Welcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try https://etherpad.wikimedia.org).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

---

Links:

Inflammation dataset: https://swcarpentry.github.io/python-novice-inflammation/data/python-novice-inflammation-data.zip

Lessons with inflammation data: https://swcarpentry.github.io/python-novice-inflammation/01-intro.html
** This curriculum is useful if you prefer numpy methods for data that is not in "tidy" format (i.e. your data is more like the inflammation data and less like the gapminder data from day 1)

2024-1-12 : Data for today: https://swcarpentry.github.io/python-novice-gapminder/files/python-novice-gapminder-data.zip

Best practices in scientific computing: https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001745
Good enough practices in scientific computing: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005510
Plotting Curriculum 1: https://swcarpentry.github.io/python-novice-gapminder/09-plotting.html
Plotting Curriculum 2: https://umcarpentries.org/intro-curriculum-python/01-python-plotting/index.html

data to download: https://swcarpentry.github.io/python-novice-gapminder/files/python-novice-gapminder-data.zip

Matplotlib cheatsheets: https://matplotlib.org/cheatsheets/
Color palettes in seaborn: https://seaborn.pydata.org/tutorial/color_palettes.html

Map code!!!

```
(
    gapminder_1997
    # if you use index_col, you must reset_index before .replace
    .reset_index()
    .replace({'country' : {'United States' : 'United States of America',
                           'United Kingdom' : 'United Kingdom of Great Britain and Northern Ireland',
                          }})
    .merge(pd.read_csv("un-report/un-report/country-iso.csv")
           .rename(columns={'name' : 'country'}),
           on='country', how='inner')
    .pipe(px.choropleth,
          locations='alpha-3',
          color='lifeExp',
          hover_name='country',
          hover_data=['lifeExp', 'pop'])
)
```

☆**Please give us feedback for day 2!**

https://forms.gle/2rhXredWf84zCMFK6

Link to data: https://github.com/UMCarpentries/intro-curriculum-python/raw/gh-pages/files/un-report.zip
Pandas functions: https://pandas.pydata.org/docs/reference/general_functions.html
Pandas DataFrame functions: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html
Data cleaning Google Slides: https://docs.google.com/presentation/d/19IDqslbhXAzz13oWpImxUYzE2Gyr5ckMQcO1md3mIvs/edit?usp=sharing
Data slicing with Pandas: https://swcarpentry.github.io/python-novice-gapminder/instructor/08-data-frames.html
********U of Michigan data cleaning curriculum: https://umcarpentries.org/intro-curriculum-python/04-python-data-analysis/index.html *******

List of functions that work with .agg

Stack overflow link: python - What are all Pandas .agg functions? - Stack Overflow python - https://stackoverflow.com/questions/53943319/what-are-all-pandas-agg-functions

Straight to the documentation: GroupBy — pandas 2.1.4 documentation (pydata.org)
https://pandas.pydata.org/pandas-docs/stable/reference/groupby.html#computations-descriptive-stats

August Workshop Links

Workshop website: Cal Poly Humboldt University: Aug 14-15, 2023 (laporpe.github.io)
https://laporpe.github.io/2023-08-14-humboldt/
This site includes hyperlinks to the curriculum in the "schedule" section

Plotting curricula for 1/12/2024

General python plotting lesson: Plotting and Programming in Python: Plotting (swcarpentry.github.io)
https://swcarpentry.github.io/python-novice-gapminder/instructor/09-plotting.html

U of Michigan plotting curriculum (seaborn package): Python for Plotting – U-M Carpentries Curriculum (umcarpentries.org) https://umcarpentries.org/intro-curriculum-python/01-python-plotting/index.html

Yoon G Kim -- Math + Statistics
Yes, excited about Day 2 (after great Day 1).

Tamara Beitzel Barriquand - physical oceanography - cereal
Yay! Data!

Kami-- math bio, I like smoothies. I am excited to see some cool plots! I am also excited to see our students again!
Bri Hagen - Library - Breakfast Burritos 4 Life
Mary-Francis -- plant genetics -- scone -- seminars to begin!

Tyler - mathematics
Abigail Penland - comp sci and math - cereal!
Young-Exercise Science-programming!
Beth-water resources engineering
Carolyn -- ecology/land use change -- attending a class on course design!

Carolyn's European min/max plot code:

- # data is from https://swcarpentry.github.io/python-novice-gapminder/files/python-novice-gapminder-data.zip
- data = pd.read_csv('data/gapminder_all.csv', index_col='country')
- # filter data
- eu = (

```
    data
    .query("continent == 'Europe'")
    .filter(like="gdpPercap")
)
years2 = eu.columns.str.replace('gdpPercap_','')
eu.columns = years2.astype(int)
plt.style.use('seaborn-v0_8-colorblind')
# make plot
eu.max().plot(label='max', marker="")
eu.min().plot(label='min', marker="o")
plt.xticks(np.arange(1952, 2008, step=5))
plt.xlabel('Year')
plt.ylabel('GDP per capita')
plt.title('European minimum and maximum GDP per capita')
plt.legend(loc='best')
for x, y, text in zip(eu.columns, eu.min(), eu.idxmin()):
    plt.text(x, y, text, rotation=45, fontsize=8)
for x, y, text in zip(eu.columns, eu.max(), eu.idxmax()):
    plt.text(x, y, text, rotation=60, fontsize=8)
```

- The above method uses a "for loop." For more information on for loops, see: https://swcarpentry.github.io/python-novice-gapminder/12-for-loops.html
- Reference for creating marker labels: https://stackoverflow.com/questions/56147039/how-to-put-labels-on-plot-markers

Convert the inflammation data to long/"tidy" format (the way seaborn likes it!)

```
inflam2 = inflam.stack().reset_index()
inflam2.columns = ["Patient", "Day", "Inflam"]

(
   so.Plot(inflam2,
           x='Day',
           y='Inflam')
   .add(so.Range(), so.Est(func='mean', errorbar='sd'))
   .add(so.Dot(), so.Agg())
)
```

plot with min, mean, max:

```
di1.min().T.plot(label = 'min')
di1.max().T.plot(label = 'max')
di1.mean().T.plot(label = 'mean')

plt.legend(loc = 'best')
plt.xlabel('Time (days)')
plt.ylabel('Inflammation Level (units TBD)')
```