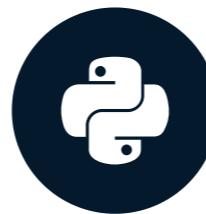


Looking at the farmers market data

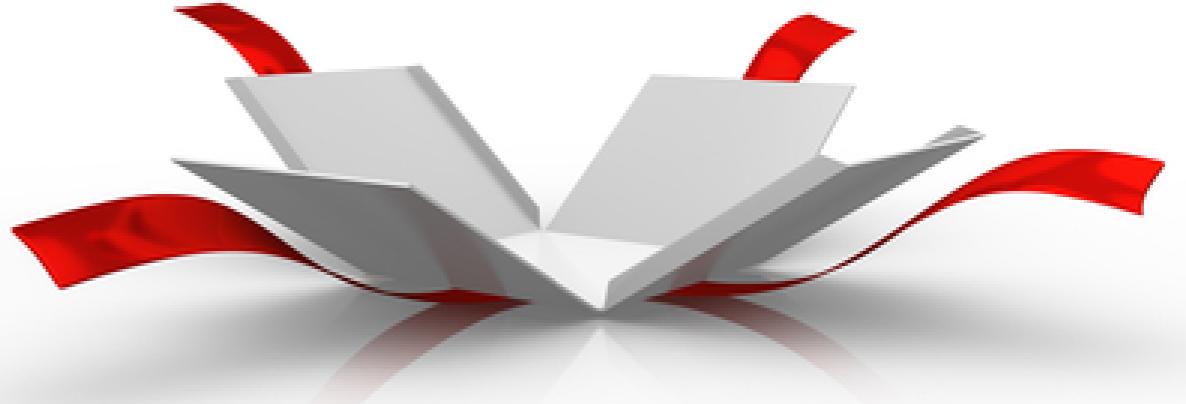
IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON



Nick Strayer
Instructor

First explorations of a dataset

- Take a broad view
- Show as much info as possible
- Don't fuss over appearances



Using your head()

