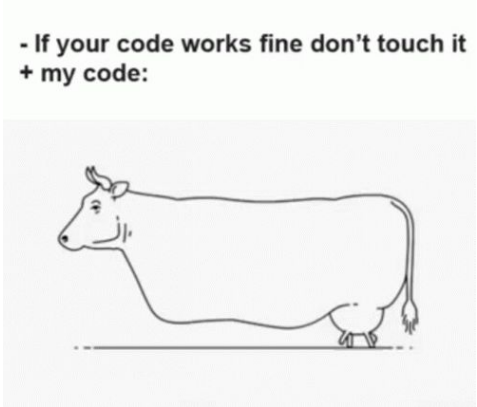
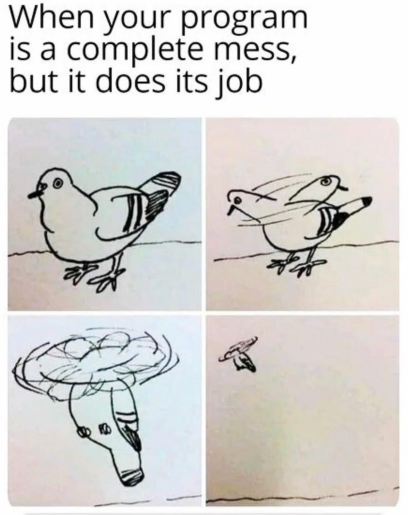


COGS 108 Week 2

FA23, A04/A07



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

Plan for Today

- Basic introductions
- Tools we will be using in COGS 108 (anaconda, jupyter, git/github)
- Github Demo

Your TA

Jason Chen

- First-year graduate student in Data Science
- B.S. in Cognitive Science and Neurobiology @ UCSD
- OH: Tue/Thu 4-5 pm (over Zoom)
- Email: xic007@ucsd.edu

A Little Bit About Yourself…

slido

How are you doing?

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

slido

What is your major?

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

slido

What is your level of Python proficiency?

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

Section Philosophy

- Attendance is not mandatory (but encouraged)
- Reasons to attend the discussion sessions:
  - Demos to help you set up and start working
  - Hands-on experience and personalized guidance
  - Time to practice technical aspects of the course
  - Discussion is not supposed to be a monologue – Please ask questions

Programming Basics

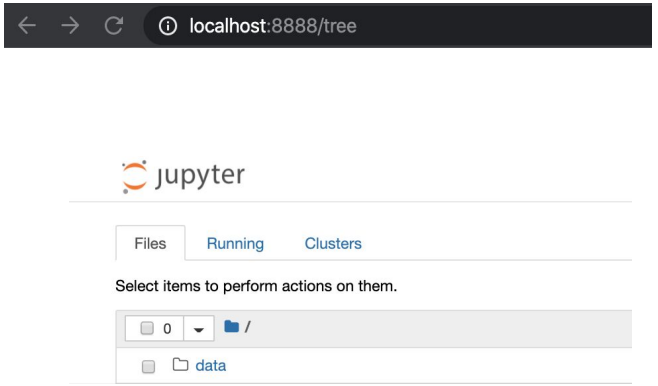
- This course assumes basic programming knowledge... but not much
- Resources:
  - codeacademy
  - Start Here: https://github.com/COGS108/Tutorials/blob/master/01-Python.ipynb
  - Python in detail: https://jakevdp.github.io/PythonDataScienceHandbook/
  - Pandas: https://www.dataschool.io/python-pandas-tips-and-tricks/
  - Git: https://guides.github.com/activities/hello-world/
  - Stack Overflow
- AI helper
  - ChatGPT (warning: DO NOT rely on it too much as it can slow you down from being a good programmer!!! Please refer to the syllabus for information about the use of LLM)

Anaconda

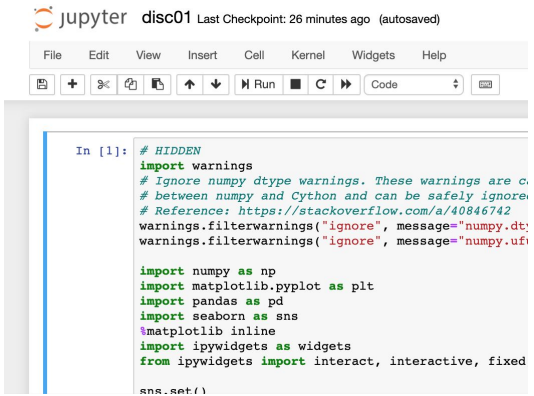
The Data Science Toolkit – contains Python and data science libraries (including jupyter notebooks)

- Download: https://www.anaconda.com/products/individual
- Installation: https://docs.anaconda.com/anaconda/install/
- Verify installation: https://docs.anaconda.com/anaconda/install/verify-install/
- Make sure anaconda is added to the system path: For mac: export PATH="/usr/local/anaconda3/bin:$PATH"

