# INFO 3501/5501: Lab Assignment #1

Due on Wednesday, October 5, 2016 at 9:00am

**Professor Brian Keegan**
Department of Information Science
College of Media, Communication, and Information
University of Colorado, Boulder

# Background & Objectives

Wikipedia makes the complete revision histories for all its content publicly available through a variety of data interfaces. Lab Assignment #1 will explore different approaches to quantitatively analyzing these data.

1. **Interactively analyzing data using Jupyter Notebook**. Jupyter Notebook is a popular and free tool among professional data scientists for exploratory data analysis. Students will learn the fundamentals about how to launch and interact with pre-written notebooks.

2. **Accessing revision history data**. The MediaWiki software on which Wikipedia runs stores all changes made to articles in a public database. Students will learn how to access a MySQL database hosted by the Wikimedia Foundation and retrieve revision history data from it.

3. **Differentiating between article and user-level behavior.** It is possible to track behavior at the level of a Wikipedia page as well as behavior measured at the level of a user. Students will learn about and propose metrics that track behavior at both levels of analysis.

4. **Making inferences from visualizations.** Using the quantitative revision history data, we can generate plots visualizing different metrics and findings. Students will learn about how to make inferences about distributions of data, changes over time, and bivariate relationships.

5. **Comparing the behavior of different articles.** Data can be collected about two or more articles and the analyses run on the data of each. Students will learn about how to generate results using data from multiple articles and compare findings across them.

# Instructions

We will be be using Jupyter Notebooks running on the Wikimedia Foundation's PAWS infrastructure to retrieve public data from replica MySQL databases containing the complete history of revisions to articles. To complete this and future lab assignments, you will need to create an account, log in to the PAWS environment, upload a Jupyter Notebook to your account, and be able to launch and interact with the elements of the notebook.

## Step 1: Create a Wikimedia account

1.a Go to the PAWS system at `https://paws.wmflabs.org/paws/hub/login`.

1.b Click the orange "Sign in with MediaWiki" button.

1.c If you already have a Wikimedia account, enter your credentials and click the blue "Log in" button. If you don't have a Wikimedia account, click the white "Join Meta" button. Complete the fields, entering your desired username, password, and email address, and click the blue "Create your account" at the bottom. Make sure not to lose this login information. Complete any additional authentication steps necessary such as clicking the link send to your email address.

1.d A dialog box asking you to grant permission to PAWS to complete your request will pop up. Click the blue "Allow" button to proceed.

1.e Click on the "My Server" button to start your server. This directory should be empty initially, but we will upload a notebook file to it in the next step.

## Step 2: Uploading a Jupyter Notebook

2.a In a different browser window or tab, download the Jupyter Notebook posted to D2L to your desktop. Note that you can also view other notebooks I will be developing for the course in `http://paws-public.wmflabs.org/paws-public/User:Cuinfostudents/` but downloading the files from here apparently will not work to upload to your PAWS account.

2.b Return to the browser tab having your empty "My Server". Click the "Upload" button in the upper right.

2.c In the system pop-up box, navigate to where you saved the downloaded "Lab 1 – Revision Histories.ipynb" file on your desktop. Select the file and click "Open".

2.d Click the blue upload button next to the file name to upload it to the PAWS server.

2.e Confirm that you can access the notebook by clicking on its title in the Home directory, which should launch a new tab/window labeled "Lab 1 – Revision Histories" and show the contents of the notebook after a few seconds.

## Step 3: Interacting with a Jupyter Notebook

3.a Jupyter Notebooks are made of cells containing different kinds of content: code, markdown (formatted text), and section headings. Any cell can be edited by double-clicking on it so that the status indicator changes from blue to green.

3.b All cells can also be executed by pressing Shift+Enter. For the cells containing code, this will execute the code in the cell. For the cells containing markdown or headings, this will format the text into HTML. The status will also advance to the next cell.

3.c Execute the first few lines in the "Confirm basic Python commands work" section.

If you have any errors at this stage, contact me immediately.

## Additional Resources

Here are some links to other resources that provide more details if you are curious and want to learn more.