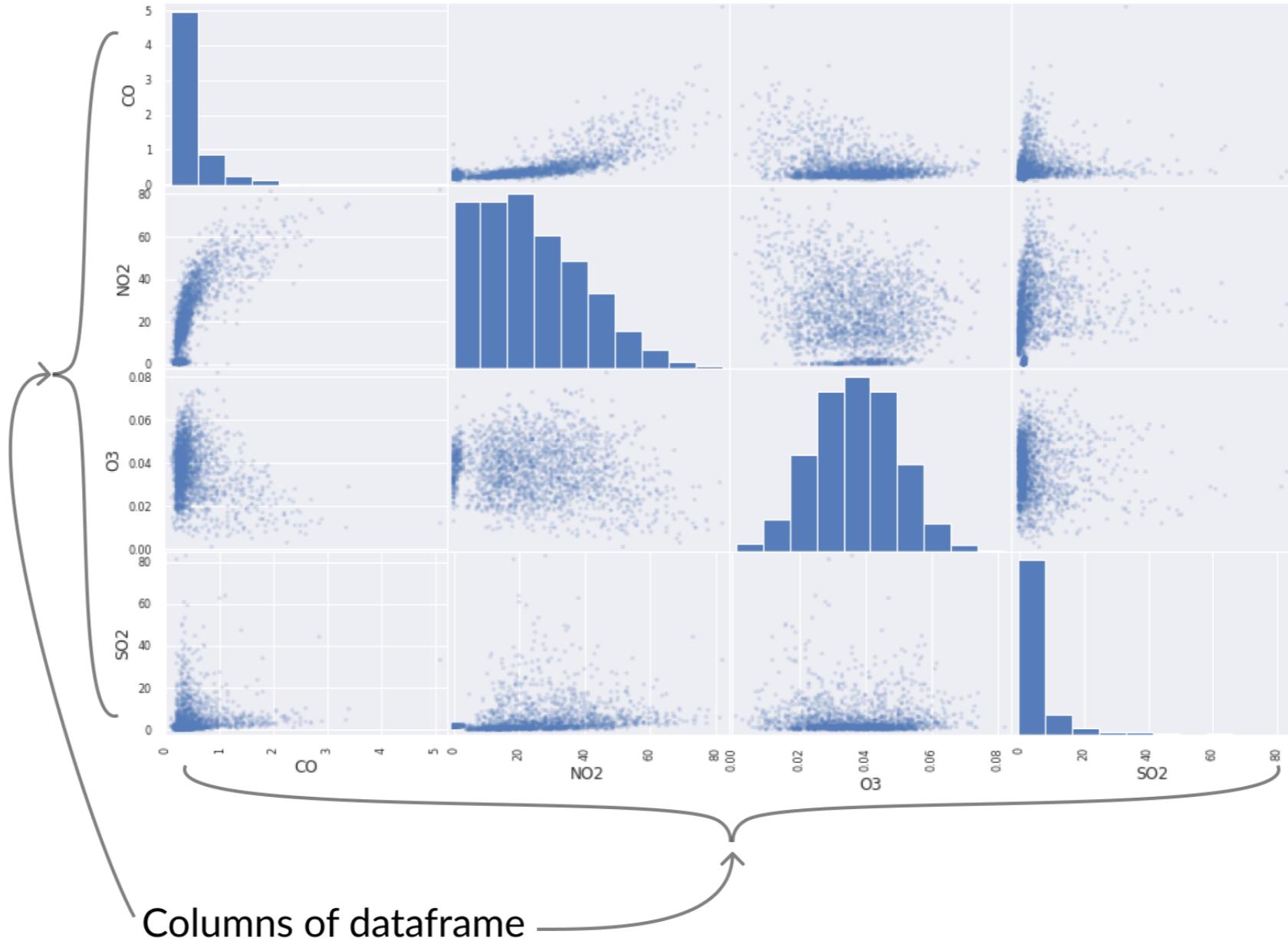
```
pollution.head()
```

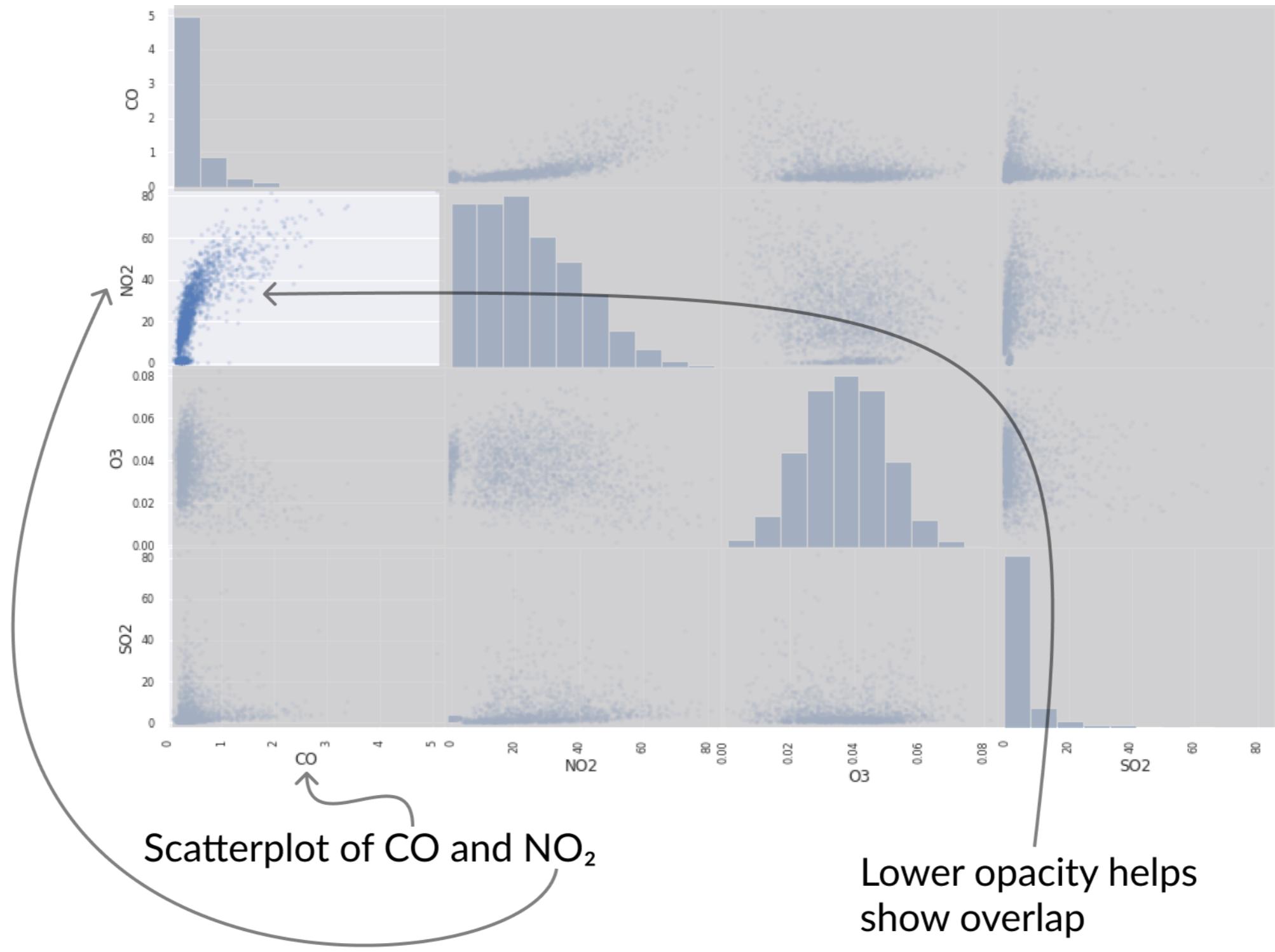
	city	year	month	day	CO	NO2	O3	SO2
0	Cincinnati	2012	1	1	0.245	20.0	0.030	4.20
1	Cincinnati	2012	1	2	0.185	9.0	0.025	6.35
2	Cincinnati	2012	1	3	0.335	31.0	0.025	4.25
3	Cincinnati	2012	1	4	0.305	25.0	0.016	17.15
4	Cincinnati	2012	1	5	0.345	21.0	0.016	11.05

