Introduction to Programming with Data

Fall 2017

Instructor: Katharine Jarmul
Email: kjarmul@jou.ufl.edu
Twitter: @kjam

About the Instructor: Katharine Jarmul is a data scientist and educator based in Berlin, Germany. Originally from Los Angeles, California, she first began working with Python for data analysis in 2008 at the Washington Post. Since then, she has worked at large and small companies working primarily on data extraction, cleaning and insights. She co-authored the O'Reilly book *Data Wrangling with Python* and has a M.A. in journalism and a M.S. in education.

Office Hours: Office hours will be held virtually via Slack (https://slack.com/is) as necessary. In order to schedule office hours, please email or message me privately -- give at least 48 hours notice if possible and include at least two available meeting times (for your schedule) and appropriate time zone details. I will coordinate several live hangout sessions for asking questions in a group discussion and post the schedule for those on the course website. They are not mandatory but will be coordinated to accommodate several different time zones. I reside in Berlin, Germany (CET), so please allow up to 24 hours for proper receipt and response of your messages.

Course Website: http://elearning.ufl.edu

Course Communication: For most communication, the course will revolve around our Slack team, with a mixture of group channels for things like #help and more general chats hosted in channels like #general. If your question is more involved than a simple chat message (i.e. more than one paragraph of text), please use email instead (kjarmul@jou.ufl.edu). Be sure to include the course number and a relevant topic in the title of the email.

Course Description: Introduction to Programming with Data provides a hands-on overview of how to program for data analysis. With the help of Python, students will learn how to write code for easy collection, analysis and sharing of data. The course offers an introduction to programming best

practices, while quickly getting started with practical data evaluation tasks like tabular reporting and data visualization techniques.

Course Objectives:
By the end of this course, students will:

- Create Python scripts to fix a problem, such as how to automatically send emails, how to generate CSV reports and how to scrape a website for data.
- Evaluate data visualization techniques and determine best practices for sharing data visually.
- Analyze data in tabular format using Pandas and NumPy.
- Assemble SQL queries for data extraction from a database.
- Identify statistical methods of data analysis and describe why they are useful and significant.
- Use Jupyter Notebook to share Python skills and data analysis techniques.
- Develop tests to evaluate Python functions.
- Define Python data types and several methods available to each type.
- Execute Python scripts via Jupyter Notebook and the shell or command prompt.
- Examine text data using basic Natural Language Processing techniques such as bag-of-words.
- Evaluate an API for use and execute commands against that API using the Python requests library.

Course Goal:
Why is this course important? Learning how to program is an important skill for those who wish to harness data at their fingertips or manage others who do so. Python is a popular language for data science objectives and is easy to learn. Whether or not students plan on using their programming skills at work or even becoming a data scientist, it's essential to understand what is both easy and hard to accomplish using programming if a person is to oversee or interact with technical teams. Finally, being self-sufficient and

able to answer their own data questions with some insight will help keep students stay ahead of the curve and somewhat independent when managing projects or fulfilling data-related tasks.

Expectations:
In order to cover the wide variety of topics included in this course, students will need to apply themselves thoroughly to the coursework and bring a willingness to try new things and a curiosity for data. In this course, students will apply self-guided learning techniques, such as how to debug without an expert by their side and how to use StackOverflow and group chats to solve programming problems. Throughout the course, students will be expected to ask questions and help others who are stuck. These are great skills beyond the scope of programming and will help students succeed in most data analysis tasks they perform or advise in the workplace.

This course requires students to perform a pre-class assessment and have a laptop with an operating system which allows them to install applications and programs (i.e. Administrative access). If you are running Windows, you will need Windows Vista or later. If you are running OS X, you will need 10.8 or later (Mountain Lion). If you are running Linux, please insure you can install Python 3.

Ownership Education:

As graduate students, you are not passive participants in this course. This class allows you to not only take ownership of your educational experience but to also provide your expertise and knowledge in helping your fellow classmates. The Canvas shell will have an open Q&A thread where you should pose questions to your classmates when you have a question as it relates to an assignment or an issue that has come up at work. Your classmates along with your instructor will be able to respond to these questions and provide feedback and help. The same applies to the course Slack team channels. This open communication and accountability also allows everyone to gain the same knowledge in one location rather than the

instructor responding back to just one student which limits the rest of the class from gaining this knowledge.

Required Text: Data Wrangling with Python by Jacqueline Kazil and Katharine Jarmul

Required Installations: You will need to have Python and several other libraries installed on your computer. I will also provide a shared server for some exercises (i.e. quizzes and tests); but it is highly recommended you set up your local computer to run all programs for testing, project work and your own use. If you have not used Python before, I recommend following the Python 3.6.X installation instructions here:

- MacOSX (https://www.python.org/downloads/mac-osx/)
- Windows (https://www.python.org/downloads/windows/)
- Other Platforms (https://www.python.org/download/other/)

If you have never used Python for data science before, I also ask that you install anaconda (https://www.continuum.io/downloads) for managing packages and different Python versions.

To properly install Python 3.6+, here are some outlines for each operating system:

- Windows Vista or later
- Apple OS X 10.8 or later (Mountain Lion)
- Linux: Please ensure you can install via normal package manager (or builds)

If you run into trouble during any installation, feel free to email -- however, I encourage you to first try searching and solving your problem. Becoming more familiar with the inner workings of your computer and how to fix computer problems is a great first step in learning to program and a skill you will hone throughout this course.

Additional Readings:
Listed in the course schedule and in each weekly module on Canvas

Prerequisite knowledge and skills:
Students should have intermediate knowledge of how to install and debug

programs on their own computers. We will be practicing these skills often throughout the course, so it's okay to be a bit slow at first. I encourage students to as soon as possible try walking through the Python and anaconda installations covered in the *Required Installations* section. This will be good practice for getting to know a bit more about your computer.

As part of the pre-course training, you should step through one of the following introduction to Python video courses that are free online. This will help you get a jumpstart on the course and allow some things to be easier repetition and review rather than immediately diving in with no background. If you have already programmed with Python before, you can skip this requirement.

