UNIVERSITY OF SASKATCHEWAN

**Department of Computer Science**
176 Thorvaldson Building
110 Science Place, Saskatoon, SK, S7N 5C9, Canada
Telephine: (306) 966-4886, Facimile: (306) 966-4884



# Assignment 1

## Getting Started

**Date Due: February 7, 2020, 5pm** **Total Marks: 56**

### Version History

- **28/01/2019**: released to students

### General Instructions

- **This assignment is individual work.** You may discuss questions and problems with anyone, but the work you hand in for this assignment must be your own work.
- Each question indicates what to hand in.
- **Assignments must be submitted to Moodle.**
- Assignments will be accepted until 11pm without penalty.

# Software

This assignment is intended to make students familiar with data processing tools in Python 3, using Jupyter Notebooks, and various standard Python modules, including NumPy, Pandas, and Sci-Kit. These tools are installed on all Computer Science computers. For students who want to download and install these tools on their personal (or lab) systems, it's easy enough to do. The Anaconda 3 distribution gives you everything we'll probably need in this course, including Jupyter, and all the Python modules demonstrated so far.

- Anaconda 3: `https://docs.continuum.io/anaconda/install`

You can also use the Computer Science Jupyter Lab, which is a cloud-based installation of the tools.

- `https://trux.usask.ca`

Sign in with your NSID and password.

**Note:** Throughout this preamble, I will link to on-line tutorial resources that seem pretty good to me at the time. There may be others, and maybe be better ones, too. I've tended to prefer tutorials that are connected to the origin of the software.

Most tutorials assume you haven't installed anything yet, and they tell you to download and install software. If you're working on departmental computers, you don't have to download or install anything. If you are working on your personal computer, you may have to install some things. As mentioned above, the Anaconda 3 distribution has most if not all of the stuff we'll use. Most importantly, if you are using Anaconda, don't do anything with a tool called `pip`, unless at the very utmost end of need. These two (`pip` the Anaconda tool `conda`) work at cross-purposes, and it can be confusing. If you are going to install extra packages, first look for installation scripts that mention `conda`.

A common theme in what follows is this: Don't learn these tools by studying them. Figure out when you would use each one, and then look up things you want to do. Imitate examples, and adapt them to your situation.

## Python

The Python language (Python 3) is friendly and easy to learn if you already know some programming language. **You will not need to implement complicated algorithms in this course.** We use Python only to open files, lightly manipulate data, and send data to learning algorithms or visualization tools. You don't need to be a deep Python expert.

If you need a Python tutorial or refresher, there are many on-line tutorials. Here's one: `https://docs.python.org/3/tutorial/` The most important sections:

- Basic syntax: Sections 2-5
- Useful practicalities: Sections 6, 7, 8.1-8.2, 9.1, 10.5, 10.6

## Jupyter and Notebooks

Jupyter is a platform for interactive computing, and while it had its origins with Python, it now includes many other systems and languages. Jupyter Notebooks allow us to combine Python code, descriptions, documentation, and visualizations in a kind of unified report document. The Anaconda 3 distribution includes Jupyter Notebooks.

If you need a Jupyter tutorial, here's a link: `https://jupyter-notebook.readthedocs.io/en/stable/`

This assignment requires you to submit the Jupyter Notebook file (file extension is `.ipynb`), as well as a PDF file. The most reliable way to create a PDF is to Export the notebook to HTML using the Jupyter menu: `Jupyter | File | Download as | HTML`. Then open the HTML file, and print/save as a PDF. Notably, the export menu shows LaTeX and PDF via LaTeX as options, but these are not really complete options right now. They are a bit primitive.

## Numerical computing with NumPy

While Python is very friendly, it can be slower than other languages like C or Java. The NumPy package implements a lot of matrix-based operations and other numerical algorithms in a way that gets the most out of modern computing: a nice language combined with an efficient library.

If you need a NumPy tutorial, try this one: https://numpy.org/devdocs/user/quickstart.html The most important sections:

- The Basics
- Copies and Views
- Less basic, Fancy indexing

The key concept is that you can write programs that involve calculations for huge numeric datasets without the need for loops. The trick is to realize when you can do that, and then to figure out how.

That said, there's too much to learn here. Be familiar with the basic approach, and look up what you need as you go.

## matplotlib for plotting data

This collection of packages gives Python users a lot of control over plotting visualizations. The MatPlotLib package is enormous, and intimidating. Don't try to learn it all. Try to find examples of what you want to do, and adjust to your purposes. You can do a lot just by imitating and cobbling ideas together. Here's a link to official user documentation: https://matplotlib.org/tutorials/index.html

It's so enormous and intimidating that packages that work on top of matplotlib are quite common. One popular package is called Seaborn https://seaborn.pydata.org/tutorial.html

## Pandas for mixed data

Pandas is a Python package for data that is not all numeric. It's got a lot of useful tools. Here's a link to an official tutorial site: https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html

## Question 1 (1 point):

**Purpose:** Jupyter Notebook

**Degree of Difficulty:** Basic

**References:** `https://jupyter-notebook.readthedocs.io/en/stable/`

Open the document `A1Q1.ipynb` using Jupyter Notebook, and complete the given task.

