, which is how I learned about the <a href="from.dict">from.dict</a> method and its orient parameter which solved my problems.

5.7.2. What keyboard key do you use to run the code so I don't have to use the mouse?

hold shift, press enter

5.7.3. How do we take the plots and save them to a separate file?

We will need an additional library to do this, we will do that on Thursday.

#### 5.7.4. how do you display different types of charts

the kind attribute can change to differen types

5.7.5. When you give feedback on an assignment is there a way to fix it to get the points you said it did not meet?

No, you can attempt in the next assignment or portfolio check.

#### 5.7.6. I want to know more about the limitations of pandas

pandas is relatively slow and cannot use accelerated hardware very well. However it is still good to learn because it has gained a lot of traction. So much so that in <a href="modian">modian</a> you can change one line of code to get those advantages.

## 5.7.7. Can Jupyter use other graphics software?

Jupyter notebooks are a file(roughly a json). You can edit it using any text editor. You can also convert to a plain text files using jupytext that is still ruppable

## 5.7.8. How do we generate different models as done in r, also is there a supernova function?

R is designed by and for statisticians. Most of the calculations can be done. They may be slightly more clunky in Python than R.

5.7.9. I had a question on the assignment, in the <u>datasets.py</u> file were we were supposed to save a function handle, should it be a function object or a string?

function object

5.7.10. Besides accepting the invite, is there any more setup we were supposed to do with the achievement tracker repository?

No, that's it.

## 1. Assignment 1: Portfolio Setup, Data Science, and Python

Due: 2023-01-30

Eligible skills: (links to checklists)

- $\bigstar^{[\underline{1}]}$  python <u>level 1</u> and <u>level 2</u>
- ★process<sup>[2]</sup> level 1

### 1.1. Related notes

- Welcome and Introduction
- Syllabus and Python Review

### 1.2. To Do

1 Important

If you have trouble, check the GitHub FAQ on the left before e-mailing  $\,$ 

Your task is to:

- 1. Install required software from the Tools & Resource page
- 2. Create your portfolio, by accepting the assignment
- 3. Learn about your portfolio from the README file on your repository.
- 4. edit \_config.yml to set your name as author and change the logo if you wish
- 5. Fill in about/index.md with information about yourself(not evaluated, but useful) and your own definition of data science (graded for level 1 process)
- 6. Add a Jupyter notebook called grading.ipynb to the about folder and write a function that computes a grade for this course, with the docstring below.
- 7. Add the line file: about/grading in your\_toc.yml file.

Important

Do not merge your "Feedback" Pull Request

## 1.2.1. Docstring

## 1.2.2. Sample tests

Here are some sample tests you could run to confirm that your function works correctly

### Note

If you get stuck on any of this after accepting the assignment and creating a repository, you can create an issue on your repository, describing what you're stuck on and tag us with @rhodyprog4ds/{{ ghinstructors

To do this click Issues at the top, the green "New Issue" button and then type away.

```
assert compute_grade(15,15,15) == 'A'
assert compute_grade(15,15,13) == 'A-'
assert compute_grade(15,14,14) == 'B-'
assert compute_grade(14,14,14) == 'C-'
assert compute_grade(4,3,1) == 'D'
assert compute_grade(15,15,6) == 'B+'
```

#### 1.2.3. Notebook Checklist

- · a Markdown cell with a heading
- your function called compute\_grade
- three calls to your function that verify it returns the correct value for different number of badges that produce at three different letter grades.

#### 1.2.4. Grading Notes:

- a basic function that uses conditionals in python will earn level 1 python
- to earn level 2 python use pythonic code to write a loop that tests your function's correctness, by iterating over a list or dictionary. Remember you will have many chances to earn level 2 achievement in python, so you do not need to do this step for this assignment if you are not sure how.
- [1] skills will be marked like this on the first time they are eligible. There will also be a 🛮 on skills for the last assignment they are eligible
- [2] process is a special skill. You'll earn level 1 in this assignment or a soon one and two in either portfolio 1 or assignments 6-10, then level 3 in portfolio 2,3, or 4

# 2. Assignment 2: Practicing Python and Accessing Data

due: 2023-02-06

## 2.1. Objective & Evaluation

Eligible skills: (links to checklists)

- first chance prepare 1 and 2
- · access 1 and 2
- python 1 and 2
- summarize 1 This assignment is an opportunity to earn level 1 and 2 achievements in python and access and begin working toward level 1 for summarize. You can also earn level 1 for process.

In this assignment, you'll practice/ review python skills by manipulating datasets and extracting basic information about them.

## 2.2. Related notes

- Grading review, Pandas, and Iterables
- Pandas and Indexing

### 2.3. Setting

Next week, we are going to learn about summarizing data. In this assignment, you are going to build a small dataset about datasets. In class next week, we will combine all of your datasets about datasets together in order to be able to answer questions like:

- how much total data did you all load
- how many students picked the same dataset?
- · how many total rows of data did each student load?

## 2.4. Tasks

First, accept the assignment. It contains a notebook with some template structure (and will set you up for grading).

## 2.4.1. Find Datasets

Find 3 datasets of interest to you that are provided in at least two different file formats. Choose datasets that are not too big, so that they do not take more than a few second to load. At least one dataset, must have non numerical (eg string or boolean) data in at least 1 column.

In your notebook, create a markdown cell for each dataset that includes:

- · heading of the dataset's name
- a 1-2 sentence summary of what the dataset contains and why it was collected
- a "more info" link to where someone can learn about the dataset
- 1-2 questions you would like to answer with that dataset.