- CodeAcademy: https://www.codecademy.com/courses/introduction-to-python-6WeG3/0/1
- DataCamp: https://www.datacamp.com/courses/intro-to-python-for-data-science
- Khan Academy: https://www.youtube.com/watch?v=husPzLE6sZc&list=PL36E7A2B75028A3D6

In addition, if you'd like to learn a little more about the command line, which I have found incredibly useful for debugging and working better with my computer and computer internals, I recommend taking a look at these resources:

- Windows MS-DOS: https://www.youtube.com/watch?v=MNwErTxfkUA
- Windows PowerShell (Windows 8.0+): https://www.youtube.com/watch?v=IHrGresKu2w (I only recommend watching this if you already know DOS)
- Unix-based (Mac, Linux): https://www.codecademy.com/en/courses/learn-the-command-line
- More in-depth Unix-based (bash): https://github.com/Idnan/bash-guide

Teaching Philosophy:

As a primarily self-taught developer, I strongly favor practice and project-based learning for computer science. Throughout this course, we will touch upon the deeper theories and academic approaches to data science and computing; however, the course will have a strong emphasis on practical use cases and projects. I believe this allows you to quickly apply and excel

at programming to help you do your work; while still allowing for questions, growth and curiosity towards the academic field. This approach will be reflected in our “Weekly Readings” which will blend the research in the field with the daily applications.

Instructional Methods:
This course will involve several different instructional methods as a way to address different learning styles and approaches. If you find your learning style is not adequately addressed, please feel free to offer feedback via email at any time. The methods are as follows:

- Video lectures
- Required readings
- Online quizzes and tests
- Discussion threads, posting and voting
- Asynchronous group discussions (via Slack)
- Coding projects (alone and in small assigned groups)
- Peer and self-assessments and code reviews

Course Policies:

Attendance Policy:
Due to its online nature, the course will not have in-person meetings or attendance in a classic sense. Students are required to:

- Check Slack for regular updates at least twice a week
- Check the course discussion board and participate in postings, voting and discussion threads at least once a week
- View and read all required content by the due date
- Send required projects in by their respective due dates
- Respond to group coordination and emails in 24 hours or less

If a student fails to meet the above attendance requirements, there will be a reduction in the participation portion of the student’s grade. I encourage students to install necessary applications (such as Slack) on their mobile device or set up alerting to ensure you can promptly respond to fellow students and instructor messages without long delays.

Late Work and Make-up Policy:
Deadlines are critical to this class. All work is due on or before the due date. Pre-approved extensions for deadlines will only be permitted for

emergencies. Minor inconveniences such as technical issues, family vacation or minor illness are not valid reasons for extensions. With this in mind there will be penalties for late work. NO LATE ASSIGNMENTS WILL BE ACCEPTED FOR FULL CREDIT without prior arrangements that are acceptable to the instructor, unless the lateness is due to an excused absence such as illness or catastrophic emergency that can be documented. This is true for all assignments, discussion boards, papers, case studies, etc. Late penalties are as follows:

Assignments <= one hour late:	15% penalty
Assignments > an hour late, but <= 12 hours late:	25% penalty
Assignments > 12 hours late, but <= 24 hours late:	50% penalty.
Assignments > 24 hours late, but <= 48 hours late:	70% penalty.
Assignments > 48 hours late:	0 points (no credit, or 100% penalty).

If you have an emergency or pre approved schedule when you will be unavailable to complete assignments (such as offline or limited access to internet), you must let the instructor know at least 5 days in advance.

Requirements for class attendance and make-up exams, assignments, and other work in this course are consistent with university policies that can be found in the online catalogue at:
https://catalog.ufl.edu/ugrad/current/regulations/info/attendance.aspx

Emergency and extenuating circumstances policy: Students who face emergencies, such as a major personal medical issue, a death in the family, serious illness of a family member, or other situations beyond their control should notify their instructors immediately.
Students are also advised to contact the Dean of Students Office if they would like more information on the medical withdrawal or drop process: https://www.dso.ufl.edu/care/medical-withdrawal-process/ .
Students MUST inform their academic advisor before dropping a course, whether for medical or non-medical reasons. Your advisor will assist with notifying professors and go over options for how to proceed with their classes. Your academic advisor is Tiffany Robbert, and she may be reached at trobbert@jou.ufl.edu.

Coursework:

Most non-coding coursework will be submitted via Canvas. There are several other services we will use throughout the course for submissions, including:

- Coding Projects — Github
- Code Reviews — Github
- Quizzes / Tests — Canvas & Jupyter Server

The weekly coursework deadlines and where to submit work will be posted to Canvas and updated as needed throughout the course.

Deadlines:
This class, like others, involves many deadlines. Here is a reminder. The new lecture starts on Mondays:

Code Practice & Quiz	11 AM EST Wednesdays the week assigned
Course Discussions	11 AM EST Thursdays the week assigned
Assignments	11 AM EST Fridays the week assigned
Final Coding Project	11 AM EST last Wednesday of the semester

Grading:
Your work will be evaluated according to this distribution. There will be opportunities in some of the assignments for extra credit. Those opportunities will only count for students with regular on-time completion of other assignments (i.e. >= 75% of work turned in on-time and complete).

- Reading Reactions / Chat Participation — 20%
- Quizzes — 10%
- Assignments — 35%
- Final Project — 35%

The final grade will be awarded as follows:

A	100%	to	93%
A-	< 92%	to	90%
B+	< 90%	to	87%
B	< 87%	to	83%
B-	< 82%	to	80%
C+	< 80%	to	77%
C	< 77%	to	73%
C-	< 72%	to	70%

D+	< 70%	to	67%
D	< 67%	to	63%
D-	< 62%	to	60%
F	< 59%	to	0%

https://catalog.ufl.edu/ugrad/current/regulations/info/grades.aspx

I will round grades to the nearest half-percent, meaning a 92.50-92.99 will result in an A; whereas a 92.00-92.49 would remain an A-.

Weekly Lectures:
This course will have weekly video lectures shared via Canvas. These videos will be a mixture of content produced by the instructor, as well as other free online videos that explain and demonstrate the course content for the week. I ask that you watch all videos and complete all reading before attempting the quiz and assignment content -- even if the content is review for you. It's likely there are some tips and other pointers covered that you will be assessed on at a future point.

In addition to the video lectures, there will be between 1-3 optional live lectures where we will meet in a live setting. This could be to share a conference or meetup talk I find engaging, or simply to have a group discussion on several topics and a demonstration of a tool via shared conversation. I will have these live events recorded for those that cannot attend in real-time.

It is highly recommended you watch and read each module in order. I have arranged the topics so they build upon one another and reference previously covered content. This will help your learning progress intuitively. If you find yourself asking numerous unanswered questions, keep track of them. If they remain unanswered at the end of the module, please post them to the module discussion board or group chat. This helps me know you are watching and following along and allows me and your fellow students to engage and react by sharing our thoughts, answers and related questions.

