title: DSI Installfest duration: "1:00" creator: name: Marc Harper

city: LA

Data Science Immersive "Installfest"

DSI Computer Setup

Welcome to GA's Data Science Immersive! Before you start class, you'll need to download and install a few tools. Follow this guide to get your computer all set up, and let us know if you have any questions. * Part 1: Operating System * Part 2: Anaconda and Python * Part 3: Confirm Your Python Installation * Part 4: Git * Part 5: PostgreSQL * Part 6: Classroom Tools * Part 7: Text Editors

Part 1. Operating System

While you can be a data scientist on any operating system, most practicing data scientists choose a Unix-type operating system, typically either Apple's OS X or a popular linux distribution such as Ubuntu or Linux Mint.

Please note that this course will be taught using Macs, and instructors may not necessarily be able to help you troubleshoot PC or Linux issues. For more information about course technology policies, please see this guide.

Part 2. Anaconda and Python

In our class, we'll be working closely with tools that utilize the Python programming language. Anaconda is a popular cross-platform tool that helps install and manage python-related data science libraries.

1. Download Anaconda **Python 2.7 version**
2. Follow the installation instructions package for your operating system.

   - Mac users
   - Windows users
   - Linux users

3. Agree to the terms and let Anaconda go through its **Python 2.7 version** installation.

4. Anaconda will install several packages by default, including:

   - **python**: a programming language very popular with data scientists
   - **jupyter**: an interface for creating interactive python notebooks, great for sharing analyses
   - **matplotlib**: a plotting library for python
   - **nltk**: a toolkit for natural language processing
   - **numpy**: a linear algebra library
   - **pip** & **setuptools**: software to manage and install python packages
   - **scikit-learn**: a toolkit for machine learning algorithms
   - **scipy** and **statsmodels**: statistical packages for python
   - **sqlite**: a popular, easy to use database

5. Once Anaconda is installed, verify your installation. You can add additional python packages from the command line.

> Note: You don't need to type the `$`

```
$ conda install jupyter python matplotlib nltk numpy pip setuptools scikit-learn scipy sqlite statsmodels
```

Adding additional python packages from the command line is simple and can be done at any time; for example:

```
$ conda install gensim seaborn spacy
```

Just for Mac Users

You can also use the `pip` command to install libraries.

> Some markets may also recommend the use of `Homebrew`.

Just For Linux Users

On Ubuntu, if the `conda install` command fails for some reason, restart your terminal or source your `.bashrc` like so:

```
$ source ~/.bashrc
```

Part 3. Confirm Your Python Installation

1. When you've gotten this far, open up a terminal and enter the Python interpreter:

   ```
   $ python
   ```

   Depending on your operating system, your terminal should return something like this:

   ```
   user@vbox:~/Downloads$ python
   Python 2.7.11 |Anaconda 2.5.0 (32-bit)| (default, Jan  1 2017, 18:08:45)
   [GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
   Type "help", "copyright", "credits" or "license" for more information.
   Anaconda is brought to you by Continuum Analytics.
   Please check out: http://continuum.io/thanks and https://anaconda.org
   >>>
   ```

2. Next, make sure that the necessary packages are installed. For example, to check that `matplotlib` is installed, type in your terminal:

   ```
   >>>> import matplotlib
   >>>> print matplotlib.__version__
   1.5.1
   ```

   You may see another version (which is OK). If you get an error like this:

   ```
   $ import matplotlib
   ImportError: No module named matplotlib
   ```

   then you'll need to try to install the Python packages again.

3. You can check the installation and versions of *all* the python libraries. You can do this a couple of different ways.

   A) By running this jupyter-notebook:

   - Fork this repository to your profile, then download the files to your local machine. The notebook is called installfest-test.ipynb
   - Open a terminal, navigate to the notebook in your download folder, and run `jupyter`

   ```
   $ cd Downloads
   $ jupyter-notebook
   ```

   - Open the notebook by selecting the notebook file
   - From the `Kernel` menu select `Restart & run all`

   If you see any errors then you'll need to reinstall the library that posts the error. Otherwise you should see a bunch of version numbers!