## 2.4.2. Store them for loading

Create a list of dictionaries in datasets.py, so that there is one dictionary for each dataset. Each dictionary should have the following keys:

url	with the url
short_name	a short name
load function	(the actual function handle) what function should be used to load the data
TORU_I UNCETON	into a pandas.DataFrame.

### 2.4.3. Make a dataset about your datasets

In a notebook called dataset\_of\_datasets.ipynb, import the list of dictionaries from the datasets module you created in the step above. Then iterate over the list of dictionaries, and:

- 1. save it to a local csv using the short name you provided for the dataset as the file name, without writing the index column to the file.
- 2. record attributes about the dataset as in the table below in a list of lists or dictionary  $\frac{1}{2}$
- 3. Use that to create a DataFrame with columns that match the rows of the following table.

#### ▲ Warning

remember the difference between side effects and returns

#### Not

when the value of the expression after assert is True, it will look like nothing happened. assert is used for testing

9 Hint

See they <u>python module docs for examples</u>

name	a short name for the dataset			
source	a url to where you found the data			
num_rows	number of rows in the dataset			
num_columns	number of columns in the dataset			
num_numerical	number of numerical variables in the dataset			

Meta Data Description of the DataFrame to build

#### 2.4.4. Explore Your Datasets

In a second notebook file called exploration.ipynb:

For one dataset that includes nonnumerical data:

- read it in from your local csv using a relative path
- · display the heading and the last 4 rows
- · make a numpy array of only the numerical data and save it to a new variable (select these programmatically)
- was the format that the data was provided in a good format? why or why not?

For any other dataset:

- read it in from your local csv using a relative path
- display the heading with the first three rows
- · display the datatype for each column
- Are there any variables where pandas may have read in the data as a datatype that's not what you expect (eg a numerical column mistaken for strings)? If so, investigate and try to figure out why.

For the third dataset:

- · read it in from your local csv using a relative path
- display the first 3 multiples of 3 rows (eg 3,6,9) of the data for two columns of your choice

## 2.4.5. Exploring data files

Continue in your exploration.ipynb. There are two files in the data folder, both can be read in with read\_csv but need some options or fixing.

- try to read in the german.data file, what happens with the default settings? What option do you need to use to make it look right?
- try to read in the .csv file that's included in the template repository (), use the error messages you get to try to fix the file manually (any text editor, including jupyter can edit a .csv), making notes about what changes you made in a markdown cell.

#### 2.5. Submission

Upload to your repository.

## 2.6. Thinking ahead



This section is not required, but is intended to help you get started thinking about ideas for your portfolio. If you complete it, we'll give your feedback to help shape your ideas to get to level 3 achievements. If you want to focus only on level 2 at this moment in time, feel free to skip this part. You could also think about this after submitting the assignment, since you do not have to get a grade for it. If you want, you could discuss these ideas in office hours.

- 1. When might you prefer one datatype over another?
- 2. How does PEP 8 standard code help you be collaborative?
- 3. Learn about <u>Datasheets for Datasets</u> and find some examples, (eg this google scholar result) How could something like this impact your work as a data scientist?

# 3. Assignment 3: Exploratory Data Analysis

\_\_Due:2023-02-14 \_\_

Template repo for submission



You have the option to work with a partner. You must plan this in advance so that you have access to collaborate

### 3.1. Objective & Evaluation

This week your goal is to do a small exploratory data analysis for two datasets of your choice.

Eligible skills: (links to checklists)

- process 1
- access 1 and 2
- first chance summarize  $\underline{\mathbf{1}}$  and  $\underline{\mathbf{2}}$
- first chance visualize 1 and 2

## 3.2. Related notes

Exploratory Data Analysis (EDA)

### 3.3. Choose Datasets

Each Dataset must have at least three variables, but can have more. Both datasets must have multiple types of variables. These can be datasets you used last week, if they meet the criteria below.

# 3.3.1. Dataset 1 (d1)

must include at least:

- · two continuous valued variables and
- · one categorical variable.



a dataset from the UCI data repository that's for classification and has continuous features would work for this

#### 3.3.2. Dataset 2 (d2)

must include at least:

- two categorical variables and
- · one continuous valued variable

## 3.4. EDA

Use a separate notebook for each dataset, name them dataset\_01.ipynb and dataset\_02.ipynb.

For each dataset, in the corresponding notebook complete the following:

- 1. Load the data to a notebook as a DataFrame from url or local path, if local, include the data file in your repository.
- 2. Explore the dataset in a notebook enough to describe its structure use the heading ## Description
  - shape
  - o columns
  - variable types
  - o overall summary statisics
- 3. Write a short description of what the data contains and what it could be used for
- 4. include overall summary for the data and interpret what that means. Are there limitations? are the variables what you expect?
- 5. Ask and answer 3 questions by using and interpreting statistics and visualizations as appropriate. Include a heading for each question using a markdown cell and H2:##. Make sure your analyses meet the criteria in the check lists below. Use the checklists to think of what kinds of questions would use those type of analyses and help shape your questions.
- 6. Describe what, if anything might need to be done to clean or prepare this data for further analysis in a finale ## Future analysis markdown cell in your notebook.

