Git & GitHub

COGS 108 Spring 2025
Jason Chen
Week 2

xic007@ucsd.edu

OH: Thu 3-5 pm

Discussion slides and materials adapted from Sam Lau (TA: WI20)

Reminders

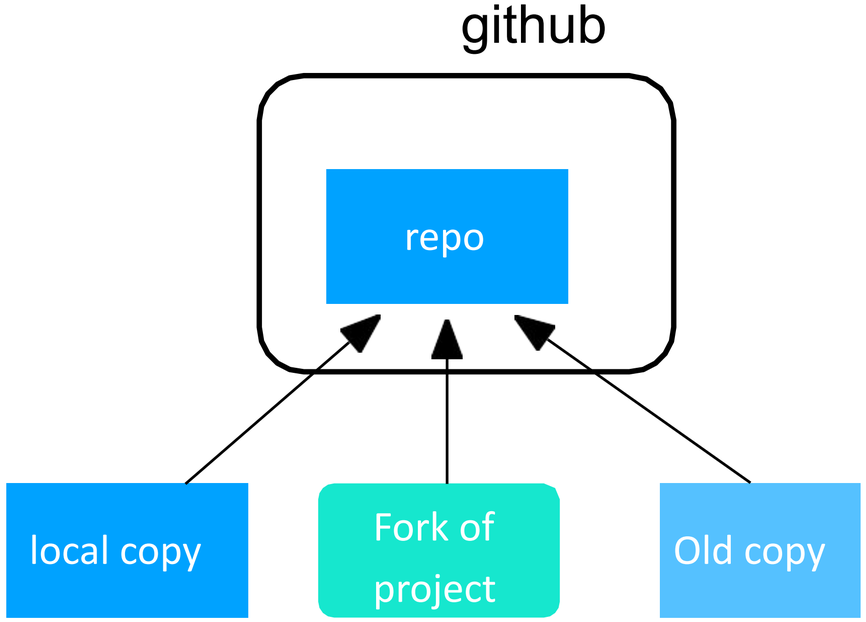
- Practice assignment & Pre-course survey Due Apr 9 11:59 PM
- D1 Due Apr 11 11:59 PM

Git

- Version control system!
- Go to https://git-scm.com/downloads
  - Choose your Operating System (Windows/OS X/Linux)
  - Follow the steps specific to your OS
  - Verify installation: In terminal type “git –-version”

What is git + GitHub?

- Somewhere online to store a copy of a project (Github)
- Plus a tool to interact with this copy (Git)
  - Command line and desktop versions
- A way of keeping track of changes you make to a project
- Does everyone have a GitHub account?



Why use git + GitHub?

- **Collaboration:** Git allows you to work on code projects with other people. It's the preferred tool for many projects, like...
  - Python: https://github.com/python/cpython
  - Jupyter: https://github.com/jupyter/
  - COGS 108: https://github.com/COGS108/
- Backup
- Version control (*undo* on a large scale)
- Code reuse

Jupyter is NOT Good in Version Control

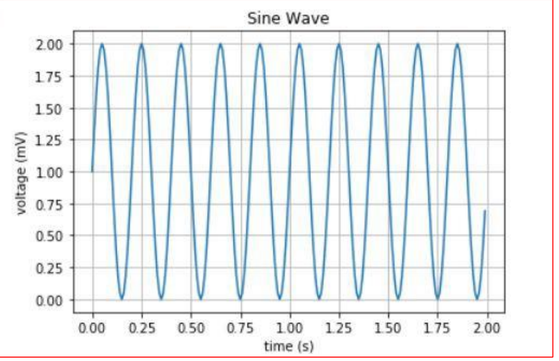
In [10]:
```
import numpy as np
import matplotlib.pyplot as plt

# Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin((5 * 2)* np.pi * t)

# Note that using plt.subplots below is equivalent to using
# fig = plt.figure() and then ax = fig.add_subplot(111)
fig, ax = plt.subplots()
ax.plot(t, s)

ax.set(xlabel='time (s)', ylabel='voltage (mV)', title='Sine Wave')
ax.grid()
```

Out[10]:



This is what the plot looks like in GitHub:

"outputs": [
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
AAlwSFlzAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uID
IuMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSeIllOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMbQQC4wYMWLEIRIYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELjBgfGojIXSLyoiXSLyoojMiMgZEXlBRH4IQCn
1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8rBrAE2Ab8ClB2Tfha4t+3zPcC7H
b57Tyk15XgsMWJEQiwwYnxY8BEApdS3lVJ1pdSCUupxpdRbCLyRRRF5PjjY1/i/JCLvichZEfktEZ
G23/8PEdnl//aYiFzWhe6zwJ0iEvDa3cC/A7Ys+u5Z/7qrReRPReSkf+0/FZGL/d8+LyLb2i8uIl8RkYf9
/2dF5F+LyEEROS4i3xCR4YjPLUaMJmKBEePDgj1AXUT+k4h8VkYcwCiwX4NICI/O
/APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAZcCiwAv+..."

