# CSC 201 – Introduction to Computer Programming — Fall 2017

***Computer characteristics, algorithms, data representation, program development.***
***Students will write several programs to solve numerical and non-numerical problems.***

## PROVISIONARY VERSION – CHANGES ARE LIKELY

| | |
|---|---|
| **Tyler Hall Lab 055 (mandatory)** | Mondays, 9:00am-10:50am |
| **Work Sessions in Pharm 180** | Tue & Thu, 11am-12:15pm |

| | |
|---|---|
| **Name** | Martin Hellwig |
| **Office** | Tyler Hall 135 |
| **Walk In Office Hours** | Mondays 11am-12pm |
| **Appointments (via Starfish)** | Mondays 12pm-1pm |
| **eMail** | hellwig@uri.edu |

| | |
|---|---|
| **Last date to drop without record** | Sep 27th |
| **Last date to withdraw from this course** | Oct-18th |
| **Last day of class** | Dec-11th |
| **Final Exam (Project)** | Project due Dec-18th |

You do not need to purchase any textbook for this class. We will heavily use free online resources, especially our wikibook:

http://intro-computing.cs.uri.edu

You can complete this entire course using just the computers in the lab. You may choose to use a different device, but your instructor will not provide any technical support.

Throughout this course you will:

1. Learn *computational thinking* to discover new information and solve problems
2. Learn to obtain data sets, clean them, visualize them, analyze them, and present data-driven answers to questions
3. Learn to program in the Python programming language
4. Learn to use Python it manipulate data sets for analysis
5. Explain how Google Trends uses visualizations to illustrate trends
6. Use a spreadsheet to generate charts and graphs to visualize data.
7. Use a spreadsheet to sort, filter, categorize, and clean data
8. Use a spreadsheet to create a summary table (often called pivot table) from the data
9. Explain how to obtain data in these ways:
    a. Using data repositories
    b. Searching the Internet
10. Obtain data in each of those ways.
11. Program in Python using variables, types, expressions, strings, functions, lists, dictionaries, loops, libraries, and file I/O.
12. Program in Python to clean data in CSV format
13. Program in Python to perform data statistics
14. Program in Python to produce data visualizations
15. Program in Python to obtain data using data service Application Programming Interfraces (APIs)

## Tentative Schedule:

| Date | Topic / Readings Due | Assignment Due |
|---|---|---|
| September 6, 2017 | Welcome, Syllabus, etc | |
| September 11, 2017 | Spreadsheets and Visualizations | Spreadsheets |
| September 18, 2017 | Obtaining and Processing Data | Visualizations |
| September 25, 2017 | First Steps in Python | Processing Data |
| October 2, 2017 | Strings | Pivot Tables |
| October 9, 2017 | Columbus Day (no makeup) | Python Expressions |
| October 16, 2017 | Conditionals | Python Strings |
| October 23, 2017 | Functions | Python Conditionals |
| October 30, 2017 | Lists, Dicts and Loops | Python Functions |
| November 6, 2017 | Data Processing in Python | Python Lists / Data Structures |
| November 13, 2017 | Veteran's Day (no makeup) | Code Academy Unit 8 |
| November 20, 2017 | Statistics | Data Cleaning in Python |
| November 27, 2017 | Visualization | Python Stats |
| December 4, 2017 | Getting Data Automatically | Python Visualization |
| December 11, 2017 | Final Project Discussion | Obtaining Data With Python |
| December 18, 2017 | | Final Project |

## Class Format

In this class the instructor does not lecture. We may briefly go over some hard to understand concepts, but the largest part of the class time is yours. Work on your assignments, ask questions and discuss problems with each other. The instructor is there to guide you along every step of the way. In pedagogical circles this is called a flipped classroom. It allows you to learn at your own pace by reading ebooks and watching videos before class. That way we are all on the same page, no matter what your style or speed of learning is. In class we will complete little projects that help you put to work what you just learned. After that you go home and work on a graded assignment.

## Deliverables

14 Assignments worth 1 point each
1 Final Project worth 6 points

You are expected to complete the weekly reading assignments before class every Monday morning. This is essential for your success in this course! Should it become apparent that some students are falling behind in their reading, the instructor reserves the right to add random short quizzes to the beginning of class. These quizzes would become additional deliverables and be worth 1 point each.

## Evaluation

In this class we use an alternate evaluation method. Rather than starting out with zero points and slowly earning your way up to a D and eventually an A, everybody in this class starts out with a solid B. Basically we give you 80 points for free. You can then earn **and lose** points throughout the semester.

Each assignment has a point value associated with it. However, you only get those points if you do exceptionally well (far exceed expectations). If you just meet the requirements, you don't get or lose any points. However, if you submit an assignment that is sub-par, your will lose the assignment's point value – meaning that we will deduct those points from your overall grade.

### Late Assignments

If you miss a deadline, you have the option of submitting your assignment up to one week late. Late assignments will incur a penalty equal to the point value they are worth. This penalty is calculated independently of the points for your assignment. If you fail to submit an assignment altogether, you lose twice the points associated with it.

### An Example

Say your current score is 80 and you are working on an assignment that is worth 1 point. Depending on when you submit this assignment and how well you do on it, your overall grade will change as follows

| | Exceed Expectations | Meet Expectations | Fall Short of Expectations |
|---|---|---|---|
| Submitted on time | 81 | 80 | 79 |
| Submitted 1 hour late | 80 | 79 | 78 |
| Submitted 6 days late | 80 | 79 | 78 |
| Submitted 8 days late | 79 | 78 | 77 |
| Never submitted | 79 | 78 | 77 |

## Grading Questions

Your grade will be frequently updated on Sakai. Assignments will be graded within one week. If you have any concern about a particular grade, you may challenge this grade within one week after it is posted. After that your grade will be "set in stone"..

## Attendance

Attendance is absolutely mandatory every Monday. Attendance will be taken no later than five minutes after class starts. Throughout the semester you get one "free" absence for any reason. Beyond that any additional unexcused absence will cost you two percentage points off your final grade.

## Accelerated Learning

If you feel that you are learning faster than your peers, you may request an accelerated course format. The instructor will evaluate each request individually and may or may not grant it based on a wide range of factors. If granted, you may complete assignments earlier than the rest of the course. However, you must still submit all assignments in order and may only work on one assignment at a time. You may not  resubmit a previously submitted assignment. Once you complete all assignments and the final project, you will receive your final grade and you will usually be excused for the rest of the semester. Please note: Since accelerated students will be ahead of the other students, the instructor may from time to time ask you to help out your peers. Also, you will be expected to complete your assignments with minimal supervision since your instructor will focus most of his attention on those students who need more help than you do.

## Things to Note

A. Please note that I like to keep things straight forward for my students and hence I do not curve any assignments or final grades. What you see is what you get.
B. The instructor reserves the right to modify the above requirements with appropriate notification should it be deemed necessary.
C. Students should be in attendance for each class session. Extenuating circumstances causing absenteeism should be prearranged with the instructor.
D. All assignments should be completed on time, as scheduled, according to criterion established by the instructor. Points will be deducted for each day a written assignment is late.
E. The instructor will be available during posted office hours for general assistance. You can also reach him by email any time.

