

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

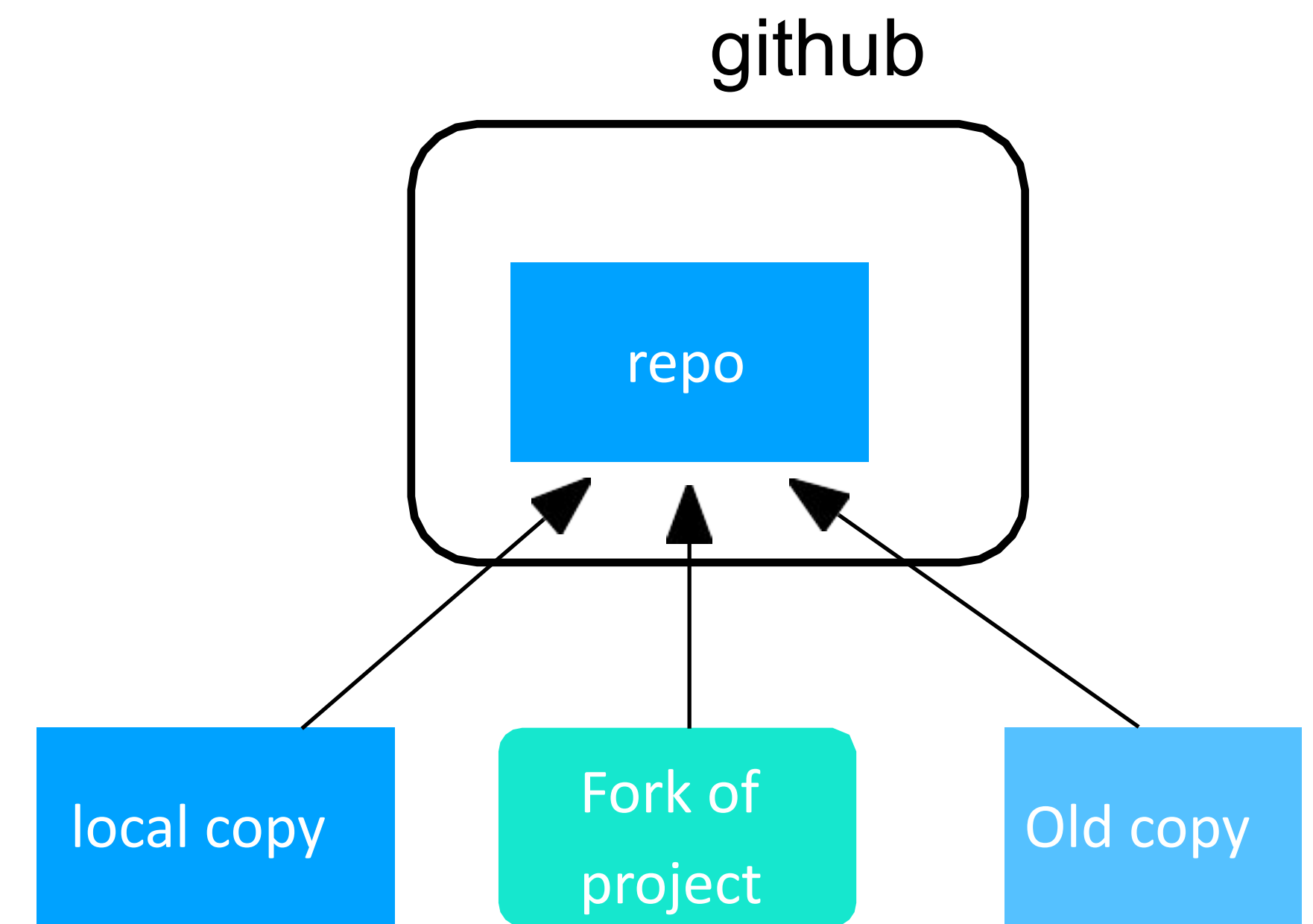
- Practice assignment & Pre-course survey Due Apr 8 11:59 PM
- D1 Due Apr 11 11:59 PM

