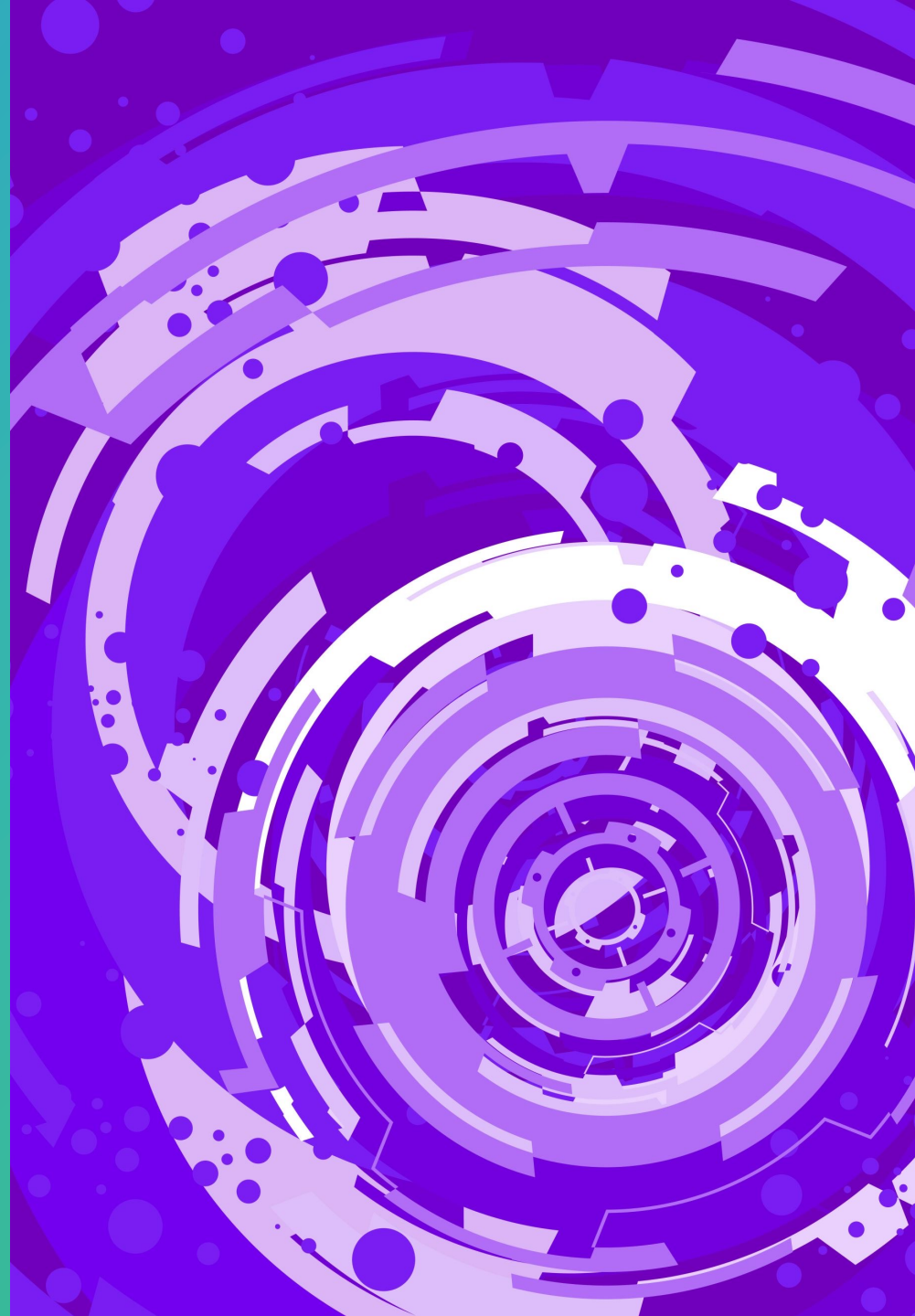
URBAN GEOGRAPHIC INFORMATION SYSTEM

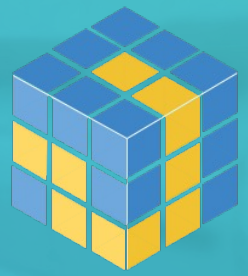
Python Visualization

Chun-Hsiang Chan
Department of Geography,
National Taiwan Normal University



Outline

- Matplotlib
- Seaborn
- Plotly
- Contextily



NumPy

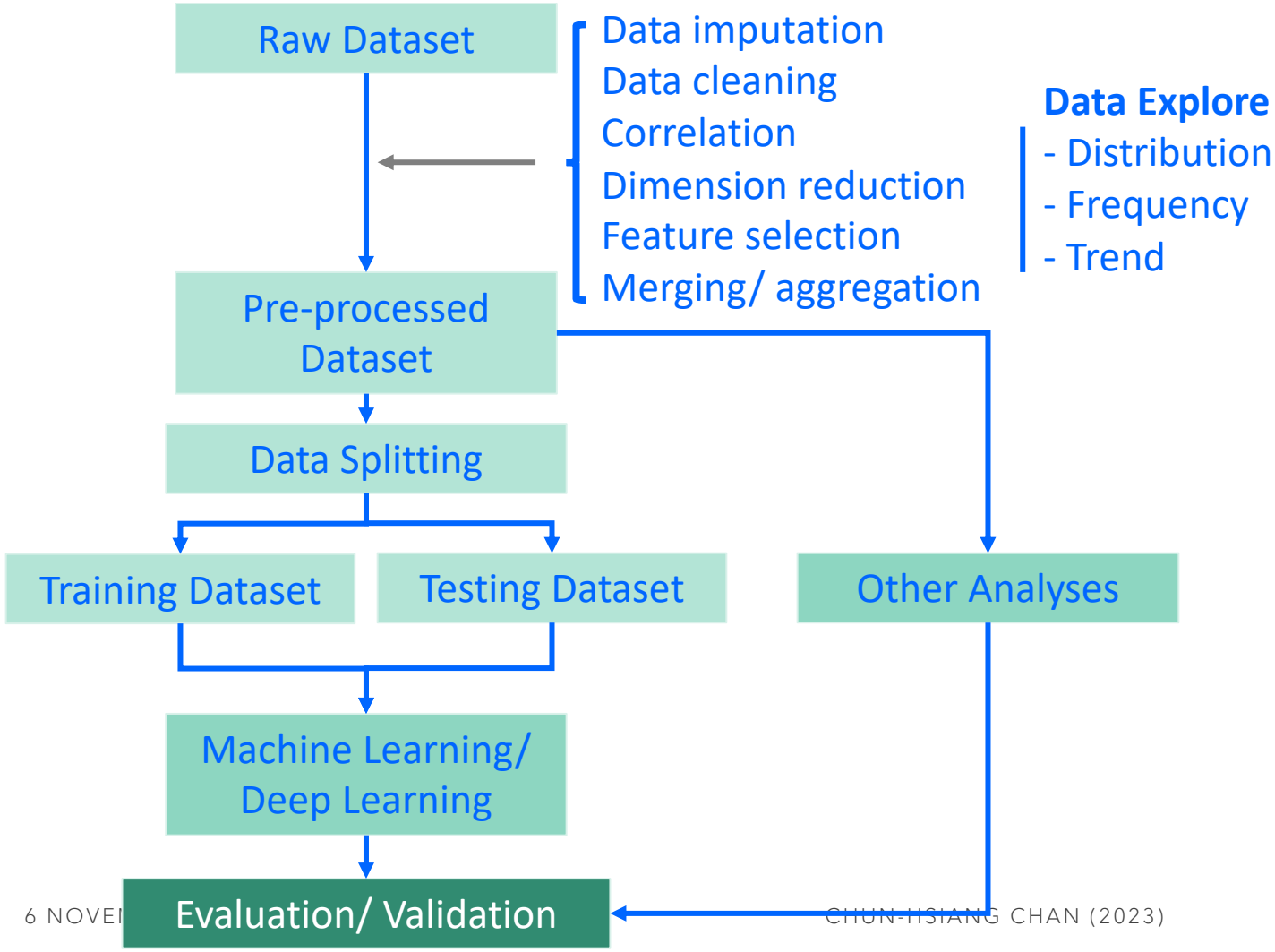
Pandas

Visualization – Methods

- Several visualization functions were developed in built-in package, such as line, scatter, boxplot, and histogram.

Distribution			
Continuous Data	**Discrete Data**	**Ordered/ Categorical Data**	**Proportional Data**
Scatter Bubble Histogram Violin Plot Box Plot Heatmap Density Map	Scatter Bubble Histogram Violin Plot Box Plot Heatmap Density Map	Group/ Stacked Bar Pie	Group/ Stacked Bar Pie

Data Science Mindset



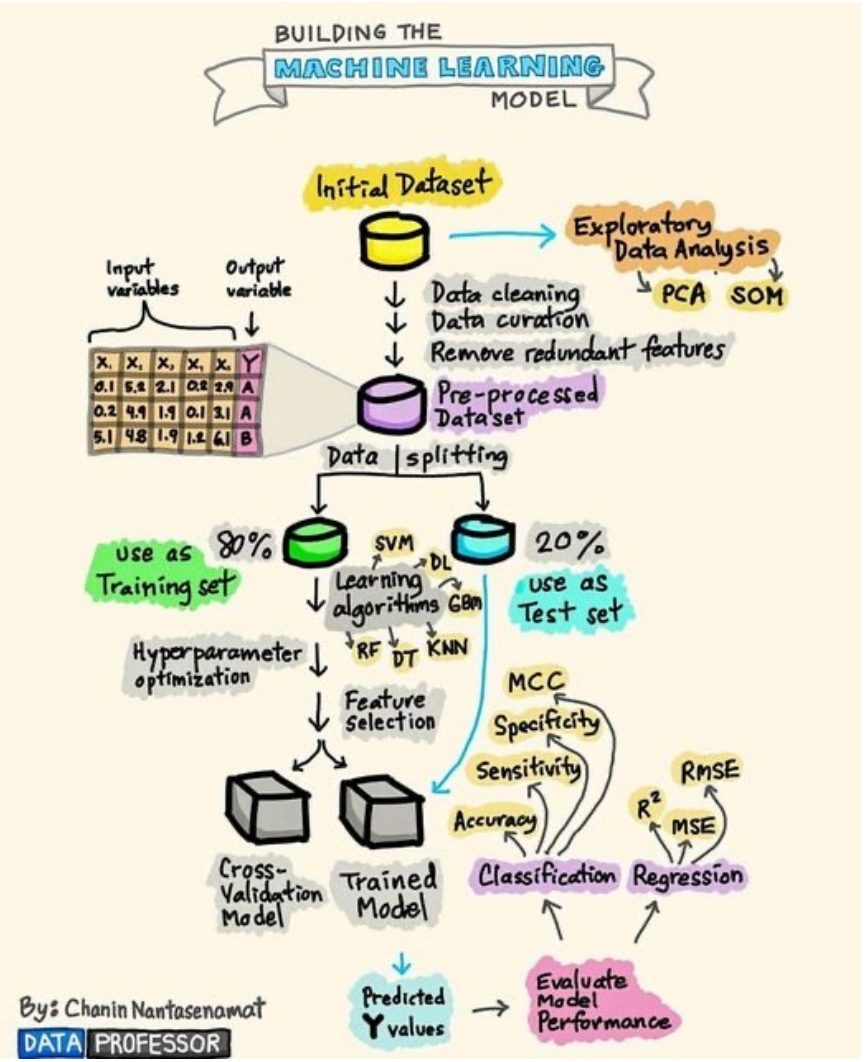
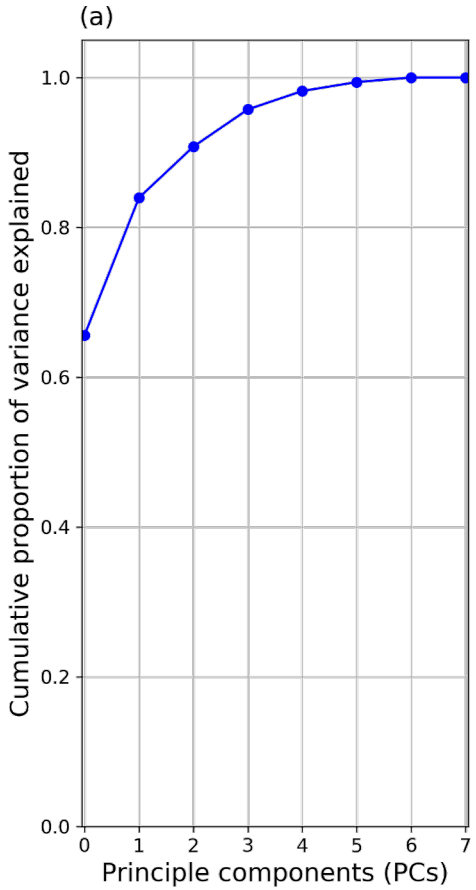
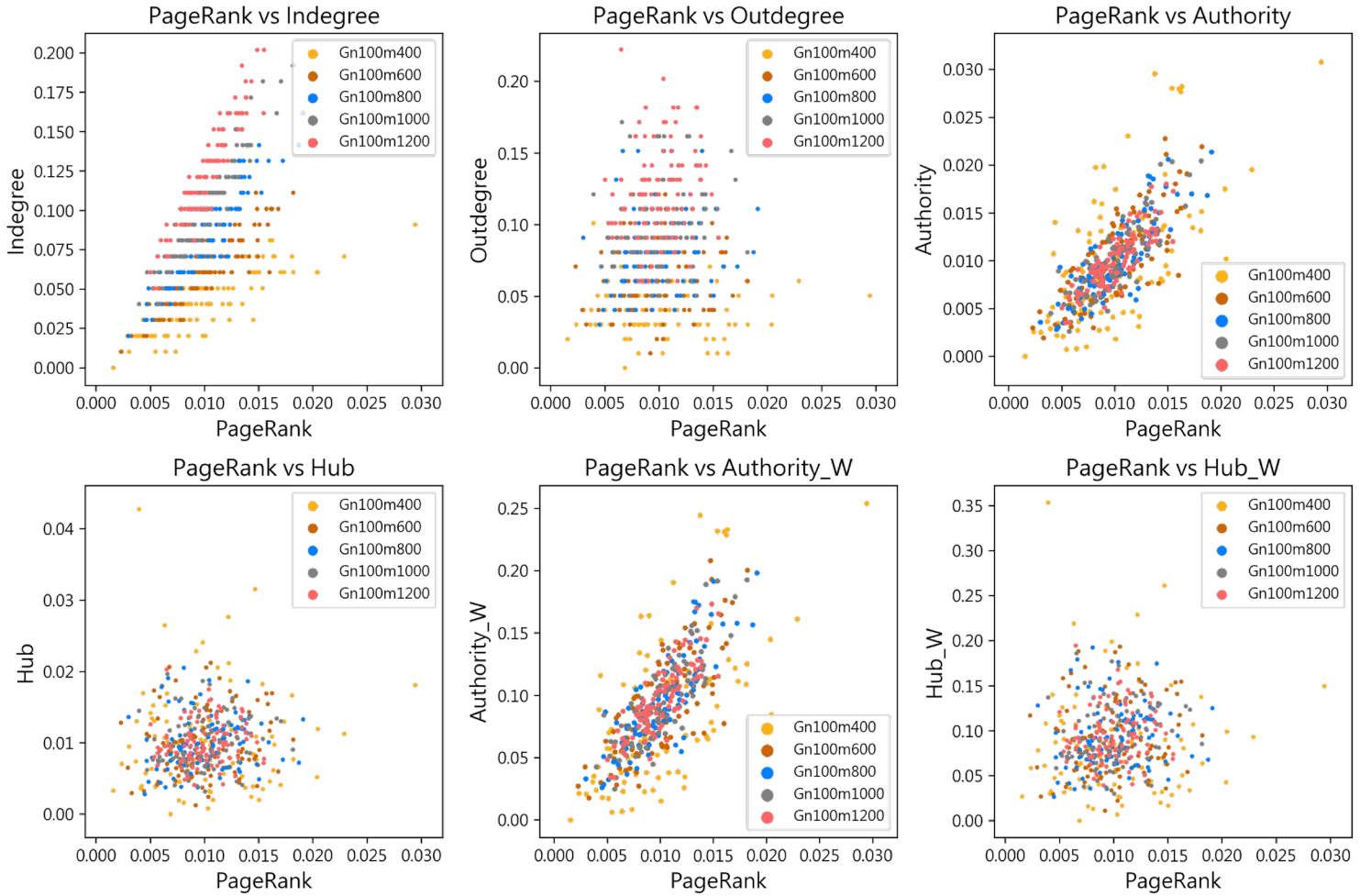