### Diversity

It is the instructor's policy to support and promote diversity in our classrooms.. One of the ways we can do this is to form an appreciation and respect for each other. Remember that diversity encompasses more than the color of one's skin; it includes diverse ideas, beliefs, religions, sexuality, age, gender, and those challenged physically or mentally. If you feel discriminated against or harassed by any other student, please inform your instructor right away. Such offensive behavior will not be tolerated in any circumstance!

### Cell Phone Usage

Cell phones and other communication devices may not be used during class periods. These devices are highly distracting, even if they are set to "vibrate only". Therefore they need to remain either completely turned off or at least switched to a completely silent state and stowed away. Should you require using a cell phone due to special circumstances such as medical emergencies, you need to obtain the instructors prior approval.

### Academic Integrity

Students are expected to maintain principles of academic integrity and conduct as defined in URI's Student Handbook. Violations may be reported to the Dean of Students.

For this class it is especially important that you complete all assignments yourself and submit only your own work. It is perfectly alright to discuss a solution with other students, but what you ultimately submit must be your own work. Submitting somebody else's work is considered plagiarism and will always result in:

a) A penalty equivalent to five times the point value associated with the assignment
b) An incident report to the Dean of Students. Please refer to the student handbook for the consequences such a report may have.

### Students with Disabilities

Any student with a documented disability is welcome to contact me as early in the semester as possible so that we may arrange reasonable accommodations. As part of this process, please be in touch with Disability Services for Students Office at 330 Memorial Union, 401-874-2098..

### Academic Enhancement Center

The work in this course is complex and intensive. To do the best you can, it's a good idea to visit the Academic Enhancement Center (AEC) in Roosevelt Hall. The AEC offers a comfortable environment in which to study alone or together, with or without a tutor. AEC tutors can answer questions, clarify concepts, check understanding, and help you to study. You can make an appointment or walk in during office hours -- Monday through Thursday from 9 am. to 9 pm, Friday from 9 am to 1 pm, and Sunday from 4 pm. to 8 pm. For a complete schedule visit http://www.uri.edu/aec/, call (401) 874-2367, or stop by the fourth floor in Roosevelt Hall.

# September 6th 2017

| | |
|---|---|
| What You Will Learn | What is this class all about?<br>What will life be like in CSC 201?<br>What assignments are there and how are they graded? |
| Read / Do Before Class (!) | No reading assignment yet. |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Spreadsheets (Due next Monday, Sep-11th !)**<br><br>Many of you will have used spreadsheets before – some more superficially for quick calculations, some more extensively for budgeting etc.. In this assignment, you will become familiar with Google's online spreadsheet processing tool. Google Sheets has some very constraining limitations, but for basic tasks it can be a great option – especially when sharing work with others. Complete the Google Spreadsheet Tutorial, and download the following sheets as Excel Files (yes, Google lets you easily convert sheets – most of the time) and submit them in Sakai:<br><br>- Checks<br>- Company<br>- Expenses<br><br>To save a Google Sheets document locally on your computer, go to "File" => "Download As" and select your favorite file format. For this assignment, you should select "Microsoft Excel (XLSX)". If you have Microsoft Excel installed on your computer, you should try to open the files just to make sure that they were properly saved.<br><br>***Don't forget to go to Sakai and to upload all three Excel files!*** |

# September 11th 2017

| | |
|---|---|
| What You Will Learn | Create a Google Doc, save it as a Word document,<br>Use spreadsheet software to perform basic mathematics and statistics.<br>Use spreadsheet software to compute values based on related values.<br>Use a spreadsheet to generate charts and graphs to visualize data. |
| Read / Do Before Class (!) | Complete the Spreadsheet Assignment<br>Review the Syllabus and make sure you understand it<br>Read: Data Visualization 101 - How To Design Charts and Graphs |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Trends**<br>*For this assignment you will create either a MS Word document or a Google Doc and answer five questions (A-E). We will call this the "assignment document". When you are done, save the document as MS Word file (DOCX) and submit it to Sakai.*<br><br>When you post information to a social network, watch a video online, or simply search for information on a search engine, some of that data is collected, and you reveal what topics are currently on your mind. When a topic is quickly growing in popularity, it is often said to be trending, but there are many different trends or patterns we might find in this data, including historical trends. These patterns might help us to identify, understand, and predict how our world is changing.<br><br>You will be using Google Trends, which is a tool that allows you to visualize data about search history across different times and locations. You will be looking for interesting patterns, trends, or relationships between multiple trends and try to tell the story that the pattern is showing. Just for the fun of it, perform a trend analysis on "United Airlines". Something about this graph should make you curious. What happened there? You can also make an educated guess of where the most United passengers live by looking at "Interest by Region".<br><br>In your own words answer these questions in your assignment document:<br>A. Where does Google Trends data come from?<br>B. How is Google Trends data adjusted? What does a value of 100 mean?<br><br>Now pick a topic you are interested in and perform a Google Trends analysis. Look for:<br>● Long-term trends: Is your topic becoming more popular over time? Less?<br>● Short-term trends: Does your topic suddenly spike or dip in popularity?<br>● Patterns: Does your topic follow some predictable repeated pattern?<br>● Relationships: Does one topic increase or decrease in popularity when another one does? For this you can "compare" search terms. For example, if "United Airlines" is your main search term, you could compare it to "Delta Airlines" and see what scandals might have occurred there – if any.<br>● Zoom-in: You can narrow your search to particular regions, times, and categories.<br><br>Answer these questions in your assignment document:<br>C. Describe what terms you compared and whether you narrowed your search by using filters.<br>D. Paste in screen shots of the charts and visualizations supplied by Google Trends for the terms you just searched. Accurately describe what the charts or other visualizations are showing.<br>E. Come up with a possible story or explanation of why the trend you described might have happened.<br>***Make sure you save or download your assignment document in DOCX format (MS Word) and submit it in Sakai!***<br><br>***YOU ARE NOT DONE YET! PLEASE CONTINUE ON THE NEXT PAGE*** |

## Visualizations

Good visualizations can help people make sense of data sets that are too large to interpret by looking at the raw data. In this part of the assignment you will learn to use a spreadsheet tool to make visualizations of your own.

There are many different kinds of charts that are used to visualize data. In this lesson, you will learn to make scatter, line, and bar charts, but there are many other types of visualizations that people use to interpret data in different ways. Look through the Data Visualization 101 Guide and familiarize yourself with the different chart types, paying particular attention to the scatter, line, and bar charts.

Google Sheets is constantly evolving. This is often a good thing since new features are added and errors eliminated rather quickly. However, it also means that your experience can change from one day to the next. This is especially noticeable when you are following instructions and suddenly a button has moved or is missing or the menu structure has changed or something was simply renamed. Keep this in mind when you create a number of different visualizations in this tutorial, You should place all of them inside the same spreadsheet. Each visualization should have appropriate ranges, a fitting title, labels and a legend. At the end of the tutorial, download the file in Excel format (XLSX) and submit it in Sakai. It should have the following graphs:

- A scatter chart
- A line chart
- A bar chart

*In this assignment you will continue working on the document you started in the Trends assignment. If you have closed it, repen it in Word or Sheets and answer these three questions (F-H)!*

F. For this scatter chart:
   - What does this scatter chart help you notice about the data?
   - Which age groups had the highest average ratings?
   - Which age groups had the lowest?
   - What other connections and trends can you see?