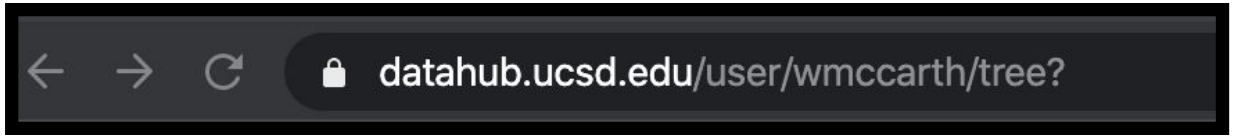
Jupyter



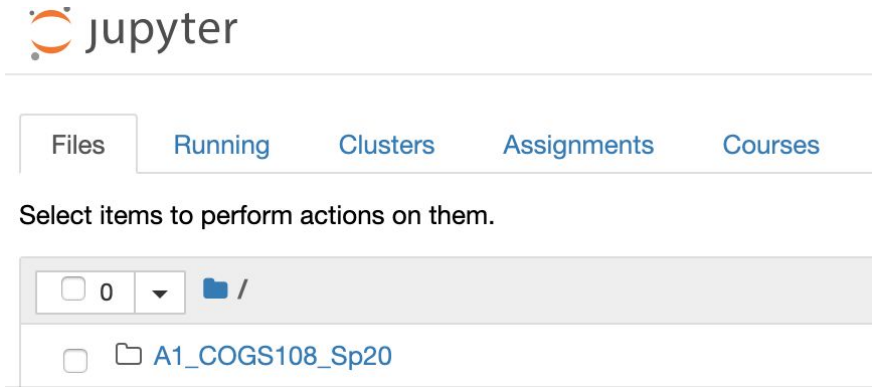
- Python code is run on a python interpreter
- Jupyter is a program that creates an interface for typing python code in a browser, that also runs that code in a python interpreter
- What does this mean?!
  - You can run python programs from a browser!



Datahub



- Datahub lets you interact with Jupyter remotely.
- What does this mean?!
  - You don't need to worry about installing Jupyter
  - You can use datahub to create and run python programs (online)
  - You can use this interface to fetch and submit assignments



Check Your Work (in general on datahub)

- The tests built into the notebook are (very) minimal.
  - To write your own tests, add a cell with assert statements below your code.
- Make sure to click Validate before turning in your notebook!
  - This replicates what our autograder will do.
  - Hidden tests on the autograder aren't validated.
- Don't click submit after deadline! (Your assignment will be marked as a late submission)

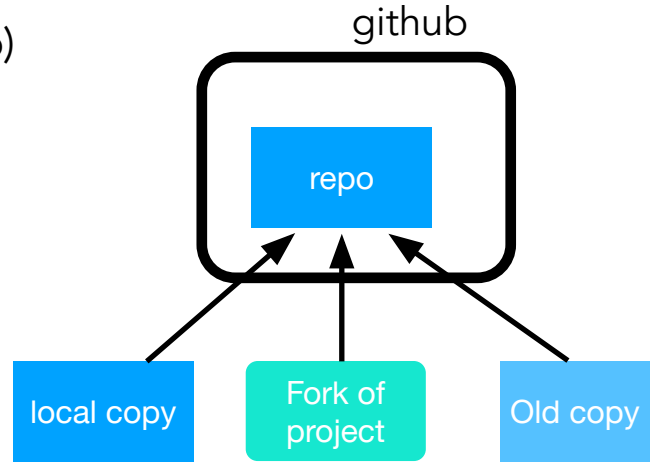
Git

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 this 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]:

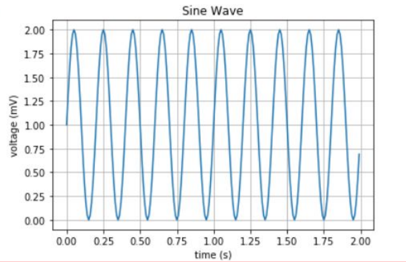
```
import numpy as np
import matplotlib.pyplot as plt

# 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/s9vc4+2
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSeIllOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCIdbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFFKE
SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMbQC4wYMWLEiKGFWGDEiBEjRgxtxAIjRowY
MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELLjBgfGojIXSLyoojMiMgZEXIBRH4IQCn
1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8Cfr8Br8BrAE2Ab8ClB2Tfha4t+3zPcC7H
b57Tyk15XgsMWJEQiwwYnxY8BEApdd3dRbACLyRRF5PjWlpcdwx14/i/JCLvichZEfktEZ
G23/8PEdnl//aYiFLLb2Li8uIl8RkY0f9RP//6FZM//qrmRUjQZDzR5Mvm9PYz7iwq/2ZdEVyuwUXCKsf7G8CfwX4NICI/O
/APwL/6sx3+ehh8VkGGcO7tORW5uFPyvQzZV1pm+LIZI+NdZNf4FSb1FBHTPgiRFLA9WkRFxd2zQm2wdXvv
/2dF5F+LyEERO4i3xCR4YjPLUaMJmKBEePDgj1ANXQO+k4H8Vzqf3NCb2tLUeDhisq5d1MAsAmX3tRvj+sKZvz3MdBAuBjdJJs8TSQuN1JWwBRUV6L+jGWLb0nCwdOZV07XThcvOcwnHQZlIzjl0h2zIS9JlPwvBi8KSjQbZxvcNWJ90J2yvLP2/eFOfLgk07OEKMWHyWVqWvUKkB2Zz1PxRvf5d8dFUzU7tPaI9xc5Y6RB+bFWg/+V8kgZPN/Fj7FBNSGAamuRGBSRaG7JCJkj5p0GsnvqZGYYYWsCsJcaCDG1Uj+M3jSjazH/99PaiBz1ZMzvbSjIXmJAB3aZ/gAhmgqRpHf7HBe12UoMg6rtKA3zZgRTM7cWzqyvM4AhR1aIxlenOlsUT7/5oCoE6eCw2fawa7tFCnxnNfd4g+7eX84UWGIt45YE+X6u9HFVbRtOZw2+a+0pFZaAIlV/UUxRuVESQCWtQkzKTQvOU5NGp/P1fBT6s5JOJFpaFVY/RS3lHpZ6yZfg8owNCkAaNuP09zRHBRwSJCRvIFW1pZOVkgkLUr5iExmtIJ86qydM1aO6umjAxgMuKVF0VjJQ==
"outputs": [
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
AAlwSFlzAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uID
IuMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9vc4+2
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSeIllOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCIdbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFFKE
SNGjBgxYvRDYrkHECNGjBgxVgZigREjRowYMbQC4wYMWLEiKGFWGDEiBEjRgxtxAIjRowYMWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELLjBgfGojIXSLyoojMiMgZEXIBRH4IQCn1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8Cfr8Br8BrAE2Ab8ClB2Tfha4t+3zPcC7Hb57Tyk15XgsMWJEQiwwYnxY8BEApdd3dRbACLyRRF5PjWlpcdwx14/i/JCLvichZEfktEZG2338PEdnl//aYiFLLb2Li8uIl8RkY0f9RP//6FZM//qrmRUjQZDzR5Mvm9PYz7iwq/2ZdEVyuwUXCKsf7G8CfwX4NICI/O

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

Jupyter is NOT Good in Version Control

When we make a little change:

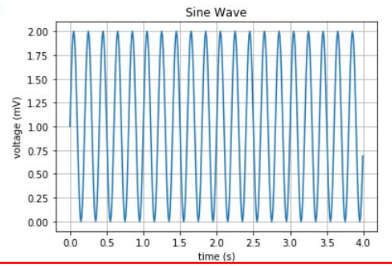
```
In [11]: import numpy as np
         import matplotlib.pyplot as plt