Assignments:

Course Discussion:

Each week we will cover a new set of materials with video content, code practice and examples and reading materials. Both I and the weekly moderator (see following *Weekly Moderator* section) will pose several questions about the assignments -- both questions that can be answered by the content and those that require you to form some opinions and thoughts related to the content. Your responses to at least 2 of these prompts are required.

I expect your response to be well-thought out and presented coherently. Your response to each prompt should be minimally one paragraph (4-7 sentences) and maximally 3 paragraphs. Your initial responses will be due by Thursday (11AM EST); but will be evaluated at the beginning of the following week -- giving you time to continue responding if another prompt or comment inspires a more thoughtful or interesting response.

Weekly (co-)Moderator: The weekly moderator assignment will be rotated so that all students have a chance to minimally co-moderate. The moderator will be evaluated on the questions and prompts chosen to share with the group; and the ability to keep the conversation focused and interesting -- encouraging students to keep writing. The moderator will also be in charge of finding one interesting related piece of content (video, article, tweet, GitHub repository, blog post) to share with the class. A list of suggested blogs for useful content and several newsletters are included in the Canvas resources for the course. The weekly moderation will be evaluated according to the moderation rubric and be part of the percentage of the reading reaction grade. In the event there are more than 12 students in class, there may be some modules that have more than one weekly co-Moderator.

Discussion Post Rubric (for more details on several of these rubric choices, I recommend reading the 6+1 Writing Traits Rubric included in the Canvas materials -- from which several of these points are adapted)

	Exceptional (90-100)	**Proficient (80-89)**	**Basic (70-79)**	**Poor (<70)**
Ideas	Main idea is clear; relevant details support	Main idea is clear; the supporting	Main idea is unclear or unsupported	Main idea is missing as are supporting

	and add value to the conversation; the topic is relevant to the discussion at hand and supports the main idea.	details are present, but some details don't add value; the topic is related, but doesn't add compelling information to the conversation.	by details. The topic is related, but not relevant.	details. Topic is missing or irrelevant.
Organization	The writing is clear, focused and organized. Details are added in a logical order with attention to transitions between items.	The writing is clear, somewhat focused and shows some organization. Details are added in a supporting order, but transitions might be missing.	The writing is unclear or unfocused. There is basic organization (i.e. paragraphs) but little attention to order of details or presentation of support.	The writing is unclear, unfocused and lacks basic organization. Details may be missing or presented with no attention to order or transitions.
Word Choice	The writing uses insightful words, with technical words and concepts from the module used appropriately and when they add meaning.	The writing uses clear terms and technical words from the reading; several perhaps misused or added without meaning.	The writing does not include more than one technical word from the module or misuses nearly all when present.	The writing misuses all technical words included, or they are completely missing.
Conventions	The writing follows good conventions for conversational posts, with few if any grammatical or	The writing follows most writing conventions, and would require at least one round of edits and	The reader is distracted by grammatical and spelling errors. Minimal 2 rounds of edits to be	The writing is rife with grammatical and spelling errors.

	spelling errors.	responses to be exceptional.	exceptional.	
Communicative & Collaborative	The writing promotes conversation and responses. The writer incorporates other responses and ideas, while still communicating their* own ideas.	The writing allows for conversation and responses, however it doesn't explicitly incorporate questions, mentions or others' ideas.	The writing includes one or fewer references to others' responses or references to others are not relevant. The writing is more statement and fact oriented and lacks attention to ongoing conversation.	The writing lacks any references towards others' ideas or responses.

*they is used to better generalize for non-cis pronouns

Moderator Rubric

	Exceptional (90-100)	**Proficient (80-89)**	**Basic (70-79)**	**Poor (<70)**
Leadership	The moderator demonstrates leadership qualities such as passion, open-mindedness, authenticity and inspiration in their* prompt and in follow up responses. They are patient and encouraging with responses.	The moderator demonstrates at least one quality of leadership in their* prompt. They encourage at least two responses.	The moderator writes a prompt, but it doesn't clearly demonstrate leadership qualities. They encourage at least one response.	There is no prompt, or the prompt lacks any qualities. The few (if any) responses lack encouragement or thoughtfulness.

Management	The moderator posts at least two prompts in a timely and organized fashion. The follow-up posts and responses show ability to encourage others' participation and a value for others' feedback.	The moderator posts at least one prompt before the deadline (but not earlier). The follow up posts are not encouraging or are too late (i.e. Sunday night before the due date) to promote conversation.	The moderator posts only one prompt. The follow up is either lacking encouragement or too late.	The moderator fails to either post prompts or responses in a timely fashion.

Assignments

Most assignments will be submitted via the Jupyter Server or GitHub / GitLab. The specifics (including data, problem set or challenge) of each assignment will be available on the Monday of the module week. The due date for each assignment will be Friday by 11AM EST. Each assignment will have some element of code included as well as some writing (usually documentation, problem solving reflection and / or debugging notes). You will be asked to, at times, use peer or self-assessment with the same rubric as part of these assignments. I encourage you to evaluate each piece submitted with the following rubric *before* actually submitting the work. Consider this rubric a checklist for items that each assignment will require. Information about PEP-8 and how to easily lint your code for issues are available in the Canvas resources. Note, that several pieces of the rubric (Documentation, Testing) depend on later modules and will therefore be required in and after the related module is completed.

Code Assignment Rubric

	Exceptional (90-100)	**Proficient (80-89)**	**Basic (70-79)**	**Poor (<70)**

Clear, legible code, logic	The code follows PEP-8 standards and is legible. Variable and function names are clear and meaningful. The logic, data types and layout are easy-to-follow.	The code follows most PEP-8 standards and is somewhat legible. Variable and function names or logic (regarding data types or organization) are sometimes unclear.	The code follows some PEP-8 standards. Variables and functions are often named unclearly. Some logic is applied to data types, but at times it might be unclear.	The code shows no regard for PEP-8, legibility or normal logic flows.
Code or Project Structure	The code (or repository) has been organized in a manner to allow readability and easy extension. If it is a repository, the folders and files follow common conventions and the structure itself is well-documented.	The code (or repository) is readable and at least somewhat extensible (ability to import and use). If a repository, most folders and files follow common conventions and the structure is documented.	The code is readable but would take work to improve for extensibility. If a repository, the folders and files follow some conventions, but not enough to easily share with others. The structure may or may not be documented.	The code is disorganized and not very legible. It might function, but it is not easily shared, readable or extensible. If a repository, little organization exists and common conventions are not followed.
Correct Functionality / End Result	The code offers exceptional or above-and-beyond comprehension and functionality for the end result. This can be in the form of a “one step further”	The code functions properly and returns the proper result.	The code has at least one error in the result, but also at least one proper step or intermediate result is achieved.	The code is missing or achieves improper or false results.