G. For this line chart:
   - What does this line chart help you notice about the data?
   - For which ages were the average ratings similar between men and women?
   - For which ages were they different?
   - What other connections and trends can you see from this chart?

H. For this bar chart:
   - What does this chart help you notice about the data?
   - i. For which ages were the number of ratings similar between men and women?
   - ii. For which ages were they different?
   - iii. What other connections and trends can you see from this chart?

*Now let's look at some more interesting data. Your instructor will assign you to a small group and give each group access to a CSV with data about a particular topic. You will also receive a readme file with information about your data. Only one of you should import the CSV file into Google Sheets and then share the sheet with the other group members. Each student should then insert the following graphs into the assignment document (labeled as question I -K). Note that you may have to switch between documents or applications several times to accomplish this.*

| | |
|---|---|
| | I. Place in your document a scatter chart similar to the one that you created in the *Do A Visual Check of Scatter Chart* part of the tutorial. It should have an appropriate title that includes your full name, an appropriate legend, appropriate ranges, and appropriate labels. You pick the data that you use for a scatter chart.<br>J. Place in your document a line chart similar to the one that you created in the *Do A Visual Check of Line Chart* part of the tutorial. It should have an appropriate title that includes your full name, an appropriate legend, appropriate ranges, and appropriate labels. You pick the data that you use for a scatter chart.<br>K. Place in your document a bar chart similar to the one that you created in the *Do A Visual Check of Bar Chart* part of the tutorial. It should have an appropriate title that includes your full name, an appropriate legend, appropriate ranges, and appropriate labels. You pick the data that you use for a scatter chart.<br><br>*Finally, answer one more question about your experience with Google Sheets and/or Docs so far:*<br><br>L. What advantages do Google Sheets and Docs have over Excel and Word? What disadvantages could you think of? Overall, which would you rather ruse in everyday life and why?<br><br>***Make sure that you upload the <u>final</u> assignment document to the Visualizations Assignment on Sakai!*** |

# September 18th 2017

| | |
|---|---|
| What You Will Learn | Use a spreadsheet to sort, filter, categorize, and clean data<br>Use a spreadsheet software to perform basic mathematics and statistics.<br>Use a spreadsheet to create a summary table (pivot table) from the data<br>Find datasets on the Internet<br>Import CSV files into Google Sheets. |
| Read / Do Before Class (!) | Complete the Trends and Visualizations Assignments<br>Read: Wikibook chapter on Data Cleaning<br>Read: Wikibook chapter on Data Statistics<br>Read: Wikibook chapter on Obtaining Data<br>Watch: How Google Search Works [3:14]<br>Watch: Google Advanced Search [2:45] |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Cleaning and Processing Data**<br>**Part 1 (Due September 25th)**<br><br>*For this assignment you will create either a MS Word document or a Google Doc. In this "assignment document", you will answer the questions below. When you are done, save the document as MS Word file (DOCX) and submit it to Sakai.*<br><br>Using computational tools to analyze data has made it much easier to find trends and patterns in large datasets. When preparing data for this kind of analysis, however, it's important to remember that the computer is much less "intelligent" than we might imagine. Small discrepancies in the data may prevent accurate interpretation of trends and patterns or can even make it impossible to use the data in computation in the first place. Cleaning data is therefore an important step in analyzing it.<br><br>Follow this tutorial to clean data from a survey that a student took of her classmates asking "What did you do last night?" and insert the following into the assignment document:<br><br>A. A screenshot showing how you filtered the data in one of the columns as described in the *Filter Your Data* part of the tutorial. Add a sentence or two describing what you did and how it filtered the data.<br>B. A screenshot showing how you did more complex filtering into two different columns. In one column choose at least two values. In the other column use a Conditional filter, as described in the *More Complex Filtering* part of the tutorial. Add a sentence or two describing what you did and how it filtered the data.<br>C. A screenshot showing sorted data in a column as described in the *Data Sorting* part of the tutorial. Add a sentence or two describing what you did and how it sorted the data.<br>D. A screenshot showing corrected errors that you can and delete the values that make no sense as described in the *Fixing Errors* part of the tutorial. Add a few sentences describing the errors and what you did to correct them.<br>E. A screenshot showing at least one new column in your dataset that standardizes a column of data collected as "free form text", as described in the *Categorizing Data* part of the tutorial. Add a sentence or two describing how you categorized the data.<br><br>Now let's look at the dataset you were assigned in the "Visualizations" assignment. The Data Cleaning Chapter, and the above tutorial, showed you how to check data for constraints using these spreadsheet capabilities:<br>● Filtering (including conditional filtering)<br>● Sorting<br>● Formatting |

- Search
- Find/replace

Consider each of the following constraints that were described in the Data Cleaning Chapter:

F. Type constraints
G. Range constraints
H. Mandatory constraints
I. Format constraints
J. Uniqueness constraints

Apply each constraint at least once to the data set you were given on your topic if you can. If you can’t find a reasonable field in your data to check the constraint, then state that, but you probably can find at least one instance of each, probably several instances of each. Apply the constraint by using the spreadsheet techniques of filtering, sorting, etc to find bad data and then either fix the bad data item, or eliminate the record with the bad data item.

In your assignment document, address each constraint (F-J) by describing an instance of that constraint that you checked in your data. Describe how you checked for it, report a value in the data that your check found to be noncompliant and explain how you fixed the problem. If no problems were found, state that.

***When you are done, upload your assignment document (in DOCX format) to Sakai!***

## Part 2 (Due October 2nd)
## Summary and Pivot Tables

*For part 2, you will continue the assignment document you created in part 1.*

A summary table is a table used to compute summary statistics of a larger dataset. Summary tables are another way that computational tools can be used to look more closely at data in order to identify trends and patterns. They can often be good data visualizations on their own, and they are quite useful when trying to make charts from larger collections of data.

Vocabulary

- *Summary Table* - a table that summarizes information about some larger dataset. It typically consists of performing computations like sums, averages, and counts on higher level groupings of information. The intent is to summarize lots of data into a form that is more useful, and easier to "see".
- *Pivot Table* - the tool used by most spreadsheet programs to create a summary table.
- *Aggregation* - a grouping of similar data within a dataset for the purposes of performing a calculation on items within each group. For example, if some dataset contained information about how many hours of television people watched and included their age, you could "aggregate the data by age" and compute the average hours watched for each age group. You could also "aggregate by hours of TV watched" and compute the average age for each number of hours.