         # 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+2
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSeIlOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCIdbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
SNGjBgxYvRDECxHECNGjBgxVgZigRErowYMbQQC4wYMWLEiKGFWGDEiBEjRgwtxAljRowY
MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELjBgzpIuMDGzSGOjIXXXSLyooojMiMgZEXlBRH4IQCn
1B0qp+x3QfExE/q+2z5tERHX5boNXQ6/D8IgiE8CfAr8BrAE2Ab8ClB2Tfha4t+3zPC7H
b57Tyk15XgsMWJEQiwwYnxY8BEApdS3lVJ1pdSCUuprxdRbEpxZPjjY1/i/JCLvichZEfktZ
G23/8PEdnl//aYi/Fz/Whe6zb0/Yi+u5Z/7qrReRPReSfkf+0/FZGL/d8+LyLb2i8Ull8RkYf9
/2dF5F+LyEERO4i3xCR4i3xCR4YjPLUaMJmKBEePDgj1AXUT+k4h8VkXVm5zzY8APAAPTcCfwX4NICI/O
/APwL+AnAB8Bzw7Bzw7S7X+AGQ9a8BnjXVAmZVAv+u...

Jupyter is NOT Good in Version Control

When we make a little change:

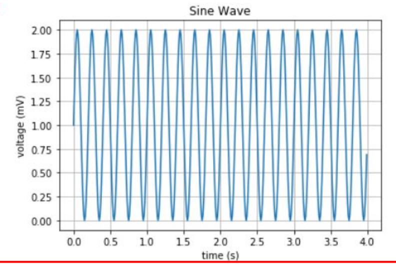
```
In [11]: import numpy as np
         import matplotlib.pyplot as plt

         # 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+2
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSellOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFFKE
SNGjBgxYvRDYHECNGjBgxVgZigREjRowYMbQQC4wYMWLEiKGFWDEiBEjRgwtxAlRowY
MWJoIRYYMWLEiBFDC7HAiBEDEJG/JiKPL/c4YsQ4nxELELjIXSLyoojMiMgZEXIBRH4IQCn
1B0qp+x3QfExE/q+2ERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8ClB2Tfha4t+3zPcC7H
b57Tyk15XgsMWJEQiwwYnxY8BEApdS3lVJ1pdSCUpdpxdRw0KhJHU2JMUhUU5JHUPRHF5PV+i/m8%D0cIc
JCLvichZEfktEZG23/bQwzUVR+ykK4L1/8W
G23/8PEdnl//aYiFzWhe6zR/7qrReRPReSkf+0/FZGL/d8+LyLb2i8uRkY9
/2dF5F+LyEEROS4i3xCR4YjPLUaMJmKBEePDgj1AXER+X0Q+LSKrq9lZtM09
/APwL+AnAB8Bzw7S7X+AGQ9a8BnjXxBLB30XfP+v9PAL8PXAZnCiwAv/O
FFFFFFFF

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

Jupyter is NOT Good in Version Control

When we make a little change:

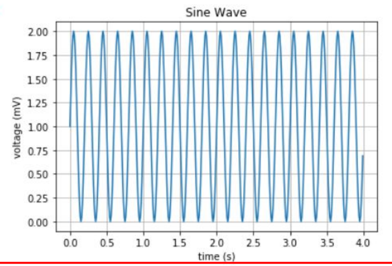
```
In [11]: import numpy as np
         import matplotlib.pyplot as plt