### 3.4.1. Question checklist

be sure that every question (all six, 3 per dataset) has:

- a heading
- · at least 1 statistic or plot
- · interpretation that answers the question

#### 3.4.2. Dataset 1 Checklist

make sure that your dataset\_01.ipynb has:

- Overall summary statistics grouped by a categorical variable
- A single statistic grouped by a categorical variable
- at least one plot that uses 3 total variables
- $\bullet\,$  a plot and summary table that convey the same information. This can be one statistic or many.

## 3.4.3. Dataset 2 Checklist

make sure that your dataset\_02.ipynb has:

- overall summary statistics
- two individual summary statistics for one variable
- one summary statistic grouped by two categorical variables
- a figure with a grid of subplots that correspond to two categorical variables



Tip

Be sure to start early and use help hours to make sure you have a plan for all of these.

# 3.5. Peer Review



This is optional, but if you do a review, you only need to do one analysis each.

With a partner (or group of 3 where person 1 review's 2 work, 2 reviews 3, and 3 reviews 1) read your partner's notebook and complete a peer review on their pull request. You can do peer review when you have done most of your analysis, and explanation, even if some parts of the code do not work.

In your review:

- Use inline comments to denote places that are confusing or if you see solutions to problems your classmate could not solve
- · Use the template below for your summary review

### 3.5.1. Review Questions

- 1. How was the analysis overall to read? easy? hard? cohesive? jumpy?
- 2. Did the data summaries tell you enough about the data to understand the analysis and anticipate what kinds of questions could be answered? If not, what questions do you still have about the data?
- ${\it 3. Do the questions make sense based on the data? Are they interesting questions? What could improve the questions}\\$
- 4. Are the statistics and plots appropriate for the questions?
- 5. Are the interpretations complete, clear, and consistent with the statistics and plots?
- 6. What could be done to make the explanations more clear and complete?
- 7. What additional analysis might make the analysis more compelling and clear?

```
<!-- delete sections that are not needed -->

## Overall

This analysis was ...

## Data Summaries
- [] complete

To understand this analysis I still need to know ...

## Checklist
- [] questions fit the data
- [] questions are in natural language
- [] chosen statistics and plots match questions
- [] all statistics and plots have an interpretation in English

## Areas of improvement
```

#### 3.5.2. Response

Respond to your review either inline comments, replies, or by updating your analysis accordingly.

#### Think Ahear

- 1. How could you make more customized summary tables?
- 2. Could you use any of the variables in this dataset to add more variables that would make interesting ways to apply split-apply-combine? (eg thresholding a continuous value to make a categorical value)

## Portfolio

/opt/hostedtoolcache/Python/3.7.15/x64/lib/python3.7/site-packages/jpykernel\_launcher.py:26: FutureWarning: Using the level keyword in DataFrame and Series aggregations is deprecated and will be removed in a future version. Use groupby instead. df.sum(level=1) should use df.groupby(level=1).sum().
/opt/hostedtoolcache/Python/3.7.15/x64/lib/python3.7/site-packages/jpykernel\_launcher.py:31: FutureWarning: Using the level keyword in DataFrame and Series aggregations is deprecated and will be removed in a future version. Use groupby instead. df.sum(level=1) should use df.groupby(level=1).sum().

This section of the site has a set of portfolio prompts and this page has instructions for portfolio submissions.

Starting in week 3 it is recommended that you spend some time each week working on items for your portfolio, that way when it's time to submit you only have a little bit to add before submission.

The portfolio is your only chance to earn Level 3 achievements, however, if you have not earned a level 2 for any of the skills in a given check, you could earn level 2 then instead. The prompts provide a starting point, but remember that to earn achievements, you'll be evaluated by the rubric. You can see the full rubric for all portfolios in the <a href="syllabus">syllabus</a>. Your portfolio is also an opportunity to be creative, explore things, and answer your own questions that we haven't answered in class to dig deeper on the topics we're covering. Use the feedback you get on assignments to inspire your portfolio.

Each submission should include an introduction and a number of 'chapters'. The grade will be based on both that you demonstrate skills through your chapters that are inspired by the prompts and that your summary demonstrates that you know you learned the skills. See the <u>formatting tips</u> for advice on how to structure files.

On each chapter(for a file) of your portfolio, you should identify which skills by their keyword, you are applying

You can view a (fake) example in this repository as a pdf or as a rendered website

## **Upcoming Checks**

## Portfolio 1

More information to follow

# Formatting Tips

### ▲ Warning

This is all based on you having accepted the portfolio assignment on github and having a cloned copy of the template. If you are not enrolled or the initial assignment has not been issued, you can view the template on GitHub

Your portfolio is a jupyter book. This means a few things:

- it uses myst markdown
- it will run and compile Jupyter notebooks

This page will cover a few basic tips.

## Managing Files and version

You can either convert your ipynb files to earier to read locally or on GitHub.

The GitHub version means installing less locally, but means that after you push changes, you'll need to pull the changes that GitHub makes.

## To manage with a precommit hook jupytext conversion

change your .pre-commit-config.yaml file to match the following:

```
repos:
    repo: https://github.com/mwouts/jupytext
    rev: v1.10.0 # CURRENT_TAG/COMMIT_HASH
    hooks:
    id: jupytext
    args: [--from, ipynb, --to, myst]
```

#### **A** Warning

This section is not required, but is intended to help you get started thinking about ideas for your portfolio. If you complete it, we'll give your feedback to help shape your ideas to get to level 3 achievements. If you want to focus only on level 2 at this moment in time, feel free to skip this part.