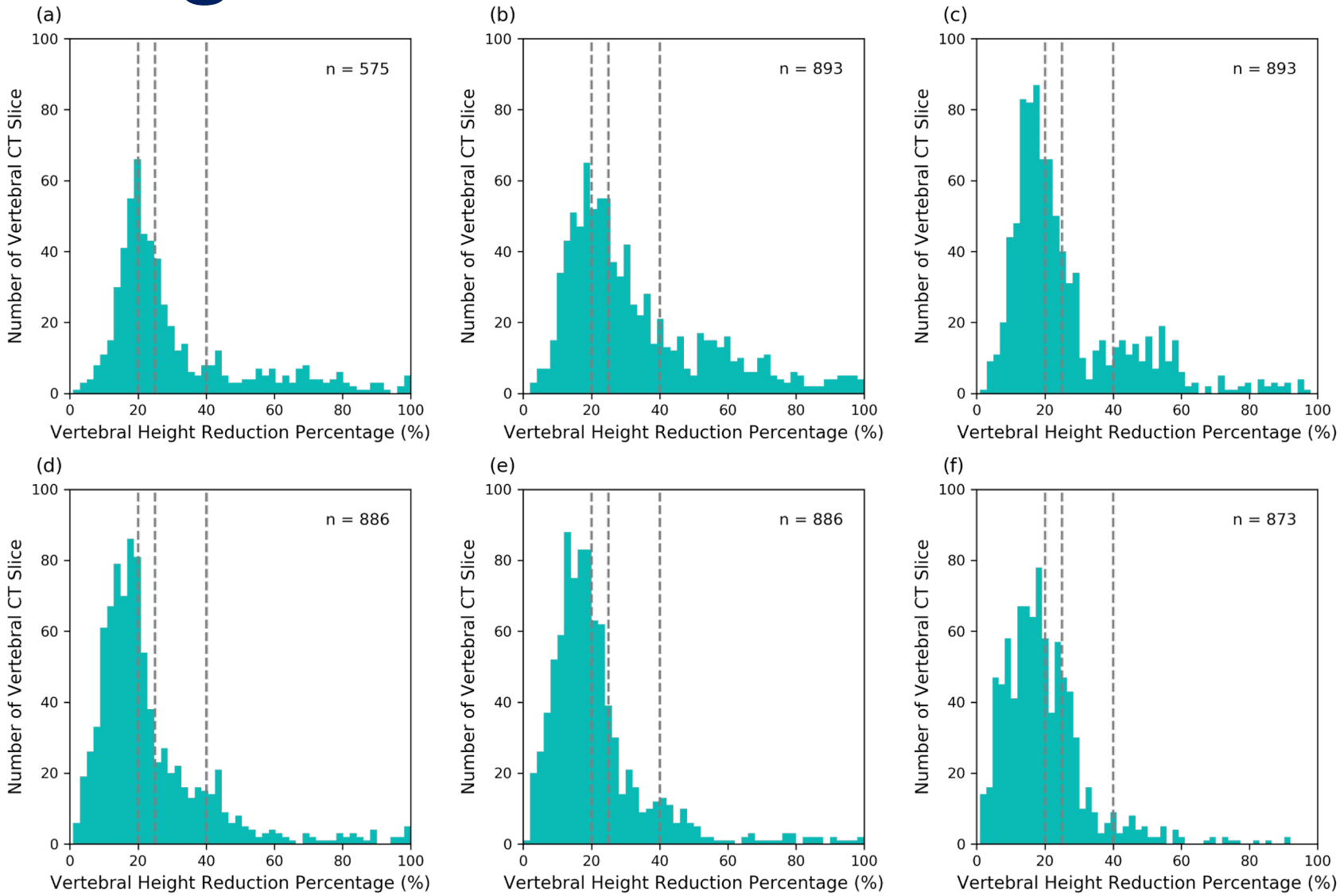
Figure source: https://bit.ly/3BJd2d4

Line Plot & Scatter Plot

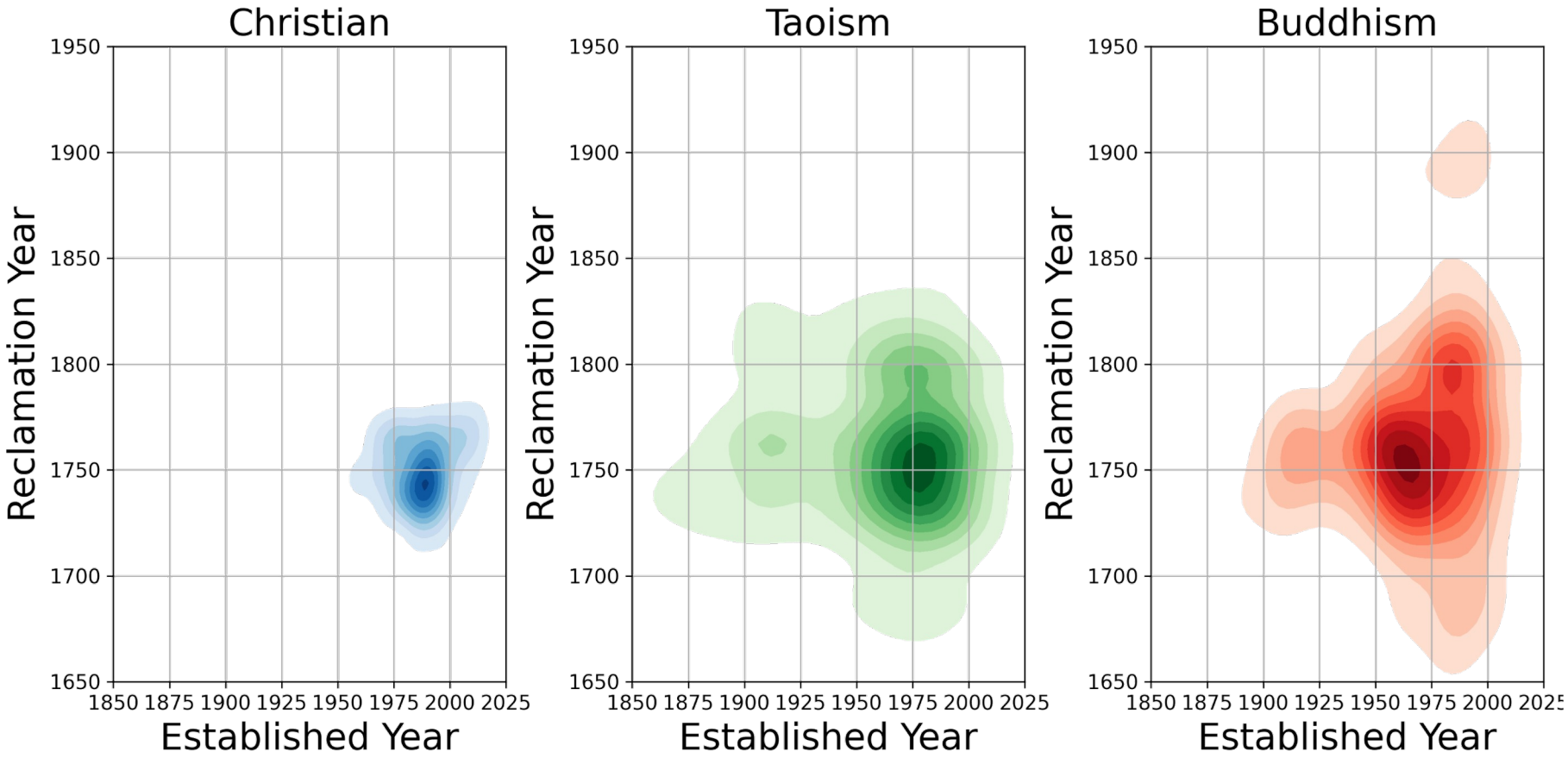




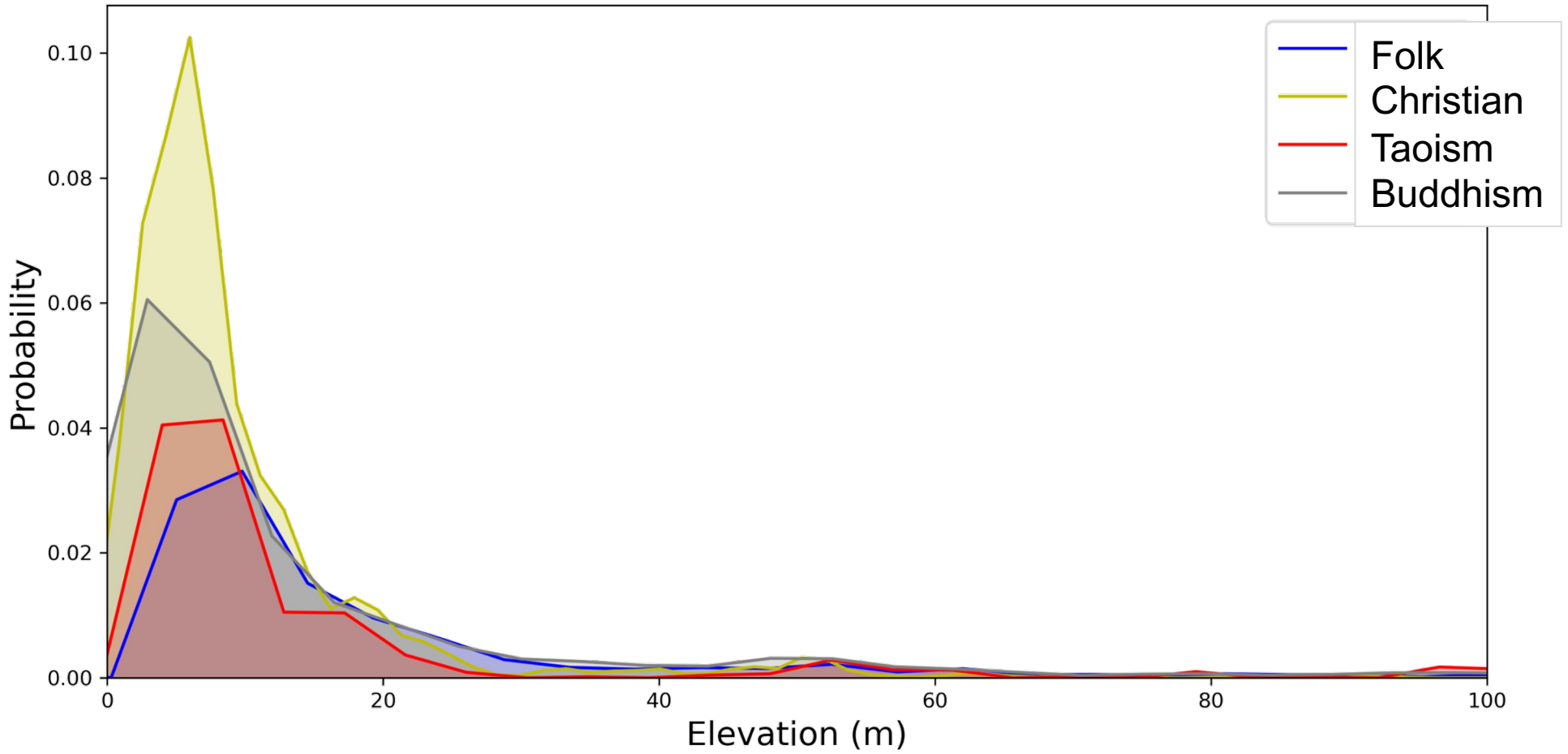
1D Histogram



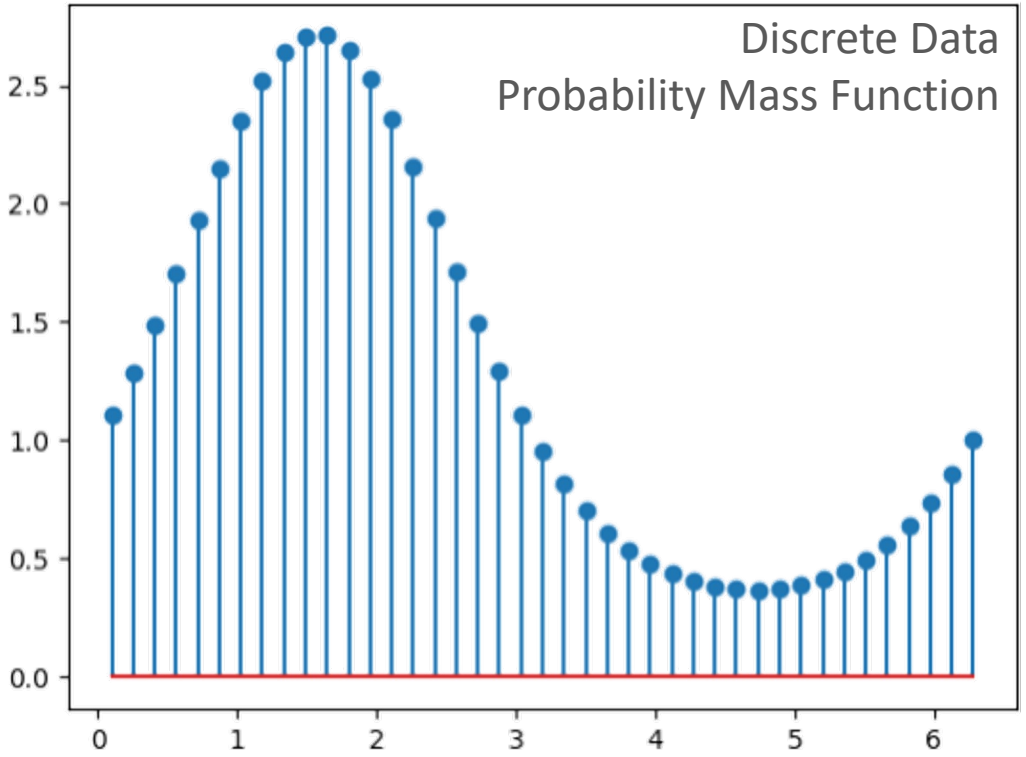
2D Histogram



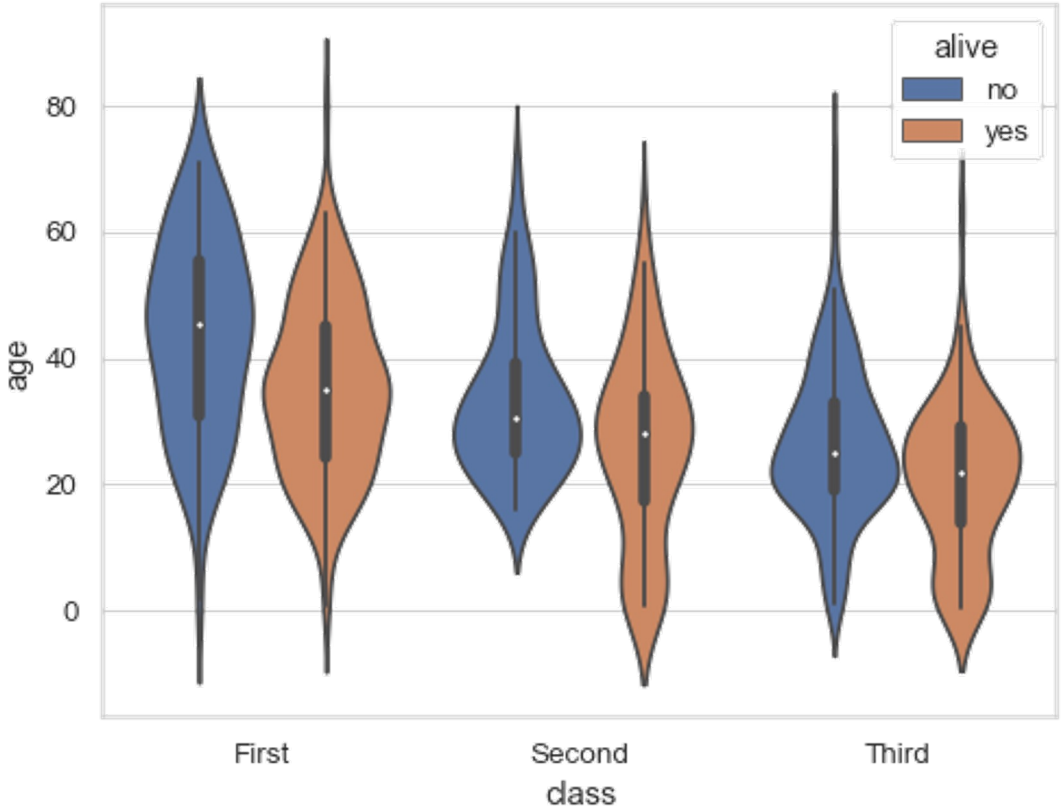
Area Plot


Folk
Christian
Taoism
Buddhism
Probability
0.00
0.02
0.04
0.06
0.08
0.10
0
20
40
60
80
100
Elevation (m)

Stem Plot & Violin Plot



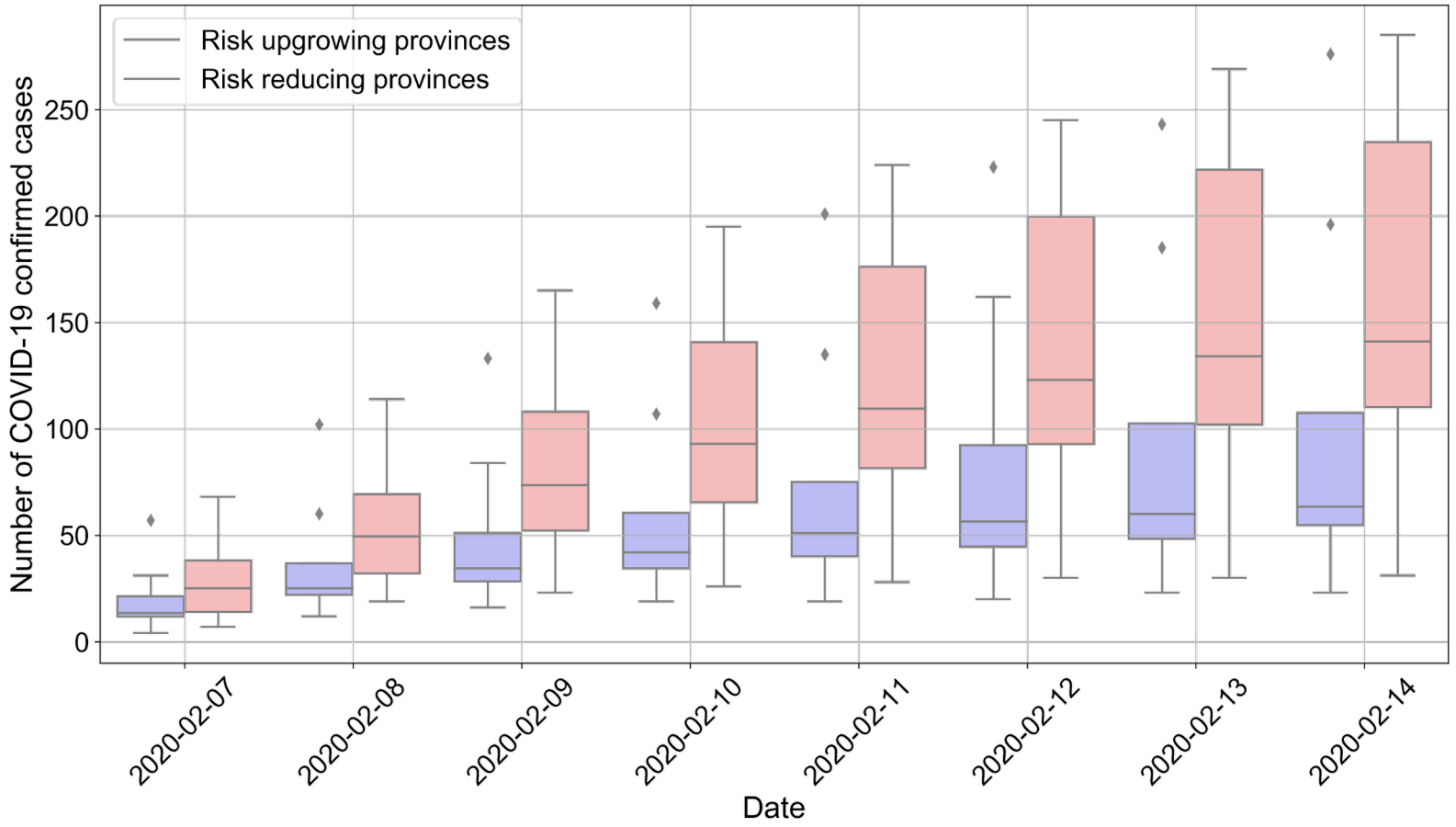
Continuous Data

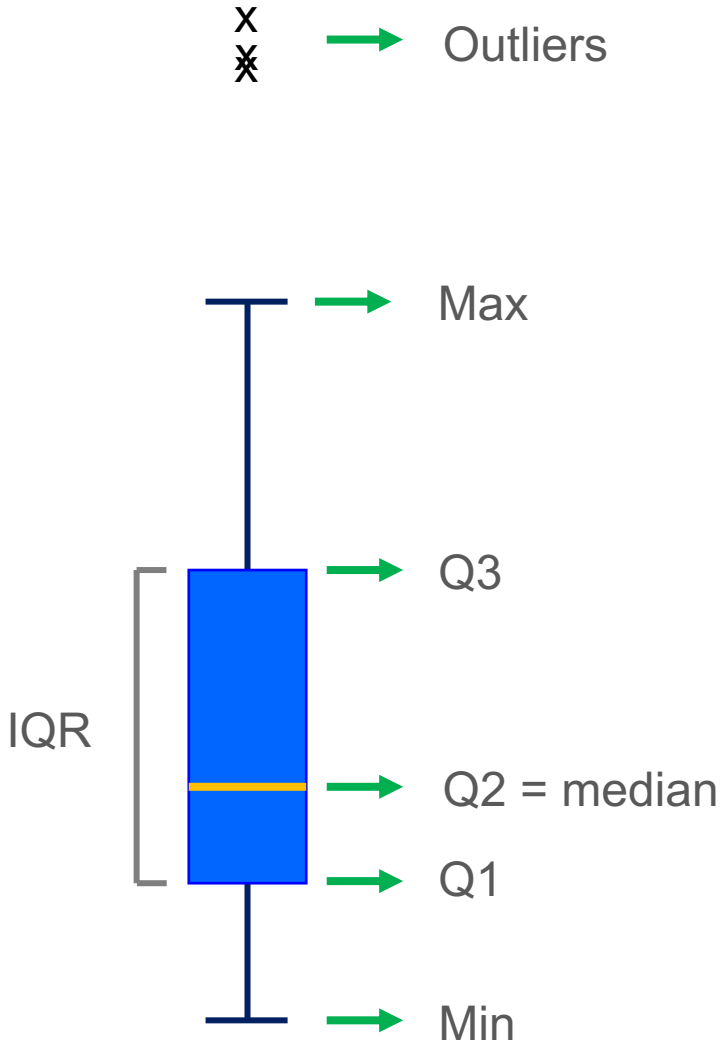


Source: https://matplotlib.org/stable/gallery/lines_bars_and_markers/stem_plot.html#sphx-glr-gallery-lines-bars-and-markers-stem-plot-py