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

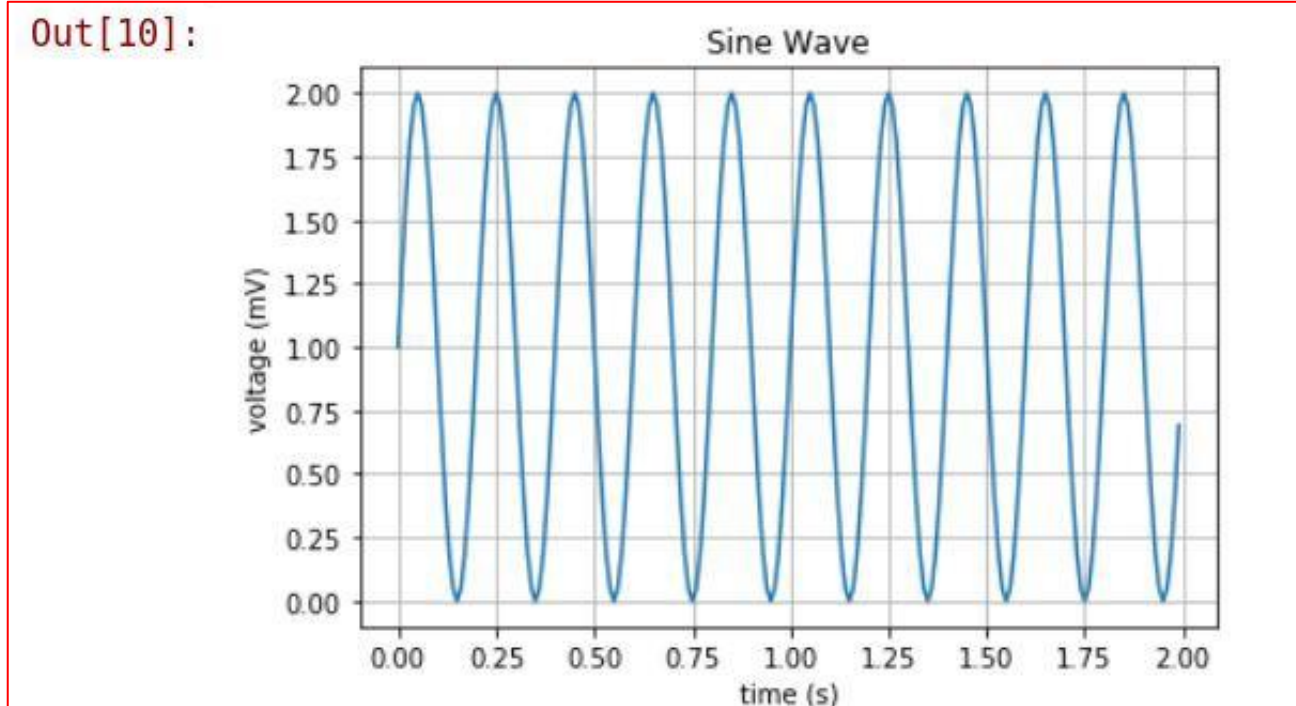
This is what the plot looks like in GitHub:

```
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()
```



```
"outputs": [
{
  "data": {
    "image/png":
      "iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
      AAlwSFZAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAbWF0cGxwdGxpYiB2ZXJzaW9uID
      luMi4yLCBodHRwOi8vbWF0cGxwdGxpYi5vcmlvLmUwAAIAAJREFUeJzsvXmcHNd13/s9vc4+2
      EgABHeQEKVSYGGRFLembFNspN7Wyy45i5UXh5ZjvcSy4xcr78WK5bwbkzvKSeIIQqaVxZKcOJL
      N+FHc0dxJEVxAAgQBAiCldBdp0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
      SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMBQQC4wYMWLEiKGFWDGEiBEjRgwtXAljRowY
      MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELjBgfGojIXSLyoojMiMgZEXIBRH4IQcN
      1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8CIB2Tfha4t+3zPcC7H
      b57Tyk15XgsMWJEQiwYnY8BEApdS3IVJ1pdSCUupxpdRbACLyRRF5PjjY1/i/JCLvichZEfktEZ
      G23/8PEdnI//aYiFzWhe6zwJ0iEvDa3cC/A7Ys+u5Z/7qrReRPRReSkf+0/FZGL/d8+LyLb2i8u1l8RkYf9
      /2dF5F+LyEEROS4i3xCR4YjPLUaMJmKBEEPDgj1AXUT+k4h8VkrWa5zzY8APATcCfwX4NICI/O
      /APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAzcCiwAv+..."
```