Run Precommit over all the files to actually apply that script to your repo.

```
pre-commit install pre-commit run --all-files
```

If you do git status now, you should have a .md file for each ipynb file that was in your repository, now add and commit those.

Now, each time you commit, it will run jupytext first.

#### To manage with a gh action jupytext conversion

create a file at .github/workflows/jupytext.yml and paste the following:

```
# Only run this when the master branch changes
   push:
      branches:
      - main
# If your git repository has the Jupyter Book within some-subfolder next to
# unrelated files, you can make this run only if a file within that specific
# folder has been modified.
      # paths:
# - some-subfolder/**
\# This job installs dependencies, build the book, and pushes it to 'gh-pages' jobs: jupytext:
      runs-on: ubuntu-latest
      steps:
      - uses: actions/checkout@v2
      # Install dependencies
- name: Set up Python 3.7
uses: actions/setup-python@v1
            python-version: 3.7
      - name: Install dependencies
run: |
   pip install jupytext
- name: convert
         run: |
      run: |
	jupytext */*.ipynb --to myst
	jupytext *.ipynb --to myst
- uses: EndBug/add-and-commit@v4 # You can change this to use a specific version
         with:

# The arguments for the 'git add' command (see the paragraph below for more info)

# Default: '.'
            \# The name of the user that will be displayed as the author of the commit \# Default: author of the commit that triggered the run author_name: Your Name
            \# The email of the user that will be displayed as the author of the commit \# Default: author of the commit that triggered the run author_email: you@uri.edu
             # The local path to the directory where your repository is located. You should use actions/checkout first to set it up
            # Default:
cwd: '.'
            \# Whether to use the --force option on 'git add', in order to bypass eventual gitignores \# Default: false
             force: true
            # Whether to use the --signoff option on `git commit' # Default: false signoff: true
            # The message for the commit
# Default: 'Commit from GitHub Actions'
message: 'convert notebooks to md'
            # Name of the branch to use, if different from the one that triggered the workflow
# Default: the branch that triggered the workflow (from GITHUB_REF)
ref: 'main'
            \mbox{\#} Name of the tag to add to the new commit (see the paragraph below for more info) \mbox{\#} Default: ^{\mbox{\tiny II}}
             tag:
          env:
# This is necessary in order to push a commit to the repo
GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }} # Leave this line unchanged
```

## Organization

The summary of for the part or whole submission, should match the skills to the chapters. Which prompt you're addressing is not important, the prompts are a starting point not the end goal of your portfolio.

### Data Files

Also note that for your portfolio to build, you will have to:

- include the data files in the repository and use a relative path  $\ensuremath{\mathsf{OR}}$
- load via url

using a full local path(eg that starts with ///file:) will not work and will render your portfolio unreadable.

# Structure of plain markdown

Use a heading like this:

```
# Heading of page
## Heading 2
### Heading 3
```

in the file and it will appear in the sidebar.

You can also make text italic or **bold** with either \*asterics\* or \_\_underscores\_\_ with \_one for italic\_ or \*\*two for bold\*\* in either case

## File Naming

It is best practice to name files without spaces. Each chapter or file should have a descriptive file name (with\_no\_spaces) and descriptive title for it.

## Syncing markdown and ipynb files

If you have the precommit hook working, git will call a script and convert your notebook files from the ipynb format (which is json like) to Myst Markdown, which is more plain text with some header information. The markdown format works better with version control, largely because it doesn't contain the outputs

If you don't get the precommit hook working, but you do get jupytext installed, you can set each file to sync.

## Adding annotations with formatting or margin notes

You can either install jupytext and convert locally or upload /push a notebook to your repository and let GitHub convert.

Then edit the .md file with a <u>text editor</u> of your choice. You can run by uploading if you don't have jupytext installed, or locally if you have installed jupytext or jupyterbook.

In your .md file use backticks to mark special content blocks

```
""\{note\}
Here is a note\!
""\{warning\}
Here is a warning\!
""\{tip\}
Here is a tip\!
""\{margin\}
Here is a margin note\!
```

For a complete list of options, see the sphinx-book-theme documentation.

#### Links

Markdown syntax for links

```
[text to show](path/or/url)
```

## Configurations

Things like the menus and links at the top are controlled as settings, in \_config.yml. The following are some things that you might change in your configuration file.

### Show errors and continue

To show errors and continue running the rest, add the following to your configuration file:

```
# Execution settings
execute:
allow_errors : true
```

# Using additional packages

You'll have to add any additional packages you use (beyond pandas and seaborn) to the requirements.txt file in your portfolio.

# FAQ

This section will grow as questions are asked and new content is introduced to the site. You can submit questions:

- via e-mail to Dr. Brown (brownsarahm) or Beibhinn (beibhinn)
- via Prismia.chat during class
- by creating an issue

# Syllabus and Grading FAQ

How much does assignment x, class participation, or a portfolio check weigh in my grade?

There is no specific weight for any activities, because your grade is based on earning achievements for the skills listed in the <a href="skills rubric">skills rubric</a>.

However, if you do not submit (or earn no achievements from) assignments or portfolios, the maximum grade you can earn is a C. If you do not submit (or earn no achievements from) your portfolio, the maximum grade you can earn is a B.

## What time are assignments due?

End of day. I could start grading at any time the next morning. If your work is not there when I start grading it will not be graded, but if it is, I won't check the time stamp.