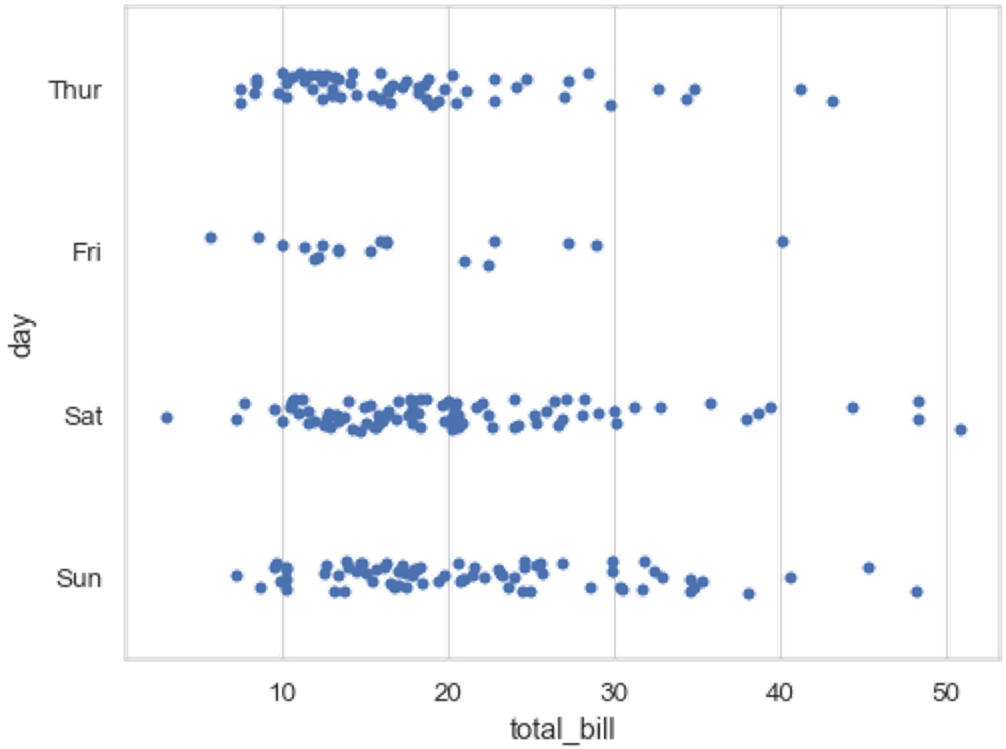
Source: https://seaborn.pydata.org/generated/seaborn.violinplot.html

Box Plot

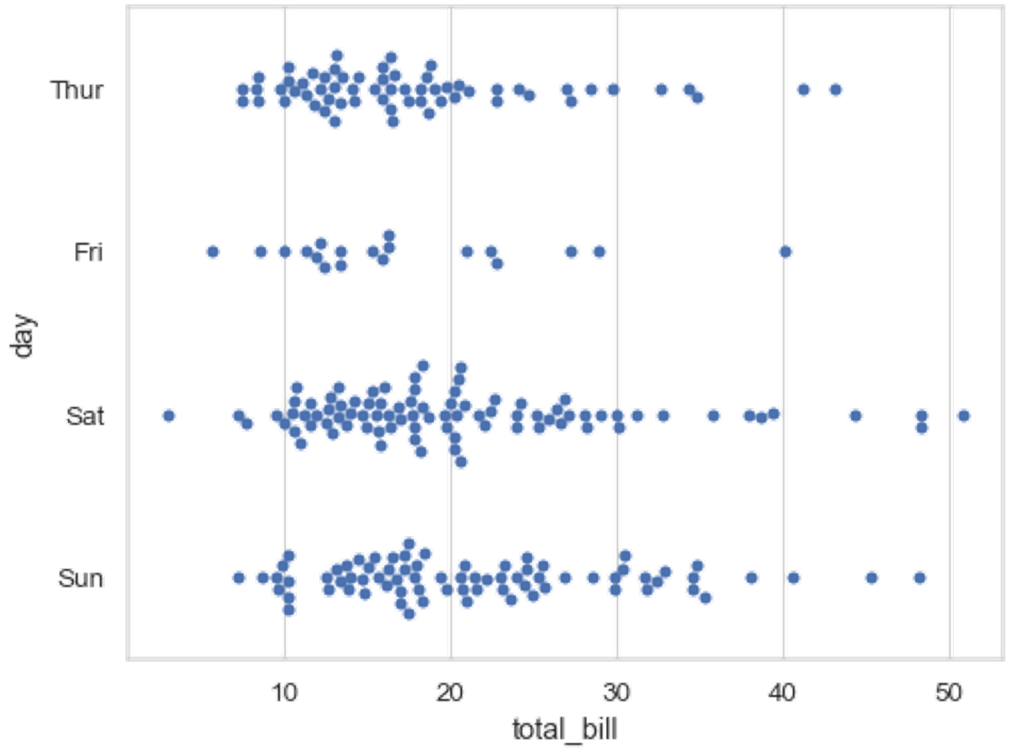

Risk upgrowing provinces
Risk reducing provinces
Number of COVID-19 confirmed cases
0
50
100
150
200
250
2020-02-07
2020-02-08
2020-02-09
2020-02-10
2020-02-11
2020-02-12
2020-02-13
2020-02-14
Date


Outliers
Max
Q3
IQR
Q2 = median
Q1
Min

Strip Plot and Swarm Plot

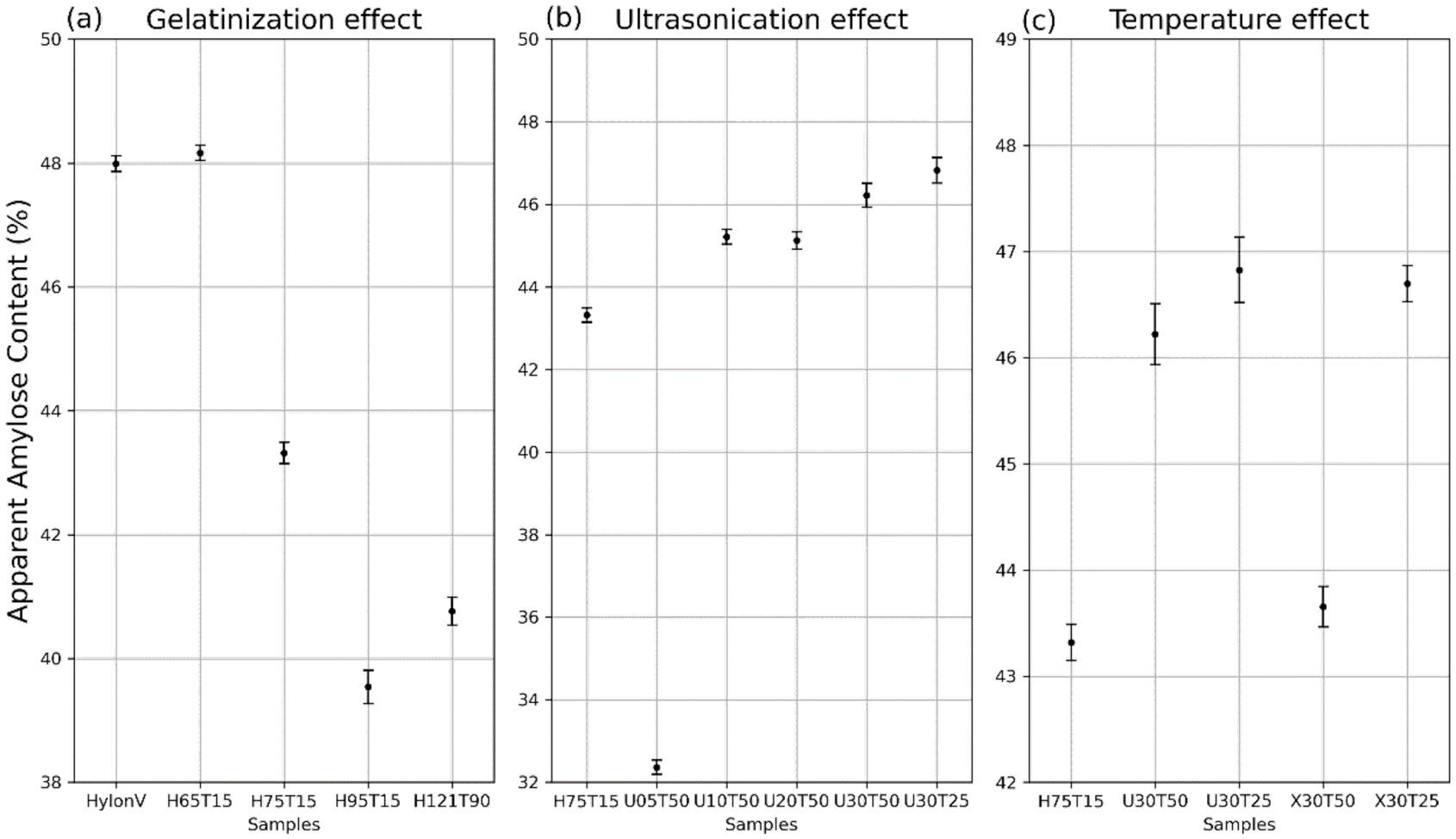


Source: https://seaborn.pydata.org/generated/seaborn.stripplot.html

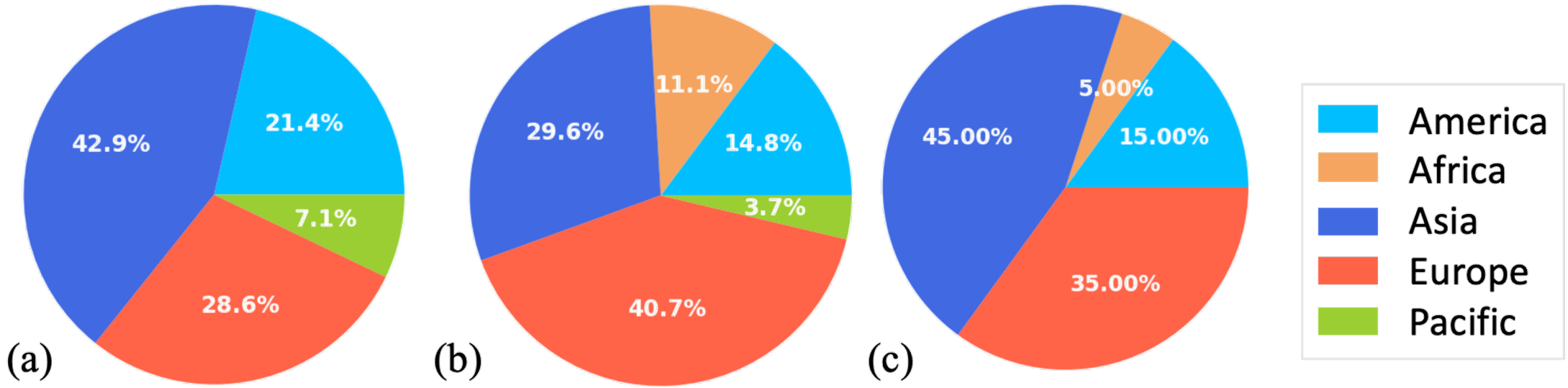


Source: https://seaborn.pydata.org/generated/seaborn.swarmplot.html

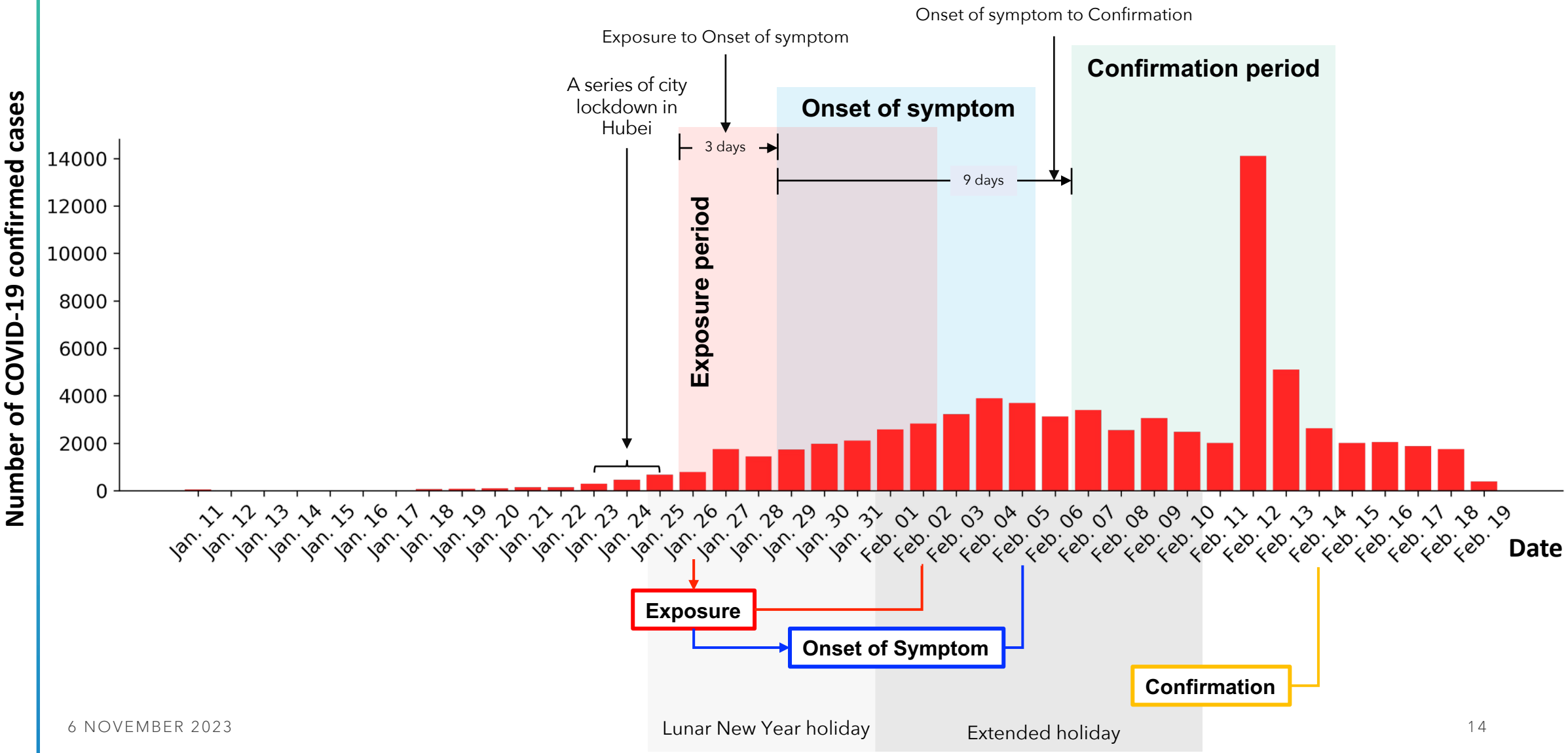
Error Bar


(a) Gelatinization effect
Apparent Amylose Content (%)
50
48
46
44
42
40
38
HylonV
H65T15
H75T15
H95T15
H121T90
Samples
(b) Ultrasonication effect
50
48
46
44
42
40
38
36
34
32
H75T15
U05T50
U10T50
U20T50
U30T50
U30T25
Samples
(c) Temperature effect
49
48
47
46
45
44
43
42
H75T15
U30T50
U30T25
X30T50
X30T25
Samples

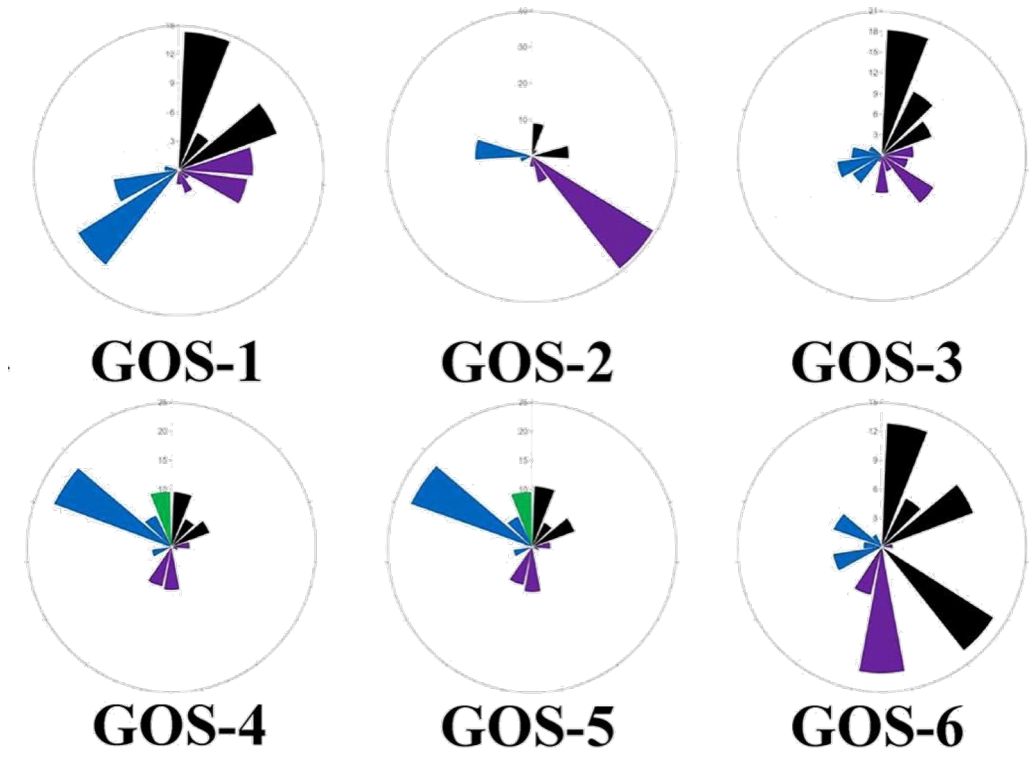
Pie Chart


(a)
21.4%
42.9%
7.1%
28.6%
(b)
11.1%
29.6%
14.8%
3.7%
40.7%
(c)
5.00%
45.00%
15.00%
35.00%
America
Africa
Asia
Europe
Pacific