         # 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:

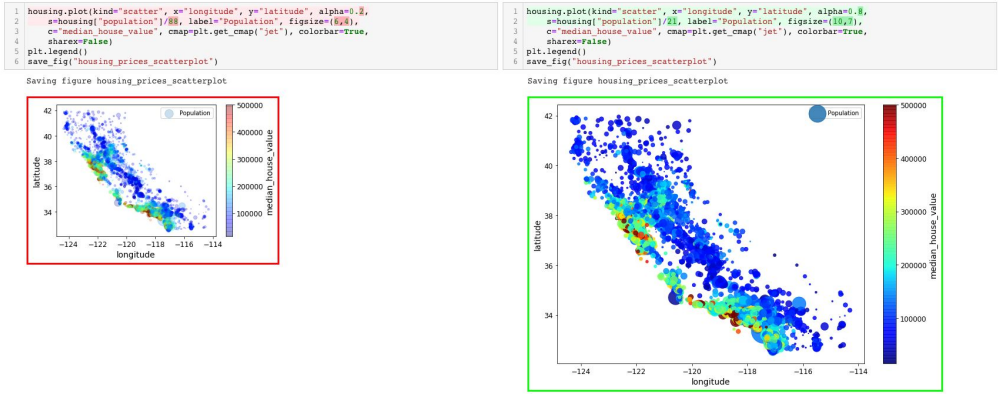
"outputs": [
  {
   "data": {
    "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYwAAAEWCAYAAAB1xKBvAAAABHNCSVQICAgIfAhkiAAA
AAlwSFlzAAALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uID
IuMi4yLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvhp/UCwAAIABJREFUeJzsvXmcHNd13/s9vc4+2
EgABHeQEkVSXGGRFLembFNSPn7Wyy45i5UXh5ZjvcSy4xcr78WK5bwkzvKSelllOqaVxZKcOJL
N+FHc0dxJEVxAAgQBAiCldbDP0tPT+80fVdXdmOnl1q17ezBm/T6f+QDdXVXnVtU996z3HFKKE
SNGjBgxYvRDYrHECNGjBgxVgZigREjRowYMbQC4wYMWLEiKGFWGDEiBEjRgwtxAlJYKPL/c4YsQ4nxELjBgfGojIXSLyoojMiMgZEXIBRH4IQCn
1B0qp+x3QfExE/q+2z5tERHX5boNt+jFi2EQsMGJ8KCAiE8CfAr8BrAE2Ab8CIE8CfAr8BrAE2Ab8ClB2Tfha4t+3zPcC7H
b57Tyk15XgsMWJEQiwwYnxY8BEApdS3lVJ1pdSCUpxpdRbACLyRRF5PjjY1/i/JCLvichZEfktEZ
G23/8PEdnl//aYiFzWkfwuVJ8pMBWBP5GOh0+LzENgwcsBRe8sk1vUhuTU7qlPCEeowTnmOUdQmYoSOpdLp3uCrTDsYiJdGGRWbkXVzTmyy3zcGHdKM7PhUBDT5AjCrl5S5ArXSXMCrVdR7TQAKTs4aCWwAWHAVLK/MzeJzHxyYohRn/9WrFxnn7Zh7ifdSh2pW1NzQxDMlrLWy/gDG1+VxvuqTOxn1TENvdndBK+nOr4T4ODlcDnKGPNAs+ZYwxRVB21LDPYyR35c0Mnzs5zBOh7zZrcRInUHqqAy/Tw6g9rpKBI9qGfgSRFzh5twsXhzWEwhjXhp0ezm0dgqSw11OIqV1USZl50ESPBmljtNvaRpm3HhPE+u1Elv6EZ8m9T0jN4i/ShNq+n9exEBANUrVfZi5cIHdF/PZtntvUtgLQAs7rPUC+xcnGEa5xNMrF7hFrmAHn7vvBmaYdUc/dKBQDe+DTXRgWc7Uw7C8wwr/TZW0pfr3GqFlo4TEk/BTIpnmnHpLLPuHQk1mAvnGzI+2Wr/b7YC/1WDn1JRjNAGhb3Y9+FaL7zP3sFaObZSxGLXLtvdPFjM7HhsPw9AAAqeJrpOyOGe2ndb0nvP5fObt/Zg5sFd5iGq9cEJ8hILJgd4xlCmxzBpzV/f3jTcnH96EknSc6YVEnuHT91OAB0ykkZtRU+TGpeRQ0TRQRwgQCjYyH0FLMwXkzKP0PNAXsc//QvUvtSMtIHPSOsrpqTIdk5x9uQvBsKwpUWLAsmYBvZXExfkNGnJlXTPw9MNaAAl4ZfTLhbIHgnB0zCnEngNUwI/J75eJudO6pvgAJbEy3jDHEyqy1aFgxg8s+NkUnPZQiaMaOe4KzTaVSbgwHUrX6S0cGYRfazRZH8QE4HedQr9Gzj+gF3Cf46+BVuuZ5PDJvGw8+P3uFqIlXSnY/r5iBGJSIZ00VqzYAaS6Q4jKyaLSNlWDmJgt5rRDxSeUhJi7OXsxIETSQSYCBxWhgqvOvWU9IcXfUqW4KTBNfZ3e0tyGRVpHUsjMzJOSUPHnu0xnEB0ITMZR4+hWtkxJ0LrtmORp7DbuH4+NhxHRHM7qW+5aj8ngtDkEvCIEuI4W1aK0nr8H6Xfm6nUPl+Va1d09Omur+ZhyM+IHfVc3LMUp0oqy7ZytHuLiwdo7jQAwU7Kag/a6pIvLvKBQTSgTsqkKHCHA0xJxk2a6NeMOsiTpGzHZWQkCpGyNn8QbOzcGTP3TPU5rq5C6XDnNWZpEQQwIc8/6eHmSUtLWQqdBaS/x2CwqxUXaiUz6jiT9tanXI1zfTSvK3/8ZeqAA9uPr8xU4VK8BtfnE4GRD1TzJHi/a/UAdCljlBy2VmCl1vcdlTCtbxXzJG5IvC4E8/u2B93b+yRN0IXJeqpKi2