Jupyter is NOT Good in Version Control

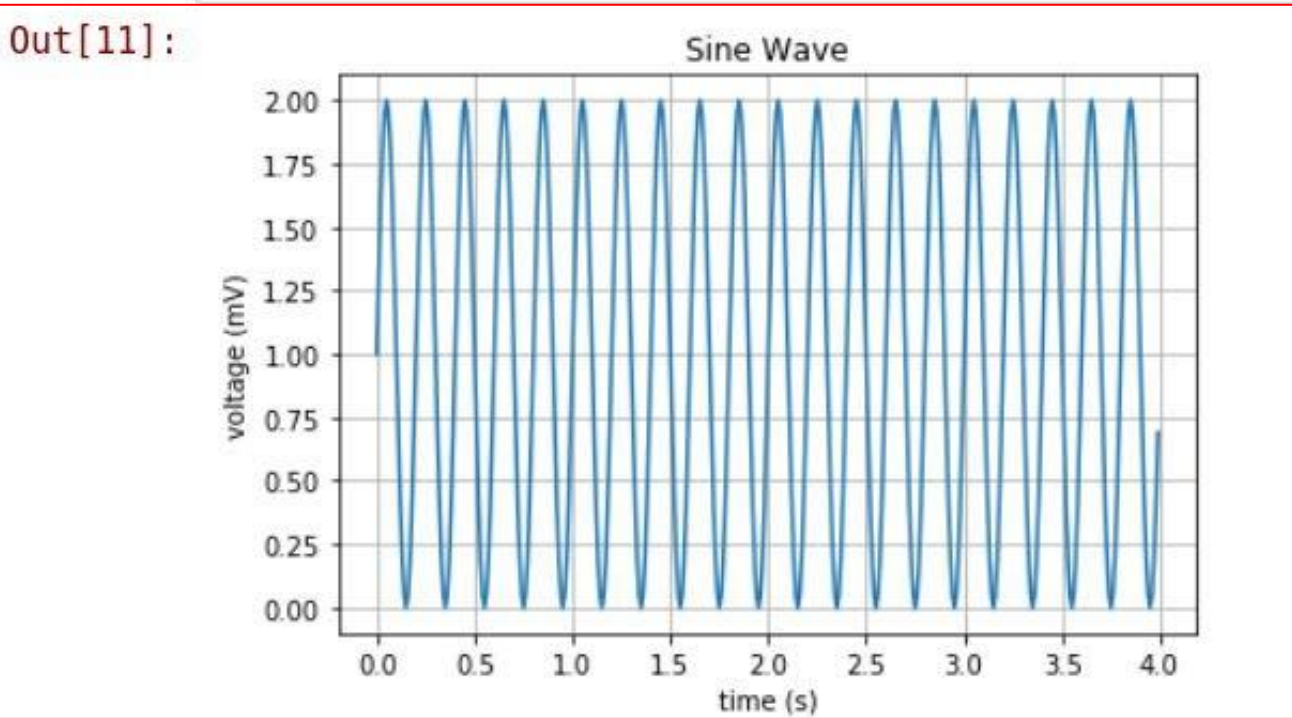
When we make a little change:

```
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()
```



The difference in plot will be marked as:

```
"outputs": [
{
  "data": {
    "image/png":
    "iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
    AA/wSF/zAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uID
    luMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9vc4+2
    EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwbkzvKSeIIQqaVxZKcOJL
    N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
    SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMBQQC4wYMWLEiKGFWDGEiBEjRgwtXAljRowY
    MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELjBgfGojIXSLyoojMiMgZEXIBRH4IQcN
    1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8CIB2Tfha4t+3zPcC7H
    b57Tyk15XgsMWJEQiwWYnxY8BEApdS3IVJ1pdSCUupxpdRbACLyRRF5PjjY1/i/JCLvichZEfktEZ
    G23/8PEdnl//aYiFzWhe6zwJ0iEvDa3cC/A7Ys+u5Z/7qrReRPRReSkf+0/FZGL/d8+LyLb2i8u1l8RkYf9
    /2dF5F+LyEEROS4i3xCR4YjPLUaMJmKBEEPDgj1AXUT+k4h8VkrWa5zzY8APATcCfwX4NICI/O
    /APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAzcCiwAv+..."
```