Bar Chart

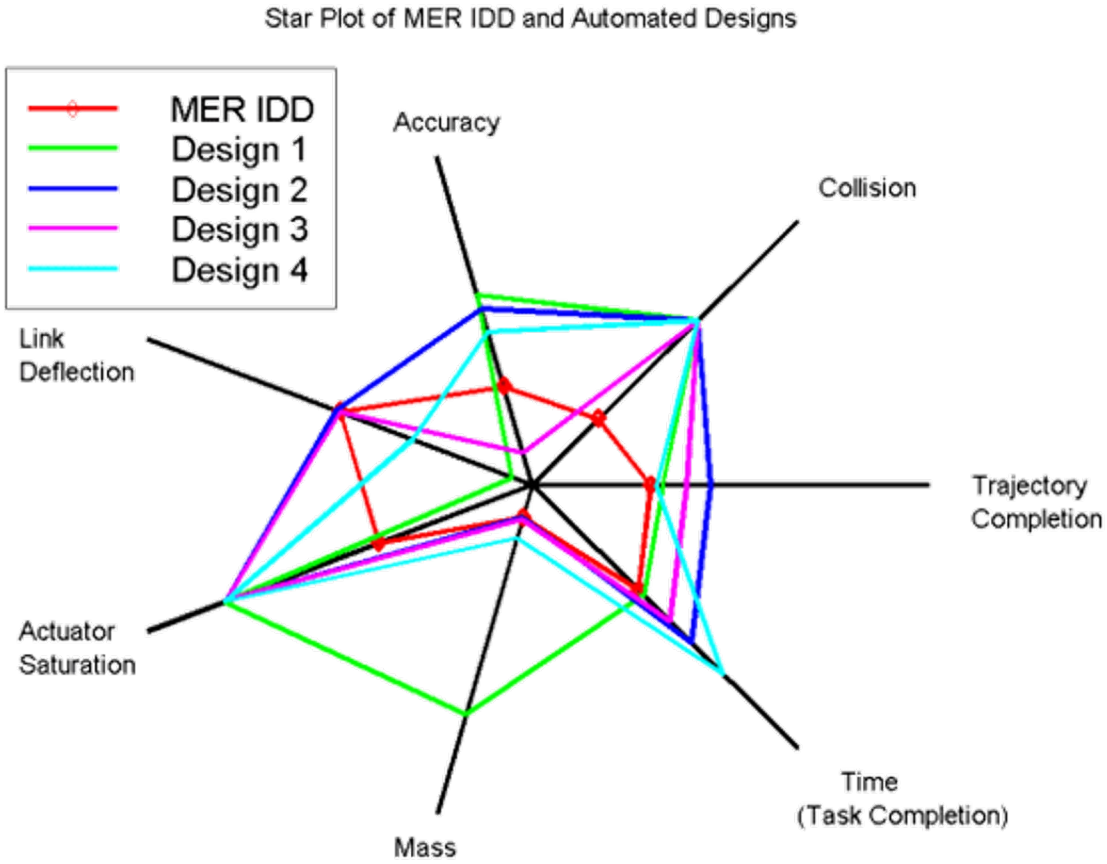

Onset of symptom to Confirmation
Exposure to Onset of symptom
A series of city lockdown in Hubei
Confirmation period
Onset of symptom
3 days
9 days
Exposure period
Number of COVID-19 confirmed cases
14000
12000
10000
8000
6000
4000
2000
0
Jan. 11
Jan. 12
Jan. 13
Jan. 14
Jan. 15
Jan. 16
Jan. 17
Jan. 18
Jan. 19
Jan. 20
Jan. 21
Jan. 22
Jan. 23
Jan. 24
Jan. 25
Jan. 26
Jan. 27
Jan. 28
Jan. 29
Jan. 30
Jan. 31
Feb. 01
Feb. 02
Feb. 03
Feb. 04
Feb. 05
Feb. 06
Feb. 07
Feb. 08
Feb. 09
Feb. 10
Feb. 11
Feb. 12
Feb. 13
Feb. 14
Feb. 15
Feb. 16
Feb. 17
Feb. 18
Feb. 19
Date
Exposure
Onset of Symptom
Confirmation
Lunar New Year holiday
Extended holiday

Rose Plot & Radar Plot



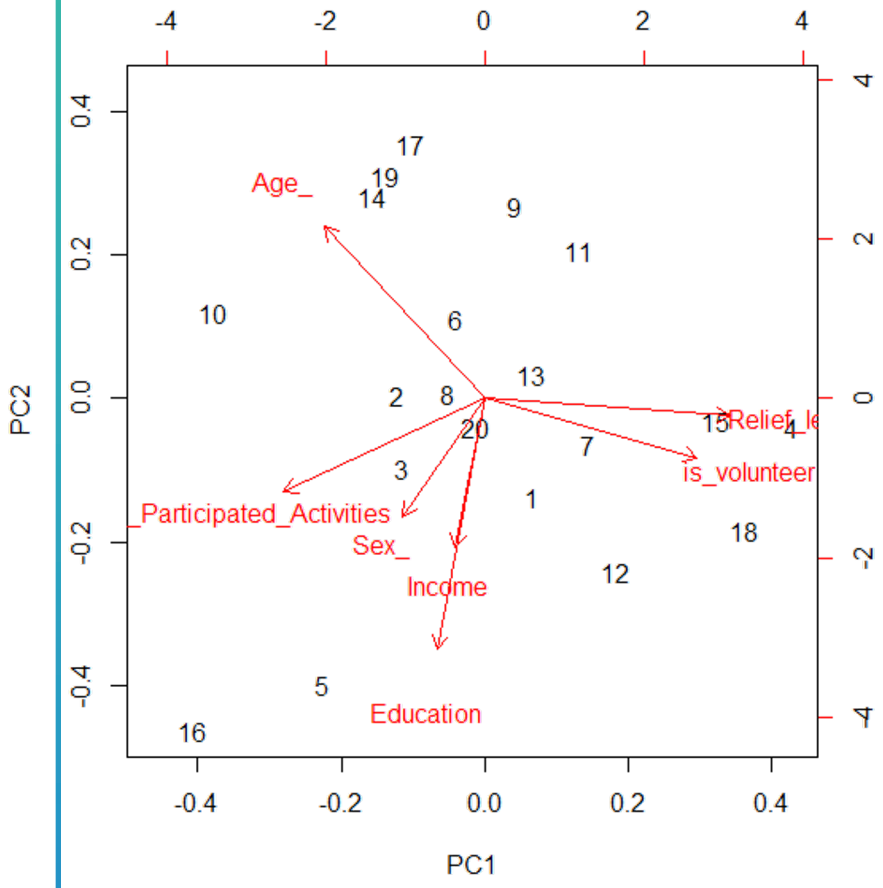
Profile diversity in various GOS samples

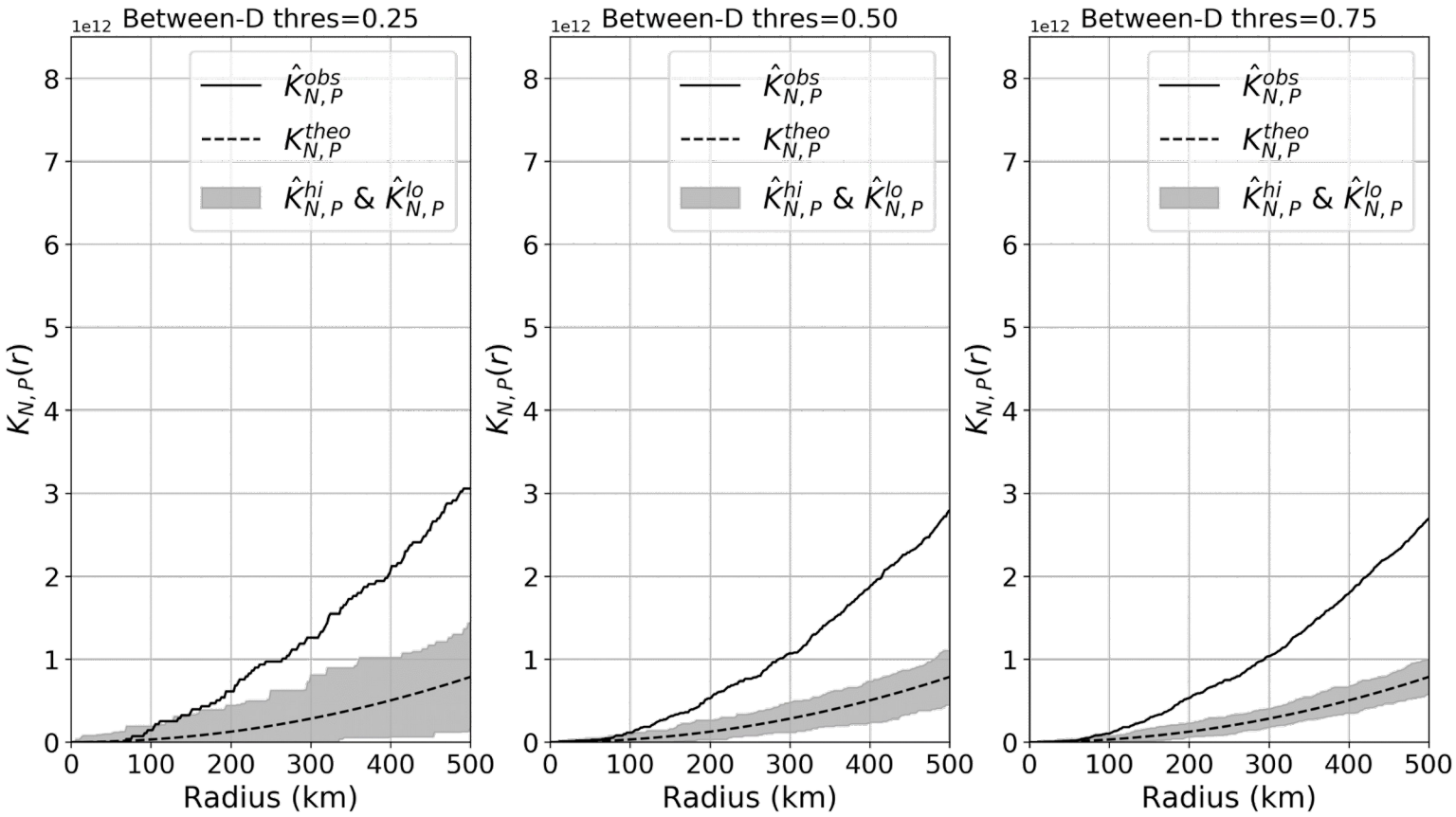
Lin et al. (2022) Profile diversity of galacto-oligosaccharides from disaccharides to hexasaccharides by porous graphitic carbon liquid chromatography-orbitrap tandem mass spectrometry. Food Chem. Vol. 390. 133151.



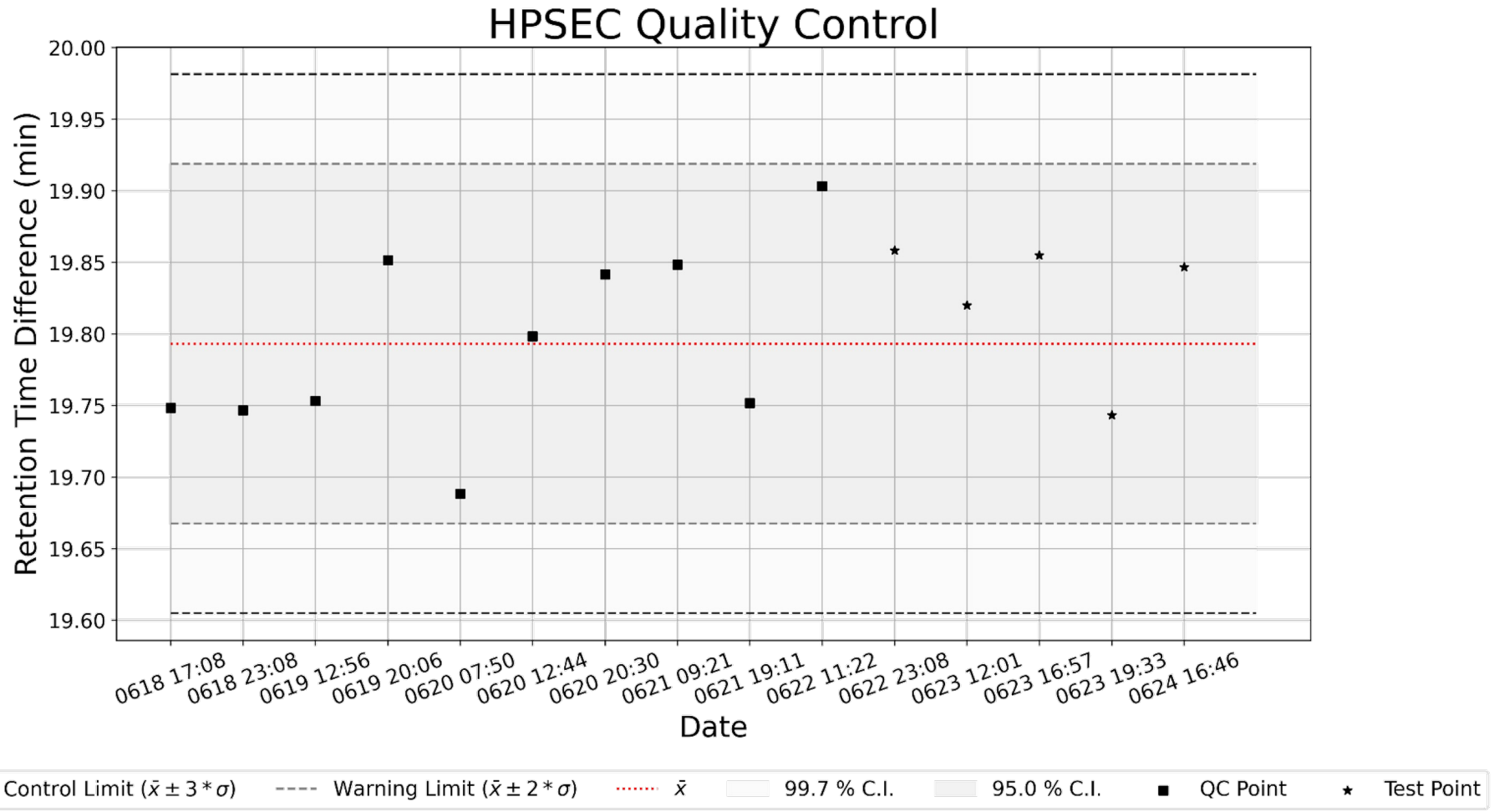
Source: https://en.wikipedia.org/wiki/Radar_chart#/media/File:MER_Star_Plot.gif

Biplot & Control Chart (I)

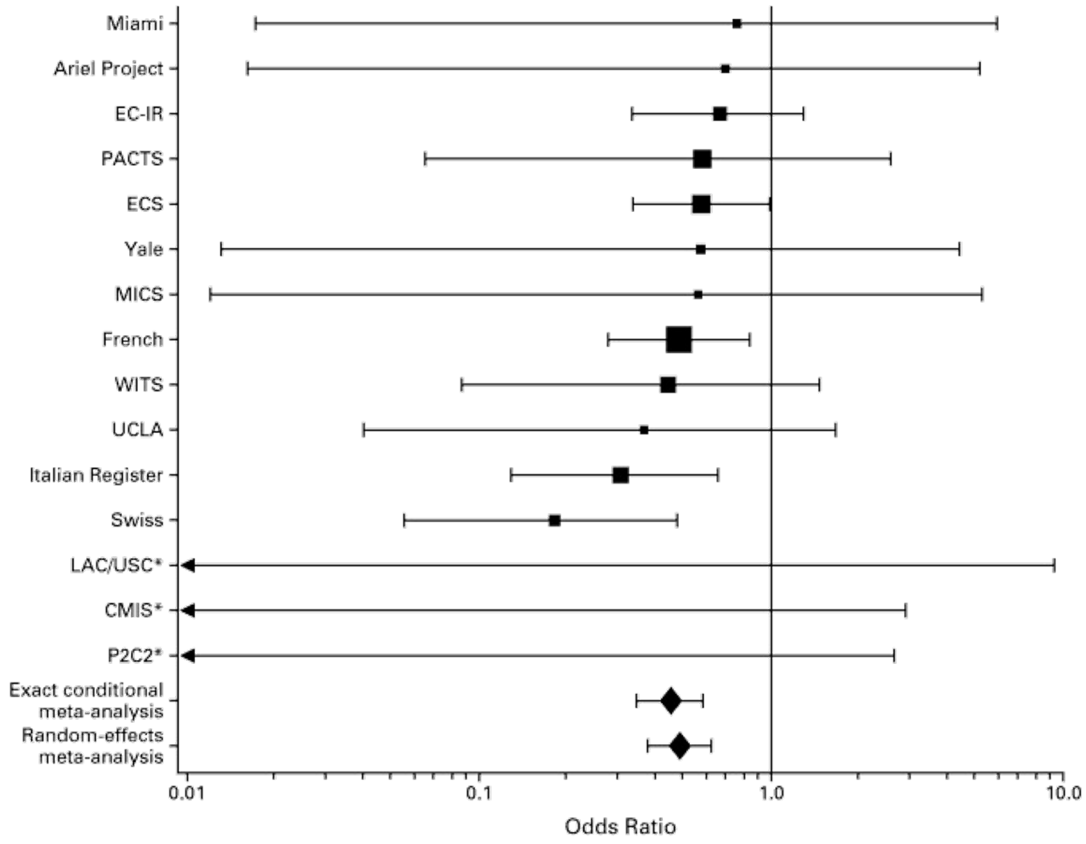




Control Chart (II)



Forest Plot



	Diseased	Healthy
Exposed	20	380
Not Exposed	10	490

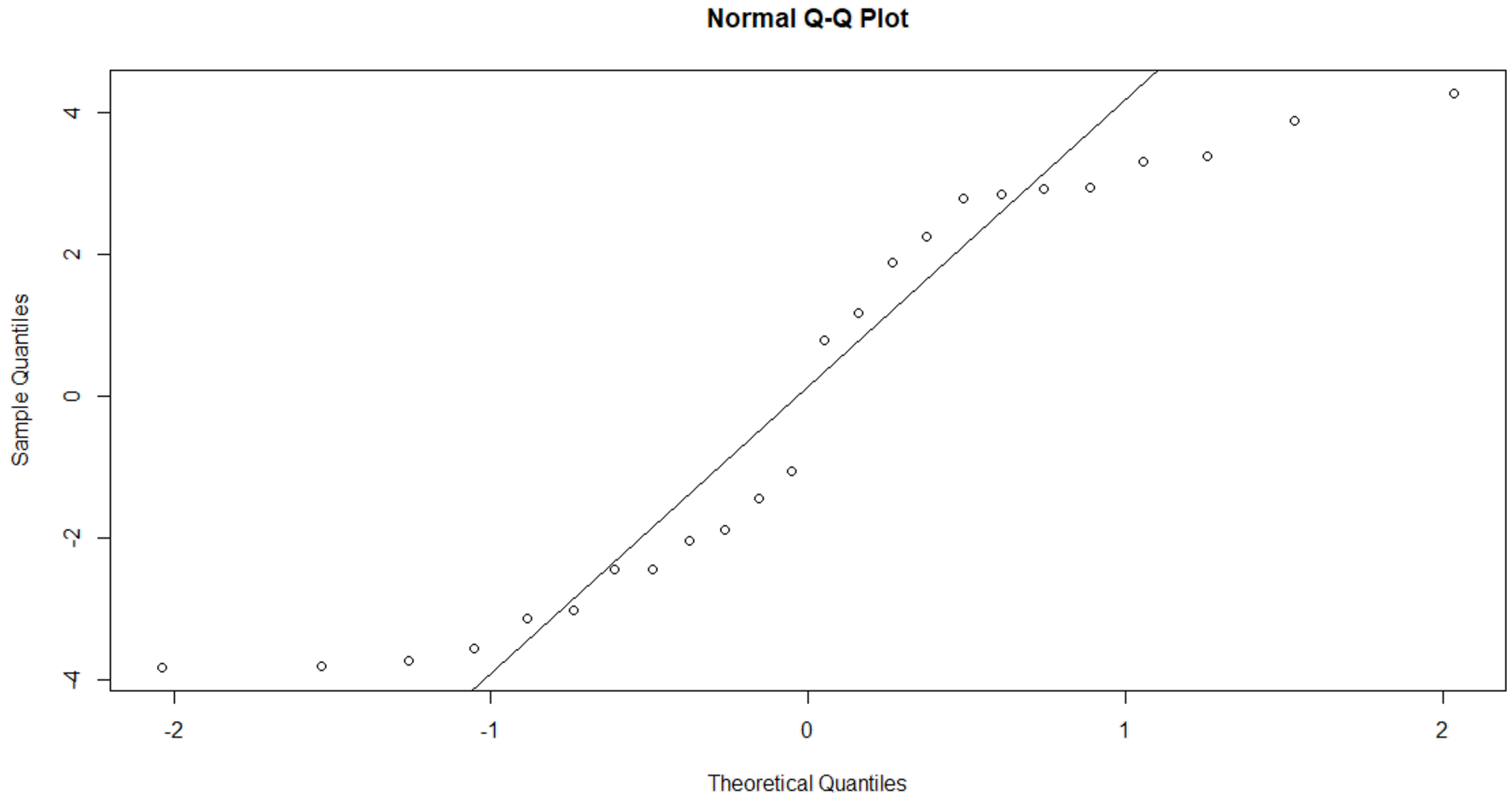
$$risk\ of\ developing\ the\ disease\ given\ exposure = \frac{DE}{VE} = \frac{20}{400}$$

$$risk\ of\ developing\ the\ disease\ given\ non-exposure = \frac{DN}{VN} = \frac{10}{500}$$