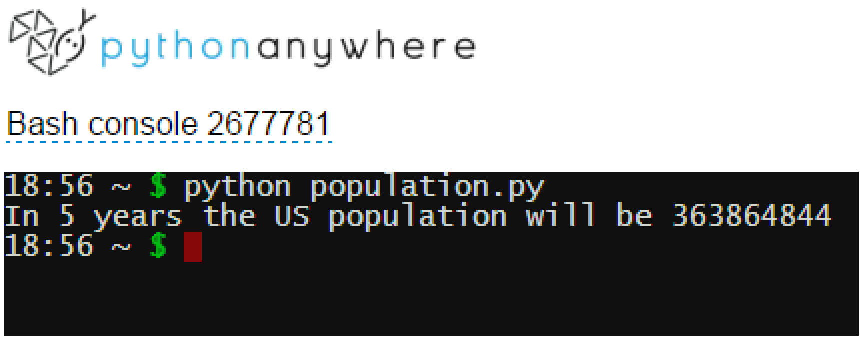
Follow this tutorial to create summary tables from the movie data set in the tutorial. Then add screenshots showing the following to the assignment document:

K. The pivot table that you created in the *Making Summary Tables* section of the tutorial.
L. A table that displays the average rating for every movie listed in the data set as described in the Adding Rows section of the tutorial.
M. A table that displays the counting for the number of ratings for each movie in the data set as described in the *Let's Change the Value -- Summarize by: COUNT* section of the

| | |
|---|---|
| | tutorial.<br><br>N. A table that displays the adding of another Values field as described in the *Add Another Field To Values* section of the tutorial.<br>O. A table that displays the adding of columns that group your data by gender as described in the *Add Columns* section of the tutorial.<br>P. A chart of movies where the differences between male and female ratings are significant as described in the *Manipulating The Pivot Table* section of the tutorial.<br><br>So far we have mostly been working with sample datasets. Now let's make this a little more exciting by analyzing a dataset that you are personally interested in. Use your favorite search engine to locate a CSV file with data about whatever topic you want to look at. Virtually all search engines allow you to specifically look for certain file types, although the exact syntax varies a little. For Google, you can add *filetype:csv* to any search to only look for CSV files (or whatever other file type you might look for in other situations).. You may need to poke around a little to find a set that you can use for this assignment. Once you have found one, add the following to your assignment document:<br><br>Q. The search string that you used to find the data set.<br>R. A screenshot of the first 10 data records (lines of the CSV file) of the data set that you found.<br>S. A screenshot of the data set loaded into Sheets showing the column headers (you don't have to show all columns if they don't fit in the screenshot) and the first 10 or so data records.<br>T. A screenshot of a pivot table that you created based on your new data set. You can freely decide what you want your pivot table to look like.<br><br>*Finally, answer these questions in your assignment document:*<br><br>U. Was the search that yielded your data set a general Google search, or did you do a local search on the website of one of the major data repositories like data.gov or github to find the data set?<br>V. Describe in a few sentences the website on which your data set was found.<br><br>***When you are done, upload your assignment document (in DOCX format) to Sakai!*** |

## September 25th 2017

| | |
|---|---|
| What You Will Learn | Basics of Python syntax, variables, types, and expressions.<br>Write a full Python program using expressions<br>Program with python data types numbers and booleans.<br>Program with python arithmetic operators including +, -, *, /, **, and %. |
| Read / Do Before Class (!) | Read: Wikibook Chapter on Introduction To Programming (Stop at Section 3 "Python Strings")<br><br>Watch: Code Stars [5:43]<br>Watch the PythonAnyWhere intro video [1 min]<br>Watch: PythonAnywhere Programming Environment [12:41]<br><br>Create a free account on CodeAcademy.<br>Do: CodeAcademy Unit 1 Python Syntax, all 13 lessons<br>Do: CodeAcademy Unit 1 Tip Calculator |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Python Expressions (Due October 9th)**<br><br>Begin by creating a free user account on PythonAnywhere. Once you are logged in, make sure to go to Account => Teacher and to enter your instructor's username ( hellwig ). This gives your instructor full access to anything you do on PythonAnywhere.<br><br>Use the PythonAnywhere File Manager to create a file called *us_popluation.py*. Write and execute your program in PythonAnywhere using Python 2.7.<br><br>**U.S. Population Estimation Program**<br><br>Write a program to estimate the population of the United States five years from now based on information found at the US census Population Clock website: http://www.census.gov/popclock/<br>That website specifies the following information:<br>● A number that represents the current population.<br>● 3 rates:<br>○ Births per second<br>○ Deaths per second<br>○ Immigrations per second<br><br>Predicting into the future using only these numbers is not very accurate since the rates are likely to change, but it does give you a general idea for short predictions (only a few years into the future). Longer predictions (100's of years) are likely to be more inaccurate.<br><br>**Program Specifications**<br><br>Your program will define an integer variable for the number of years into the future you are predicting (5 years for the assignment, but feel free to play with it). Your program will print out the predicted population size as an integer. You must use the values from the US Population Clock web site and add a comment to your python program stating the date and time you retrieved those values from the US Census site.<br><br>Use the following information about time conversion:<br>● 365 days in a year (not quite true due to leap years, but let's use it anyway)<br>● 24 hours in a day<br>● 60 minutes in an hour |

| | |
|---|---|
| | ● 60 seconds in a minute<br><br>Hint - create three variables: one for births, deaths, and immigrations over the number of years your program is calculating.  Then create a fourth variable *expectedPopulation* for the population estimate and set it to an expression which calculates the expected population by using the current population adding births and immigration over the years and subtracting deaths over the years.<br><br>Use the python 2.7 print command<br>*print "In 5 years the US population will be %d" % (expectedPopulation)*<br>to print the value of the variable *expectedPopulation* to the screen.<br><br>**Sample Output**<br><br><br>pythonanywhere<br>Bash console 2677781<br>`18:56 ~ $ python population.py`<br>`In 5 years the US population will be 363864844`<br>`18:56 ~ $` |

# October 2nd 2017

| | |
|---|---|
| What You Will Learn | Create strings three ways in python: with single quotes, double quotes, and the str function.<br>Program with python string functions len, upper, lower.<br>Print strings and variables with various formats. |
| Read / Do Before Class (!) | Complete the Data Processing and Cleaning Assignment<br>Read: Wikibook Chapter on Python Strings (stop at "Python Conditionals")<br>Read/Do: CodeAcademy Unit 2: Strings and Console Output, all 16 lessons<br>Do: CodeAcademy Unit 2: Date and Time |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Python Strings**<br><br>Use the PythonAnywhere File Manager to create a file called *csv_strings.py*. Write and execute your program in PythonAnywhere using Python 2.7.<br><br>**CSV Files**<br><br>We have worked with CSV files before, but we have not looked at what they actually are. CSV in an acronym for "comma separated values". It is a file format commonly used to store data in plain text, so that many applications including spreadsheets and databases can read them. A CSV file consists of lines where each line has values separated by commas. For instance, the data on latitutes, lognitudes, and time zones of cities in the world at http://www.infoplease.com/ipa/A0001769.htmlhttp://www.infoplease.com/ipa/A0001769.html displays as a table. To store and transmit that data a CSV would be a common choice. The first few lines of a CSV might have lines that look like:<br><br>Aberdeen, Scotland, 57, 9N, 2, 9W, 5:00 p.m.<br>Adelaide, Australia, 34, 55S, 138, 36E, 2:30 a.m.<br>Algiers, Algeria, 36, 50N, 3, 0E, 6:00 p.m<br><br>with hundreds of similar lines for all of the cities.<br><br>The latitude and longitude are shown using GPS coordinates. The time is shown corresponding to 12:00 noon, Eastern Standard Time.<br><br>**Program Specifications**<br><br>Use http://www.latlong.net/ to determine the latitude and longitude of your hometown as well as a city/town in the US where the city name starts with the first letter of your first name (e.g. if my name is "Vic", I might pick Victoria, California and determine that the latitude rounded to the nearest integer is 48 and the longitude is -123. Use the single latitude number and single longitude number, not the GPS coordinates. Round the latitude and the longitude to the nearest integer.<br><br>In your program make 8 variables:<br>● my_town - the name of your home town as a string<br>● my_state - the name of your home state as a string<br>● my_lat - the latitude of your hometown rounded to the nearest integer<br>● my_long - the longitude of your hometown rounded to the nearest integer<br>● us_town - the name of the town that shares your name's first letter as a string<br>● us_state - the name of the town that shares your name's first letter as a string<br>● us_lat - the latitude of the town that shares your name's first letter rounded to the nearest integer<br>● us_long - the longitude of the town that shares your name's first letter rounded to the |