Jupyter is NOT Good in Version Control

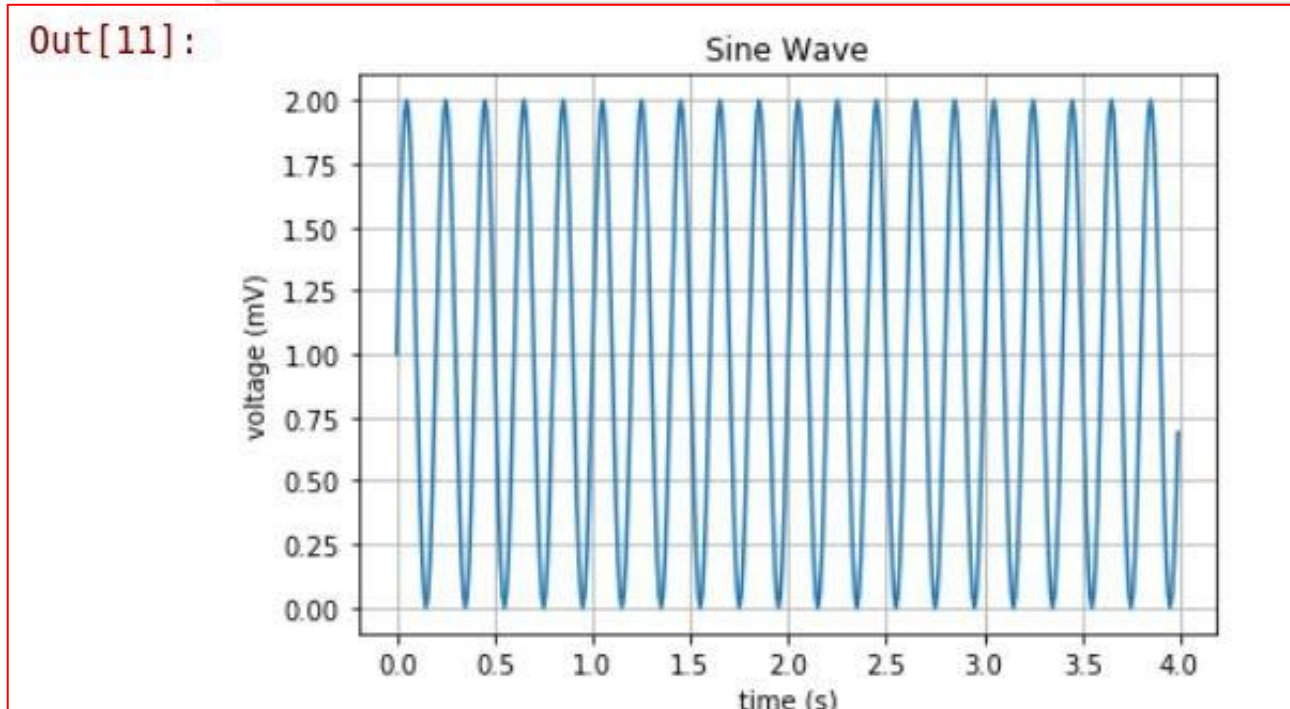
When we make a little change:

```
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()
```



The difference in plot will be marked as:

```
"outputs": [
{
  "data": {
    "image/png":
      "iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
      AAlwSFlzAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uID
      luMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9vc4+2
      EgABHeQEKV SXGGRFLembFN SPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvK SellOqaVxZKcOJL
      N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
      SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMBQQC4wYMWLEiKGFWDGEiBEjRgwtXAljRowY
      MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELjBgfGojIXSLyoojMiMgZEXIBRH4IQcN
      1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8CIB2Tfha4t+3zPcC7H
      b57Tyk15XgsMWJEQiwWYnxY8BEApdS3IVJ1pdSCUupxpdRbACLyRRF5PjjY1/i/JCLvichZEfktEZ
      G23/8PEdnl//aYiFzWhe6zwJ0iEvDa3cC/A7Ys+u5Z/7qrReRPRReSkf+0/FZGL/d8+LyLb2i8u18RkYf9
      /2dF5F+LyEEROS4i3xCR4YjPLUaMJmKBEePDgj1AXUT+k4h8VkrWa5zzY8APATcCfwX4NICI/O
      /APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAzcCiwAv+..."
```