	approach or applying new data to ensure general usability.			
Library Knowledge	The code shows a clear understanding of when and how to apply outside or standard libraries. The author has taken time to learn and apply outside libraries for the problem at hand.	The code uses outside libraries demonstrated in the module appropriately. Only one or two sections of the code could benefit from more library utilization.	The code utilizes at least one outside library, but is not effective or efficient in its use. There are more than two sections of code which could benefit from more library utilization.	The code uses no outside libraries or applies them to irrelevant or improper uses.
Mathematical / Statistical Reasoning	The code demonstrates a strong grasp of mathematical and statistical logic. The data types and output chosen reflect appropriate mathematical reasoning and can be strongly supported by work in the field.	The code demonstrates a basic grasp of mathematical and statistical logic. Most data types and chosen output reflect mathematical reasoning.	The code shows little grasp of mathematical or statistical logic. The data types chosen show a rudimentary understanding of the mathematical concepts.	The code has few (if any) elements showing any mathematical or statistical reasoning. Data types are misused or rarely used.
Documentation	The documentation is clear, concise and covers relevant topics. There is both module	The documentation is clear, concise and covers most relevant topics. There is	The documentation is unclear or too short or long for manageable reading.	The documentation is incomplete, missing or illegible for at least one if not all module,

	level documentation and class / function level. If necessary, the inline documentation clarifies the logic choices.	complete module level documentation although not all classes and functions are documented.	Several modules, functions or classes lack documentation.	class, function and inline code sections.
Testing	There are unit or data tests and validation provided that are clear, easy-to-use and documented.	There are unit or data tests and validation provided. Some lack clarity or documentation.	There are missing unit or data tests or they are so unclear and undocumented that they cannot be used.	There are no unit or data tests.
Debugging Notes	The author has included appropriate debugging notes in the ReadMe either in the form of a narrative (how I solved X?) or an FAQ. These notes are ready for public consumption and use.	The author has included some debugging notes in a file. They have at least one outside reference but are difficult to follow.	The author has included only one note or reference re: debugging and it is not included in a separate document or resource.	There are no debugging notes or they are indistinguishable from other notes.

Final Project

The final project will be a group project with group work (in the form of peer review, code review, issue assignment and delegation) being a large part of the tangible grade. Of course, a functional product is also a requirement and non-functional code will be returned for further work. There will be several stages to this project and reviews along the way, to ensure the groups are working well and progress is steady and obvious (I am hoping

this also discourages procrastination). Final projects will be presented and reviewed in the final week of the course and will all be on GitLab. The code and ReadMe's will be reviewed using the Code Assignment review rubric. The overall project must meet the defined requirements (posted on Canvas), and the group work will be reviewed using the following "Data Science Team" rubric. For the rubric, team members will be evaluated based on the overall team performance (see "As A Team" criteria) as well as individually (see "As a Member" criteria). These evaluations will be based on review of the GitLab project as well as peer and individual reviews.

Teams will be assigned by the instructor after a survey. If you have concerns about your team, a particular team member or your ability to contribute to the team, please contact the instructor immediately. You will be evaluated based on your ability to work together as well as the individual strengths you bring to the table; so I encourage you to attempt to resolve team issues internally before bringing them to the instructor (i.e. as you would for a true Data Science team if the instructor was instead the CEO/CTO/CIO).

Data Science Team Rubric

	Exceptional (90-100)	**Proficient (80-89)**	**Basic (70-79)**	**Poor (<70)**
As a Team: Collaborative & Communicative	The team communication is collaborative and timely. Issues that are brought to light are quickly addressed. Peer feedback is constructive and positive.	The team communication is collaborative or timely. Issues are eventually addressed. Peer feedback is lacking constructive or positive elements.	The team communication lacks timeliness or collaboration. Peer feedback is missing or overwhelmingly negative.	Team communication is infrequent, incomplete and lacking collaboration.
As a Team:	The team utilizes all tools	The team uses most of the	The team uses at least one tool	The team does not utilize the

Use of Tools	available within GitLab in appropriate capacity; including Issues, planning, and documentation.	tools available in an appropriate capacity.	in an appropriate capacity.	tools available or uses them in unclear, inappropriate ways.
As a Team: Accepts Challenges & Able to Delegate	When faced with inevitable challenges, the team is adaptive, uses issues to report and inform other members and appropriately and clearly delegates work and review.	When facing challenges, the team is responsive; but may not always utilize issues or communication to handle problems or delegate issues.	When facing challenges, the team is unresponsive or unclear. One or two members end up carrying the group to the finish.	The team has little to no ability to delegate or respond to challenges or does so in an incomplete or untimely fashion.
As a Member: Responsible	The team member takes on tasks and issues and helps delegate what they cannot do well. They ensure the work assigned to them is completed in a timely manner.	The team member responds to delegated or assigned tasks (and might at times volunteer). They ensure their work is completed in time.	The team member responds to some, but not all, delegated or assigned tasks. The work is completed mostly on time.	The team member shows little to no responsibility towards their team or tasks.
As a Member: Responsive	The team member is both timely and open-minded when responding to others' mentions, comments and	The team member is often timely and open-minded when responding to others; with some visible deviations.	The team member is at least twice not on time or responsive to others feedback, comments or mentions.	The team member is often unresponsive to other members of the team.

	feedback.			
As a Member: Productive	The team member commits meaningful and contributive code and comments frequently.	The team member commits and comments at least once a week. Some contributions or comments lack thought or attention to detail.	The team member commits and comments at least every other week. The contributions are mostly helpful.	The team member does not contribute as a productive member via code and / or comments.