nearest integer

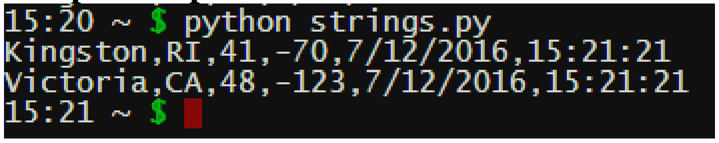
Add two print statements to your program:

1. One to print out a line with 6 data values: hometown, home state, home town latitude, home state longitude, the current date in mm/dd/yyyy form and the current time in hh:mm:ss form. Have all 6 values separated by commas, as in a CSV file (see sample output below).
2. One to print the same data for the town that shares your name’s first letter (and also the current date and time).

Print only the data values as specified above, no other text. For instance, do not print “My home town is” - just print the town name, then print the comma, then print the state name - all in one line on the screen, which may wrap to a second line if you have a narrow screen window. See the sample output below.

Note that this is similar to how applications like spreadsheets and databases produce CSV files from the data that they store and manage.

## Sample Output



```
15:20 ~ $ python strings.py
Kingston,RI,41,-70,7/12/2016,15:21:21
Victoria,CA,48,-123,7/12/2016,15:21:21
15:21 ~ $
```

## October 9th 2017

| | |
|---|---|
| What You Will Learn | Columbus Day – There is no class, BUT there is an assignment due!!!! |
| Read / Do Before Class (!) | Complete the Python Expressions Assignment! |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | Continue workingon the Python Strings Assignment |

## October 16th 2017

| | |
|---|---|
| What You Will Learn | Program with Python Comparators <, >, <=, >=, !=, ==.<br>Program with Python Boolean operators and, or, not.<br>Program with Python conditional statements formed with if, elif, else.<br><br>Read: Wikibook Chapter on Python Conditionals<br>Read/Do: Code Academy Unit 3: Conditionals and Control Flow all 15 lessons<br><br>Do: Code Academy Unit 3: PygLatin<br><br>Watch: Bill Gates Explains Conditional Statements [1:46] |
| Read / Do Before Class (!) | Complete the Python Strings Assignment |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Data Cleaning with Python Conditionals**<br><br>Use the PythonAnywhere File Manager to create a file called *conditionals.py*. Write and execute your program in PythonAnywhere using Python 2.7.<br><br>**Data Cleaning**<br>When data values are collected, errors can occur. For example consider a case where you collect data on people's favorite color using a Google Form survey on the Internet that has two questions like this:<br>*Your age:*<br>*Your favorite color*:<br>Where people can type in any text as their answers. For age, you would expect an integer, but you might get answers like:<br>● 5 (invalid, too young, maybe they left out a digit)<br>● 16 (a valid answer)<br>● 21years (a valid integer but they added the unit "years")<br>● 17.6 (a valid answer but they assumed fractions were OK)<br>● -18 (a valid integer, but negative ages are invalid)<br>● 917 (a valid integer but they might have accidently typed the 9 before putting their age)<br>If your program is doing calculations, many of these inputs could cause problems, such as the program not working, or generating wrong answers. For instance, if you wanted to know the average age of the people responding, if your calculations allowed the 917 value, the average age calculation could yield a value that is much too high because that 917 value skewed the result.<br><br>Data scientists have several ways to handle bad data values. One is to discard the data value. Another is to make an assumption of what the right data value is and clean it. In the example above, a data scientist could do the following:<br>● 5 - discard it under the assumption we expect respondents to be 10 years of age or older. |

- 16 - accept it as valid data.
- 21years - remove the “years” to yield a data value of 21.
- 17.6 - truncate the fractional part to yield a value of 17.
- -18 - this is not a clear decision. The data scientist could assume the leading negative sign was a mistake and remove the leading sign to yield a data value of 18, or she could just discard the value as invalid.
- 917 - discard this as being an invalid age.

### Program Specifications

Write a Python program that accepts input from the user meant to represent an age. The program applies these data value rules:

A. Age data values must be positive two-digit integers from 10 to 99.
B. Input that contains a number followed by text must have the text removed. For example “21years” must yield 21.
C. Input that is a number with a fractional part must be truncated to the integer part. For example 17.6 must yield 17, and 23.9 must yield 23.
D. Any other input is invalid.

More than one rule may be applied. For instance, the input *17.6 years old* should yield a data value of 17 after the “years old” is removed and 17.6 is truncated to 17.

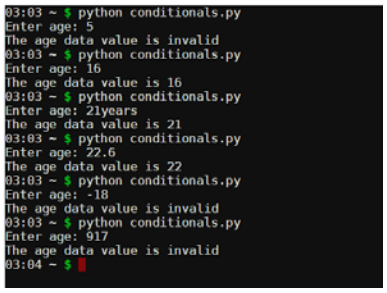
After accepting the input and applying the rules, your program should print “The age data value is” and then either the valid data value, or the word “invalid”.

Here is an outline to the program

```
# accept the input into a variable called “original”
# set a variable “valid” to be True. If any rules A-D from
above
  Are violated, set “valid” to False.
# test if “original” is less than 2 characters long. If so,
  set “valid” to False
# set a variable “age” to be the first two characters of
“original”
# if any characters in “age” aren’t digits (Google Python’s
isdigit()
  function), then set “valid” to False.
# if “original” is more than 2 characters long and the
third character
  is a digit, then it is something like 917, so set “valid”
to False
# Finally, if “valid” is still True, then print the data
value,
  otherwise print it as invalid.
```

Test your program using the sample input listed above and see that it does the right thing. Make up your own test cases as well.