**Jupyter Notebook** Official documentation about how to configure and use the Jupyter Notebook: `http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html`

**PAWS** Official documentation about how to setup and access the PAWS environment: `https://wikitech.wikimedia.org/wiki/PAWS`

**Tool Labs database** Official documentation about the databases available for analysis: `https://wikitech.wikimedia.org/wiki/Help:Tool_Labs/Database`

**MediaWiki manual: "revision"** Official documentation describing the fields in the "revision" table: `https://www.mediawiki.org/wiki/Manual:Revision_table`

**MediaWiki manual: "user"** Official documentation describing the fields in the "user" table: `https://www.mediawiki.org/wiki/Manual:User_table`

**Example MySQL queries** Examples of queries that can be run against the Tool Labs replica databases: `https://wikitech.wikimedia.org/wiki/Help:MySQL_queries#Example_queries`

## Problem 1

Identify a pair of articles that you believe have interesting editing dynamics. Why did you choose these articles? What kinds of editing behavior would you expect to see on these articles? How would this behavior manifest in each article's revision history? Write down your expectations for (1) the number of editors, (2) number of revisions, (3) year the article was created, and (4) year that editing activity on the article peaked. Write up at least two paragraphs answering the questions above and referencing either social behaviors we discussed over the past five weeks or technical features from the documentation in the additional resources.

## Problem 2

Retrieve the revision histories for one of your articles. Report the number of revisions, unique users, and minimum and maximum revision timestamps for the article. Discuss these descriptive metrics: is this more or less than you expected in the previous step? What would having more/less or earlier/later revisions than you expected indicate about editor behavior? Speculate about what these basic metrics can or cannot tell us about the conflict, quality, or popularity of the article.

## Problem 3

How do you save the Matplotlib figures that you generate within a Jupyter Notebook? When you save a figure in the PAWS environment, where is the image file stored? Please cite the source (Jupyter Notebook or Matplotlib documentation, StackOverflow answer, example from another tutorial or code repository, advice from a peer, etc.). There are multiple acceptable answers and approaches, but the goal is for you to become acquainted with how to formulate questions about a programming problem you are having and where you can find resources to solve them. Include a plot with a narrative about what you find exciting, unexpected, or troubling about it.

## Problem 4

What was the most active day of editing? How has edting activity on the article changed over time? Around what date did editing activity on your article begin to slow down? Around what date did the size of the article begin to stabilize? Who was responsible for the revision that made the article its longest? Look up the content of a few of the largest revision by entering the corresponding "rev_id" numeric value into the URL: `https://en.wikipedia.org/w/index.php?oldid=rev_id_value` Look at the preceding or subsequent (click the "Previous revision" or "Newer revision" links beneath the red banner box) edits to examine whether this large contribution was reverted completely, reverted partially, or subsequent revisions extended/ignored this content.

## Problem 5

Who were the most active editors on the article? Are they bots or do they appear to be people? Does the distribution of editing activity show a "long tail" property? What does the distribution of editors' tenure on the article look like? Where do the most active editors fall on this distribution? Is there a convincing relationship between editors' tenure on the article and the number of revisions they make?

## Problem 6

Are there any "VIPs" who have contributed to the article based on their user information: especially old editors, editors who've made many revisions, etc.? What could the "rev_fraction" tell us about user behavior on the article?

Problem 6 continued on next page...

How is your article's distribution similar or dissimilar from the distribution for the Mitt Romney article? What could the "first_rev_account_age" tell us about user behavior on the article? How is your article's distribution similar or dissimilar from the distribution for the Mitt Romney article?

# Problem 7

**Required for graduate students, extra credit for undergraduates.** Duplicate one of the analyses for the other article you selected. Include a single plot that combines data from both articles, making sure to scale axes appropriately to show the detail in both. Compare the results of the analysis from both articles' data and discuss the implications of any differences or similarities in the context of your motivations for selecting these articles in Problem 1.

# Miscellaneous

## Acknowledgements

I want to thank the Wikimedia Foundation for the PAWS system and related Wikitech infrastructure that this workbook runs within. Yuvi Panda, Aaron Halfaker, Jonathan Morgan, and Dario Taraborelli have all provided crucial support and feedback.

## Licensing