hYGRodptfgA8y2C+7DPNRK7m3Rz8k+yOl3TfZ9U5ShpNSUHQJaJdA8gW62lsb3eu4AT3A5KHhs2W3Dsz3VkhG8WIlTbZTp58MG06ZMvrkG0YzS37z4JjJbm2ozABCXh9nUCF1ZDdoAY0x1EhHoxzhR1d9g8f89fUhZJ/LAB37aKHJhVaGCsrjeTTsaWWEsxWagg81G+qv0AeUoSCQc91kntKnHpfDMXo9hfIwtLIpmlgRyLHdTK0unVqZqF9oExfyJHlaMG8xQcm6Oqq6Hgeb13yG46GGfcOYzqohr/7RHY1nyzvUvEspkmoeSajAYs6sF2ArUdXw6zlAJH6MP/+QwTMhnUyGi6W3K0fB5c0wg44dcLS/VQM+Ht/gHMnXxdX6+3VbZ+rHNWGtJdk/iP5Ptoj9hOwyYYfCbDCDzfBm7BC3odK+3hPJ2gcJ6iDQzoCXZ6jjBZ4Ejyfb/W5DUsd1jvkejX/OQFm8Un9i0hFpNfk3jCUn0CP4C2GAXrEqHGWh7zYBgTPnkZKYqUkf5XSXJYnsJNw8jnIVnBRJOxq9aRBrT9yHK+h8nGs13tKMg0AtrfS2FqzbZ6WmI5xpfbTVBXBTcPdxN/9EzQHQArwrTrPyhmZtsL5vo8g5Ydr2JkxxTPNUgR9x9BjoqjHBxj8i4Ym4HJ24T8ti73ASbl0JhtWuJe6iwkyg3fPAo+eB7Svr4zsq0A1y2xowzwDiZWYqYb4KHWNdtHtKalaFbHD7E+X5U00+HXjlp4Ll3Bb+x34aRJRWQNXh7e+ybaHcUhHy1iHlX6oNNpBJWxLpW8HLHghZkF3JOH29/YC7ebPpnMfghbXHyoEI6cUwCowgyQDwH2V0Sk6+Fz0NjUTc7jfAPkWZH9MtjdOyIh9KNtpvrrEWVL8gzc5J8dDqUd5c3fvN1a35CQEj0xmq2Uw+kI9xYu+hoZzY/5tdY9mygHHk6AuRjbO9BkamrikKsq0Os9vX8fXtSQhxgfAm98/NHHR+1AWemeNH3PaDRcOVi+QqC5H87aSUSMpxeALRUt3d5e01xEyhW2ylcRzqlh2fBqg6NNTmcF3E2bSKFhbZy61k06JMZ/eiq+kLlr5Y+Rz5jZ8NrkFhgVk6eAtEI3+4Wvn88p8t/L+8AoHPj7OJpqrbqjH3tOAfrEfi3WqhbFmy0lvtLX6XtKNs9xTvzXGvAd5W7gpY8kmmG8MNUF2yXt1/Bqf1WNa76ck+ReIJz7xmDYHUCm2nBfHdwS1OE9YpQMQTGYSXzQWjkjYXdIvizYXSWfxzYkzz8X8/2JQFkGmyZCQgmDUSqRONTuTd9Oja7g1iXOhzbczZO5BvubuhiSgZ4g4qMrYPgfoCjMHCZm15ZwEqK5DtSsm5J+fkGuQjUj3OAm4EESBvZbWIpXmUU8YsjhpwU0HEVuyiqxg0eXpbyyMc2VfhEP3bnxCdgCgq0TXWzD9/Q+TuRC0Zjre0T6IYeyX7eLNaFZIhimCzsWHsb0qm3bHl4DRNxGOIsRmCQpvFgjAAw/svFjQcuYXo6H0YjKfdiYyw9+cfhUIjJYAT7vA+wBj2TmTVTtUH1ndMiuzqGd6XUJiSTyCaXt5tGVpWdjPVcLw95Hvfl4LZ+jNJXBcn5kF0+lXRFOwMAFa0RKtUtdiu3AtIwHwcu9Rau8h9nXQbAZDnNnFjkk4qPNCT7HodVdWIvrh6XbPWKk2BU6UVUvP1k+A+zmtuCFtiu98KV/VHyDpDCU+oXlx+wevqnk8E4Uzwmhat6klBugGSxYsQfbdg5mOoRKUx/chNaaEIFmFX3GOFSFB+gaHs2OozKHg0HeYcCeVBwOWeg2QN7W4MrK8Dl2E8JaE9UFA27cELWTCrQG3fIGv9/ylNKwgu76lxJMvUEZT4+8Yg6Up7uDpDGTHl9yMoK9WkH+H/nHnxsKmF2O/UNSaKSZ6u1j8ldrpAWzgDeeaMihVWj0Z9ZNIkoaUAAgglvbaLehEA5TO5NEdMmzrXxoGiVZpV4bWyLLQHTKVEaRhUB/bBrCzuI/X+S3QUwkC3IR6LXMMt4aOLHM04lRWvywxm49tp/tZaCfn+i+5E6NGPStupAdbwnp8xHSxoPzDVk+23jeO0kxR+s0vTHAMtQP20mbMy3LTYRBE6ZUZJQJ8CoWXQClZF3F8nPp8v/f+vjH6KQ4dXQvJbfB+pQHoxdNHo6Q7Bh2eDGHsgVEMDhvoX15kJOvy1iY3MmHgxcYy0GOuCsAIOG1KpUzn4kSK62XJqwDgjwmYSHzMXcDyFTr7FKTNlfJ9GXrSP80qtDAnFnThFdvbfiqAUX1IGxEKlVcqUd6ePFxHmoH88RQnXtCvWvRAEl8EJ+/nnNCh9Pvr/yBnRMbR9LwS0s7hc/H+8sVs0ijW4Zvhzt1PmIAe5hNtqf2DhYc0JeAZCMx30QAU5mmi4mpkpvN4cIyEL1xs7t80ApRAA+ZfEjF3KgG+iA/eBd7LiJV2NNgJFrmWo5Ye+Fx4tr4oI8nj3m3J4hFzTDhBgdH14s8d2GeqAGr0wo4CzrNRuXffd8X2NL3ZQ1exy6XvxLDHv+2enuqnUj9Tr1FJ0T9Z+kGZFjWvJG2Jk3z4AVD9WSZlFtN0uH5PsX7XCN1bbnZvCvBDNXohbk1Jl7+sAZBjfpygGmHuNEsbjdbX/O95PY7KyjDs4kXewT4XfV6SoT+6TbMxbCnYkq42xBOiEEpzyDqGlpPaoxmeEETZD+vBD4AlPBMBOsT2gOTEgAzegfexhyW2GaewrjHJWXQARIaqRbPHEGYVUpJuTF5EPX5OJVPbbh4NwNRE+EcFq9I/mKd6dT+tsQxMkXkYlWahEDW5kF+pp3qRr6Zm8MMh4/NQdMSyN24/3CzMUYi2rJ9y2vIM1FsqLC6R6Wjqbcx6sRAd+QSM5HxsUjWZb0OHtmh9knu2qSOfd/bE5fC4OY7HK11FPqQkOkIIAtEDhN+Kb6OA+lEofg9+zBdhwcS27ERBMEbDV+B8H0j+hU4aVwoSNmi46hymdYzOI8q7dVdWnSsH0qFCTQ/Bfco+J7EuMyNJ1kNWGw2prc+BiRvmNOSmWl1fSo5/aTLSVB8cmj+DdxGYmiXzg5kZvonrqhp6nTekv7Ag+MIz4GVSOxp+P1Cb+5TTSMXpNcaSDe7ryZ8d1UmwXFM1sckqRDWt6Oe3PJIHOQA4FJUYeOY2/dm0/2g9HXqpMDQfO0TIfyhnlwMQJmNkAf+ZrpIMbW4QapSL0BsTnoOKr6vjnKoxcMTbCVakEdh5bnX7OVtHAeSdf50oVjlj61oE+PFAyLNQIOaeAF5+tkdyP8NtJpwjYfNA7ND7krvBgyOjuOv3UdOcG2xs6kWh+Ez+ViswBF0WUyGpSIc51nCH8lYcoJyW7HHGLCN3Up1ylJj+c0JTtvybfmEzdImIXCRcIh6zUWdJfBh+YFTebOWxUw6ROW/+20nJqLa/F1Gs3LxZ0jwPLzYWnahVkzHT8j8OaWo9Iw1kw2Eiq5fvwrfP4YIWaiRAxmOPNbcqCNoyKCbYSMkK+lSjbA6IzhXdT2VvsVtowgUfp1TOxz+KhH4ov9xrOpjUxWAg2wW26mJOFx4RUbsinF/09/W6fBCmmPu2xpC4D1t1UOY2Ln/h3JCDPn+vFmcS5WrDbmvIFwyFBT1OCikykjWLKkGiEYYzA7KPWMB/4u9bBrg1d3Of4QfIMyOEb0HUdk0ye/c9DKNCRNsEDyRrpmeNo1D/bBttdE1u41PtZKwrd7MQjkiLSIJB1U3D4jgGq1EKO+8k5Mg5txMZzqGNYBYhuo47b1o0q/BQCOTChqJ7N44awLrj0IJJvIfabyXgm9ihTvhICngpQ6TbuX30dQ8+xa2yqAGYsStWs64mMsx+8+fsP73zlrPaAc5EOsQG7F96pq56Sh/4Ir8dXZHb7g0/pRjC4i/DX51+37NHEpUzrL4SWOtXExRdmQcIn8ilCZA/0WTEDYP/bHqPu2ZBALEkQBb/1wzMRNjlhNmCfeVHe/OiCJdqFCmLYmBO6ilFpuJPb19v0q7o60sRQgvD4spaQ9fIgZE0kKyhMLy8ZEl/6ef8uLDjHHkWSkfMOb7/j9Ih6yMMdWyWlnM7NYJMoeZYkZBhY/WhUP23U59z7cvAiX3PONrEHB1ZtX7GVn+fTnxFnHVE0vVeSVuX56t9gT7NCe1IlSDXsBEG5sWbE4OBfdl+O1b3NzmU1SqONN/Xx2C61c08d1Y12/IxzYUIX6mQNsmMZ2wPJOVZ9dc0ttPPXApHe5J0VDdNkY5mXjyAhnTtnMJU5wxHBYDWkkqs5o5S/2pcqLQqgd0AVH/8s6hWcYObzf9Qx6q2dNYJI3RX/k/1BO/F85lR+aKrZx5IlEs0cjsBVF8KfbvNbrKN1wfE2PdVGIKHRwhBdSpOI+3ILGuwL8X8J5FyWT8CAVVUDnT7P+AYBPcR2z2AGU0CJJaHQPtKyltHmpm/LNwJ6cIdqzzRKmUqY4mCTsWrBPtDrOJpgI