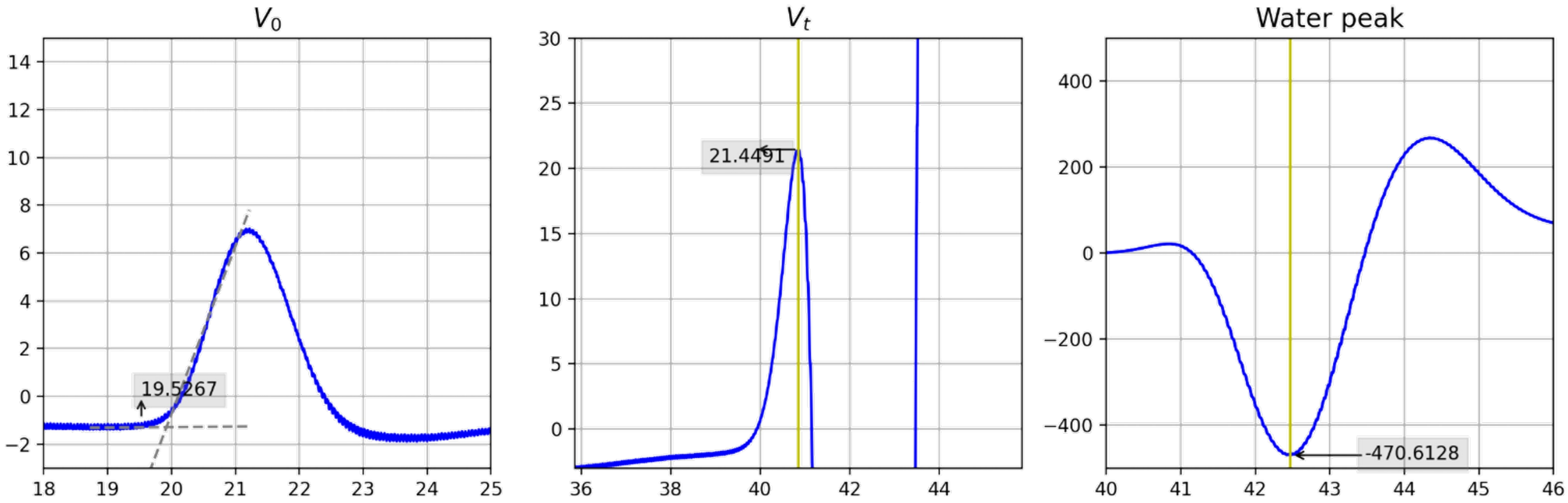
$$relative\ risk = \frac{\frac{DE}{(DE+HE)}}{\frac{DN}{(DN+HN)}} = \frac{DE/VE}{DN/VN} = \frac{20/400}{10/500}$$

$$odds\ ratio = \frac{DE/HE}{DN/HN} = \frac{20/380}{10/500}$$

Quantile-quantile Plot (Q-Q Plot)

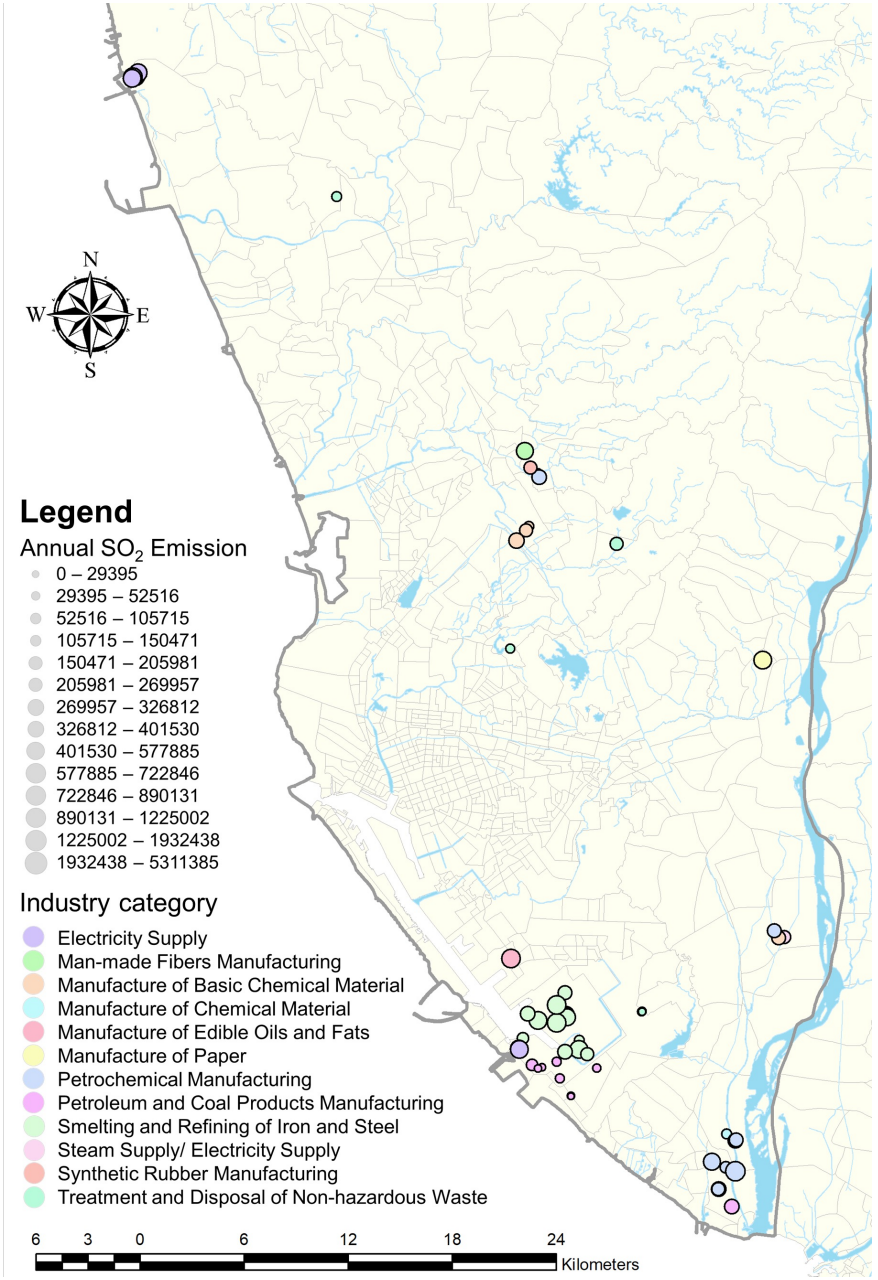


Combination Chart



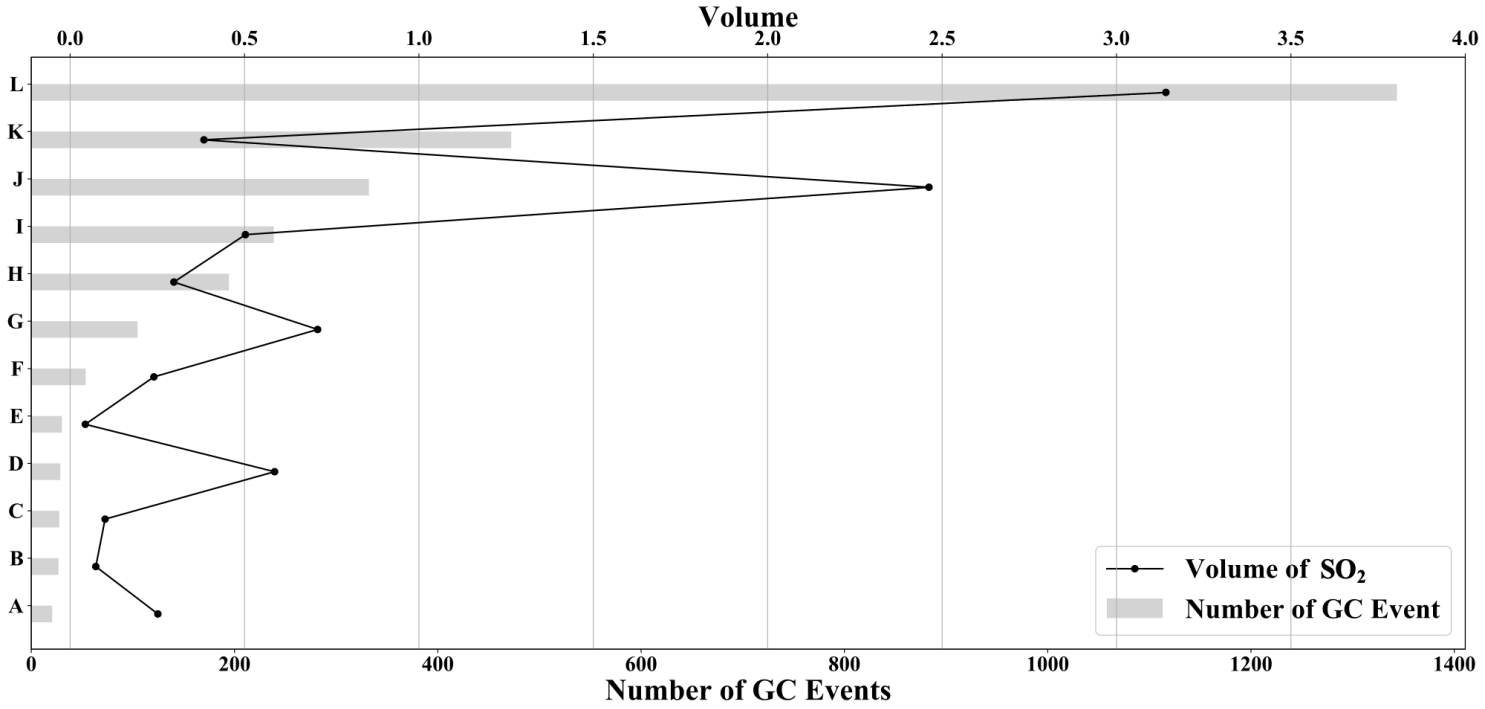
Map #1

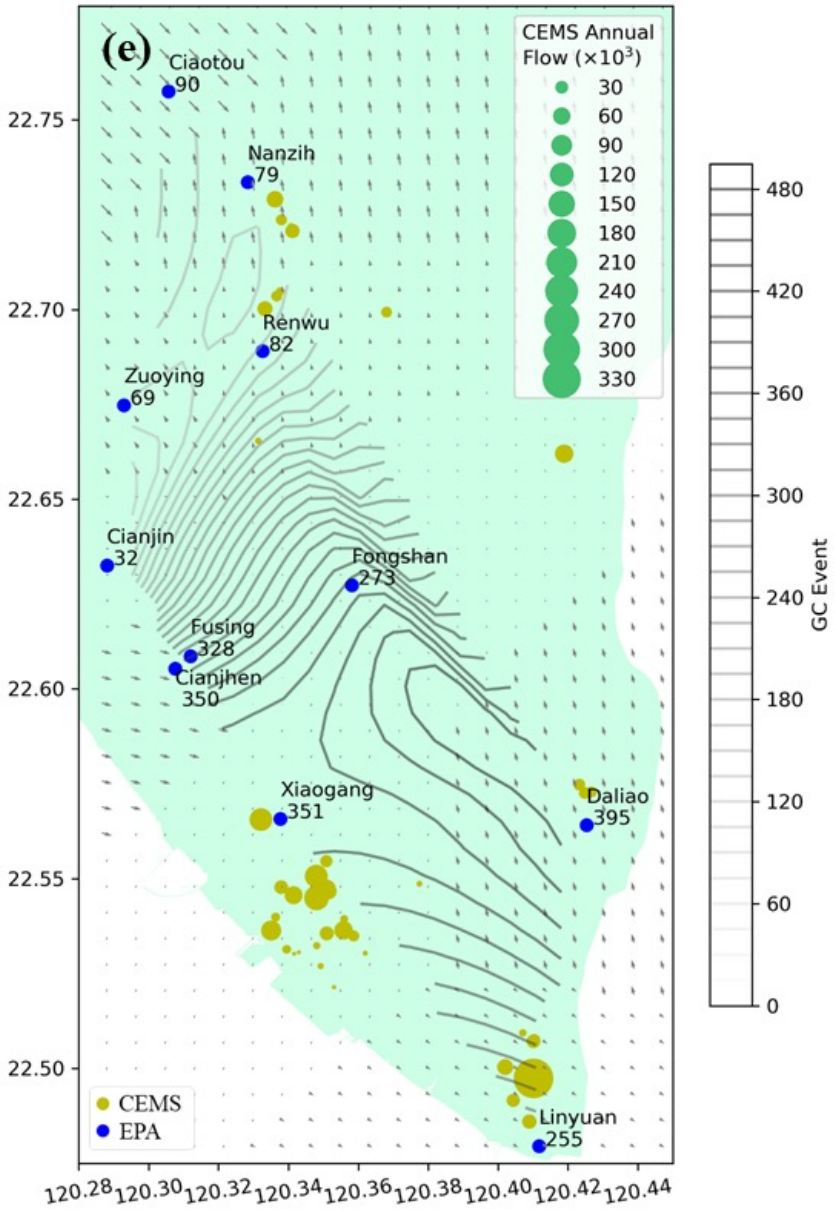
- Important elements in the figure.
    1. Title
    2. X/ Y tick label
    3. X/ Y label
    4. Legend (size, color, symbol)
    5. Grid (optional)
    6. Error bar (optional)
    7. Confidence interval (optional)
    8. Colormap (optional)
    9. Compass icon(optional)
    10. Scale bar(optional)



Figures are from Chan et al. (2022). A novel evaluation of air pollution impact from stationary emission sources to ambient air quality via time-series Granger causality. Earth Data Analytics for Planetary Health. Springer.

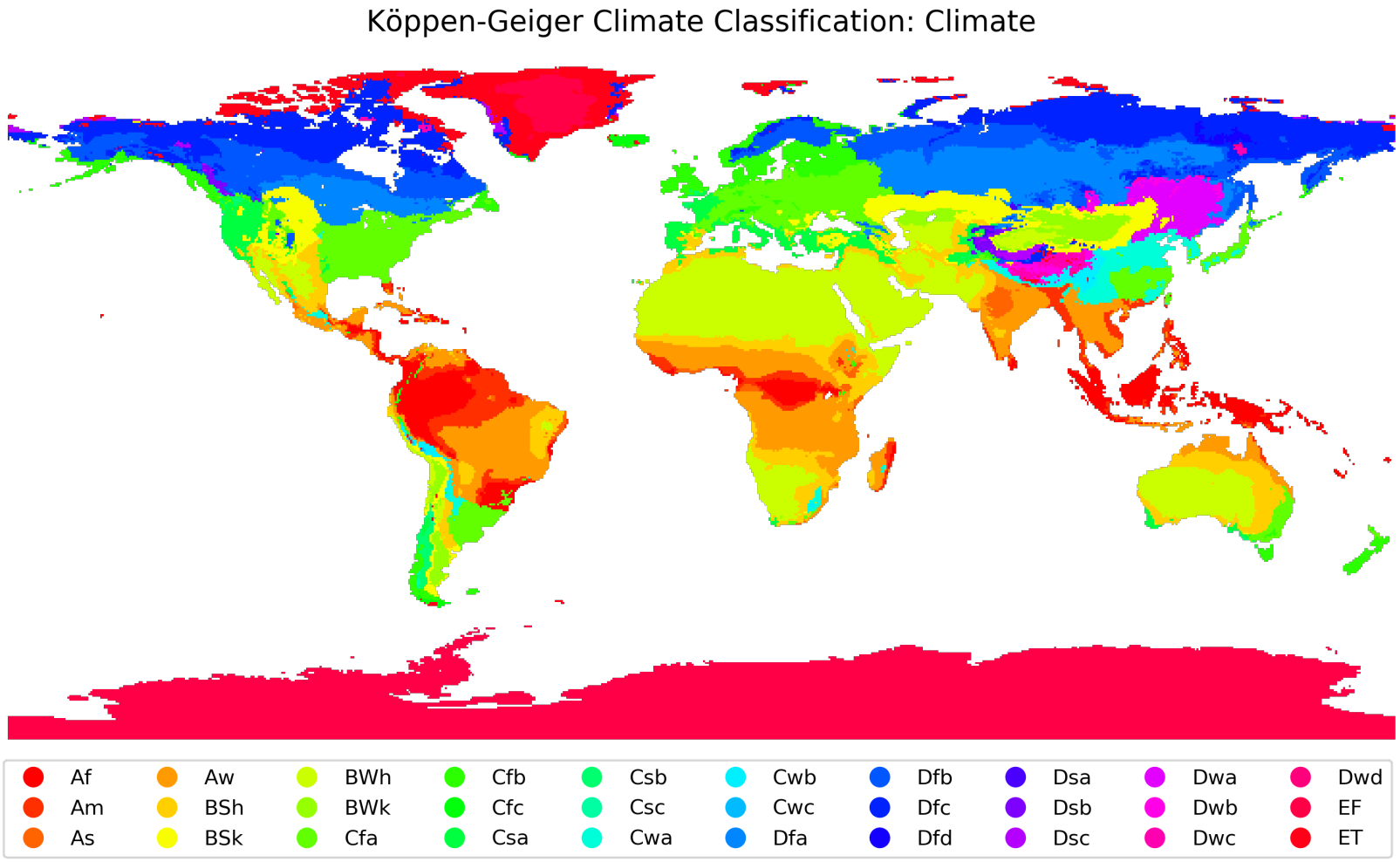
Map #2



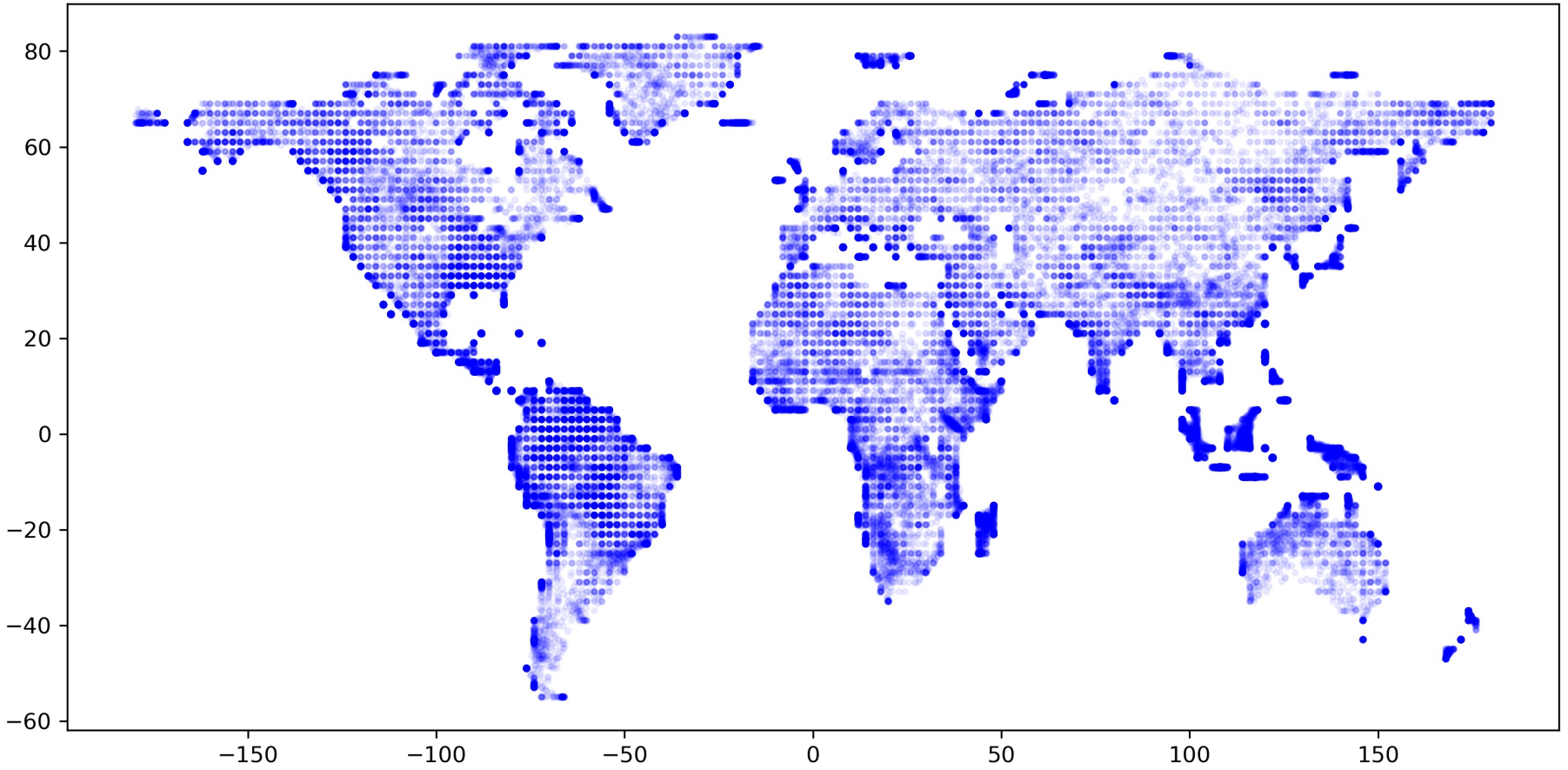


Figures are from Chan et al. (2022). A novel evaluation of air pollution impact from stationary emission sources to ambient air quality via time-series Granger causality. Earth Data Analytics for Planetary Health. Springer.