### Errata

1. None so far!

### What to Hand In

- A PDF document exported from Jupyter Notebook, containing original document, with your name and student number.

### Evaluation

- 1 mark: Your version of the notebook is named A1Q1.pdf, and it contains your name and student number at the top.

## Question 2 (4 points):

**Purpose:** To demonstrate the level of Python programming needed.

**Degree of Difficulty:** Basic

**References:** https://docs.python.org/3/tutorial/

Open the document `A1Q2.ipynb` using Jupyter Notebook, and complete the given task.

### Errata

1. None so far!

### What to Hand In

- Your PDF version of this notebook named A1Q2.pdf, containing solutions to Tasks 1 and 2, and your name and student number at the top.

### Evaluation

- 2 marks: Your code cell for Task 1 uses Python (only – no imported modules) to show the given output.
- 2 marks: Your code cell for Task 2 uses Python (only – no imported modules) to show the given output

## Question 3 (10 points):

**Purpose:** Python and Numpy

**Degree of Difficulty:** Basic

**References:** https://numpy.org/devdocs/user/quickstart.html

Open the document `A1Q3.ipynb` using Jupyter Notebook, and complete the given task.

### Errata

1. None so far!

### What to Hand In

- Your PDF version of this notebook named `A1Q3.pdf`, containing solutions to Tasks 1 and 2, and your name and student number at the top.

Be sure to include your name, NSID, student number, and course number at the top of all documents.

### Evaluation

- 1 mark: Your answer two Task 1 Part 1 was correct.
- 2 marks: Your answer two Task 1 Part 2 was correct.
- 4 marks: Your code cell for Task 2 Part uses Python and Numpy to calculate the 30th Fibonacci number correctly.
- 3 marks: Your explanation for the behaviour of this script for large N is correct.

## Question 4 (10 points):

**Purpose:** Python and Numpy

**Degree of Difficulty:** Basic

**References:** https://numpy.org/devdocs/user/quickstart.html

Open the document `A1Q4.ipynb` using Jupyter Notebook, and complete the given task.

### Errata

1. None so far!

### What to Hand In

- Your PDF version of this notebook named `A1Q4.pdf`, containing solutions to the 10 parts of the question, and your name and student number at the top.

Be sure to include your name, NSID, student number, and course number at the top of all documents.

### Evaluation

- 10 marks: Your answers show that you have basic mastery of Numpy.

## Question 5 (9 points):

**Purpose:** Python and MatPlotLib

**Degree of Difficulty:** Basic

**References:** https://matplotlib.org/tutorials/index.html

Open the document `A1Q5.ipynb` using Jupyter Notebook, and complete the given task.

### Errata

1. None so far!

### What to Hand In

- Your PDF version of this notebook named `A1Q5.pdf`, containing solutions to the 3 tasks in the question, and your name and student number at the top.

### Evaluation

- 3 marks: Your plot for task 1 shows the 3 data sets on a single set of axes. The plot has labels, a title, and a legend.
- 3 marks: Your plot for task 2 shows the 3 data sets on individual axes arranged horizontally. The plots have labels, and there is a title. It looks good.
- 3 marks: Your plot for task 3 shows the 3 data sets on individual axes arranged vertically. The plots have labels, and there is a title. It looks good.

## Question 6 (5 points):

**Purpose:** Python and Pandas

**Degree of Difficulty:** Basic

**References:** https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html

Open the document `A1Q6.ipynb` using Jupyter Notebook, and complete the given task.

### Errata

1. None so far!

### What to Hand In

- Your PDF version of this notebook named `A1Q6.pdf`, containing solutions to the 3 tasks in the question, and your name and student number at the top.

### Evaluation

- 1 mark. For Task 1, you used `read_csv()` to load a datafile into the notebook.
- 1 mark. For Task 2, you used `describe()` to display some information about the DataFrames.
- 1 mark. For Task 3, you used `cov()` to display some covariance information about the dataframe.
- 1 mark. For Task 4, you used `hist()` to display histograms for the dataframe.
- 1 mark. For Task 5, you used `scatter_matrix()` to display a visualization of the dataframe

## Question 7 (12 points):

**Purpose:** Python and Pandas

**Degree of Difficulty:** One or two parts might be a little challenging.

**References:** https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html

Open the document `A1Q7.ipynb` using Jupyter Notebook, and complete the given task.

### Errata

1. None so far!

### What to Hand In

- Your PDF version of this notebook named `A1Q7.pdf`, containing solutions to the 3 tasks in the question, and your name and student number at the top.