Source: https://nextjournal.com/schmudde/how-to-version-control-jupyter

Jupyter is NOT Good in Version Control

When we make a little change:

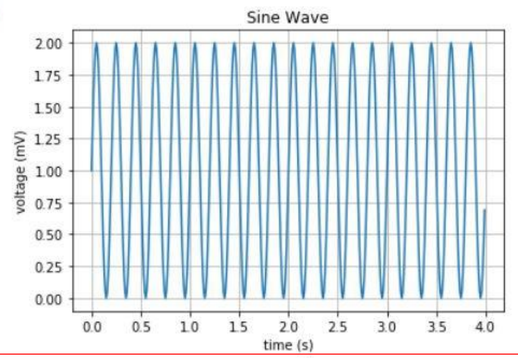
```
In [11]: import numpy as np
         import matplotlib.pyplot as plt

         # Data for plotting
         t = np.arange(0.0, 4.0, 0.01)
         s = 1 + np.sin((5 * 2)* np.pi * t)

         # Note that using plt.subplots below is equivalent to using
         # fig = plt.figure() and then ax = fig.add_subplot(111)
         fig, ax = plt.subplots()
         ax.plot(t, s)

         ax.set(xlabel='time (s)', ylabel='voltage (mV)', title='Sine Wave')
         ax.grid()
```

Out[11]:



The difference in plot will be marked as:

"outputs": [
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
AAlwSFlzAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uID
IuMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9Vc4
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSeIllOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMbQC4xELjBgfGojIXSLyoojMiMgZEXlBRH4IQCn
1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8ClB2Tfha4t+3zPcC7H
b57Tyk15XgsMWJEQiwwYnxY8BEApdS3lVJ1pdSCUupxpdRbACLyRRF5PjjY1/i/JCLvichsh
G23/8PEdnl//aYiFzWhe6zwJ0iEvDa3cC/A7Ys+u5Z/7qrReRPReSkf+0/FZGL/d8+LyEEROS4i3xCR4YjPLUaMJmKBEePDgj1AXUT+k4h8VkWa5zzY8APATcpNIC/O
/APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAZcCiwAv+..."

Source:

Jupyter is NOT Good in Version Control

When we make a little change:

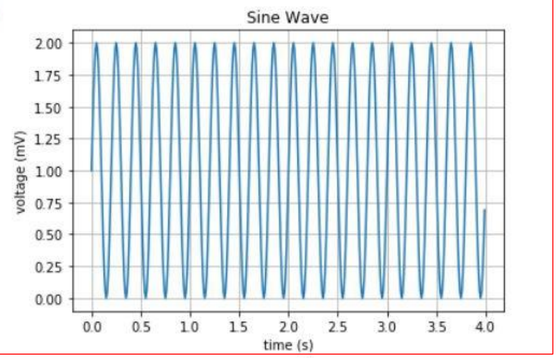
In [11]:
```
import numpy as np
import matplotlib.pyplot as plt

# Data for plotting
t = np.arange(0.0, 4.0, 0.01)
s = 1 + np.sin((5 * 2)* np.pi * t)

# Note that using plt.subplots below is equivalent to using
# fig = plt.figure() and then ax = fig.add_subplot(111)
fig, ax = plt.subplots()
ax.plot(t, s)

ax.set(xlabel='time (s)', ylabel='voltage (mV)', title='Sine Wave')
ax.grid()
```

Out[11]:



The difference in plot will be marked as:

"outputs": [
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
AAlwSFlzAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uID
IuMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9Vc4A
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSeIllOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMbQQC4wYMWLEiKGFWGDEiBEjRgwtxAIjRyYMW
MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELjBgfGojIXSLyoojMiMgZEXlBRH4IQCn
1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8ClB2Tfha4t+3zPcC7H
b57Tyk15XgsMWJEQiwwYnxY8BEApdS3lVJ1pdSCUupxpdRbACLyRRF5PjjY1/i/JCLvichZEfktEZ
G23/8PEdnl//aYiFzWhe6zwJ0iEvDa3cC/A7Ys+u5Z/7qrReRPReSkf+0/FZGL/d8+LyLb2i8uIl8RkYf9
/2dF5F+LyEEROS4i3xCR4YjPLUaMJmKBEePDgj1AXUT+k4h8VkRWa5zzY8APATcNICI/O
/APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAZcCiwAv+..."