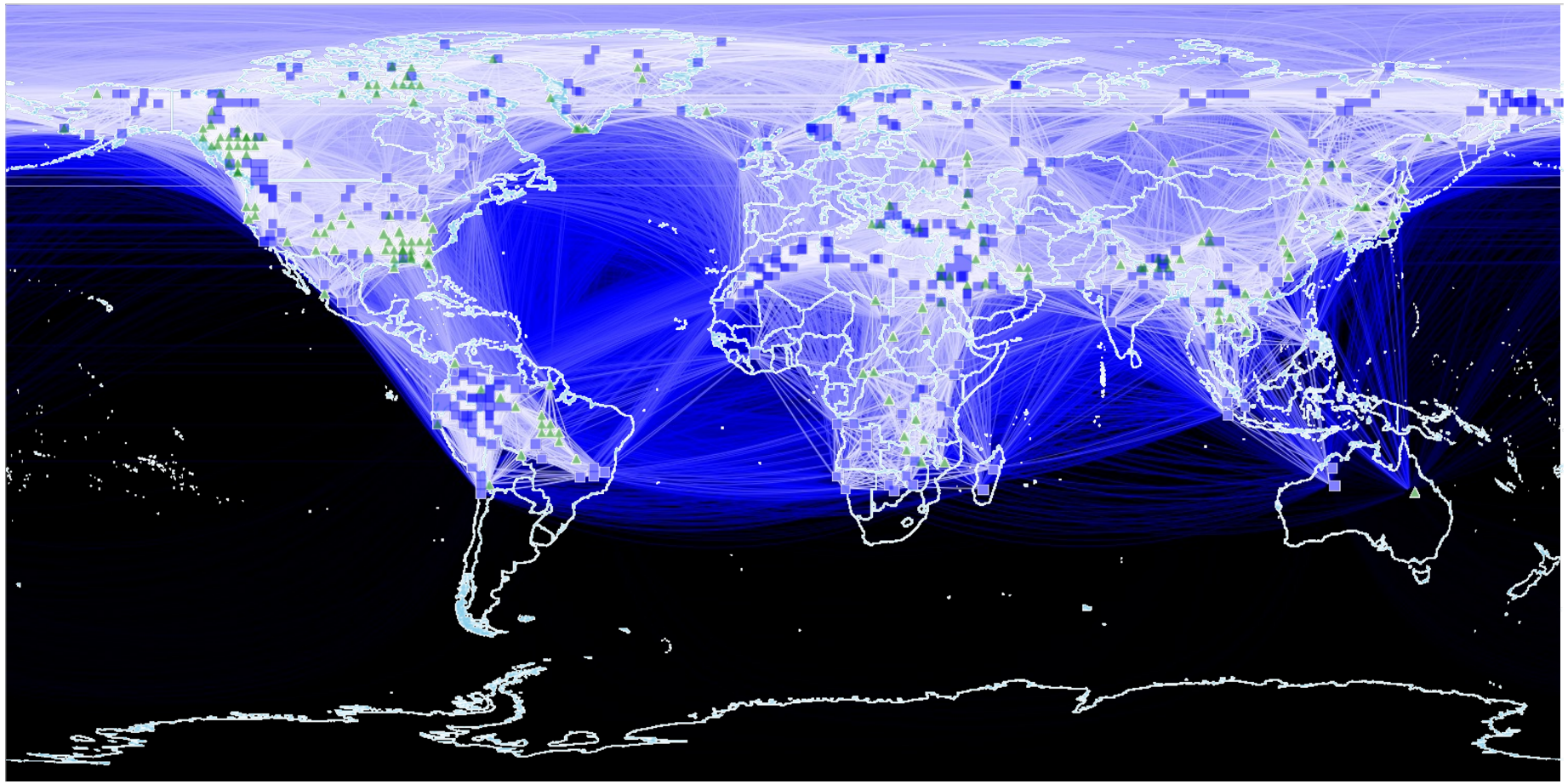
Map #3


Köppen-Geiger Climate Classification: Climate
Af
Am
As
Aw
BSh
BSk
BWh
BWk
Cfa
Cfb
Cfc
Csa
Csb
Csc
Cwa
Cwb
Cwc
Dfa
Dfb
Dfc
Dfd
Dsa
Dsb
Dsc
Dwa
Dwb
Dwc
Dwd
EF
ET

Map #4



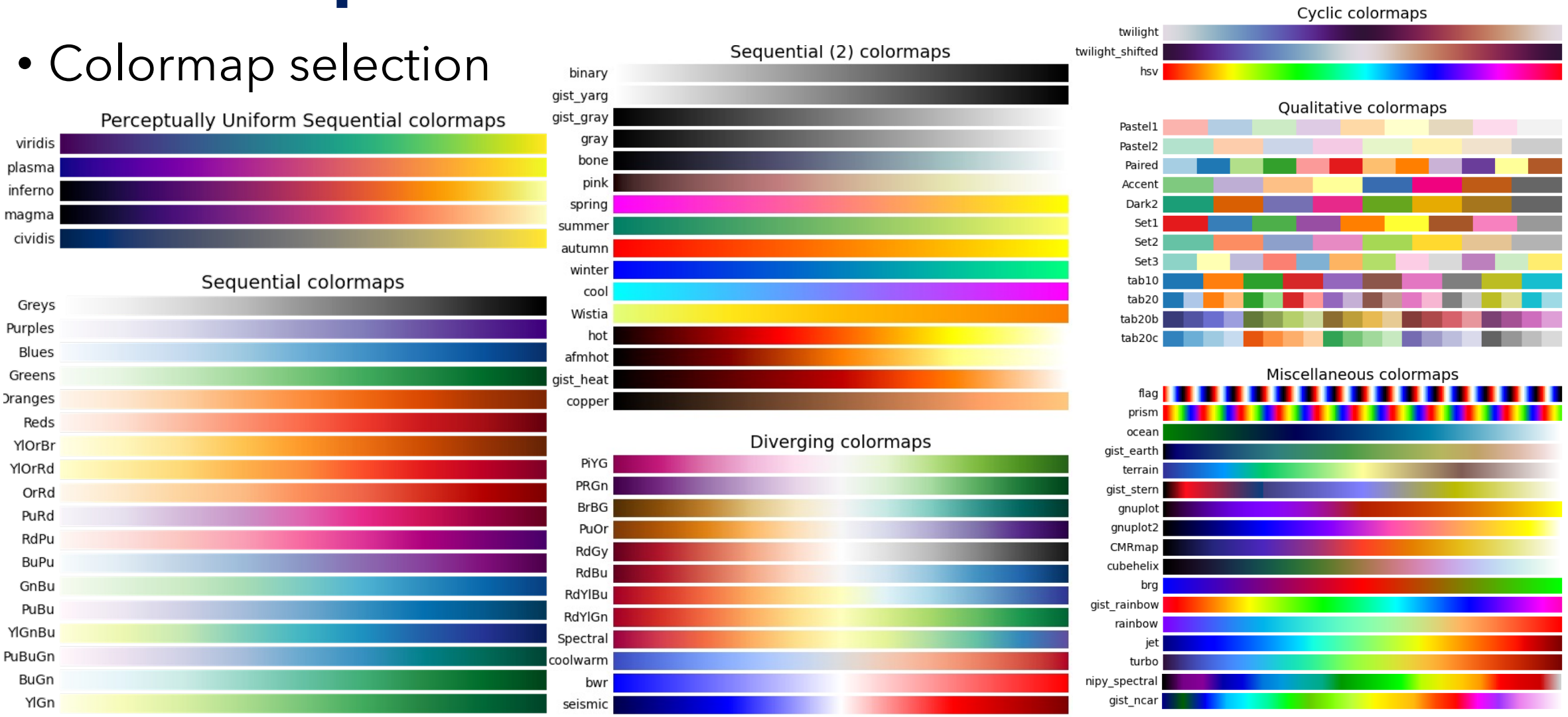
Map #5



Colormap

- Colormap selection

Figure source:
https://matplotlib.org/stable/gallery/color/colormap_reference.html#sphx-glr-gallery-color-colormap-reference-py



Colormap

- Colormap selection
  - Do not pick more than 8 colors from graduated colormap
  - Graduated colormaps are for continuous values
  - Discrete colormaps are for categorical values
  - … (**think about it!)**

- To better the understanding of figures…
  - Use different sizes or symbols to represent different data
  - Plot different data into the same subplot/ figure
  - Use subplot with fixed x and y ranges

Matplotlib

- Matplotlib is a useful package for visualization; however, we cannot demonstrate all functions in this class. But we still introduce the most useful functions.

```
# import python packages
import matplotlib.pyplot as plt
```

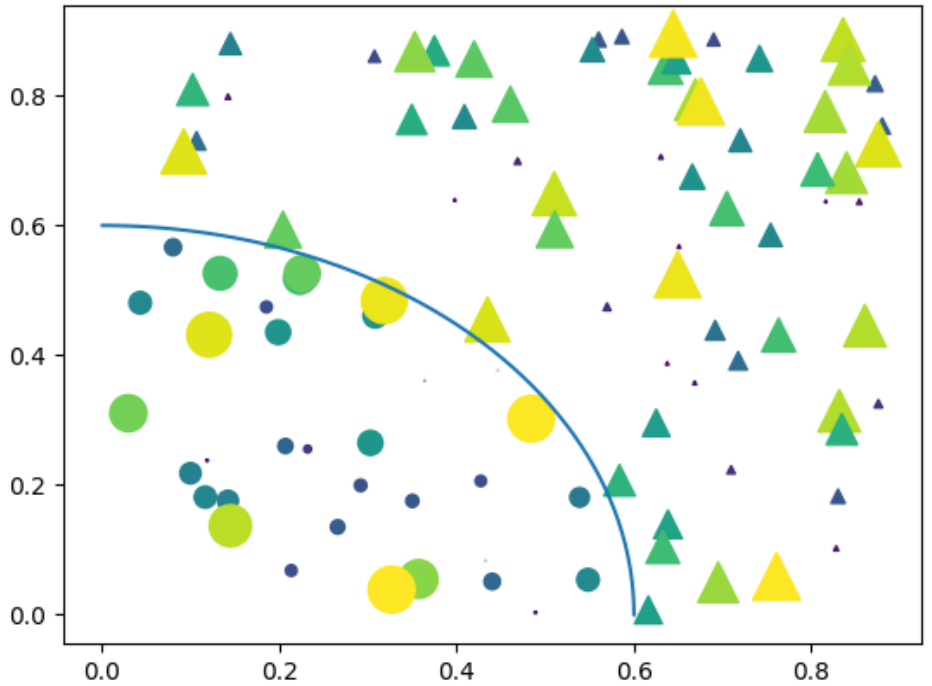
matplotlib.pyplot.scatter

```
matplotlib.pyplot.scatter(x, y, s=None, c=None, marker=None, cmap=None,
norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, *,
edgecolors=None, plotnonfinite=False, data=None, **kwargs)        [source]
```

A scatter plot of *y* vs. *x* with varying marker size and/or color.

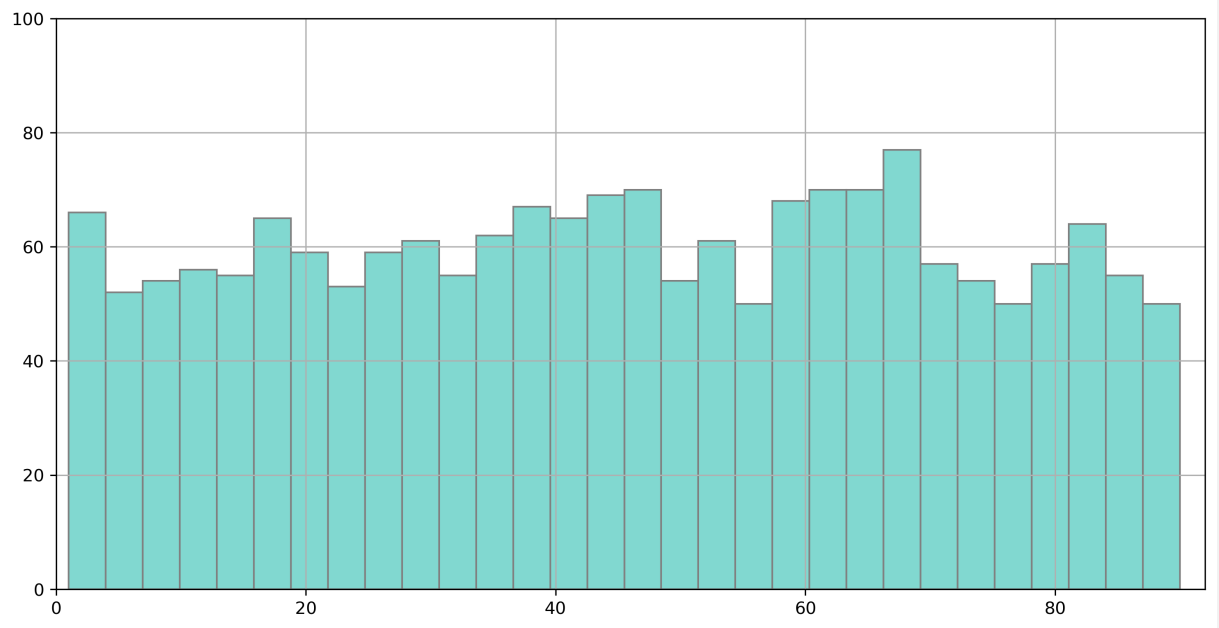
Scatter and Line

```
import matplotlib.pyplot as plt
import numpy as np
# Fixing random state for reproducibility
np.random.seed(19680801)
N = 100
r0 = 0.6
x = 0.9 * np.random.rand(N)
y = 0.9 * np.random.rand(N)
area = (20 * np.random.rand(N))**2  # 0 to 10 point radii
c = np.sqrt(area)
r = np.sqrt(x ** 2 + y ** 2)
area1 = np.ma.masked_where(r < r0, area)
area2 = np.ma.masked_where(r >= r0, area)
plt.scatter(x, y, s=area1, marker='^', c=c)
plt.scatter(x, y, s=area2, marker='o', c=c)
# Show the boundary between the regions:
theta = np.arange(0, np.pi / 2, 0.01)
plt.plot(r0 * np.cos(theta), r0 * np.sin(theta))
plt.show()
```



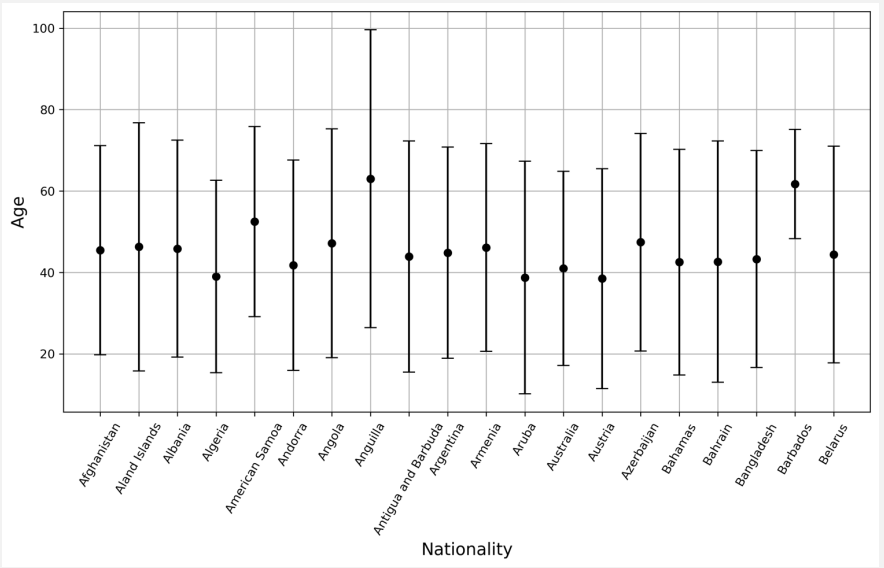
Histogram

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

plt.figure(figsize=[12,6], dpi=300)
plt.hist(jp['Age'], bins=30, facecolor='#81D8D0', edgecolor='gray')
plt.axis([0,92,0,100])
plt.grid(True)
plt.show()
```



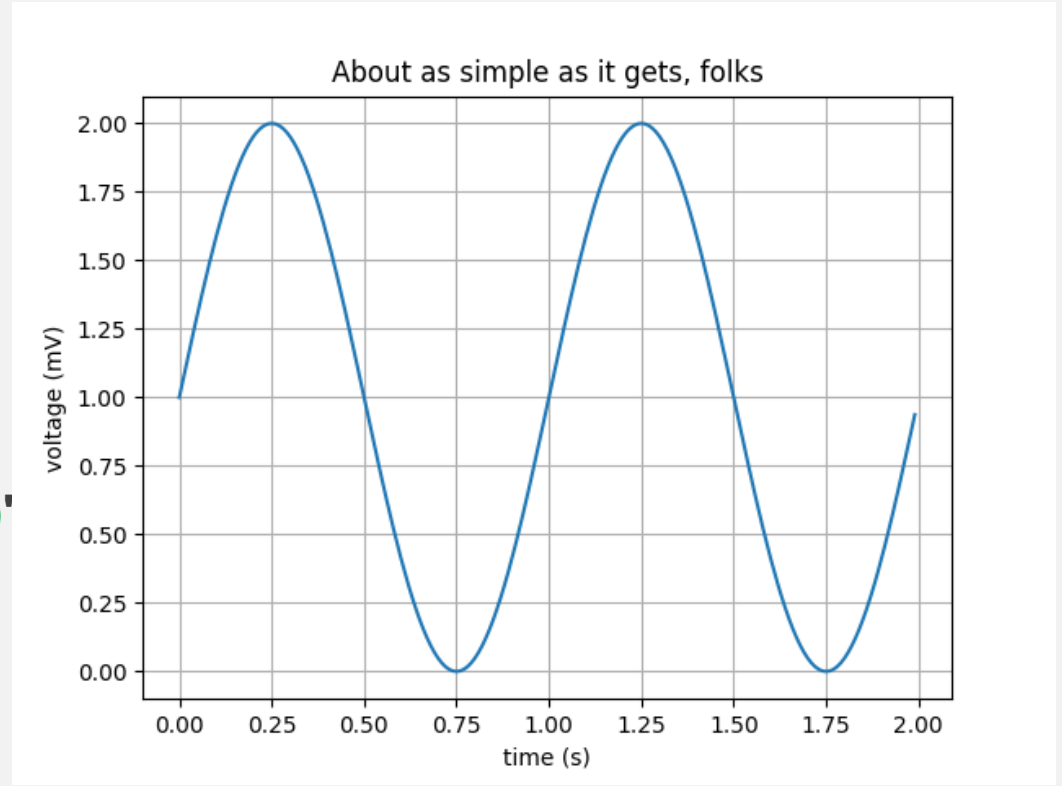
Error Bar

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

plt.figure(figsize=[12,6], dpi=300)
plt.errorbar(np.arange(20), mean['Age'], yerr=std['Age'], capsize=5, linestyle='',
color='k')
plt.scatter(np.arange(20), mean['Age'], c='k')
plt.xticks(np.arange(20), mean['Nationality'],
                rotation=60)
plt.ylabel('Age', fontsize=14)
plt.xlabel('Nationality', fontsize=14)
plt.grid(True)
plt.show()
```