## Can I submit this assignment late if ...?

Late assignments are not accepted, however, your grade is based on the skills, not the assignments. All skills are assessed in at least two <u>assignments</u>, so missing any one will not hurt your grade. If you need an accommodation because you cannot submit multiple assignments, contact Dr. Brown.

## I don't understand my grade on this assignment

If you have questions about your grade, the best place to get feedback is to reply on the Feedback PR. Either reply directly to one of the inline comments, or the summary.

Be specific about what you think you should have earned and why.

## Git and GitHub

## I can't push to my repository, I get an error that updates were rejected

```
! [rejected] main -> main (fetch first)
error: failed to push some refs to <repository name>
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Your local version and github version are out of sync, you need to pull the changes from github to your local computer before you can push new changes there.

After you run

```
git pull
```

You'll probably have to resolve a merge conflict

## The content I added to my portfolio isn't in the pdf

There was an error in the original \_toc.yml file, change yours to match the following:

uncomment the later lines and add any new files you add.

## My command line says I cannot use a password

GitHub has <u>strong rules</u> about authentication You need to use SSH with a <u>public/private key</u>; HTTPS with a <u>Personal Access Token</u> or use the <u>GitHub CLI auth</u>

### My .ipynb file isn't showing in the staging area or didn't push

.ipynb files are json that include all of the output, including tables as html and plots as svg, so, unlike plain code files, they don't play well with version control.

Your portfolio has \*/\*.ipynb in the .gitignore file, so that these files do not end up in your repository. Instead, you'll convert your notebooks to <a href="Myst Markdown">Myst Markdown</a> with <a href="jupytext">jupytext</a> via a <a href="precommit hook.">precommit hook.</a>

Your portfolio has the code to do this already, what you should do is make sure that pre-commit is installed and then run pre-commit install (see your portfolio's README.md file for more detail)

(See your portions 3 NEADWELING THE IOI THOSE detail)

If this doesn't work, you can follow the alterntive in the porfolio readme.

If that doesn't work, and you have time before the deadline, create an issue to get help.

As a last resort, use the jupyter interface to download (File > Download as > ...)your notebook as .md if avialable or .py if not and then move that file from your Downloads folder to your repository. We'll set up another workflow for future work

## My portfolio won't compile

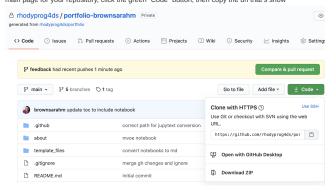
If there's an error your notebook it can't complete running. You can allow it to run if the error is on purpose by changing settings as mentioned on the formatting page.

## Help! I accidentally merged the Feedback Pull Request before my assignment was graded

That's ok. You can fix it.

You'll have to work offline and use GitHub in your browser together for this fix. The following instuctions will work in terminal on Mac or Linux or in GitBash for Windows. (see Programming Environment section on the tools page).

First get the url to clone your repository (unless you already have it cloned then skip ahead): on the main page for your repository, click the green "Code" button, then copy the url that's show



Next open a terminal or GitBash and type the following.

```
git clone
```

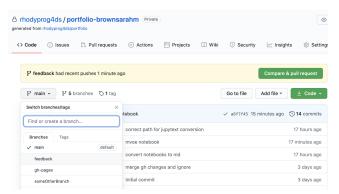
then past your url that you copied. It will look something like this, but the last part will be the current assignment repo and your username.

```
git clone https://github.com/rhodyprog4ds/portfolio-brownsarahm.git
```

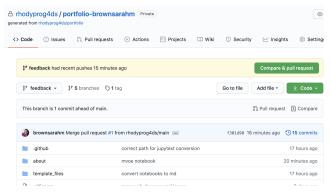
When you merged the Feedback pull request you advanced the feedback branch, so we need to hard reset it back to before you did any work. To do this, first check it out, by navigating into the folder for your repository (created when you cloned above) and then checking it out, and making sure it's up to date with the remote (the copy on GitHub)

```
cd portfolio-brownsarahm
git checkout feedback
git pull
```

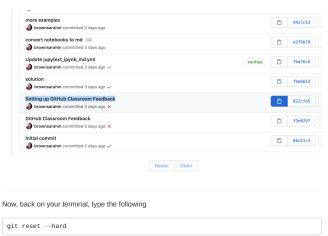
Now, you have to figure out what commit to revert to, so go back to GitHub in your browser, and swithc to the feedback branch there. Click on where it says main on the top right next to the branch icon and choose feedback from the list.



Now view the list of all of the commits to this branch, by clicking on the clock icon with a number of commits



On the commits page scroll down and find the commit titled "Setting up GitHub Classroom Feedback" and copy its hash, by clicking on the clipboard icon next to the short version.



```
git reset --hard
```

then paste the commit hash you copied, it will look something like the following, but your hash will be different.

```
git reset --hard 822cfe51a70d356d448bcaede5b15282838a5028
```

If it works, your terminal will say something like

```
HEAD is now at 822cfe5 Setting up GitHub Classroom Feedback
```

but the number on yours will be different.

Now your local copy of the feedback branch is reverted back as if you had not merged the pull request and what's left to do is to push those changes to GitHub. By default, GitHub won't let you push changes unless you have all of the changes that have been made on their side, so we have to tell Git to force GitHub to do this.

Since we're about to do something with forcing, we should first check that we're doing the right thing.