   B) Alternatively, try typing `pip freeze` in your terminal:

   - Open a terminal window

   ```
   $ pip freeze
   ```

   You should see a list of all the python packages currently installed with their version numbers in the terminal window.

Part 4. Git

1. We'll also be using git -- a popular version control system used to share code with others -- extensively along with Github. We strongly recommend hosting your portfolio and polished materials on Github, but for the purposes of this class we will actually be using *GA's Github* - a special version of Github running on GA's servers. We'll use this private, enterprise version of Github for all of our in-class materials.

- To join, go to `git.generalassemb.ly` (if you haven't already), sign up for an account, and submit your username to your instructional team.

1. Download git here by clicking on the version for your operating system.
2. Windows users

3. Check if your git installation is successful by opening a new terminal window and try to run `git` from the command line:

   ```
   $ git --version
   ```

The output should be something like this:

```
$ git --version
git version 2.5.0
```

4. Next, you'll want to tell git your name and email. Make sure to use the same email address that you use to sign up for GA's Github:

```
$ git config --global user.name "Your Name"
$ git config --global user.email your.name@example.com
```

Github Enterprise

The following suggested instructions are intended to help students setup Github Enterprise as a default remote from their command line.

1. First, we recommend that students create an access token instead of using a password. For a sample walkthrough, please see the tutorial here.

2. Second, students should cache their Github password so that they don't have to continue reentering it.

3. Third, use the following commands to connect command line to a student's individual Github Enterprise repo. `bash $ git config --global credential.helper osxkeychain $ git remote add origin https://git.generalassemb.ly/example_user/example_re`

Part 5. PostgreSQL

PostgreSQL is a databas that we'll be using later in class. Install Postgres with the following steps:

1. Follow the instructions for your operating system below

Mac Users

- Download Postgres.app from www.postgresapp.com
- Move the Postgres.app to your 'Applications' folder.
- Open the Postgres.app (using "right-click + open" since it is an application that isn't from the Mac App Store)
- Look for the elephant in the the menu bar.

Windows Users

- Download Postgres

Linux Users

- Download Postgres

- On Ubuntu, you can also install Postgres from the package manager:

```
$ sudo apt-get install postgresql postgresql-contrib postgresql-client
```

1. You need to add yourself as a user in postgres so you can access the `psql` console seamlessly. Following the commands below, replace `dsi-student` with *your own user-name* and type *your own password* when prompted.

If you are running Ubuntu, use "**ilovedatascience**" as your password.

```
$ sudo -i su - postgres
$ createuser dsi-student --superuser --password
$ createdb dsi-student
$ exit
```

Test that this works by typing `psql`. You should be presented with the postgres shell. To exit type `\q`.

Part 6. Classroom Tools

> Note: Some regions and instructors may require additional classroom tools.

1. We'll be using Slack, a popular messaging platform, for our class communications. If you haven't installed this already, we'll remind you to do so now.

   - Click on the installation instructions for your platform to install the Slack desktop app. You can also sign into Slack using a web interface or via their mobile app!

2. Chrome is Google's popular web browser, and it comes with a complete set of developer tools built-in. We'll use Chrome to examine code, debug scripts, and view back-end processes. If you don't already have Chrome, make sure to download and install it now.

Part 7. Text Editors

A data scientist frequently writes scripts to process data, perform analysis, and create visualizations, webpages, and other products, so you'll need a good text editor. If you don't already have a preference, try Atom or Sublime. Both editors are available for most platforms.

> Instructors should modify these options based on their preferences.

1. Download the editor of your choice from their website.
2. Install the package by double clicking the file icon or from the command line
3. Run your editor from the applications menu, or from the command line, like so:

```
$ subl
```

or

```
$ atom
```

This example would open up Sublime or Atom, respectively. Whichever editor you choose, be sure to practice using it!

Configure Git with your Text Editor

Finally, you'll want to tell `git` which editor it should use for your commits.

- If you choose to use Sublime, you would type:

  ```
  $ git config --global core.editor "subl --wait --new-window"
  ```

- If you choose to use Atom, you would type:

  ```
  $ git config --global core.editor "atom --wait"
  ```

That's it! Now you're ready to begin GA's Data Science Immersive. See you on the first day of class!