### Sample Output



```
03:03 ~ $ python conditionals.py
Enter age: 5
The age data value is invalid
03:03 ~ $ python conditionals.py
Enter age: 16
The age data value is 16
03:03 ~ $ python conditionals.py
Enter age: 21years
The age data value is 21
03:03 ~ $ python conditionals.py
Enter age: 22.6
The age data value is 22
03:03 ~ $ python conditionals.py
Enter age: -18
The age data value is invalid
03:03 ~ $ python conditionals.py
Enter age: 917
The age data value is invalid
03:04 ~ $
```

## October 23rd 2017

| | |
|---|---|
| What You Will Learn | Define Python functions including those with parameters and return values.<br>Call Python functions including those with parameters and return values.<br>Define Python functions that call other Python functions.<br>Import Python functions from libraries - either the entire library or just single functions.<br>Use Python functions from libraries such as sqrt, abs, max, and min from the math library.<br>Write a full Python Program using functions |
| Read / Do Before Class (!) | Complete the Python Conditionals Assignment<br><br>Watch: Chris Bosch Explains Functions [1:00]<br><br>Read/Do: Code Academy Unit 4: Functions all 19 lessons<br>Read: Wikibook Chapter on Python Functions (Stop at “Python Lists and Dictionaries)<br><br>Do:  CodeAcademy Unit 4: Taking A Vacation |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Data Processing With Python Functions**<br><br>**Part 1: Data Processing Program**<br><br>Use the PythonAnywhere File Manager to create a file called *functions.py*.  Write and execute your program in PythonAnywhere using Python 2.7.  Your program must have:<br><br>● A function called `clean_age_data` that accepts a string representing the age data from a survey form, cleans the data using the rules from the Conditionals Assignment, and returns a string which is a valid age, or returns “invalid” if the age in invalid. Your code from the Conditionals Assignment is a good starting point.<br>● A function called `make_csv_line` that accepts three parameters: `first_name`, `last_name`, `age`. This function should call your `clean_age` function on the `age` value that is passed in.  If the age is valid, the function should return a string representing a CSV file line with the values for `first_name, last_name`, `clean_age` with `clean_age` being the cleaned age, and the three values separated by commas.  That is, return all three values separated by commas in one string.  If the age in invalid, then `make_csv_line` function should return the string “invalid”<br>● Call your `make_csv_line` function with seven names (first name and last name pairs) that you make up and use the following age values (without the commas, use the exact value provided here - e.g. call it with “21years” and let the `clean_age_data` function remove the “years”): 5, 16, 21years, 22.6, 19, -18, 917.  Print the result of each function call on a line in the output.  That is, when you run the program, 7 lines should be printed with each line either being a CSV line or the string “invalid”<br><br>**Part 2: Python Plotting Library**<br><br>The CodeAcademy tutorials showed you how to import libraries. It had you import the *math* library and print the functions available in it. Some python libraries are made up of several libraries, called sub-libraries. A very useful library in data science is the sub-library *matplotlib.pyplot*.  Search the Internet and find a description of what the sub-library *matplotlib.pyplot* does. |

| | |
|---|---|
| | Use the PythonAnywhere File Manager to create a file called *libraries.py*. Write and execute your program in PythonAnywhere using Python 2.7. Your program must have:<br>● Python comments to describe matplotlib.pyplot in 1-3 sentences at the beginning of your program.<br>● Code similar to the code in CodeAcademy Lesson 11 “Here Be The Dragons” from their Unit 4 Python Functions chapter. Instead of printing the functions in the *math* library as their code does, your code should print the functions in the *matplotlib.pyplot* sub-library. Hint substitute *matplotlib.pyplot* for *math* in their code.<br>● You will see many functions displayed when you run the code from the previous step. The Matplotlib Plotting Commands documentation provides a description of each function. Add Python comments at the end of your program that tell the name of the *plyplot* functions that:<br>○ Make a bar plot<br>○ Plots a histogram<br>○ Plots a pie chart |

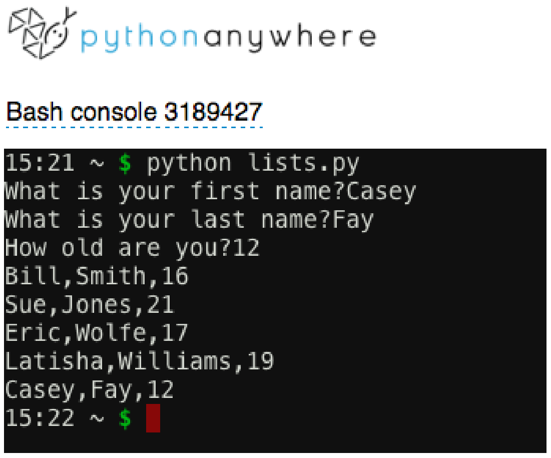
## October 30th 2017

| | |
|---|---|
| What You Will Learn | Program using Python Lists and Python dictionaries<br>Adding items, removing items, and finding items by index (lists), or key value (dictionaries).<br>Use a Python for loop to do something to every item in a list or in a dictionary.<br>Write Python functions with loops, lists, and dictionaries.<br>Update data in a Python program in response to changes in the environment. |
| Read / Do Before Class (!) | Complete the Python Functions Assignment<br><br>Read: Wikibook Chapter on Python Lists and Dictionaries (Stop at "Python Loops")<br>Read/Do: Code Academy Unit 5: Lists and Dictionaries all 14 lessons<br>Do: CodeAcademy Unit 5: A Day At The Supermarket<br>Read/Do: Code Academy Unit 6 Student Becomes The Teacher |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Data Processing With Python Lists**<br><br>When working with data sets, Python programs read data in and store it in data structures, like the lists and dictionaries that you learned about in CodeAcademy. Keeping the data in data structures in the program makes it easier to work with the data to do things like find statistics and save the data in CSV format. Use the PythonAnywhere File Manager to create a file called *lists.py*. Write and execute your program in PythonAnywhere using Python 2.7. Your program must have:<br><br>● Declare these list variables, each with seven values in the list:<br>○ `firstNames` : a list with seven string variables of first names. Make the first firstname be your own first name.<br>○ `lastNames` - a list with seven string variables of last names. Make the first lastname be your own lastname.<br>○ `ages` - a list of strings (not integers!) which will have in it: X,5, 16, 21years, 17.6, 19, -18, 917. Where instead of X put your actual age.<br><br>These lists must be in the same order. That is, the first name at index 0 should correspond to the last name at index 0 and the age at index 0; and so on. Each list is a list of strings so don't forget the single quotes around each data value.<br><br>● Read input from the user to accept another first name, last name, and age. Add this data to the data lists of the program (make a total of 8 names with their age).<br><br>● Write a for loop to go through the 8 names and print a CSV line with first name followed by last name followed by age, for all names that have valid ages. If an age is not valid and can't be fixed (e.g. a negative number can not be fixed, but "21years" should be fixed by removing the "years" and then using the 21), then do not print the data record. Use your code from the Conditionals assignment to check for valid ages.<br><br>For instance, the data:<br>First names: Tim, Bill, Sue, Eric, Latisha, Juan, Jen<br>Last names: Hill, Smith, Jones, Wolfe, Williams, Gomez, Johnson<br>Ages: 5, 16, 21years, 17.6, 19, -18, 917<br>And the input from the user:<br>Casey<br>Fay<br>12 |

Would yield output:

Bill,Smith,16
Sue,Jones,21
Eric, Wolfe, 17
Latisha, Williams,19
Casey, Fay, 12

**Sample Output**



Bash console 3189427

```
15:21 ~ $ python lists.py
What is your first name?Casey
What is your last name?Fay
How old are you?12
Bill,Smith,16
Sue,Jones,21
Eric,Wolfe,17
Latisha,Williams,19
Casey,Fay,12
15:22 ~ $
```