```
git status
```

and it should show something like

```
On branch feedback
Your branch is behind 'origin/feedback' by 12 commits, and can be fast-forwarded.
 (use "git pull" to update your local branch)
```

Your number of commits will probably be different but the important things to see here is that it says On branch feedback so that you know you're not deleting the main copy of your work and Your branch is behind origin/feedback to know that reverting worked.

Now to make GitHub match your reverted local copy.

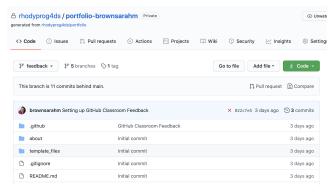
```
git push origin -f
```

and you'll get something like this to know that it worked

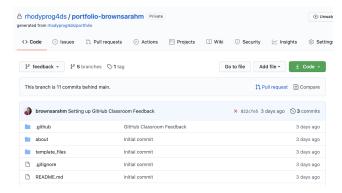
```
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/rhodyprog4ds/portfolio-brownsarahm.git
+ f301d90...822cfe5 feedback -> feedback (forced update)
```

Again, the numbers will be different and it will be your url, not mine.

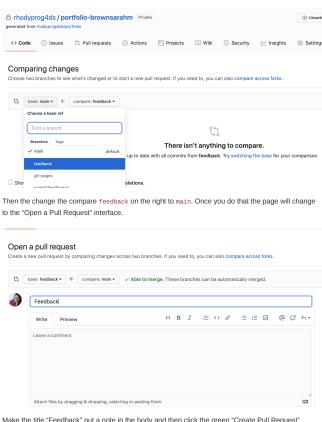
Now back on GitHub, in your browser, click on the code tab. It should look something like this now. Notice that it says, "This branch is 11 commits behind main" your number will be different but it should be 1 less than the number you had when you checked git status. This is because we reverted the changes you made to main (11 for me) and the 1 commit for merging main into feedback. Also the last commit (at the top, should say "Setting up GitHub Classroom Feedback").



Now, you need to recreate your Pull Request, click where it says pull request.



It will say there isn't anything to compare, but this is because it's trying to use feedback to update main. We want to use main to update feedback for this PR. So we have to swap them. Change base from main to feedback by clicking on it and choosing feedback from the list.



Make the title "Feedback" put a note in the body and then click the green "Create Pull Request" button.

Now you're done!

If you have trouble, create an issue and tag @@rhodyprog4ds/fall22instructors for help.

## Code Errors

# **Key Error**

If you get a key error for a pandas operation, it means that the column name as you typed it is not in the DataFrame. Check the spelling, leading or trailing whitespace can be especially troubling.

### <bul>bound method

You're probably missing () on a method, so Python returned the method itself as an object instead of calling it and returning the output.

# Glossary



Contribute glossary items and links for further reading using the suggest an edit button behind the GitHub menu at the top of the page.

### <u>aggregate</u>

to combine data in some way, a function that can produce a customized summary table

a function that's defined on the fly, typically to lighten syntax or return a function within a function. In python, they're defined with the lambda keyword.

#### BeautifulSoup

a python library used to assist in web scraping, it pulls data from html and xml files that can be parsed in a variety of different ways using different methods.

#### conditional

a logical control to do something, conditioned on something else, for example the if, elif else

#### corpus

(NLP) a set of documents for analysis

#### **DataFrame**

a data structure provided by pandas for tabular data in python.

#### dictionary

(data type) a mapping array that matches keys to values. (in NLP) all of the possible tokens a model knows

#### document

unit of text for analysis (one sample). Could be one sentence, one paragraph, or an article, depending on the goal

#### ah

GitHub's command line tools

#### git

a version control tool; it's a fully open source and always free tool, that can be hosted by anyone or used without a host, locally only

#### GitHub

a hosting service for git repositories

#### index

(verb) to index into a data structure means to pick out specified items, for example index into a list or a index into a data frame. Indexing usually invovlees square brackets [] (noun) the index of a dataframe is like a column, but it can be used to refer to the rows. It's the list of names for the rows.

#### interpreter

the translator from human readable python code to something the computer can run. An interpreted language means you can work with python interactively

#### iterate

To do the same thing to each item in an <u>iterable</u> data structure, typically, an iterable type. Iterating is usually described as iterate over some data structure and typically uses the for keyword

#### iterable

any object in python that can return its members one at a time. The most common example is a list, but there are others.

### kerne

in the jupyter environment,  $\underline{\text{the kernel}}$  is a language specific computational engine

### lambda

they keyword used to define an anonymous function; lambda functions are defined with a compact syntax <name> = lambda <parameters>: <body>

### PEP 8

Python Enhancement Proposal 8, the Style Guide for Python Code.

### repository

a project folder with tracking information in it in the form of  $\boldsymbol{a}$  .git file

### suffix

additional part of the name that gets added to end of a name in a merge operation  $% \left( 1\right) =\left( 1\right) \left( 1\right)$ 

### <u>Series</u>

a data structure provided by pandas for single columnar data with an index. Subsetting a Dataframe or applying a function to one will often produce a Series

## Split Apply Combine

a paradigm for splitting data into groups using a column, applying some function(aggregation, transformation, or filtration) to each piece and combinging in the individual pieces back together to a single table

### stop words

Words that do not convey important meaning, we don't need them (like a, the, an,). Note that this is context dependent. These words are removed when transforming text to numerical representation

## test accuracy

percentage of predictions that the model predict correctly, based on held-out (previously unseen) test data

### **Tidy Data Format**

Tidy data is a database format that ensures data is easy to manipulate, model and visualize. The specific rules of Tidy Data are as follows: Each variable is a column, each row is an observation, and each observable unit is a table.

### token

a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing (typically a word, but more gneeral)

### TraceBack

an error message in python that traces back from the line of code that had caused the exception back through all of the functions that called other functions to reach that line. This is sometimes call tracing back through the stack

#### training accuracy

percentage of predictions that the model predict correctly, based on the training data

#### Web Scraping

the process of extracting data from a website. In the context of this class, this is usually done using the python library beautiful soup and a html parser to retrieve specific data.

# References on Python

## Official Documentation

- Python
- Pandas
- Matplotlib
- Seaborn

# **Key Resources**

- Course Text this book roughly covers things that we cover in the course, but since things change quickly, we don't rely on it too closely
- Real Python this site includes high quality tutorials
- Towards Data Science this blog has some good tutorials, but old ones are not always updated, so always check the date and don't rely too much on posts more than 2 years old.



If you find other high quality, reliable sources that you want to share, you can earn ram tokens.

## Cheatsheet

Patterns and examples of how to use common tips in class

#### How to use brackets

symbol	use
[val]	indexing item val from an object; val is int for iterables, or any for mapping
[val : val2]	slicing elemtns val to val2-1 from a listlike object
[ item1,item2 ]	creating a list consisting of item1 and item2
(param)	function calls
(item1,item2)	defining a tuple of item1 and item2
{item1,item2}	defining a set of item1 and item2
{key:val1, key2: val2}	defining a dictionary where key1 indexes to val2

## Axes

First build a small dataset that's just enough to display