### Evaluation

- 1 mark. For Task 1, you used `read_csv()` to load a datafile into the notebook.
- 1 mark. For Task 2, you used `density()` to display density estimation plots for the dataframe.
- 3 marks. For Task 3, you used Boolean array indexing to create separate DataFrames (one for each label value) and then used `density()` to display density estimation for each dataframe.
- 4 marks. For Task 4, you plotted the density estimation for each of the columns in the original DataFrame, and you have 3 densities allowing you to compare the distribution for each label in one plot. You have 4 such plots.
- 1 mark. For Task 5, you used Seaborn's `pairplot()` to display visualization of the original dataframe.
- 2 marks. For Task 6, you commented on the two kinds of plots, and what they represent. You also made some observations about some patterns in the data.

## Question 8 (5 points):

**Purpose:** To exercise the derivation of formulae involving probability.

**Degree of Difficulty:** Easy. The actual derivation is easier than the explanation.

**References:** Lecture Notes 04, Math with LaTeX in Jupyter Notebook

There is a notion in Bayesian statistics that yesterday's posterior probabilities are today's prior probabilities. It's an idea that suggests that learning should naturally combine data collected over time. It also reassures us that choosing a prior can be based on data seen previously. To understand this notion we need to do some math.

**Task** Suppose we collected data $\mathbf{X}_1$ yesterday, and used the data to calculate $P(y|\mathbf{X}_1)$. Yesterday, the prior that we assumed was $P(\mu) = \text{Beta}(\mu|a, b)$. Today, we collected data $\mathbf{X}_2$, and we wish to calculate $P(y|\mathbf{X}_1, \mathbf{X}_2)$.

Derive an expression for $P(\mu|\mathbf{X}_1, \mathbf{X}_2)$ in terms of yesterday's posterior $P(\mu|\mathbf{X}_1)$. This expression shows how yesterday's posterior can be used as if it were a prior.

**Elaboration** *In the following, we'll start with a review of the lecture material. Then we'll think about what happens with data $\mathbf{X}_1$ collected yesterday, and then more data $\mathbf{X}_2$ today. We could just throw away the model based on $\mathbf{X}_1$, and start over with all the data. But we can be cleverer than that here.*

In class we derived the following equation using Bayes' Rule:

$$P(\mu|\mathbf{X}) = \frac{P(\mathbf{X}|\mu)P(\mu)}{P(\mathbf{X})}$$

This was one of the steps in determining $P(y|\mathbf{X})$ for a binary event $Y$. In this expression, $P(\mu)$ is the prior distribution for $\mu$, and $P(\mu|\mathbf{X})$ is the posterior. We assumed that $P(\mu) = \text{Beta}(\mu|a, b)$, and we learned that $E[\mu] = \frac{a}{a+b}$. We also saw (skipping some of the mathematical details) that:

$$E[\mu|\mathbf{X}] = \frac{m + a}{N + a + b}$$

where $m$ and $N$ come from the data $\mathbf{X}$. The posterior $P(\mu|\mathbf{X})$ turns out also to be a Beta distribution over $\mu$, but it's $\text{Beta}(a_1, b_1)$, where $a_1 = m + a$ and $b_1 = N - m + b$ are new hyper-parameters. In effect, $\text{Beta}(\text{a}_1, \text{b}_1)$ summarizes everything we learned about $\mu$ from $\mathbf{X}$.

Suppose we collected data $\mathbf{X}_1$ yesterday, and used the data to calculate $P(y|\mathbf{X}_1)$. Yesterday, the prior that we assumed was $P(\mu) = \text{Beta}(\mu|a, b)$. Today, we collected data $\mathbf{X}_2$, and we wish to calculate $P(y|\mathbf{X}_1, \mathbf{X}_2)$.

In class, we were able to show, by significant hand-waving, that

$$P(y|\mathbf{X}_1, \mathbf{X}_2) = \frac{m_1 + m_2 + a}{N_1 + N_2 + a + b}$$

This is exactly what we'd get if we combined $\mathbf{X}_1, \mathbf{X}_2$ together. But it also shows that we can update our uncertainty about $y$, by counting $m_2, N_2$ in $\mathbf{X}_2$, and re-using the counts $m_1, N_1$ from yesterday. Your work in this question replaces some of the handwaving with a more formal derivation.

**Hint:** There's no calculus involved, only the basic rules of probability; start with Bayes Rule. One possibly confusing idea is that $P(\mu|\mathbf{X}_1)$ doesn't look like a prior, because it is conditional. That's true. But in this context, we can look at the prior more in terms of its use, than by its notational syntax.

## Errata

1. None so far!

## What to Hand In

- A Jupyter Notebook named `A1Q8.ipynb` with your derivation encoded in markdown, using LaTeX for the math.
- A PDF document `A1Q8.pdf` exported from Jupyter Notebook, containing the derivation above. The marker will primarily look at this, so make it presentable, like a report.

Be sure to include your name, NSID, student number, and course number at the top of all documents.

## Evaluation

- 5 marks: Your derivation has $P(\mu|\mathbf{X}_1, \mathbf{X}_2)$ on the left hand side, $P(\mu|X)$ on the right hand side, and your derivation applied valid rules of probability.