# November 6th 2017

| | |
|---|---|
| What You Will Learn | Program with a Python while loop.<br>Program a Python for loop that does something for every element in a list, dictionary, or range.<br>Break Python loops with break.<br>Add else to Python loops.<br>Write a Python program that works on a list of numbers and has functions to do the sum, average, standard deviation, and variance of those numbers. |
| Read / Do Before Class (!) | Complete the Python Lists Assignment<br><br>Read/Do: Code Academy Unit 8: Loops all 19 lessons<br>Read: Wikibook Chapter on Python Loops (Stop at "Python File I/O") |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | Complete CodeAcademy Unit 8<br>Submit a screenshot of your basge. Your name must be legible,<br><br>If you like, you may take a peek at next week's assignment and get started on that as well. You should complete next week's reading assignments first. |

# November 13th 2017

| | |
|---|---|
| What You Will Learn | No class today, BUT there are readings and assignments !<br><br>Write Python programs that perform file input and output by opening files, reading from files, writing to files, and closing files. |
| Read / Do Before Class (!) | Read/Do: Code Academy Unit 12: File Input and Output all 9 lessons<br>Read: Wikibook Chapter on Python File Input/Output<br>Read: Wikibook Chapter on Data Cleaning in Python (through Chapter 3.3 only) |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Data Cleaning In Python** |

## Data Cleaning In Python

We have been working with CSV format where data is shown as *data records*, one data record per line, with values separated by commas. For instance:

Latisha, Williams,19

Is a data record with data values ‘Latisha’, ‘Williams’, and ‘19’ representing that Latisha Williams is 19 years old.

One of the amazing things happening in our world now is that we can get data on just about anything from Internet sources.  The most common format to find data is in CSV text files. Here is part of a CSV file that was generated by surveying people on how they rated a movie from 1(lowest) to 5(highest).

```
age,avg rating,number of ratings,avg rating women,number of
women,avg rating men,number of men
10,3.59,22,,,3.59,22
11,,20,,,2.85,20
13,3.45,137,2.98,40,3.65,97
14,2.64,84,2.64,84,,
fifteen,3.23,275,3.42,163,2.95,112
16,3.28,256,3.61,62,3.17,194
17,3.64,530,3.44,36,3.65,494
18,3.78,1350,3.91,665,3.65,685
19,3.37,2577,3.26,479,3.39,2098
20,3.71,2797,3.84,804,3.66,1993
210,3.49,1910,3.40,521,3.52,1389
22,3.19,2607,2.51,891,3.55,1716
23,3.30,1825,3.14,340,3.34,1485
```

Each line is a data record for information about the ratings in an age group. Each data record has seven values separated by commas. Some of the data values may be blank, which is shown by nothing between two commas. The first line has text strings which are the “headers” describing what each data value means.  For instance, the first value on the header record is “age” indicating that the first value on each line is the age group.  This means that on the second line the 10 that we see as the first value means that line is information about the movie ratings from people 10 years old.  Look at the line for 18 year olds (in red), it means:

- Age: 18
- Average rating: 3.78
- Number of people rating the move: 1350
- Average rating among women: 3.91
- Number of women rating it: 665
- Average rating among men: 3.65
- Number of men rating it: 685

CSV format takes out all the labeling and just stores data so that it is easy for computers to read it.
 Note that there are intentional errors in this data set:

- The age value for age 15 is not an integer
- age 21 has the value 210
- the average rating for age 11 is missing

You will write a program to catch these errors.

In the CodeAcademy tutorials, you saw that we want data in lists and dictionaries in our programs so that we can work with it. In a previous CodeAcademy tutorial you saw how to import libraries like the *math* library so your program has functions it can use to do specific things (like many math functions when you imported the *math* library). Getting data from CSV data files is so important, that Python provides a library called *csv* to make it easy.

**Create the CSV data file.** In your PythonAnywhere file manager, create a text file called *myMovieRatings.csv* . Copy the ratings data from this document (shown above) and paste them into my*MovieRatings.csv*. Be sure to include the header row of data, and to leave the errors.

**Write a program to import the CSV data and check data constraints.** Create a program called *movieRatingAverage.py* using the base code from the wikibook chapter on Data Cleaning In Python Section 1.

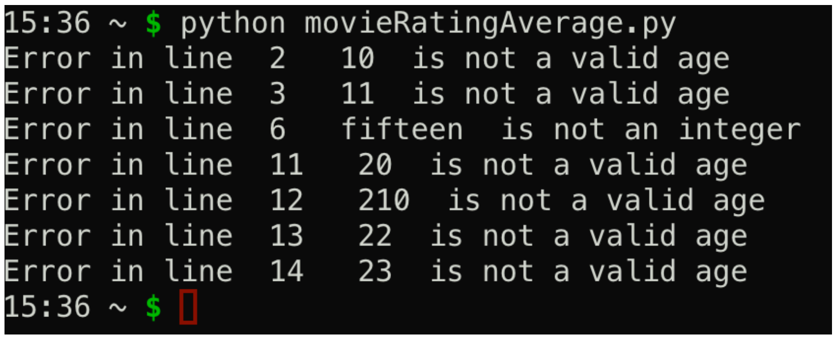
Modify the *movieRatingAverage.py* program to check these constaints:

- The age must be present and be an integer between 13 and 19
- The average rating must be present and be a float between 0 and 5

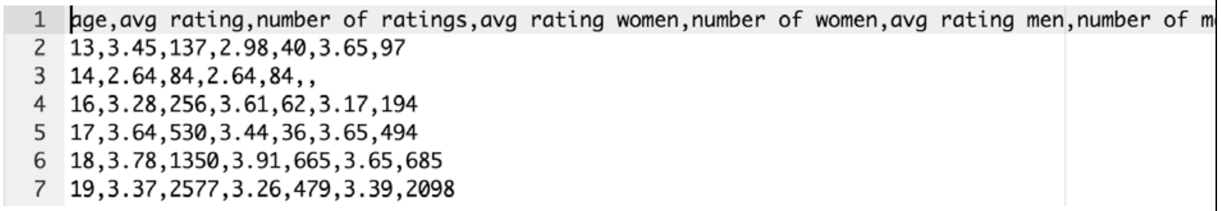
The wikibook chapter on Data Cleaning In Python Sections 3.1, 3.2, and 3.3 provide code on how to check for these constraints.

Your program must alert the user using Python print statements of all these errors that it finds. It must also create a new file called *myMovieRatingsNew.csv* that has all of these bad data records removed.

## Sample Output



```
15:36 ~ $ python movieRatingAverage.py
Error in line  2   10  is not a valid age
Error in line  3   11  is not a valid age
Error in line  6   fifteen  is not an integer
Error in line  11   20  is not a valid age
Error in line  12   210  is not a valid age
Error in line  13   22  is not a valid age
Error in line  14   23  is not a valid age
15:36 ~ $
```



```
age,avg rating,number of ratings,avg rating women,number of women,avg rating men,number of m
13,3.45,137,2.98,40,3.65,97
14,2.64,84,2.64,84,,
16,3.28,256,3.61,62,3.17,194
17,3.64,530,3.44,36,3.65,494
18,3.78,1350,3.91,665,3.65,685
19,3.37,2577,3.26,479,3.39,2098
```