2c7R6NXbMWyWwWcfaNj9jOXvrzCp91REoxd4cfrZEF1O9Stb4BR0p2xMAdR2XXOWJQ52vqKfOWG1q/Y+7F1UH7EPjONTBB87UVkEdUhlHcJfHggP9WzUdwCZhd6cCMiHRsbT9fAQMI8/bvEh2rSGJjZ8aXyn8D+N3HSbeulrAG4IE6lJCT+aZUNRwe7P9ChYz1/zN8frVrVYLvN1tKemnkdbO3zrH5nB1EbAAgF/B3ZYL3qDnNY8jzhTs//NY7qwV4w+RSuWRmdVdUO6Pg2xaCKjEFhemzlj9aLUhbDTVT/PTy3I5hhEy9jvntiGOjqtdeNgpqZwDCRjzSKdXjcb7t3JgGNhNKBX3AeBrXZZvb9b24H+KIbrvL+v/DCM0AGcJw1xF4eHqkBy1AwM+z77l5kiQyFcNFpFRuDP86AQOg1tJTumdmLcEt0p4Ep3zv17NvgTxuMwYNIf//7MB8YOFC9h39FPgeKxy+Azk4ufQsgY0yK3SAbyHNOb2rRwiiTkdS4kpmvkwaF93XxPnffdqTrbiuTm4HTS+Qp4jfJKfx4N30Nuk/5TjTwPCAKcXLBTKmZyQrGdobJGtyZ7ou25DFxO4sgeoqyIEW3AE6L2MfdlBoDj5+oBuSNNejwN9yLsGQG1yBUVfJS1HmLhFfLeiWVl2t7WkuoxKtXEO9NKuKqmE+OL3NSIA07a0WCZKiqJi5Tm/tsb+hl8hh4B8b8WhjJhFYyxq/iwpngnrDcz0h9x7Bj7AHA/v4iHh3Ncr5amzYl6BV8GLV5ea4jgx6WBSFKxlU70nbi/3qHGIBFsybh5O6gGaPGPLR6CEaz9LYg3l4BxF7I8n06RMXbzSfdtqbv7jowM4wyDE54VtgVOsw5VmNuI6Q3GnjiGGIjwcPafA3x9UhbfM5gVtS29oE7C7fX+sfSg82bQ4sdv6EbJhz6Ki9MaOMGG1B4KsigPuZoJm0p4wfLzHhvI3Ifwqq9jsd0hGtlB5y1Gvfm/WcYzYG+tTgcVHvnYT9QtKBK5XEyZ77hXOjaYYnL/ktEzNj5TOXrkVX7/CPCNgS4MgrW4TnM6GhDPx9KR9XA8XunkK1kBP6mJCzyaC/S4zwRYgkaDLlxU4Fn0ANVvkCwhoS1Df9vCzi8JSi8iqWyfQdOrKVfSpNrxf5fIBSuGJrcQCb5hBF8IOkqNa6FUQBg+ZTnzn2Oco3/Hio0WSKwoxHQtuG/d8xdZpBL3fHQcq8nxuvOjpjmkTe7DNn1EHpBuhZ4njUO/3GnvjtYaxPbxwB1y3bzyeGCf38pPTlW92gzUdrBU6zd1JEjwFh+Fyv2IGQpIU+QVWQcxyo4IbH6zTDnPEg0wPlEqtDfLBfmO0BBuzn/lxNS7BWR5bEafjMbuoSvhfgTDrgi4L8g1uOkNkwZB0gUXAy2EAPsVV2B8uLunvIbD7OL1lPsIyuybD/DvPWT1OvFZqvuqOM6M4QelBmPi3vvtzg6ZpcO2y9kI6kUHG1IHKNzNmLnvUEsO4yAH6LlcxCh2YNHhYO2a8uY3THu3kaR5YePsF9MZ+Ncxob6LHvLfj8j2KVyMPUfyAE3y+B+/kXjpKb7n+XaoXNxF7rX/sX4DRTsz7hODzCHRDNBq7ARvKx6RWn0fBJgUBYbfyEMLeO3zbfeOsPz3gv6uX7W3r/OTMGJYf3SiECXfQPj65CCDR5C6WCy0tWy8zB99BNUbTAlakxZVTHB6yYZBIlGuG/MTQkbWUI66KmIXBCQqbBE5sqwI48aY3CKqc8GAk3ogwk20hmVLa9Sr5q3T/HRQAmbzHs7mBC1m9zd4W5lfwsG38d//fGpV+9eWlq3n2KaCSr5IbZtmfKMJIU7q/rhVZNrZZp7pAH5d4HQaBdy2tmOBBKwg3H/TrywgE4ttjQ3XlQ3eUGF8JvYMPfH2h/fhXxv89tNRtT3eAO4Hgmrj+/jiU98zhm0nDxiuuwvtAkcVNvQ1wLLv/NzHtRehVxEeXR0EszK7ra1NQu4wtgzIAzH0ay1fk3+sFbVAZFhoz2Wx2HM8tKEq7ooHywQb8ZSJ7qa1DxcS+JtBg7Q93+IoxK0NgKdsqfh3PNmcbw+H4yP0PtefUXt9tThm8lj7tc4mORyAeiGhLRVWaLaEK4BtVd95WN6e8nqH2OeP1Oqgi0e4g5PNqnSL8Nqj6F0zbtT0B+fgd1EXbqcu7T8VjLnn/PsZBynuOA4i8T7BKkuY5f9oNEhOt7Ue53S5SpzG/IMJ6z2s7Uq