﻿import matplotlib

import numpy as np

import matplotlib.pyplot as plt

plt.subplots(figsize=(2,2))

Out[4]:

(<Figure size 144x144 with 1 Axes>,

<matplotlib.axes.\_subplots.AxesSubplot at 0x117360518>)



﻿t = np.linspace(-np.pi, np.pi, 64, endpoint=True)

t

Out[5]:

array([-3.14159265, -3.04185955, -2.94212645, -2.84239335, -2.74266025,

-2.64292715, -2.54319405, -2.44346095, -2.34372785, -2.24399475,

-2.14426165, -2.04452855, -1.94479545, -1.84506235, -1.74532925,

-1.64559615, -1.54586305, -1.44612995, -1.34639685, -1.24666375,

-1.14693065, -1.04719755, -0.94746445, -0.84773135, -0.74799825,

-0.64826515, -0.54853205, -0.44879895, -0.34906585, -0.24933275,

-0.14959965, -0.04986655, 0.04986655, 0.14959965, 0.24933275,

0.34906585, 0.44879895, 0.54853205, 0.64826515, 0.74799825,

0.84773135, 0.94746445, 1.04719755, 1.14693065, 1.24666375,

1.34639685, 1.44612995, 1.54586305, 1.64559615, 1.74532925,

1.84506235, 1.94479545, 2.04452855, 2.14426165, 2.24399475,

2.34372785, 2.44346095, 2.54319405, 2.64292715, 2.74266025,

2.84239335, 2.94212645, 3.04185955, 3.14159265])

c= np.cos(t)

s = np.sin(t)

fig, ax = plt.subplots(figsize=(3,3))

ax.plot(t, s)

ax.plot(t, c)

Out[7]: [<matplotlib.lines.Line2D at 0x11762bba8>]



﻿fig, ax = plt.subplots(figsize=(6,3), ncols=2)

ax.plot(t, s)

#BAD, instead do:

Traceback (most recent call last):

File "<ipython-input-8-a09d17426ef9>", line 2, in <module>

ax.plot(t, s)

AttributeError: 'numpy.ndarray' object has no attribute 'plot'



﻿fig, ax = plt.subplots(figsize=(6,3), ncols=2)

ax[0].plot(t, s)

ax[1].plot(t, c)

Out[9]: [<matplotlib.lines.Line2D at 0x117833748>]



﻿fig, ax = plt.subplots(figsize=(4,4))

ax.plot(t, c, label='cosin', linewidth=1, linestyle='-', color='#A9A9A9', marker='o', markersize=2)

ax.plot(t, s, label='sin', linewidth=1, linestyle='-', color='red', marker='+', markersize=2)

plt.legend()

Out[10]: <matplotlib.legend.Legend at 0x1176f56d8>



﻿fig, ax = plt.subplots(figsize=(4,4))

ax.plot(t, c)

ax.plot(t, s)

ax.set\_title('Sins and Cosins')

ax.set\_xlabel('Time')

ax.set\_ylabel('Magnitude')

ax.set\_ylim([-2,2])

ax.set\_xlim([-10,10])

Out[11]: (-10, 10)



﻿﻿plt.style.available

Out[12]:

['\_classic\_test',

'bmh',

'classic',

'dark\_background',

'fast',

'fivethirtyeight',

'ggplot',

'grayscale',

'seaborn-bright',

'seaborn-colorblind',

'seaborn-dark-palette',

'seaborn-dark',

'seaborn-darkgrid',

'seaborn-deep',

'seaborn-muted',

'seaborn-notebook',

'seaborn-paper',

'seaborn-pastel',

'seaborn-poster',

'seaborn-talk',

'seaborn-ticks',

'seaborn-white',

'seaborn-whitegrid',

'seaborn',

'Solarize\_Light2',

'tableau-colorblind10']

﻿﻿plt.style.use('fivethirtyeight')

fig, ax = plt.subplots(figsize=(4,4))

ax.plot(t, s)

ax.set(title="Sine", xlabel = 'Time', ylabel='Magnitude')

Out[13]: [Text(0, 0.5, 'Magnitude'), Text(0.5, 0, 'Time'), Text(0.5, 1.0, 'Sine')]



﻿import xarray as xr

infile = open("/Users/roryeggleston/Downloads/CESM.003.SST.1980.nc")

data = xr.open\_dataset('/Users/roryeggleston/Downloads/CESM.003.SST.1980.nc')

lat = np.array(data.lat)

lon = np.array(data.lon)

data = np.array(data.SST)

data.shape

meandata = np.mean(data, axis=0)

anom = data - meandata

fig, ax = plt.subplots()

p = ax.pcolormesh(lon, lat, meandata)

cb = plt.colorbar(p)

cb.set\_label('Sea Surface Temperature [$^{o}$C]')

ax.set(title='CESM Sea Surface Temperature (1970-1980)', ylabel='Latitude', xlabel = 'Longitude')