Jupyter is NOT Good in Version Control

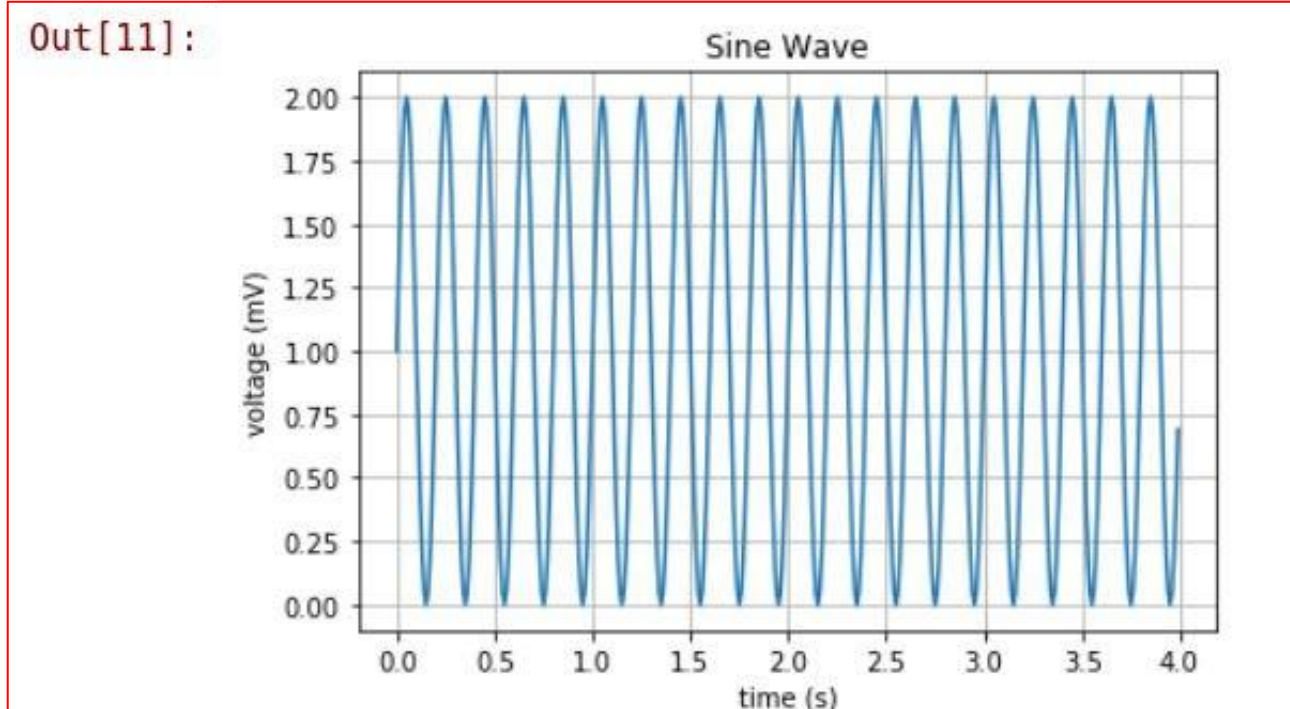
When we make a little change:

```
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()
```



The difference in plot will be marked as:

```
"outputs": [
{
  "data": {
    "image/png":
      "iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
      AAlwSFlzAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uID
      luMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9vc4+2
      EgABHeQEhVSXGGRFLembFNspN7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSeIIQqaVxZKcOJL
      N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
      SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMBQQC4wYMWLEiKGFWDGEiBEjRgwtXAljRowY
      MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELjBgfGojIXSLyoojMiMgZEXIBRH4IQcN
      1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8CIB2Tfha4t+3zPcC7H
      b57Tyk15XgsMWJEQiwYnX8BEApdS3IVJ1pdSCUupxpdRbACLyRRF5PjjY1/i/JCLvichZEfktEZ
      G23/8PEdnI//aYiFzWhe6zwJ0iEvDa3cC/A7Ys+u5Z/7qrReRPRReSkf+0/FZGL/d8+LyLb2i8u18RkYf9
      /2dF5F+LyEEROS4i3xCR4YjPLUaMJmKBEePDgj1AXUT+k4h8VkrWa5zzY8APATcCfwX4NICI/O
      /APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAzcCiwAv+..."
```

