Dev Setups -- Python and SciPy stack

This document is iterated on every session, and a lot of work goes into making sure it is clear and as simple as possible. Please do not share outside of Insight. Thanks!

Part 1: install Python and packages

choose your own adventure...

for Ubuntu

at the command line
sudo apt-get install python-distribute
sudo apt-get install python-setuptools
sudo apt-get install python-dev
sudo apt-get install ipython-notebook
sudo apt-get install python-matplotlib
sudo easy_install pip
sudo pip install ipython[all]
sudo pip install flask
sudo pip install pandas
To Open ipython notebook in the directory where this notebook is stored on your local machine:
ipython notebook master_dev_setups.ipynb

now test your installs! Proceed to part 2 inside the opened notebook (should be visible in a web browser)

Mac OS-X 10.11

The instructions below work for **Mac OS-X 10.11** . If you are having issues, there are directions for 10.9+ located here (https://sites.google.com/a/insightdatascience.com/datascience/devsetups/predevsetup)

Install homebrew package manager
at the command line
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" (https://raw.githubusercontent.com/Homebrew/install/master/install)")
install python 2.7.10
brew install python
numpy stack
pip install numpy
pip install scipy
pip install scikit-learn
pip install matplotlib
pip install ipython[all]
pip install jupyter[all]
pip install pandas
pip install flask
To Open ipython notebook in the directory where this notebook is stored on your local machine:
ipython notebook master_dev_setups.ipynb

now test your installs! Proceed to part 2 inside the opened notebook (should be visible in a web browser)

Part 2: Test your installs

Go ahead and do a shift+enter in the cell below,

- if you get an error message, your packages didn't install correctly.
- If you get no error (a number appears next to the box on the left, the computaiton is finished), great job! Continue to part 3

In [1]:

```
import numpy #this one is for scientific computing
import scipy # this one is for open-source software for mathematics, scienc
e, and engineering (higher level than numpy)
import sklearn # this one is for machine learning
import matplotlib # this one is for plotting
import pandas # this one is for making easy-to-use data structures
```

Part 3: creating Insight graphic

In [2]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

dpi = 80
fig = plt.figure(figsize=(4, 1.6),dpi=dpi)

def add_background():
    ax = fig.add_axes([0., 0., 1., 1.], axisbg='black')
    rect = mpatches.Rectangle([0, .86], 0.05, 0.14, ec="none", facecolor='b
lack')
    ax.add_patch(rect)
    rect = mpatches.Rectangle([0.05, .86], 0.05, 0.14, ec="none", facecolo
r='gray')
    ax.add_patch(rect)
    rect = mpatches.Rectangle([0, 0.72], 0.05, 0.14, ec="none", facecolo
r='gray')
    ax.add_patch(rect)
    rect = mpatches.Rectangle([.95, 0], 0.05, 0.14, ec="none", facecolor='b
lack')
    ax.add_patch(rect)
    rect = mpatches.Rectangle([.90, 0], 0.05, 0.14, ec="none", facecolor='g
ray')
    ax.add_patch(rect)
    rect = mpatches.Rectangle([.95, .14], 0.05, 0.14, ec="none", facecolo
r='gray')
    ax.add_patch(rect)
    ax.set_axis_off()
    return ax

def add_insight_text(ax):
    ax.text(0.52, 0.5, 'INSIGHT', color='black', fontsize=45,
               ha='center', va='center', alpha=1.0, transform=ax.transAxes)


if __name__ == '__main__':
    main_axes = add_background()
    add_insight_text(main_axes)
    plt.show()
```

Part 4: Make coding pretty

ipython notebooks are great for documentation, as you will see throughout the session. Many previous fellows and companies use them to keep their work clean and well-commented. Other popular editors include:

- Sublime Text (http://www.google.com/url?q=http%3A%2F%2Fwww.sublimetext.com%2F&sa=D&sntz=1&usg=AFrqEzeFkIlRQBBF0Ik9XAy6A)

Also, making sure you use version control during your projects is ***essential*** to making sure you don't lose your work. We will have an introduciton to git soon in the session

super optional If you already have a working environment on your computer that includes a particular install of python, you might consider using a virtual environment. On the mac, this can be installed using
pip install virtualenv
and the wrapper can be installed with
pip install virtualenvwrapper
and you can find information on how to do this here (http://www.google.com/url?q=http%3A%2F%2Fvirtualenvwrapper.readthedocs.org%2Fen%2Flatest%2Finstall.html%23quick-start&sa=D&sntz=1&usg=AFrqEzdYumPm3A6KlKHg9iPOPD3ojfu0qg)