# November 20th 2017

| | |
|---|---|
| What You Will Learn | Write a Python program to perform statistics on values in a CSV with Python including mean, median, mode, standard deviation, and counts of values. |
| Read / Do Before Class (!) | Complete the Data Cleaning with Python Assignment<br>Re-Read: Data Statistics<br>Read:Wikibook chapter on Data Statistics in Python<br>Do Unit 9 Exam Statistics all 9 parts |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Data Satistics**<br><br>In this assignment, you will read a CSV file using real world flight data from January 2017. You will learn how to filter data and calculate averages. You will also calculate a simple probability.<br><br>**Getting the Data**<br><br>The Bureau of Transportation Statistics provides an incredible choice of transportation related data including commercial aviation statistics. Later in the semester you will obtain your own data from there (https://www.transtats.bts.gov/homepage.asp). But for today we want to all work on the same data set. I have provided one here: pvdFlights.csv . Make sure you upload this file to your Python Anywhere account.<br><br>**Take a Look!**<br><br>Whenever you get a new set of data, you should take a look at it. Open it in Notepad++ or TextWrangler (or even Excel if you like). Look at the headers to get a feel for what kind of information you have available. You will need much of it throughout this assignment.<br><br>**Reading the Data**<br><br>We have read CSV files before. Refer to the previous assignments if you need a refresher. For this assignment you will create a new program called *flightStats.py* that will answer a few simple questions about the data. Your program should just print out the responses to the console (using the *print* command).<br><br>**Answering the questions**<br><br>You should only create one python file and submit the final version. For example, in question 1a we are asking you to modify your code from question 1. In your final submission we only need to see the final code – we do not care about what code you had after question 1, but before you completed question 1a. You **must** complete the questions in order since later questions depend on earlier ones!<br><br>**Question 1: How Many Flights?**<br><br>Create a loop to count how many flights departed from Providence (PVD) to Fort Lauderdale (FLL) in January. Print out a response saying something like “In January, 25 flights operated from PVD to FLL”. Only count flight that went TO Fort Lauderdale.<br>If you take a close look at the data, you will notice that some lines are missing entries. This is not a result of unclean data (BTS data is very clean), but usually means that the flight was cancelled. You can check the “Cancelled” column to make sure. At this point you need not worry about that. We will address it later.<br><br>**Question 1a: Custom Destination**<br><br>Adapt your code in a way that allows the user to select a destination airport. You can do this using the input command: |

```
destination = input(“Please supply a destination airport”)
```

(Hint: For those of you who are as lazy as I am and who just copy and pasted that line, you may need to replace the quotation marks. Depending on your text editor, it may have used “smart quotes” which Python does not understand.)
Your code should print out an answer very similar to that in question 1, only now it needs to provide the flight count and the proper destination airport. Like “96 flights went to ATL” for example.

## Question 2: How late?

Many of you will have experienced the joy of dealing with delayed flights. So if you are going to Chicago O’Hare (ORD), what is the average delay of your flight going to be? Create a loop that looks at each flight to Chicago and adds up the arrival delay in a variable. You will need a second variable to count the flights to Chicago because at the end you should display the **average** delay. Your ouput should be something like “The average arrival delay of flights from PVD to ORD was 6 minutes”.

You may notice that some delays are negative. This happens when a flight arrived early. While this is awesome for the passengers, it doesn’t help the poor souls on the delayed flight. Hence in aviation we don’t care about early flights. You should ignore them and **not** add them to the variable you are using to calculate total delay.

## Question 2a: A little more detail

Modify your code from question 2 to allow the user to specify a particular airline. (Use the input command again). Notice that we are using airlines’ IATA codes in this data! Also adapt your code to look at flights to any airport the user might have supplied under question 1a.

## Question 3: What are the odds?

To make things a little more interesting, we would like to know how likely we are to get to our destination on time. Building on your previous answers, print out a sentence like “Out of the 23 flights to ORD, 16 were delayed”.

## Question 3a: Cut them some slack

Nobody really cares about a tiny delay of 5 minutes, right? (Well, actually I do when I have a tight connection…) Usually we can consider any flight that arrived within 5 minutes of its scheduled arrival time as “on time”. Adapt your code from question 3 to only count flights a slate when they were delayed by more than 5 minutes.

## Question 3b: You’re going nowhere

Oh, how frustrating it can be! They tell you your flight is 20 minutes late. Then it becomes 60. Two hours. And then you get the bad news. Your flight has been cancelled. How likely are you going to run into this problem in Providence? Let’s find out! Count all the flights by the selected airline that were supposed to go to the selected destination and count how many of those were cancelled (look at the CANCELLED column). Print out something like “12 % of the flights were cancelled”. Yes, it would be nice to see a percentage here!

## Finally: Bring it all together

Use what you have written so far to complete the program. Ultimately it should ask the user two questions: “Where are you going?” and “What airline are you flying?”. It should print one final response saying something like:
“AA was supposed to operate 124 flights from PVD to CLT. 16 of those (13%) were canceled. 24 (19%) were delayed by more than 5 minutes. The average delay was 17 minutes.”

## November 27th 2017

| | |
|---|---|
| What You Will Learn | Use Python to create a simple visualization of data read from a CSV file |
| Read / Do Before Class (!) | Complete the Python Statistics Assignment<br><br>Read: Wikibook chapter on Visualizing Data With Python |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Data Visualization Assignment**<br><br>In the material for this week, you have learned how to visualize two different series of data in the same graph. Let's put this skill to work by visualizing the temperature and the dew point for January 2017. I have supplied the data for you here:<br>pvdWxJan.csv<br><br>This is a very comprehensive report. We are only interested in the daily average temperature. For this assignment we will use "wet bulb" measurements. We also want to look at daily average dew points. You will notice that there are many observations for each day and that only one of them lists the averages for each day. This makes sense since we don't know averages until the end of the day.<br><br>Your job is to create a line graph with the days of the month (1-31) on the abscissa (fancy Math speak for the x-axis). We want to see two lines – one representing the average wet bulb temperature and one showing the average dew point.<br><br>When you are done, upload your python file on Sakai and also leave it on PythonAnywhere for us to test. |

## December 4th 2017

| | |
|---|---|
| What You Will Learn | Write a Python program to convert JSON data to CSV format<br>Write a Python program obtain data from a web service application programming interface (API) |
| Read / Do Before Class (!) | Read: Wikibook chapter on Obtaining Data With Python |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Getting data automatically**<br><br>**Part 1: JSON Conversion**<br>Find a data set that is relevant to your topic in the JSON format by doing a google search that includes the term *filetype:json* Write a python program called *json_to_csv.py* to convert the data set to a CSV file. Call your file *my_csv.csv.*<br><br>**Part 2: Graphing Stock Prices**<br>- Write a Python program called *stocks.py* to access Yahoo Finance's API to get stock price data for the last two months for a stock whose code starts with the first letter of your first name. For instance, if your name is Mary, you could use Microsoft's stock price (code MSFT).<br>- Have your program write the stock data to a CSV file called *stock.csv.*<br>- Open the CSV file in Google Sheets and generate a line graph showing the stock's price per day (y axis) over the last two months (each day on the x axis). Insert your graph into the spreadsheet. Save the Google sheet (data and graph) with the name *stock* on your Google drive.<br>- Submit the spreadsheet to Sakai in Excel format (XLSX) |

## December 11th 2017

| | |
|---|---|
| What You Will Learn | We will discuss the final project |
| Read / Do Before Class (!) | Complete the JSON assignment<br>Work on the final project |
| Assignment (Usually due the following week – please refer to the syllabus for due dates!) | **Final Project (Due Dec 18th)** |