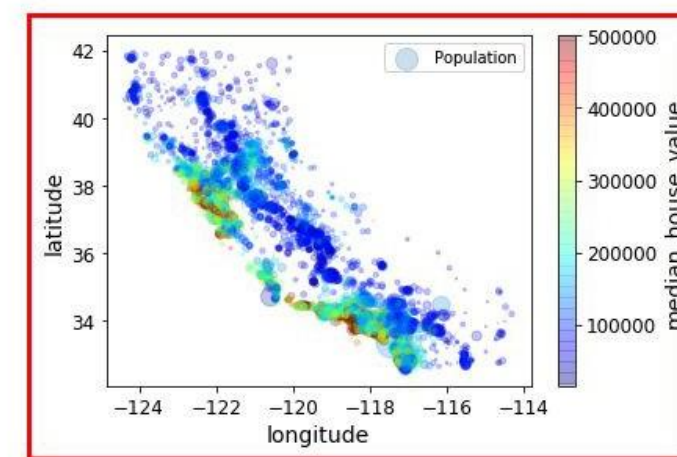
When you have a lot of plots, your changes in git will be thousands-pages long... And it is impossible to tell what the differences are...

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!)

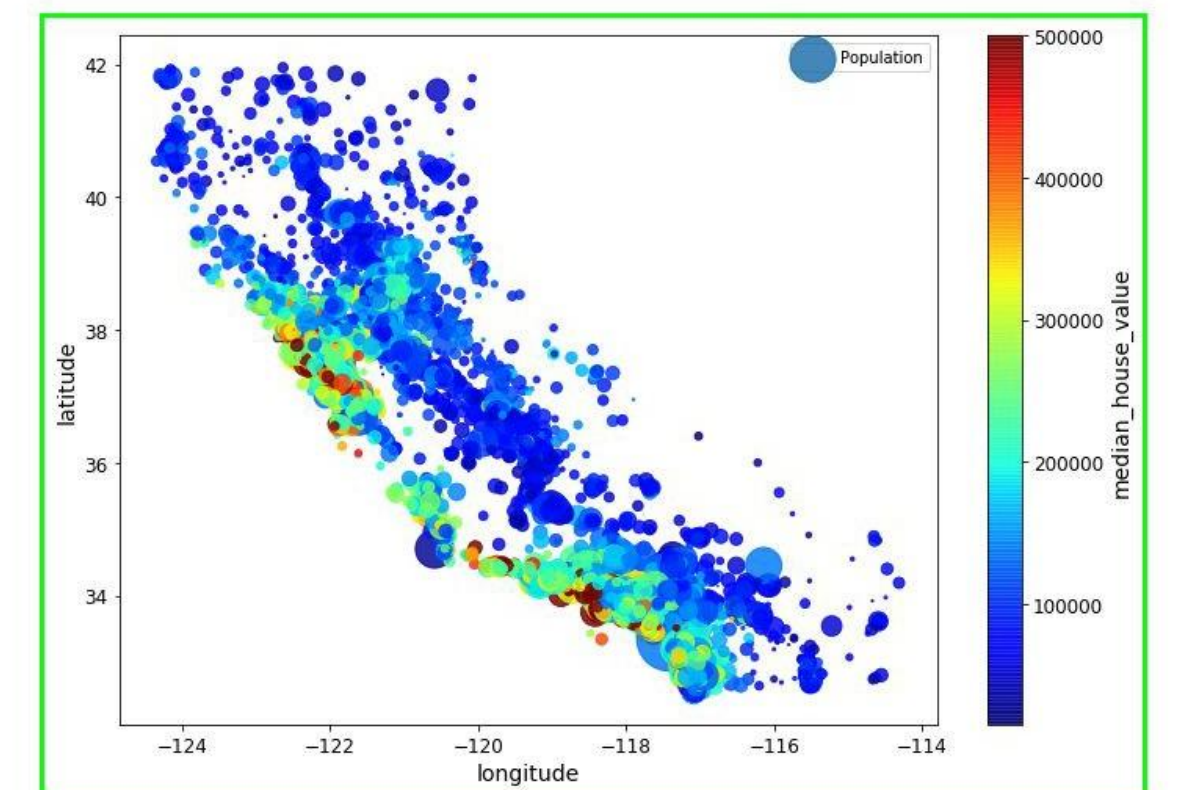
```
1 housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.2,  
2 s=housing["population"]/88, label="Population", figsize=(8,4),  
3 c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True,  
4 sharex=False)  
5 plt.legend()  
6 save_fig("housing_prices_scatterplot")
```

Saving figure housing_prices_scatterplot



```
1 housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.8,  
2 s=housing["population"]/21, label="Population", figsize=(10,7),  
3 c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True,  
4 sharex=False)  
5 plt.legend()  
6 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

Questions?