Source: https://nextjournal.com/schmudde/how-to-version-control-jupyter

Jupyter is NOT Good in Version Control

When we make a little change:

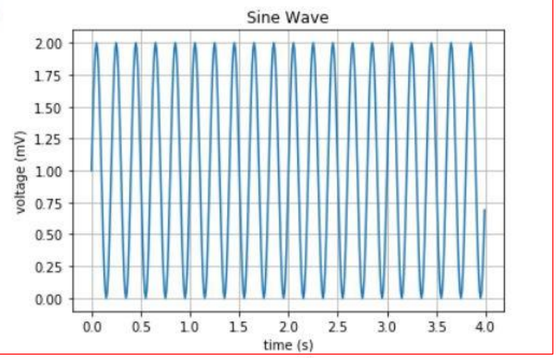
```
In [11]: import numpy as np
         import matplotlib.pyplot as plt

         # Data for plotting
         t = np.arange(0.0, 4.0, 0.01)
         s = 1 + np.sin((5 * 2)* np.pi * t)

         # Note that using plt.subplots below is equivalent to using
         # fig = plt.figure() and then ax = fig.add_subplot(111)
         fig, ax = plt.subplots()
         ax.plot(t, s)

         ax.set(xlabel='time (s)', ylabel='voltage (mV)', title='Sine Wave')
         ax.grid()
```

Out[11]:



The difference in plot will be marked as:

"outputs": [
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
AAlwSFlzAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uID
IuMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9vc4
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSeIllOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMbQQC4wYMWLEiKGFWGDEiBEjRgwtxAIjRowY
MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELjBgfGojIXSLyoojMiMgZEXlBRH4IQCn
1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8ClB2Tfha4t+3zPcC7H
b57Tyk15XgsMWJEQiwwYnxY8BEApdS3lVJ1pdSCUupxpdRbACLyRRF5PjjY1/i/JCLvichZEfktEZ
G23/8PEdnl//aYiFzWhe6zwJ0iEvDa3cC/A7Ys+u5Z/7qrReRPReSkf+0/FZGL/d8+LyLb2i8ull8RkYf9
/2dF5F+LyEEROS4i3xCR4YjPLUaMJmKBEePDgj1AXUT+k4h8VkRWa5zzY8APATcpCfwAPGNICI/O
/APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAZcCiwAv+..."

When you have a lot of plots, your changes in git will be thousands-pages long… And it is impossible to tell what the differences are…

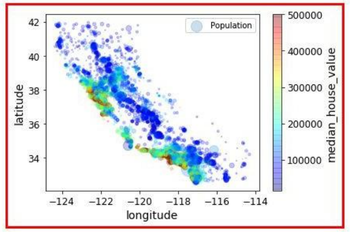
Source: https://nextjournal.com/schmudde/how-to-version-control-jupyter

How to Avoid This Issue?

- Clear your outputs before pushing back your notebook to GitHub. (You will only be able to see changes in your CODE)
- Use other GitHub Apps, like ReviewNB.
- This app will VISUALIZE the plots, instead of showing those nonsense words (Yay!)

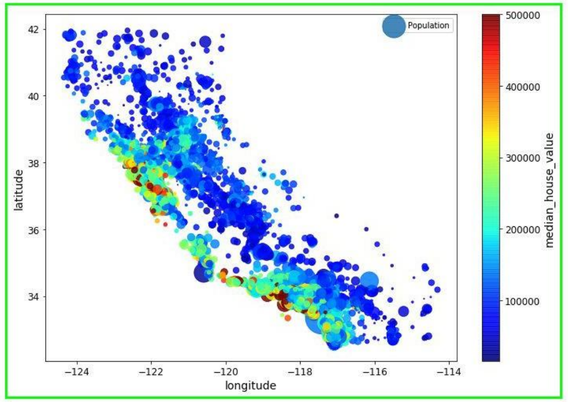
```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.2,
    s=housing["population"]/88, label="Population", figsize=(6,4),
    c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True,
    sharex=False)
plt.legend()
save_fig("housing_prices_scatterplot")
```

```
Saving figure housing_prices_scatterplot
```




```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.8,
    s=housing["population"]/21, label="Population", figsize=(10,7),
    c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True,
    sharex=False)
plt.legend()
save_fig("housing_prices_scatterplot")
```

```
Saving figure housing_prices_scatterplot
```



Source: https://nextjournal.com/schmudde/how-to-version-control-jupyter

Demo of GitHub Repo and Commands

- Cloning a repo on your local machine
- Working on your local repo (making changes to files)
- Stage, commit and push these changes to your Github repo
- Commands you should know:
  - Git clone -This will download the latest version of the repo to your local PC
  - Git status (not really needed – but really helpful)
  - Git add - This adds the changes in the staging area
  - Git commit -m “” - This will save a “Snapshot” of your most recent changes
  - Git push - This will upload your local changes to your GitHub Repo
  - Git pull - This will update your local repo to the latest version which is on GitHub

Section Materials

https://github.com/JasonC1217/COGS_108_B03-B04_Sp25/tree/master

or:

https://tinyurl.com/4d8wx3ne

Questions?