Final Presentation + Findings Rubric

	Exceptional (90-100)	**Proficient (80-89)**	**Basic (70-79)**	**Poor (<70)**
Presentation: Organized	The presentation has a clear beginning, middle and end with attention paid to building on previous concepts and findings.	The presentation builds on previous concepts, but has an unclear beginning or end.	The presentation either does not build on concepts or findings, or does so in a disjointed manner.	The presentation pays little attention to organization with little or no building on previous findings and concepts.
Presentation: Audience -Appropriate	The presentation content and language used shows attention and care given the audience. Time is spent on topics engaging to decision makers and	The presentation content and language is appropriate for the given audience. Topics are related to decision makers and peers.	The presentation uses some appropriate language and content, but also includes material too easy or too difficult for the target audience.	The presentation content and language do not fit the audience. The topics are either not present or do not include topics for decision

	peers.			makers or peers.
Presentation: Data Visuals	Data visualization is featured prominently in the presentation and the quality of the visuals improves the content. The visualizations are clear and effective at communicating the findings. The visuals are included in an engaging way and the audience wants to further explore the data based on these visuals.	Data visualization is included in the presentation and the visuals add to the content. The visualizations are effective at communicating the findings. The visuals are included appropriately, but perhaps not in an engaging way.	Data visualization is included in the presentation, but at least one visual is challenging to understand or ineffective at communicating the findings.	Either data visualization is not included in the presentation, or a majority of the visuals are ineffective, inaccurate or unethical.
Findings: Clear and well-written*	The presentation deck and supplemental final findings materials are well-written and follow convention. The materials make the findings clear and easy-to-understand. Technical and mathematical words are used appropriately	The presentation deck and supplemental final findings materials are written clearly and follow most conventions. There are a few unclear places or words used improperly, but the overall usage and clarity is not hindered by	The presentation deck and supplemental final findings are lacking clarity or are poorly written. Technical and mathematical words are sometimes misused.	The presentation deck and supplemental final findings materials are incomplete or are unclear or error-prone.

	and add meaning.	these mistakes.		
Findings: Statistically Accurate & Reproducible	The findings and visualizations included in the material and slide deck are accurate and easily reproduced. They show an understanding of the statistical principles and are well chosen to convey the outcome.	The findings and visualizations included in the material and slide deck are accurate and able to be reproduced.	The findings and visualizations included in the material and slide deck have minor inaccuracies or are not able to be reproduced.	The statistics and visuals are either missing or are have major inaccuracies.
Findings: Actionable	The findings have a clear actionable result, even if this result is in the form of a new design plan. The importance and relevance of the findings and the topic as a whole is made clear to the audience.	The findings have a clear and actionable result, but perhaps lacking a few areas of follow through or plan. The importance of the findings is referenced, but perhaps not made clear to the audience.	The findings have either an unclear result or little engagement in “what happens next.” The importance of the findings is either missing or unclear.	There is little or no attention paid to what to do with the findings. The importance of the findings is lacking clarity or is entirely missing.

* What does well-written mean? See Discussion Posts Rubric: Organization, Word Choice and Convention

Any clarification needed on the rubrics should be done before the end of the first week of class. If you have questions, suggestions or need help, please message the instructor or post in group chat to clarify the problem or question immediately.

University Policies

University Policy on Accommodating Students with Disabilities:

Students requesting accommodation for disabilities must first register with the Dean of Students Office (http://www.dso.ufl.edu/drc/). The Dean of Students Office will provide documentation to the student who must then provide this documentation to the instructor when requesting accommodation. You must submit this documentation prior to submitting assignments or taking the quizzes or exams. Accommodations are not retroactive, therefore, students should contact the office as soon as possible in the term for which they are seeking accommodations.
Students with Disabilities who may need accommodations in this class are encouraged to notify the instructor and contact the Disability Resource Center (DRC) so that reasonable accommodations may be implemented. DRC is located in room 001 in Reid Hall or you can contact them by phone at 352-392-8565.
University counseling services and mental health services:

**Netiquette: Communication Courtesy:
All members of the class are expected to follow rules of common courtesy in all email messages, threaded discussions and chats.
http://teach.ufl.edu/wp-content/uploads/2012/08/NetiquetteGuideforOnlineCourses.pdf

Class Demeanor:
Mastery in this class requires preparation, passion, and professionalism. Students are expected, within the requirements allowed by university policy, to attend class, be on time, and meet all deadlines. Work assigned in advance of class should be completed as directed. Full participation in online and live discussions, group projects, and small group activities is expected.

My role as instructor is to identify critical issues related to the course, direct you to and teach relevant information, assign appropriate learning activities, create opportunities for assessing your performance, and communicate the outcomes of such assessments in a timely, informative, and professional way. Feedback is essential for you to have confidence that you have mastered the material and for me to determine that you are meeting all course requirements.

At all times it is expected that you will welcome and respond professionally to assessment feedback, that you will treat your fellow students and me with respect, and that you will contribute to the success of the class as best as you can.

Getting Help:
For issues with technical difficulties for E-learning in Canvas, please contact the UF Help Desk at:

- Learning-support@ufl.edu
- (352) 392-HELP - select option 2
- https://lss.at.ufl.edu/help.shtml

** Any requests for make-ups due to technical issues MUST be accompanied by the ticket number received from LSS when the problem was reported to them. The ticket number will document the time and date of the problem. You MUST e-mail your instructor within 24 hours of the technical difficulty if you wish to request a make-up.

Other resources are available at http://www.distance.ufl.edu/getting-help for:

- Counseling and Wellness resources
  http://www.counseling.ufl.edu/cwc/Default.aspx
  352-392-1575
- Disability resources
- Resources for handling student concerns and complaints
- Library Help Desk support

Should you have any complaints with your experience in this course please visit http://www.distance.ufl.edu/student-complaints to submit a complaint.

Course Evaluation:
Students are expected to provide feedback on the quality of instruction in this course based on 10 criteria. These evaluations are conducted online at https://evaluations.ufl.edu
Evaluations are typically open during the last two or three weeks of the semester, but students will be given specific times when they are open. Summary results of these assessments are available to students at https://evaluations.ufl.edu/results

University Policy on Academic Misconduct:

Academic honesty and integrity are fundamental values of the University community. Students should be sure that they understand the UF Student Honor Code at http://www.dso.ufl.edu/students.php

The University of Florida Honor Code was voted on and passed by the Student Body in the Fall 1995 semester. The Honor Code reads as follows:

Preamble: In adopting this Honor Code, the students of the University of Florida recognize that academic honesty and integrity are fundamental values of the University community. Students who enroll at the University commit to holding themselves and their peers to the high standard of honor required by the Honor Code. Any individual who becomes aware of a violation of the Honor Code is bound by honor to take corrective action. A student-run Honor Court and faculty support are crucial to the success of the Honor Code. The quality of a University of Florida education is dependent upon the community acceptance and enforcement of the Honor Code.

The Honor Code: “We, the members of the University of Florida community, pledge to hold ourselves and our peers to the highest standards of honesty and integrity.”

On all work submitted for credit by students at the University of Florida, the following pledge is either required or implied:

"On my honor, I have neither given nor received unauthorized aid in doing this assignment."