```
data = [[1,0],[5,4],[1,4]]
df = pd.DataFrame(data = data,
    columns = ['A','B'])
df
```

This data frame is originally 3 rows, 2 columns. So summing across rows will give us a <u>Series</u> of length 3 (one per row) and long columns will give length 2, (one per column). Setting up our toy dataset to not be a square was important so that we can use it to check which way is which.

## Indexing

df['A'][1]	
5	
df.iloc[0][1]	
0	

## **Data Sources**

This page is a semi-curated source of datasets for use in assignments. The different sections have datasets that are good for different assignments.

## Best for loading directly into a notebook

- <u>Tidy Tuesday</u> inside the folder for each year there is a README file with list of the datasets. These are .csv files
- Json Datasets
- National Center for Education Statistics Digest 2019 These data tables are available for download as excel and visible on the page.
- · Lots of wikipedia pages have tables in them.

## **General Sources**

These may require some more work

- Stackoverflow Developer Survey. This data comes with readme info all packaged together in a .zip. You'll need to unzip it first.
- · Google Dataset Search
- . Kaggle most Kaggle datasets will require you to download and unzip them first and then you can copy them into your repo folder.
- . UCI Data Repository Machine Learning focused datasets, can filter by task
- <u>A curated list of datasets by task</u> It includes datasets for cleaning, visualization, machine learning, and "data analysis" which would align with EDA in this course.
- Hugging Face NLP Datasets lots of text datasets

## Datasets in many parts

- Makeup Shades
- Kenya Census
- Wealth and Income over time
- UN Votes
- Deforestation
- Survivor
- <u>Billboard</u>
- Caribou Tracking
   Video games from steam 2021 and from 2019
- BBC Rap Artists

# Datasets with time

Superbowl commercials

## **Databases**

SQLite Databases

If you have others please share by creating a pull request or issue on this repo (from the GitHub logo at the top right, suggest edit).

# General Tips and Resources

This section is for materials that are not specific to this course, but are likely useful. They are not generally required readings or installs, but are options or advice I provide frequently.

## on email

how to e-mail professors

# How to Study in this class

This is a programming intensive course and it's about data science. This course is designed to help you learn how to program for data science and in the process build general skills in both programming and using data to understand the world. Learning two things at once is more complex. In this page, I break down how I expect learning to work for this class.



Remember the goal is to avoid this

Learning to program requires iterative practice. It does not require memorizing all of the specific commands, but instead learning the basic patterns. Using reference materials frequently is a built in part of programming, most languages have built in help as a part of the language for this reason. This course is designed to have you not only learn the material, but also to build skill in learning to program. Following these guidelines will help you build habits to not only be successful in this class, but also in future programming.

A new book that might be of interest if you find programming classes hard is the Programmers Brain As of 2021-09-07, it is available for free by clicking on chapters at that linked table of contents section.

Where are your help tools?

In Python and Jupyter notebooks, what help tools do you have?

#### Learning in class



My goal is to use class time so that you can be successful with minimal frustration while working outside of class time.

Programming requires both practical skills and abstract concepts. During class time, we will cover the practical aspects and introduce the basic concepts You will get to see the basic practical details and real examples of debugging during class sessions. Learning to debug something you've never encountered before and setting up your programming envrionment, for example, are high frustration activities, when you're learning, because you don't know what you don't know. On the other hand, diving deeper into options and more complex applications of what you have already seen in class, while challenging, is something I'm confident that you can all be successful at with minimal frustration once you've seen basic ideas in class. My goal is that you can repeat the patterns and processes we use in class outside of class to complete assignments, while acknowledging that you will definitely have to look things up and read documentation outside of class

Each class will open with some time to review what was covered in the last session before adding new material.

To get the most out of class sessions, you should have a laptop with you. During class you should be following along with Dr. Brown, typing and running the same code. You'll answer questions on Prismia chat, when you do so, you should try running necessary code to answer those questions. If you encounter errors, share them via prismia chat so that we can see and help you

#### After class

After class, you should practice with the concepts introduced.

This means reviewing the notes: both yours from class and the annotated notes posted to the course website

When you review the notes, you should be adding comments on tricky aspects of the code and narrative text between code blocks in markdown cells. While you review your notes and the annotated course notes, you should also read the documentation for new modules, libraries, or functions introduced that day

In the annotated notes, there will often be extra questions or ideas on how to extend and practice the concepts. Try these out.

If you find anything hard to understand or unclear, write it down to bring to class the next day.

#### Assignments

In assignments, you will be asked to practice with specific concepts at an intermediate level. Assignments will apply the concepts from class with minimal extensions. You will probably need to use help funcitons and read documentation to complete assignments, but mostly to look up things you saw in class and make minor variations. Most of what you need for assignments will be in the class notes, which is another reason to read them after class.

### **Portfolios**

In portfolios, your goal is to extend and apply the concepts taught in class and practiced in assignments to solve more realistic problms. You may also reflect on your learning in order to demonstrate deep understanding. These will require significant reading beyond what we cover in class.

## Getting Help with Programming

## **Asking Questions**



One of my favorite resources that describes how to ask good questions is this blog post by Julia Evans, a developer who writes comics about the things she learns in the course of her work and publisher of wizard zines

### Describing what you have so far

Stackoverflow is a common place for programmers to post and answer questions.

As such, they have written a good guide on creating a minimal, reproducible example

Creating a minimal reproducible example may even help you debug your own code, but if it does not, it will definitely make it easier for another person to understand what you have, what your goal is, and what's working.

# **Understanding Errors**

Error messages from the compiler are not always straight forward.

The <u>TraceBack</u> can be a really long list of errors that seem like they are not even from your code. It will trace back to all of the places that the error occurred. It is often about how you called the functions from a library, but the compiler cannot tell that.

To understand what the traceback is, how to read one, and common examples, see this post on Real Python.

One thing to try, is friendly traceback a python package that is designed to make that error message text more clear and help you figure out what to do next.



## Terminals and Environments

### Why all this work?

Managing environments is **one of the hardest parts of programming** so, as instructors, we often design our courses around not having to do it. In this class, however, I'm choosing to take the risk and help you all through beginning to manage your own environments.

These issues will be the most painful in the course, I promise.

I think it's worth this type of pain though, because all fo the code you ever run must run in some sort of environment. By giving you control, I'm hoping to increase your indepence as a programmer. This also means responsibility and some messy debugging, but I think this is a good tradeoff. This is an upper level (300+) level course, so increasing some complexity is expected and I want as much as possible to keep you close to realisitc programming environments; so that what you see in this course is directly, and immediately, applicable in real world contexts. You should be able to pick up data science side projects or an internship with ease after this course.

I know some of these things will be frustrating at times, but I want you to feel supported in that and know that your grade will not be blocked by you having environment issues, as long as you ask for help in a timely matter.

### Windows

Windows has a sort of multiverse of terminal environments.

The least setup required involves using anaconda prompt and conda to manage you python environment and GitBash to work with git (and it can also do other bash related things).

Instead of managing two terminals, you may configure your path in GitBash to make Anaconda work

## MacOS

MacOS has one terminal app, but it can run different shells.

On MacOS You may want to switch to bash (using the bash command or make it your default and update bash.

# Getting Organized for class

The only **required** things are in the Tools section of the syllabus, but this organizational structure will help keep you on top of what is going on.

Your username will be appended to the end of of the repository name for each of your assignments in class.

### File structure

I recommend the following organization structure for the course:

```
CSC310
|- notes
|- portfolio-username
|- 02-accessing-data-username
|- ...
```

This is one top level folder will all materials in it. A folder inside that for in class notes, and one folder per repository.

Please do not include all of your notes or your other assignments all inside your portflio, it will make it harder to grade.

## Finding repositories on github

Each assignment repository will be created on GitHub with the rhodyprog4ds organization as the owner, not your personal acount. Since your account is not the owner, they do not show on your profile.

Your assignment repositories are all private during the semester. At the end, you may take ownership of your portfolio[^pttrans] if you would like.

If you go to the main page of the organization you can search by your username (or the first few characters of it) and see only your repositories.

# **▲** Warning

Don't try to work on a repository that does not end in your username; those are the template repositories for the course and you don't have edit permission on them.

#### Note

A fun version of this is rubber duck debugging

#### Note

We know that we don't currently teach a lot of this in our department, so in Spring 22 I'm teaching a brand new course on Computer Systems, that will help you understand the underlying concepts that make all of this stuff make sense, instead of just following recipes and debugging here and there.

If, for example, you come to me in week 5 and have never got an any environment working and you're trying for the first time, your grade will be hurt because you will be very far behind at that point Ask for help early and often. This section is a place for students enrolled in Fall 2020 to write letters to future students taking this class with Professor Brown. The websites for future sections will link back here for them to read.

# **Contributed Notes**

## Reviewing notes

Especially when it comes to the difficult topics make sure you go back through the notes and try and add to them yourself. Actively using the new topics will help you learn a lot better than just copying the notes.

## Attend Office Hours

Attending office hours will help you better understand course material, complete homework assignments, and learn new things that might not normally come up in class.

• David

# To contribute

Via GitHub directly:

- 1. Use the edit button above to add a note to this file following the example that's commented out
- 2. create a pull request

By Professor Sarah M Brown © Copyright 2022.