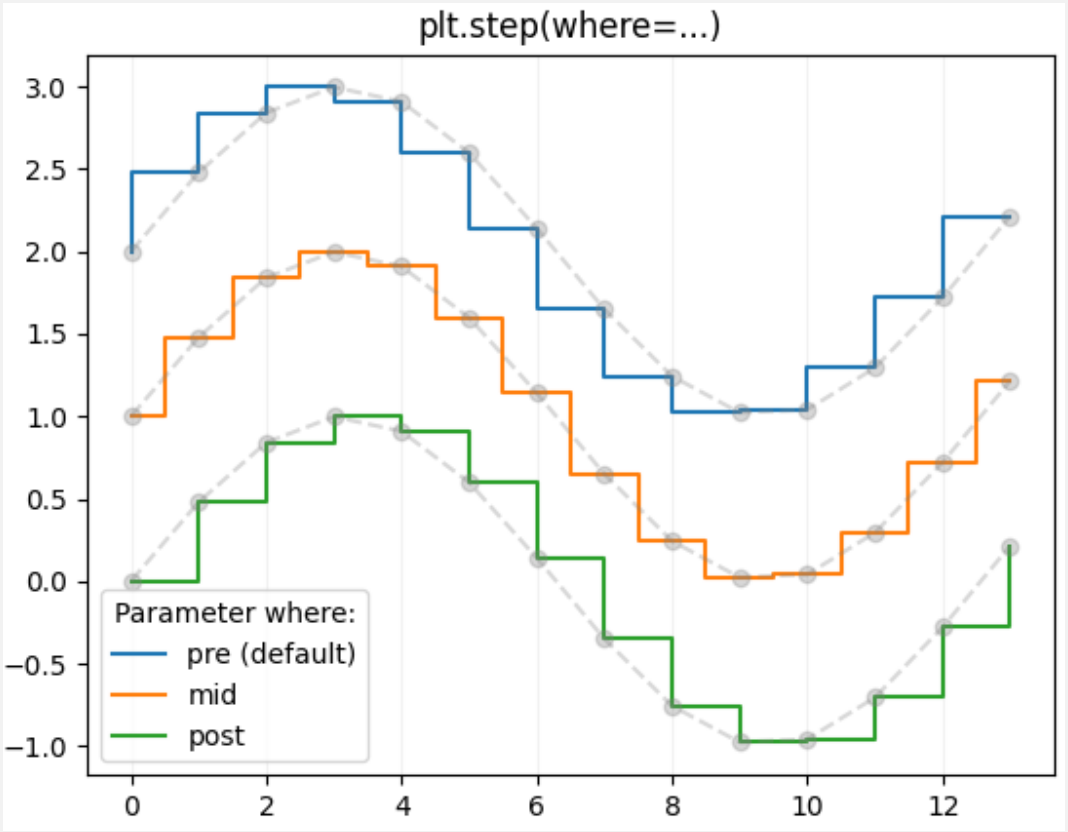
Line

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

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

fig, ax = plt.subplots()
ax.plot(t, s)

ax.set(xlabel='time (s)', ylabel='voltage (mV)'
       title='About as simple as it gets, folks')
ax.grid()

fig.savefig("test.png")
plt.show()
```



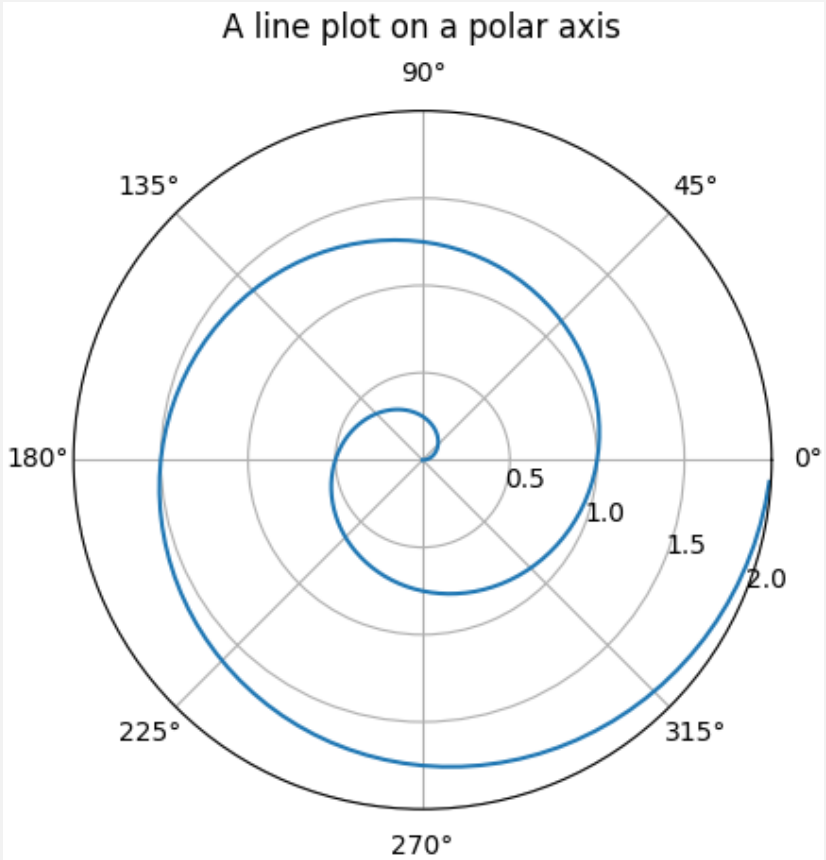
Step

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

x = np.arange(14)
y = np.sin(x / 2)

plt.step(x, y + 2, label='pre (default)')
plt.plot(x, y + 2, 'o--', color='grey', alpha=0.3)

plt.step(x, y + 1, where='mid', label='mid')
plt.plot(x, y + 1, 'o--', color='grey', alpha=0.3)

plt.grid(axis='x', color='0.95')
plt.legend(title='Parameter where:')
plt.title('plt.step(where=...)')
plt.show()
```

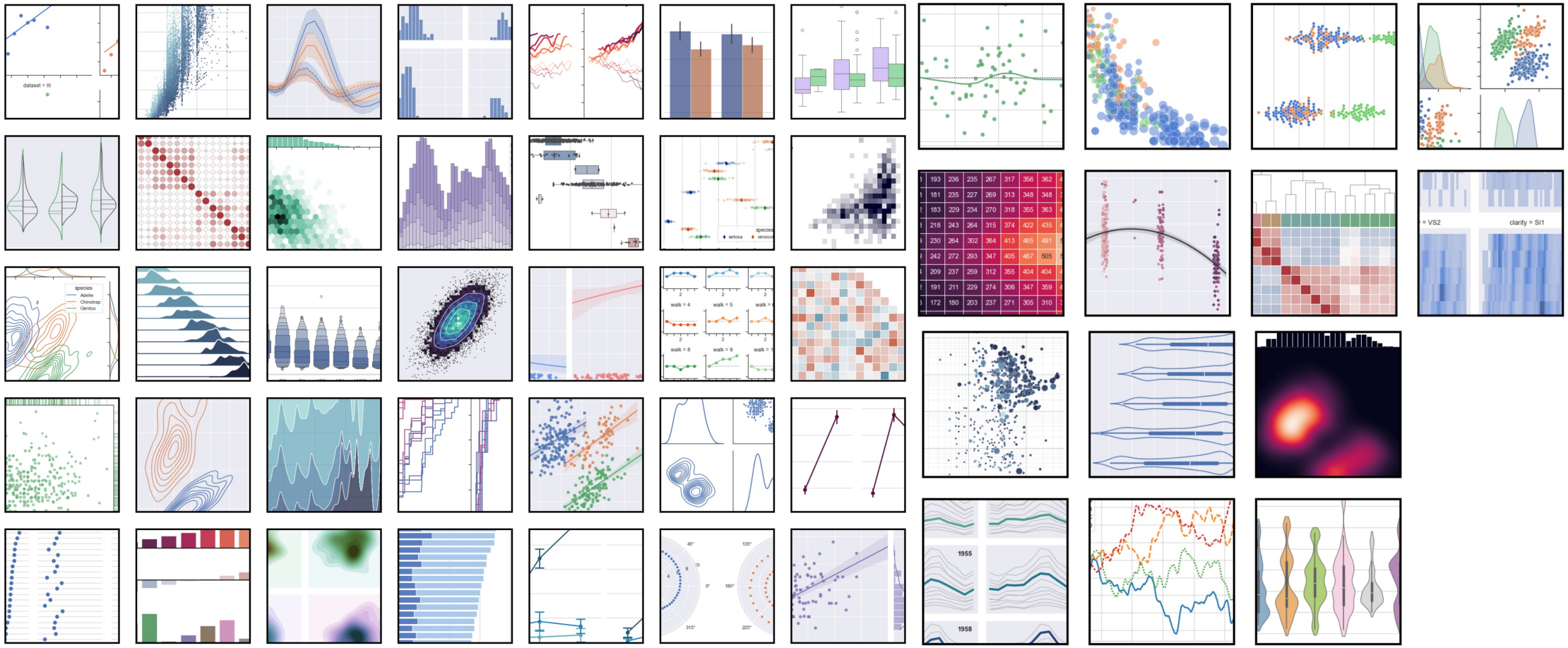


Plot R Theta Phi

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

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r

fig, ax = plt.subplots(subplot_kw={'projection': 'polar'})
ax.plot(theta, r)
ax.set_rmax(2)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.set_rlabel_position(-22.5)
ax.grid(True)

ax.set_title("A line plot on a polar axis", va='bottom')
plt.show()
```

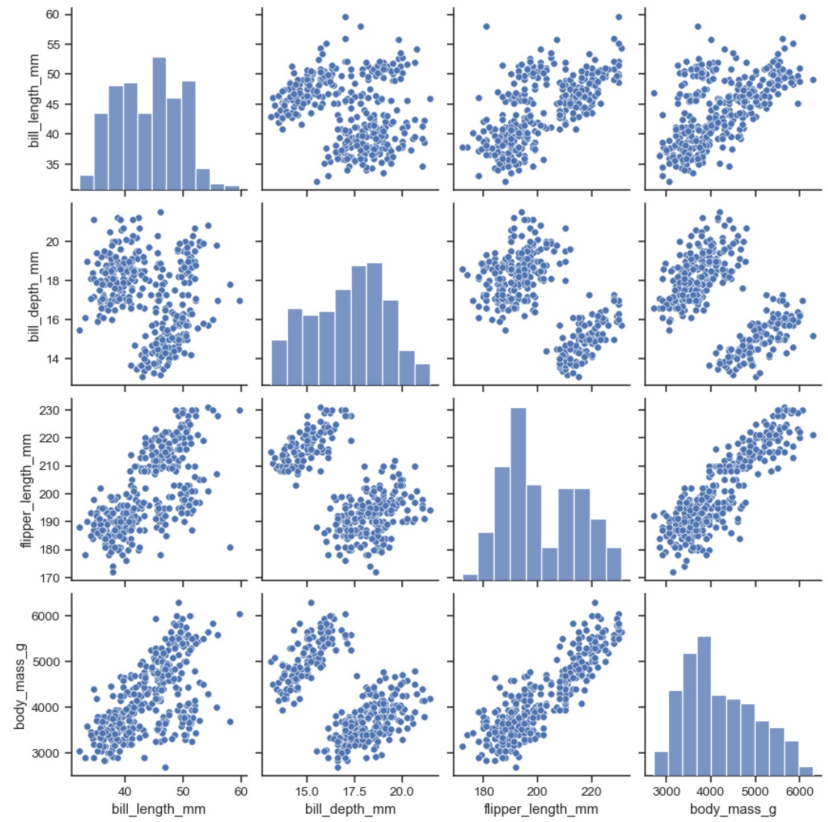


Seaborn

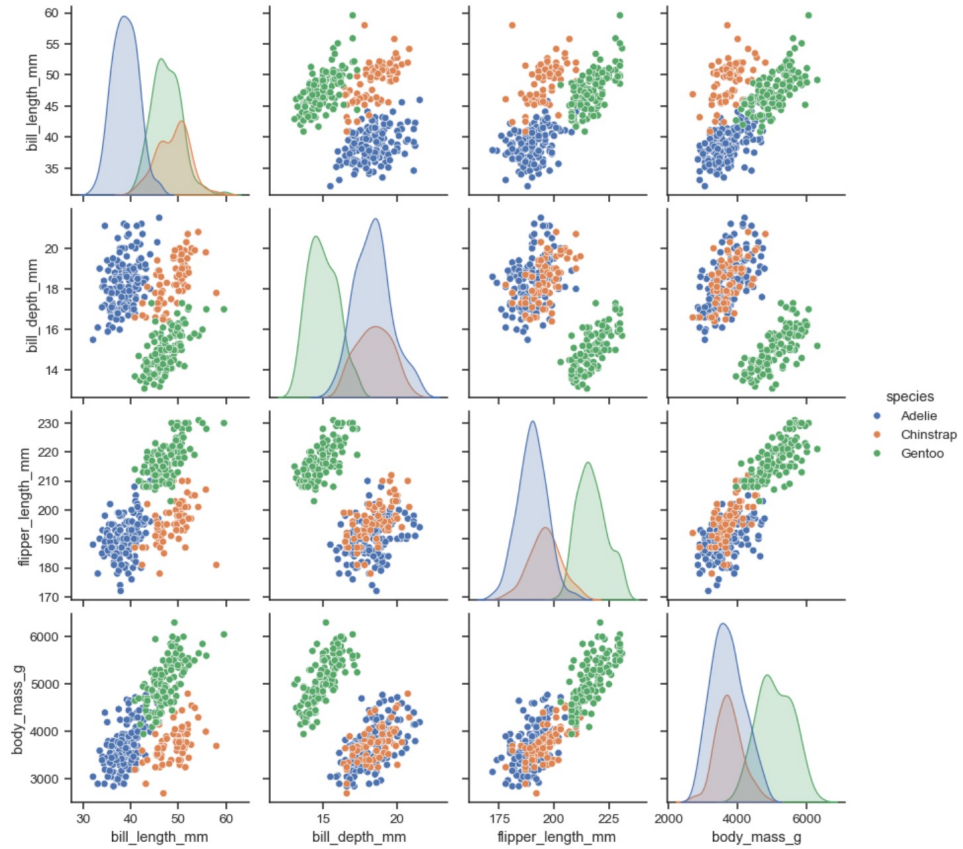


Pairplot

```
penguins = sns.load_dataset("penguins")
sns.pairplot(penguins)
```

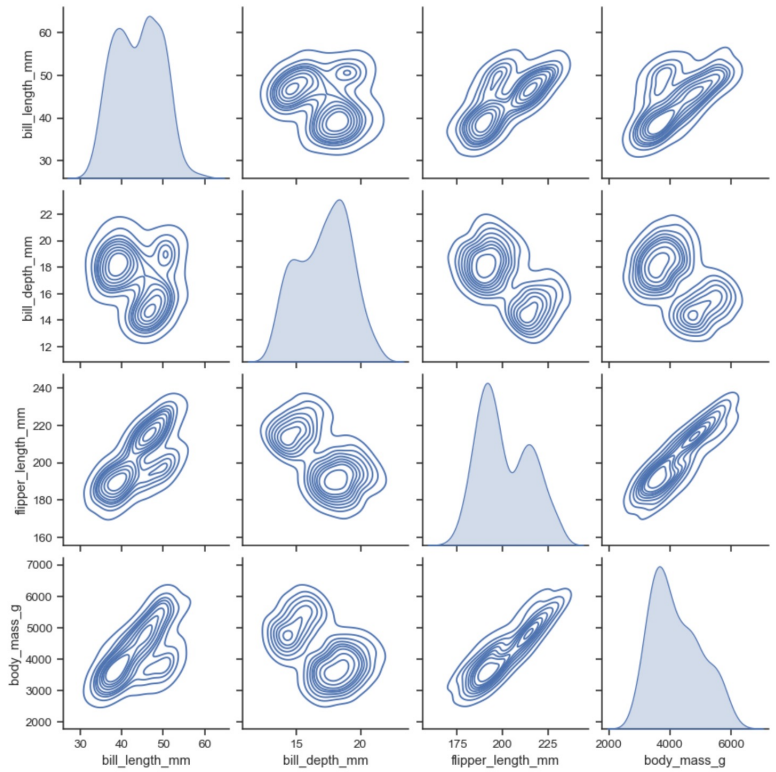