For more information about academic honesty, contact Student Judicial Affairs, P202 Peabody Hall, 352-392-1261.

ACADEMIC HONESTY

All graduate students in the College of Journalism and Communications are expected to conduct themselves with the highest degree of integrity. It is the students’ responsibility to ensure that they know and understand the requirements of every assignment. At a minimum, this includes avoiding the following:

Plagiarism: Plagiarism occurs when an individual presents the ideas or expressions of another as his or her own. Students must always credit others' ideas with accurate citations and must use quotation marks and citations when presenting the words of others. A thorough understanding of plagiarism is a precondition for admittance to graduate studies in the college.

Cheating: Cheating occurs when a student circumvents or ignores the rules that govern an academic assignment such as an exam or class paper. It can include using notes, in physical or electronic form, in an exam, submitting the work of another as one's own, or reusing a paper a student has composed for one class in another class. If a student is not sure about the rules that govern an assignment, it is the student's responsibility to ask for clarification from his instructor.

Misrepresenting Research Data: The integrity of data in mass communication research is a paramount issue for advancing knowledge and the credibility of our professions. For this reason any intentional misrepresentation of data, or misrepresentation of the conditions or circumstances of data collection, is considered a violation of academic integrity. Misrepresenting data is a clear violation of the rules and requirements of academic integrity and honesty.

Any violation of the above stated conditions is grounds for immediate dismissal from the program and will result in revocation of the degree if the degree previously has been awarded.

Students are expected to adhere to the University of Florida Code of Conduct https://www.dso.ufl.edu/sccr/process/student-conduct-honor-code

Although it should not need to be said, I will state that any student failing to abide by the Code of Conduct towards any other student or faculty member will be immediately dismissed from the course communications and placed in mediation.

If you have additional questions, please refer to the Online Graduate Program Student Handbook you received when you were admitted into the Program.

Schedule

Course Introduction:

Course Introduction Video:

- Welcome to the world of data wrangling with code!
- Introduction Discussion: Fears, Goals and What is Programming?
- Extra video: Why code? https://www.youtube.com/watch?v=cUWzRdxD6sM

Course Syllabus Video:

- Explanation of course requirements, rubrics and assignments
- Course Syllabus Discussion

Python Orientation Videos:
Please complete one of the following introduction to Python video courses that are free online. This will help you get a jumpstart on the course and allow some things to be easier repetition and review rather than immediately diving in with no background.

- CodeAcademy: https://www.codecademy.com/courses/introduction-to-python-6WeG3/0/1
- DataCamp: https://www.datacamp.com/courses/intro-to-python-for-data-science
- Khan Academy: https://www.youtube.com/playlist?list=PL36E7A2B75028A3D6

Tasks:

- Please sign up for GitHub! https://github.com/
- Install Slack and join the team via email invite

Week One: Introduction to Programming

Learning Objectives:

- Students will use the Python interpreter and Jupyter notebook to write and run Python code
- Students will select packages and install them via pip
- Students will determine how to use new functions, classes and methods based on documentation
- Students will list different Python data types and differentiate between

them

Watch:

- Introduction to your terminal and Python interpreter
- Installing Packages
- Welcome to IPython and Jupyter
- Python data types
- Complex python data types
- Executing a Python script
- Extra videos:
    - Khan Academy: Python Data Types (https://www.youtube.com/watch?v=husPzLE6sZc&t=3s)
    - Introduction to Jupyter (https://www.youtube.com/watch?v=HW29067qVWk)
    - Python functions (https://www.youtube.com/watch?v=NE97ylAnrz4)
    - Introduction to git (https://www.youtube.com/watch?v=0fKg7e37bQE)

Required Readings:

Data Wrangling with Python, Chapter 1 and 2*

*to note: You will need to run the code with Python 3 adaptations! In these chapters, this means using print(1) rather than print 1.

- Why Python? https://www.codeschool.com/blog/2016/01/27/why-python/
- Python 2 vs 3: https://www.digitalocean.com/community/tutorials/python-2-vs-python-3-practical-considerations-2
- How to ask good questions on technical and scientific forums: https://www.biostars.org/p/75548/
- Python classes and Object Oriented Programming: https://jeffknupp.com/blog/2014/06/18/improve-your-python-python-classes-and-object-oriented-programming/
- Extra reading:
    - Python Data Types: http://www.diveintopython3.net/native-datatypes.html
- Resources:

    - Markdown documentation: https://daringfireball.net/projects/markdown/syntax
    - How to Fork a repository on GitHub: https://help.github.com/articles/fork-a-repo/

Assignments:

- Complete module one checklist: installing Python, Jupyter and IPython packages, signing up for GitHub and StackOverflow and generating an SSH key and putting it in your GitHub profile (https://help.github.com/articles/connecting-to-github-with-ssh/)
- Module one quiz
- Module one discussion posts
- Code Assignment: Data Types in Python

Week Two: Introduction to Statistics

Learning Objectives:

- Students will select appropriate statistical measures for analyzing a chosen problem set
- Students will define common statistical terms and match them with their appropriate equations
- Students will assemble code to answer a series of mathematical and statistical questions

Watch:

- Statistical reasoning in everyday life
- Descriptive Statistics with Python and Pandas
- Data Exploration with Python
- Extra videos:
    - Basic Probability: https://www.khanacademy.org/math/precalculus/prob-comb/basic-prob-precalc/v/basic-probability
    - Probability and Addition Rule https://www.khanacademy.org/math/statistics-probability/probability-library/addition-rule-lib/v/probability-with-playing-cards-and-venn-diagrams
        - Practice: https://www.khanacademy.org/math/statistics-probability/probability-library/addition-rule-lib/a/addition-rule-for-probability-basic

- Mean, Median, Mode: https://www.youtube.com/watch?v=onSebaCChTg
- Mean, Median and Mode -- the rap song (https://www.youtube.com/watch?v=A7MxGyEaN64)
- Interquartile Range: https://www.youtube.com/watch?v=DGAXeX42eoE and https://www.youtube.com/watch?v=ZAE-5TJy9kU
- Outliers: https://www.youtube.com/watch?v=9aDHbRb4Bf8
- Box and Whisker plot: https://www.youtube.com/watch?v=Fhk5lDGpivo&v=Ive6_u1-b8o
- Khan Academy: measuring spread (https://www.khanacademy.org/math/probability/data-distributions-a1/summarizing-spread-distributions/v/range-variance-and-standard-deviation-as-measures-of-dispersion)
- Correlation: https://www.youtube.com/watch?v=ugd4k3dC_8Y

