

# Assignment 1: Keeping Up With Social Information (Part 1)

© Cristian Danescu-Niculescu-Mizil 2020

CS/INFO 4300 Language and Information

Due by midnight on Wednesday January 29th

You must completely this assignment **individually**.

In this assignment we are practicing with post-processing on a conversational dataset taken from the reality TV show “Keeping Up With The Kardashians” and gathering some basic statistics about it.

In the next assignment (Assignment 2) we will extend these tools to analyze conversational behavior.

## Instructions

Follow the instructions below to get the necessary packages installed and set up your Python environment, then open the attached Jupyter notebook.

Run the notebook and complete the tasks contained in it, then upload the completed notebook and an HTML copy of it to CMS.

## Learning Objectives

This project aims to help you get comfortable working with the following tools / technologies / concepts:

- word tokenization
- histogram plotting using `matplotlib`
- character analysis via conversational language
- familiarize yourself with basic numpy usage

## Academic Integrity and Collaboration

Note that these projects should be completed individually. As a result, all University-standard academic integrity guidelines must be followed.

## Setting up your environment

### System Configuration

Perform the following steps in order:

**1. Check your Version of Python (should be 3.7.6)** You can check via:

```
> python3 --version
Python 3.7.6
```

If your version differs, then download 3.7.6 [here](#).

**2. Check that Pip is Installed and Up-to-date.** You should already have pip installed if you have Python downloaded from python.org. Make sure that yours is up-to-date.

Upgrade pip :

```
> python3 -m pip install -U pip
```

If not, install it following instructions. (also found [here](#))

```
> curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
> python3 get-pip.py
```

**3. Install Virtualenv via Pip** [Virtualenv](#) helps establish an isolated Python environment. The environment allows you to separate project-specific dependencies and their versions from the Python modules installed locally on your computer.

Once you have virtualenv, run the following:

```
> virtualenv -p python3 venv
```

This creates a virtual environment called `venv`.

In order to enter than virtual environment run the following:

Linux or MacOS:

```
> source venv/bin/activate
```

Windows:

```
> venv\Scripts\activate
```

The following command line prompt will indicate that you're in the virtual environment:

```
(venv) >
```

To deactivate the virtual environment, run the following:

```
(venv) > deactivate  
>
```

Whenever you work with this project, you should **always** be in your virtual environment. Without this isolation, we might run into module versioning issues and other problems when trying to run your project, which creates administrative overhead.

**4. Install Dependencies** At the root of directory of the project skeleton code, run the following:

```
(venv) > pip install -r requirements.txt
```

This installs within your virtual environment all the necessary modules that are required at the beginning of the project.

**5. Setup Jupyter Notebook** To use your virtualenv as the kernel for your Jupyter Notebook you run the following:

```
(venv) > python3 -m ipykernel install --user --name=venv
```

**6. Open Jupyter Notebook and start working** Open the Jupyter Notebook in your virtual environment and complete the assignment.

Linux or Mac:

```
> source venv/bin/activate  
(venv) > jupyter notebook
```

Windows:

```
> venv\Scripts\activate  
(venv) > jupyter notebook
```

In your Jupyter Notebook, make sure to set your kernel to your virtualenv. To change kernels, go to **Kernel > Change Kernel** and click **venv** as the option.