```
sns.pairplot(penguins, hue="species")
```

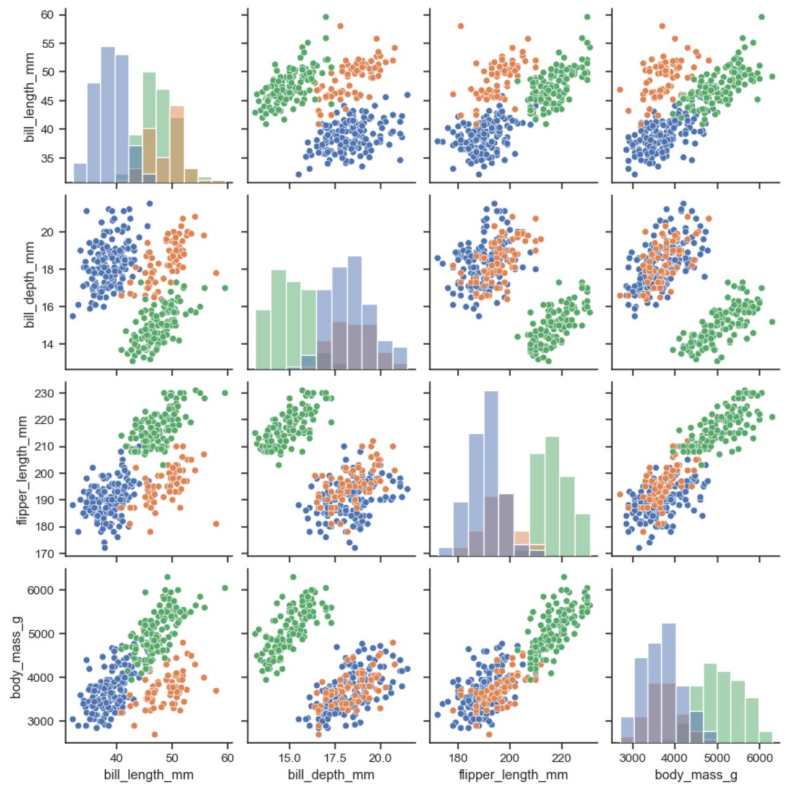


Pairplot

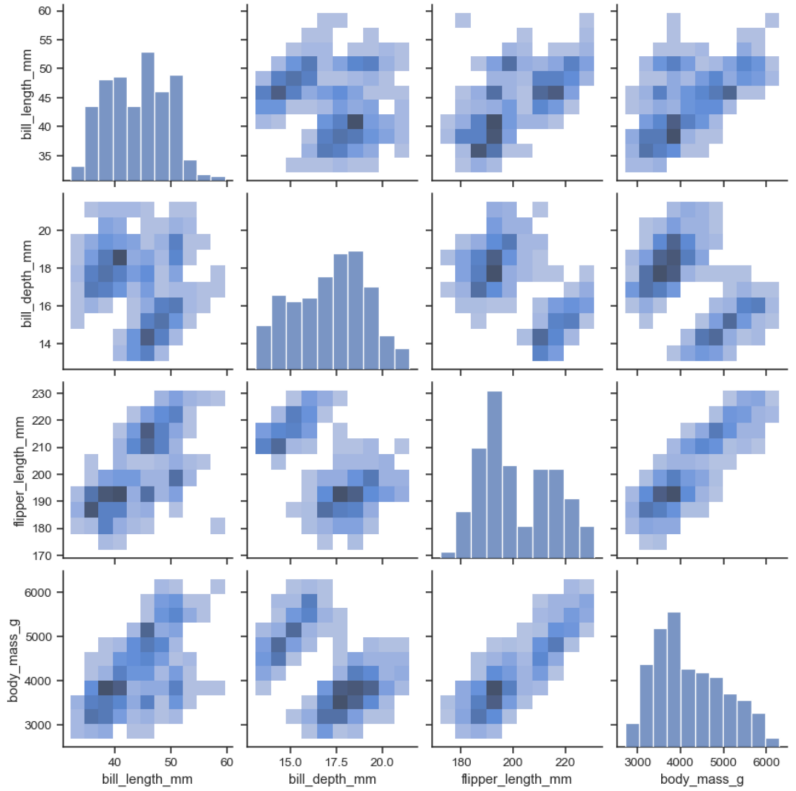
```
sns.pairplot(penguins, kind="kde")
```




```
sns.pairplot(penguins, hue="species", diag_kind="hist")
```




```
sns.pairplot(penguins, kind="hist")
```

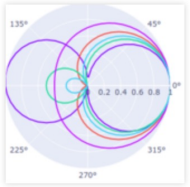


Plotly

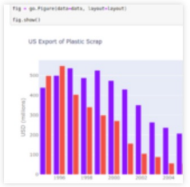
Fundamentals

More Fundamentals »

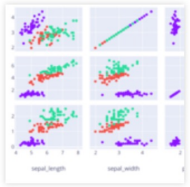
The Figure Data Structure



Creating and Updating Figures



Displaying Figures

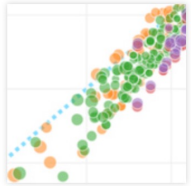


Plotly Express

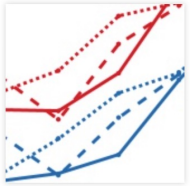
Analytical Apps with Dash

Basic Charts

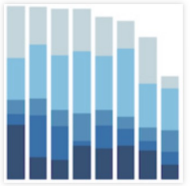
More Basic Charts »



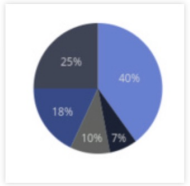
Scatter Plots



Line Charts



Bar Charts



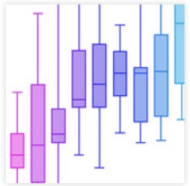
Pie Charts

Bubble Charts

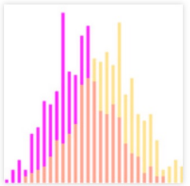
Statistical Charts

More Statistical Charts »

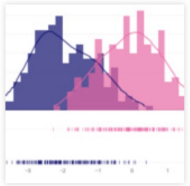
Error Bars



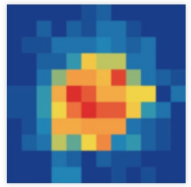
Box Plots



Histograms



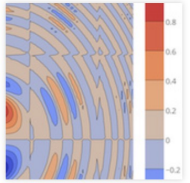
Distplots



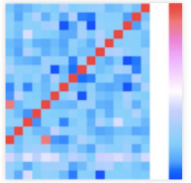
2D Histograms

Scientific Charts

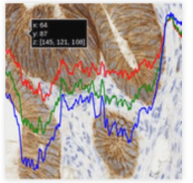
More Scientific Charts »



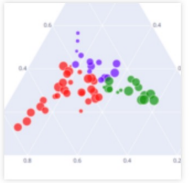
Contour Plots



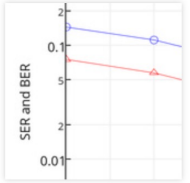
Heatmaps



Imshow



Ternary Plots



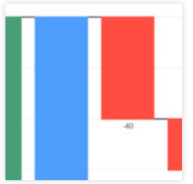
Log Plots

Financial Charts

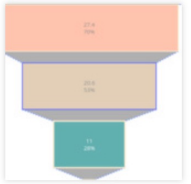
More Financial Charts »

Time Series and Date Axes

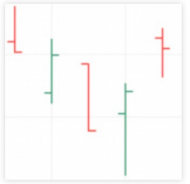
Candlestick Charts



Waterfall Charts



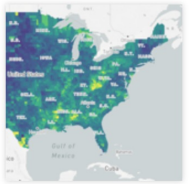
Funnel Chart



OHLC Charts

Maps

More Maps »

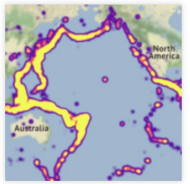


Mapbox Choropleth Maps

Lines on Mapbox

Filled Area on Maps

Bubble Maps

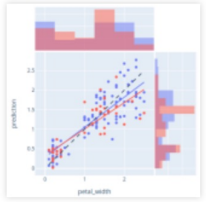


Mapbox Density Heatmap

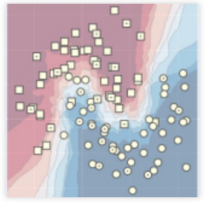
Plotly

Artificial Intelligence and Machine Learning

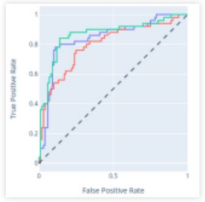
More AI and ML »



ML Regression



kNN Classification



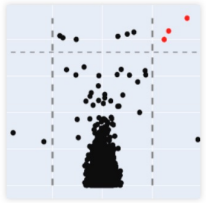
ROC and PR Curves

PCA Visualization

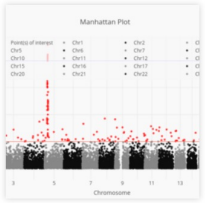
AI/ML Apps with Dash

Bioinformatics

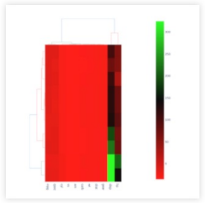
More Bioinformatics »



Volcano Plot



Manhattan Plot

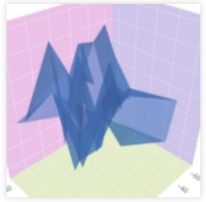


Clustergram

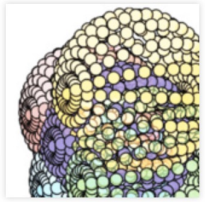
Alignment Chart

3D Charts

More 3D Charts »

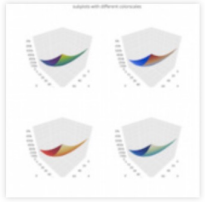


3D Axes

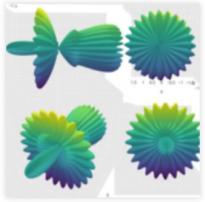


3D Scatter Plots

3D Surface Plots



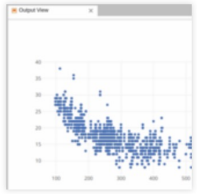
3D Subplots



3D Camera Controls

Jupyter Widgets Interaction

Plotly FigureWidget Overview



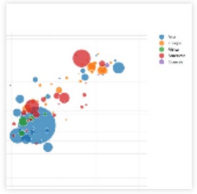
Jupyter Lab with FigureWidget

Interactive Data Analysis with FigureWidget ipywidgets

Click Events

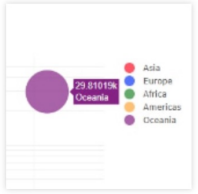
Transforms

Filter



Group By

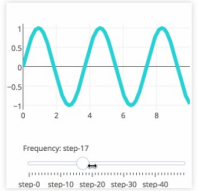
Aggregations



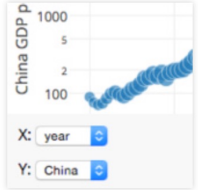
Multiple Transforms

Add Custom Controls

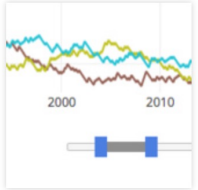
Custom Buttons



Sliders



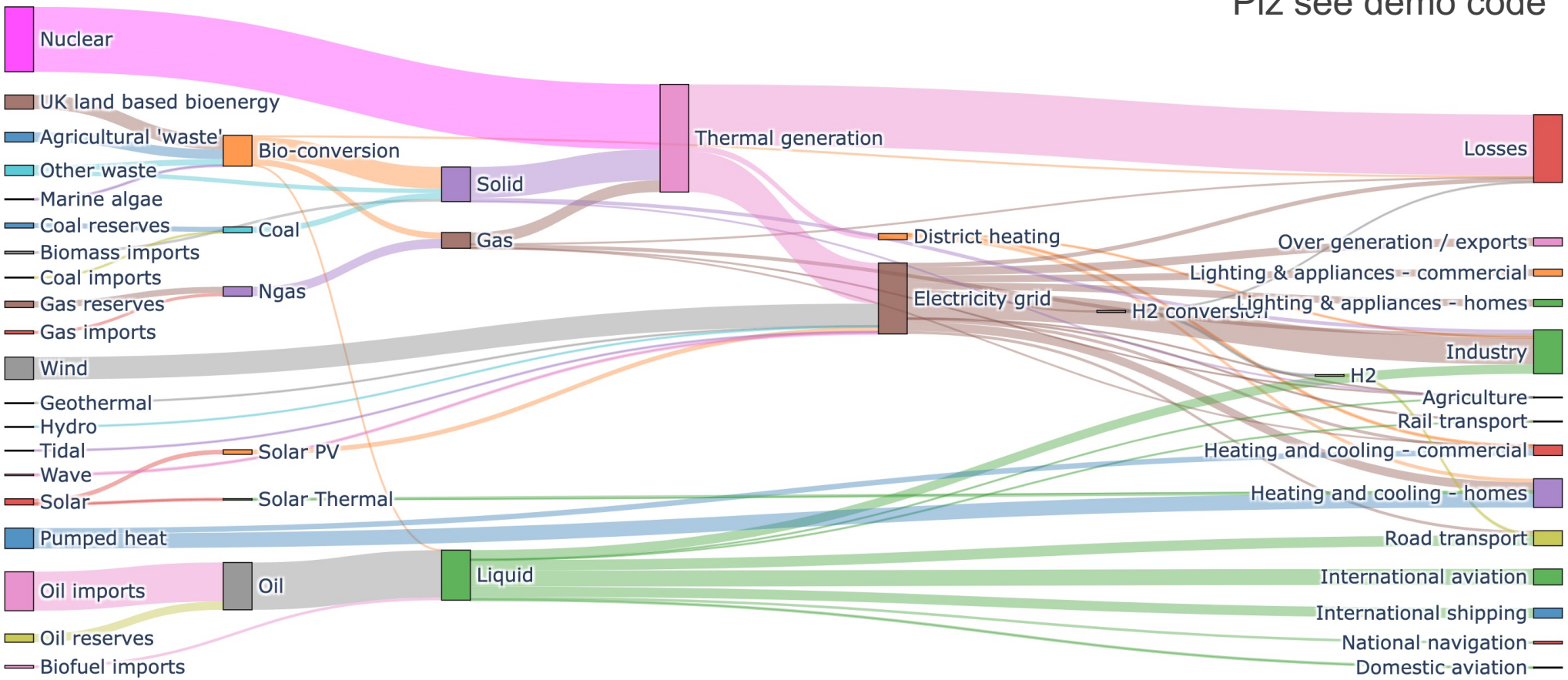
Dropdown Menus



Range Slider and Selector

Sankey Diagram

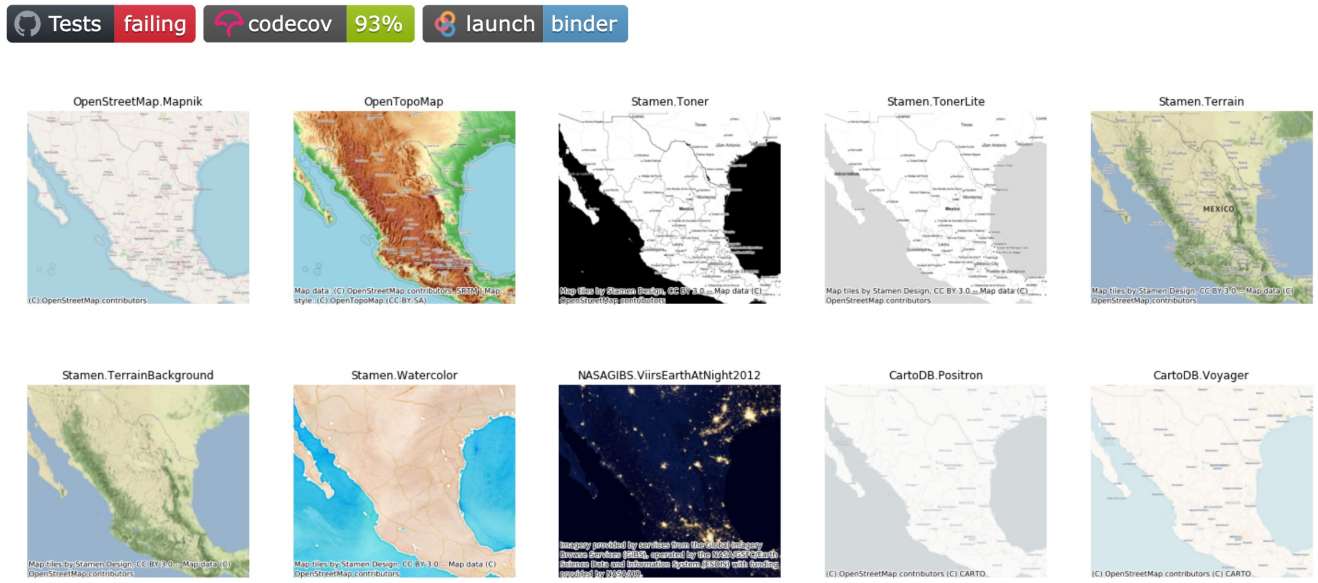
Plz see demo code



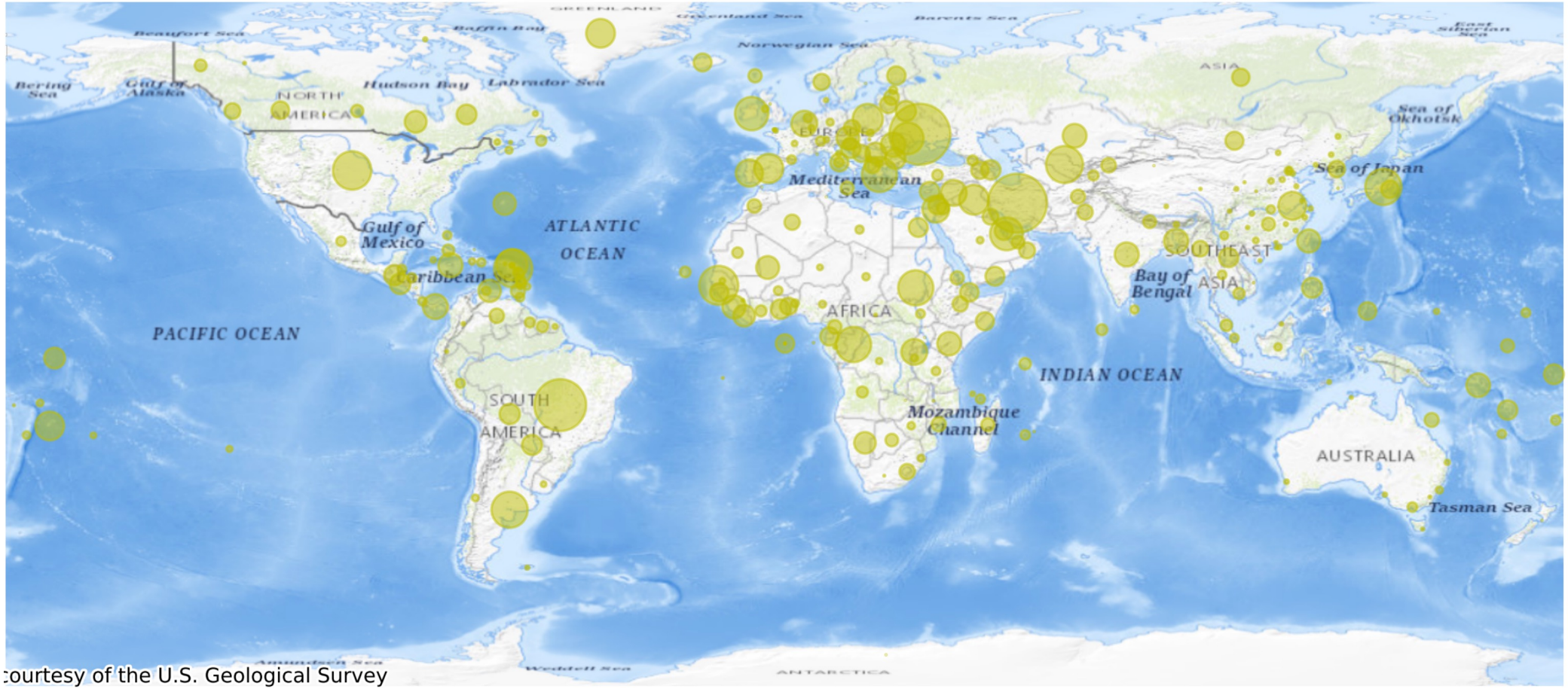
Contextily

`contextily`: context geo tiles in Python

`contextily` is a small Python 3 (3.7 and above) package to retrieve tile maps from the internet. It can add those tiles as basemap to matplotlib figures or write tile maps to disk into geospatial raster files. Bounding boxes can be passed in both WGS84 (`EPSG:4326`) and Spheric Mercator (`EPSG:3857`). See the notebook `contextily_guide.ipynb` for usage.



Contextily



courtesy of the U.S. Geological Survey

Contextily


ry provided by services from the Global Imagery Browse Services (GIBS), operated by the NASA/GSFC/Earth Science Data and Information System (ESDIS
unding provided by NASA/HQ.

Question Time

- **Assignment:**
- **Download today's lab practice and upload to moodle.**
- **Thx**

The End

Thank you for your attention!

Email: chchan@ntnu.edu.tw
Web: toodou.github.io

SCAN ME

SCAN ME