Required Readings:

- Quora: What is the difference between a data scientist and a statistician? https://www.quora.com/What-is-the-difference-between-a-data-scientist-and-a-statistician
- WikiHow: How to Calculate Outliers http://www.wikihow.com/Calculate-Outliers

Assignments:

- Group questionnaire & survey
- Code Assignment: Using the stats and math libraries in Python
- Module two discussion
- Module two quiz

Week Three: SQL and Databases

Learning Objectives:

- Students will create their own SQL database and store and select data from the database
- Students will write proper SQL syntax for selections, updates and insertion
- Students will describe database schema
- Students will differentiate between different database types

Watch:

- Introduction to SQL: https://www.khanacademy.org/computing/computer-programming/sql#sql-basics
- Basic SQL with SQLite and dataset
- Joins with Pandas
- What is NoSQL? https://www.lynda.com/Cassandra-tutorials/What-NoSQL/111598/117567-4.html
- Extra videos:
    - sqlite3 with Python3 (https://www.youtube.com/watch?v=o-vsdfCBpsU)
    - Khan academy: Introduction to SQL (https://www.khanacademy.org/computing/computer-programming/sql)

Required Readings:

- Data Wrangling, Chapter 6
- Install Sqlite: https://www.tutorialspoint.com/sqlite/sqlite_installation.htm
- SQLZoo Tutorials: http://sqlzoo.net/
- SQL vs NoSQL: https://www.sitepoint.com/sql-vs-nosql-differences/
- Recommended: Visual Guide to SQL Joins: https://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins

Assignments:

- Team Communication Guidelines
- SQLZoo Tutorials
- Code Assignment: Querying customers with Python
- Module three discussion
- Module three quiz

Week Four: Pandas and NumPy

Learning Objectives:

- Students will identify, list and classify different numpy data types
- Students will generate a pandas dataframe from given data
- Students will evaluate several statistical criteria from a pandas dataframe

- Students will visualize data using pandas

Watch:
- Numpy Data Types
- Fixing Types with Pandas
- Split Apply Combine in Pandas
- Data Science with Python Pandas (PyData Carolinas): https://www.youtube.com/watch?v=POe1cufDWFs
- Extra Videos:
    - Numpy vs Lists: https://www.youtube.com/watch?v=AGzB7_vsLbE
    - Selecting data with NumPy: https://www.youtube.com/watch?v=rnw1qixAv1s
    - Basic Statistics with Numpy: https://www.youtube.com/watch?v=WUZlyG42ko0

Required Readings:
- 100 exercises with NumPy: http://www.labri.fr/perso/nrougier/teaching/numpy.100/
- Pandas Tutorial: https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python#gs.x7HZ6gg

Assignments:
- Group Question Brainstorm and Peer Code Review
- Code Assignment: Importing, Investigating and Grouping Dataframes
- Module four discussion
- Module four quiz

Week Five: Data Visualization

Learning Objectives:
- Students will create data visualizations using Pandas and matplotlib
- Students will identify qualities of a good data visualization
- Students will design interactive charts using the Bokeh library
- Students will determine appropriate visualizations for different data types and distributions

Watch:
- What makes a good / bad data visualization?

    - The best stats you've never seen https://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen
    - Ethical Data Visualization
        - https://www.coursera.org/learn/dataviz-design/lecture/YqHHo/practicing-good-ethics-in-data-visualization
- Data Visualization with Pandas
- Data Visualization with Matplotlib
- Data Visualization with Bokeh
- Extra videos:
    - Scatter and Line Plots with Matplotlib https://www.youtube.com/watch?v=SiCyTcudoSE
    - Data Visualization landscape in Python: https://www.youtube.com/watch?v=OC-YdBz8Llw
    - Fixing ineffective visualization:
        - https://www.coursera.org/learn/dataviz-design/lecture/x3x31/ineffective-visuals-and-how-to-improve-them

Required Readings:

- Data Wrangling, Chapter 10
- http://matplotlib.org/users/recipes.html
- Reddit: Top Posts from Data is Beautiful https://www.reddit.com/r/dataisbeautiful/top/
- Pandas Visualization: http://pandas.pydata.org/pandas-docs/stable/visualization.html
- Effective Data Visualization:
    - http://paldhous.github.io/ucb/2016/dataviz/week2.html
- Optional:
    - Information is Beautiful
        - http://www.informationisbeautiful.net/
    - Dataviz -- you're doing it wrong https://www.youtube.com/watch?v=i93iWza8sG8
    - Bokeh Tutorial
        - https://www.youtube.com/watch?v=9FIUFLmaWvY

Assignments:

- Final Project Design Documentation & Group Discussion

- Code Assignment: Flat Charts with Matplotlib & Interactive Charts with Bokeh
- Module five quiz
- Module five discussion

Week Six: Writing your First Stand-Alone Script

Learning Objectives:

- Students will create a stand alone Python script with proper documentation and command line use.
- Students will design documentation and evaluate best practices for documenting code.
- Students will reflect on code design principles via peer and self-evaluation.

Watch:

- What's in a script?
- Structuring a project
- CookieCutter: https://www.youtube.com/watch?v=nExL0SgKsDY&index=51&list=PLGVZCDnMOq0p55DKM5a0BOR6vwvKcAB56
- Documentation, Disrupted: How Two Technical Writers Changed Google Engineering Culture https://www.youtube.com/watch?v=EnB8GtPuauw&list=PLkQw3GZ0bq1JvhaLqfBqRFuaY108QmJDK&index=1

Required Readings:

- Dive into Python: http://www.diveintopython3.net/your-first-python-program.html
- Beginner's guide to documentation from write the docs: http://www.writethedocs.org/guide/writing/beginners-guide-to-docs/
- Structuring your Project: http://docs.python-guide.org/en/latest/writing/structure/
- Code Review checklist: https://blog.fogcreek.com/increase-defect-detection-with-our-code-review-checklist-example/
- Example Google Docstrings: http://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html
- Example NumPy Docstrings: http://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_numpy.html#example-

numpy

Assignments:

- Group: Creating and Assigning Tasks and Issues in GitLab & Kanban usage
- Code Assignment: Building your first Python Script
- Code Review
- Module six quiz
- Module six discussion

Week Seven: Scraping Data

Learning Objectives:

- Students will create web scrapers with Python using lxml or BeautifulSoup and Scrapy.
- Students will classify different scrapers according to the type of scraping performed.
- Students will use browser tools to evaluate the ease and difficulty of scraping a website.