73DkhKE4P6ROOqAOsbnqqFBd0t6zJ3zijJ7W9EyKJD0mdPLfXRnnr0cHu4gmP4T7Lur5p5zqf2+D95hHAavq4iGqazzT1LVvnz+B79HWr+9iW8ddnrpf8Tw9r7Pnz/EPxyNdfuNWsRyoSxpzrSU1dnd2D6yXe5IRrq9pqFcJ+t4Sd2nW8KpJHQG3y7HJn/b4rX33IV/Rm/DrPO6p1xKuVtZq+WiZvZ7GPTsXXnpnT0GEHLx3MWrjDfz6L8iHX8y/yzqpHG3HeTv6EN9RIWD3+U/ZnuRPGdBd4HBBpY2aOO/FN7I1yycx6xuQm89XcJBvjvh0FSpHm4YKvL+jHuhSm9RaF6xiHvN3ZnZXVfK+XnEH9jsbxTNb8e6+5/t3Q5+tb9u/Vf2+u4hRPXrqJoCuPf/r37a06RpjrfPTWpP9tx9dfyj2P70d/T29+Hm+HetN9LOBWgFnlvyH4zBtyn+5z9JADdZ+P69e+X4EAPGp/znn5+/LT1U7+WlY1Yv4F55+v+G94+F3P6iP7X/S5ZZ/Xf1d/M6cfkJLL5b2W5fh4BXYfq1IuCvwKbAWYdoWrF3FjD1aXQ/g3mPyvzxTfrxNz1+m5yvGx4Kyh6/Nf4wPAt5/Te8I+Tym//z/3dXfo3WvcD0wxPvXsBuqnZXdfoVp37H/V//Y5VpXf/1OmT8/mpI88PT9QQQPS/eO+v+xNpT/7L/jQ/s9N6vPrT/e33v/7/nD6bvuw/fp3v/j7+dv/p/hqN/hv2wU/0/f/2f9/nv+f/9/mT/X7/H/7/6b/eP9v9t7/i8g/v/5f/4Pv/3/9/59P/2//3f/v/s/n8AbXNyexmNNYS5OHzfOd/2y/Mdk7Kf5aW9j7bLdmZpO3hXiMyzkEKvkhrq3//t/AyPZ8/OuI/9P/8//1/f6/MfuxmVJ+P9n/+v/+/z/5/7n/79fm39f/7f/8v/nf5/+7/PfXm+sP+H9/+/w9Tv/X/+c/9fb/4gPv4A8AnoZ7w/EeBPAAB+zngH9v9Ppv8/1/1vBffdz+P8eAP0fDz/fvdvfX/4Hvf/r7/7B98Azw/2/vB/e4/VH+ee/8D/7P/jc/D/P/fBp/n3/d/Hf2L/7/vF//f+f/D/+/dH9f6w9/+D/D/5/3/mf+/zR/n3+Y/dL/6n/mcfT/w/3f5/3/f/4+//7/+/v/+/2/+/ff+f/9/+/v/+/3/8b+/7/9Av+/8E/+fwGiL1ERnx9OQbMA/gNr/v/8P/c/1Gf+f/HP/9//v/9f/3+//E/+D/+e9/7v/z/+n/8f/7/7/r/gP/i/+/v+ovH8/vP/4f/+f+v/q/6/9D/7/+//4/3rvwzP/n2dz/+Pb/v+f+bn/j4/+f/m/8f/L/+/8P/l//zz/g//+//wf/L/+r//f/8/9/+//+/8z/9X/4//v+//z//j/9f/hf//c//9P/i//3/+//x/+v/z//X//d//9//2/+/f/+/+//6/9X//f/9/+f/3//P/o//n/8f/v/+/+//7/8//3+/f/+//9//f//f/9/8//5+//7/v/+f/5/+3/7//3/9/+b/+v/3+//8f/v/+/9/+//v//f//v//f/9/+v//v/9//f/9+//7//P/+/9/+v/9/+//3/7//v/7/+f/3//3//3//+/9/+//1//n//b/9f/3//v/7//v/9//n/+//////f/9/+//7/+//+//5//f/9//f/6//7//f/9//v/8f/3//f/9//f//f/9//3//f/8//3/+//v/+//f/+//3//////f/+//+//+//3//////f//+//+//f//////////f//////////////////v//////////////////////////////9//+/////////+/////////////////////////////////9/////3////////////////////+///9//"

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!)



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

A Few Reminders

- Lecture Quiz Q1 Due **TODAY 11:59 PM**
- Practice Assignment Due **Wednesday 11:59 PM**
- Pre-course Survey Due **Wednesday 11:59 PM**
- Discussion Lab D1 Due **Friday 11:59 PM**