Out[18]:

[Text(0, 0.5, 'Latitude'),

Text(0.5, 0, 'Longitude'),

Text(0.5, 1.0, 'CESM Sea Surface Temperature (1970-1980)')]



﻿﻿fig, ax = plt.subplots()

p = ax.pcolormesh(lon, lat, meandata, cmap='autumn\_r', vmin=-2, vmax=30)

cb = plt.colorbar(p, extend='both')

cb.set\_label('Sea Surface Temperature [$^{o}$C]')

ax.set(title='CESM Sea Surface Temperature (1970-1980)', ylabel='Latitude', xlabel = 'Longitude')

Out[19]:

[Text(0, 0.5, 'Latitude'),

Text(0.5, 0, 'Longitude'),

Text(0.5, 1.0, 'CESM Sea Surface Temperature (1970-1980)')]



﻿fig, ax = plt.subplots()

p = ax.pcolormesh(lon, lat, anom[10], cmap='RdBu\_r')

cb = plt.colorbar(p, extend='both')

cb.set\_label('Sea Surface Temperature Anomaly [$^{o}$C]')

ax.set(title='CESM Sea Surface Temperature (1970-1980)', ylabel='Latitude', xlabel = 'Longitude')

Out[20]:

[Text(0, 0.5, 'Latitude'),

Text(0.5, 0, 'Longitude'),

Text(0.5, 1.0, 'CESM Sea Surface Temperature (1970-1980)')]



﻿import cartopy.crs as ccrs

import cartopy.feature as cfeature

from cartopy.util import add\_cyclic\_point

fig, ax = plt.subplots(subplot\_kw=dict(projection=ccrs.Robinson()))

ax.add\_feature(cfeature.LAND, color='k')

ax.pcolormesh(lon, lat, meandata, transform=ccrs.PlateCarree())

Out[22]: <matplotlib.collections.QuadMesh at 0xd1b618518>



﻿cyclic\_data, cyclic\_lons = add\_cyclic\_point(meandata, coord=lon)

fig, ax = plt.subplots(subplot\_kw=dict(projection=ccrs.Robinson()))

ax.add\_feature(cfeature.LAND, color='k')

p = ax.pcolormesh(cyclic\_lons, lat, cyclic\_data, transform=ccrs.PlateCarree(),

cmap="plasma")

plt.colorbar(p, orientation='horizontal', pad=0.05, fraction=0.05,

extend="both")

Out[24]: <matplotlib.colorbar.Colorbar at 0xd1c5ae390>



﻿fig, ax = plt.subplots(subplot\_kw=dict(projection=ccrs.Geostationary()))

p = ax.pcolormesh(cyclic\_lons, lat, cyclic\_data, transform=ccrs.PlateCarree(),

cmap="plasma")

ax.add\_feature(cfeature.LAND, color='k', zorder=4)

plt.colorbar(p, orientation='horizontal', pad=0.05, fraction=0.05,

extend="both")

Out[25]: <matplotlib.colorbar.Colorbar at 0xd1c440c88>



﻿

fig, ax = plt.subplots(subplot\_kw=dict(projection=ccrs.Geostationary()))

p = ax.pcolormesh(cyclic\_lons, lat, cyclic\_data, transform=ccrs.PlateCarree(),

cmap="plasma")

ax.add\_feature(cfeature.LAND, color='k', zorder=4)

plt.colorbar(p, orientation='horizontal', pad=0.05, fraction=0.05,

extend="both")

gl = ax.gridlines(crs=ccrs.PlateCarree(), draw\_labels=False,

linewidth=1.5, color='gray', linestyle='-')



﻿from cartopy.mpl.gridliner import LONGITUDE\_FORMATTER, LATITUDE\_FORMATTER

fig, ax = plt.subplots(subplot\_kw=dict(projection=ccrs.PlateCarree()))

p = ax.pcolormesh(cyclic\_lons, lat, cyclic\_data, transform=ccrs.PlateCarree(),

cmap="plasma")

ax.add\_feature(cfeature.LAND, color='k', zorder=4)

plt.colorbar(p, orientation='horizontal', pad=0.05, fraction=0.05,

extend="both")

gl = ax.gridlines(crs=ccrs.PlateCarree(), draw\_labels=True,

linewidth=1.5, color='gray', linestyle='--')

gl.xlabels\_bottom = False

gl.ylabels\_left = False

gl.xformatter = LONGITUDE\_FORMATTER

gl.yformatter = LATITUDE\_FORMATTER