Watch:

- Scraping etiquette and legality
- Evaluating a website using browser tools
- Extra videos:
    - Intro to XPath: https://www.youtube.com/watch?v=jP4YIT9Ex_s
    - Brief intro to Selenium: https://www.youtube.com/watch?v=bhYulVzYRng
    - Data Mining with Web Scraping: https://www.youtube.com/watch?v=wT66i7jeyL8
    - Getting Started with Scrapy
    - https://www.youtube.com/watch?v=vkA1cWN4DEc&list=PLZyvi_9gamL-EE3zQJbU5N3nzJcfNeFHU (videos #1-3, but feel free to watch more if you like!)

Required Readings:

- Chp 11-12 Web Scraping
- Introduction to XPath: https://blog.scrapinghub.com/2016/10/27/an-introduction-to-xpath-with-examples/
- Mozilla Developer - CSS Selectors - https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Selectors

- Selenium Documentation for Navigation http://selenium-python.readthedocs.io/navigating.html

Assignments:
- Group: Identify data sources and begin data extraction
- Code Assignment: Scraping User Comments
- Module seven discussion
- Module seven quiz

Week Eight: APIs

Learning Objectives:
- Students will assess API documentation to evaluate useful methods
- Students will utilize helpful library wrappers for the Twitter API to run simple queries

Watch:
- What is an API? Using tweepy
- Is that API secure? https://www.infoq.com/presentations/http-api-security
- Choose one (or two) of the following:
    - Facebook Graph API: https://www.youtube.com/watch?v=WteK95AppF4
    - Weather API: https://www.youtube.com/watch?v=sbYXa6HJJ5M
    - Google Maps API: https://www.youtube.com/watch?v=sI8py6soTWs
    - Google APIs: https://www.youtube.com/watch?v=IVjZMIWhz3Y
    - Wolfram Alpha API: https://www.youtube.com/watch?v=tW1TM8m429Q

Required Readings:
- Useful APIs for developers: http://www.creativebloq.com/web-design/apis-developers-need-know-121518469
- Data Wrangling with Python, Chapter 13
- Pick one (or two) of the following to investigate: https://github.com/toddmotto/public-apis

Assignments:

- Initial Group Project documentation and file structure
- Group: Data exploration
- Code Assignment: Google Maps API usage
- Module eight discussion
- Module eight quiz

Week Nine: Natural Language Processing

Learning Objectives:

- Students will create code to preprocess text by removing stop words, resolving case and punctuation differences and tokenizing the content.
- Students will define important natural language processing words such as POS tagging, tokenization, bag-of-words, tf-idf and sentiment analysis and determine good applications for each.
- Students will evaluate a series of documents using doc2vec.

Watch:

- Why Natural Language Processing?
- Introduction to NLP:
  - https://www.youtube.com/watch?v=IMKweOTFjXw
- DataCamp NLP Fundamentals
  - Regex & Tokenization: (link to come)
  - Bag of Words: (link to come)
- Word Vectors and Intro to Gensim
  - https://www.youtube.com/watch?v=thLzt3D-A10

Required Readings:

- Categorizing and Tagging words: http://www.nltk.org/book/ch05.html
- A Word is Worth a Thousand Vectors: http://multithreaded.stitchfix.com/blog/2015/03/11/word-is-worth-a-thousand-vectors/
- Research: Semantics derived automatically from language corpora contain human-like biases http://science.sciencemag.org/content/356/6334/183.full?dom=icopyright&src=syn

Assignments:

- Group Assignment: v 0.10 release

- Code Assignment: Keywords with Python
- Week nine discussion
- Week nine quiz

Week Ten: Tests

Learning Objectives:

- Students will create unit tests to find bugs, identify faulty logic and create secure code.
- Students will write data validation tests to find incorrect data.
- Students will utilize property based testing to evaluate constraints of a given function or method.

Watch:

- Getting Started Testing: https://www.youtube.com/watch?v=FxSsnHeWQBY
- PyTest Introduction:
    - https://www.youtube.com/watch?v=l32bsaIDoWk&list=PLeo1K3hjS3utzQYDNRNluzqJqpMXx6hHu
- Data Validation Tests
- Data Unit Testing
- Extra videos:
    - More Hypothesis! https://www.youtube.com/watch?v=mg5BeeYGjY0

Required Readings:

- Cerebus Documentation: http://docs.python-cerberus.org/en/stable/
- Hypothesis QuickStart: http://hypothesis.readthedocs.io/en/latest/quickstart.html

Assignments:

- Group Project: Adding Tests
- Code Assignment: Adding Tests to Your Script
- Week ten quiz
- Week ten discussion

Week Eleven: Automation

Learning Objectives:

- Students will create new cron tasks and run them on a server

- Students will write a celery task with proper asynchronous execution
- Students will categorize tasks as easy or difficult to automate based on criteria and questions

Watch:
- Why automate? What to automate?
- Automation via cron: https://www.youtube.com/watch?v=UlVqobmcPuM
- Distributed Task Pipelines with Celery
    - https://www.youtube.com/watch?v=fg-JfZBetpM
    - https://www.youtube.com/watch?v=UHveawwmi-o
- Testing automation and continuous integration
    - https://www.youtube.com/watch?v=xSv_m3KhUO8
- Optional:
    - Introduction to Airflow: https://www.youtube.com/watch?v=60FUHEkcPyY&t=54s

Required Readings:
- Data Wrangling, Chapter 14
- HN: Do you automate your routine tasks? https://news.ycombinator.com/item?id=2390845

Assignments:
- Group Project: v 0.2
- Code Assignment: Automation Conventions
- Week eleven discussion
- Week eleven quiz

Week Twelve: Final Project and What's Next?

Learning Objectives:
- Students will share and analyze what they have learned throughout the course and in the production of their final project.
- Students will evaluate others learning via peer evaluation.
- Students will judge personal areas of interest for future study

Watch:
- What happens next?
- Live Project Presentations
- Any video of interest from PyDataTV

https://www.youtube.com/user/PyDataTV

Required Readings:

- Data Wrangling, Chapter 15
- Data Science for Social Good: http://www.kdnuggets.com/2015/07/guide-data-science-good.html

Assignments:

- Final group evaluations
- Group project presentation and review
- Code Assignment: Code Review and Improvements

Disclaimer:
This syllabus represents my current plans and objectives. As we go through the semester, those plans may need to change to enhance the class learning opportunity. Such changes, communicated clearly, are not unusual and should be expected.