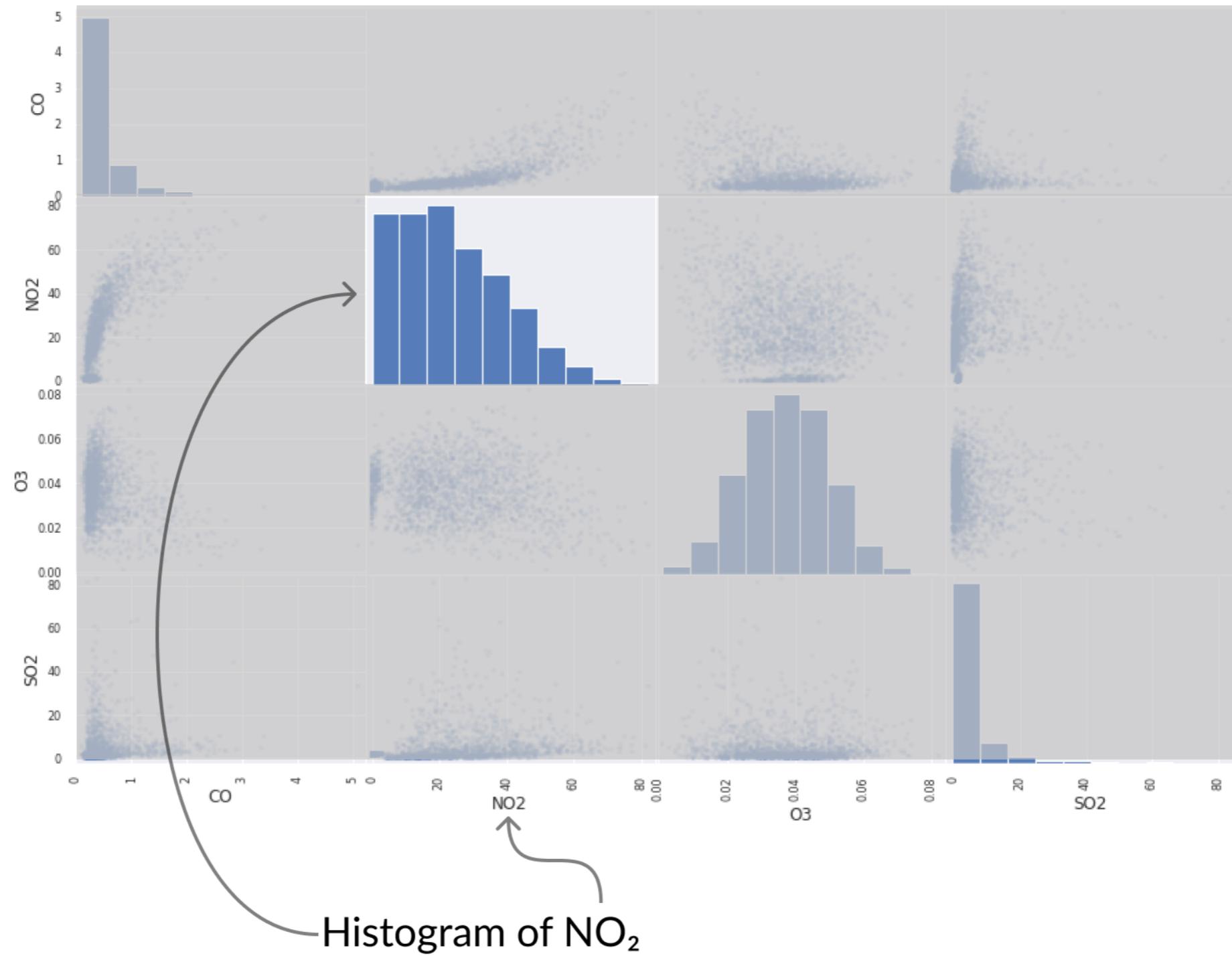
```
# Just show median  
pollution.describe(percentiles=[0.5]  
                    # Describe all columns  
                    include='all')
```

	city	year	month	day
count	8888	8888.000000	8888.000000	8888.000000
unique	8	NaN	NaN	NaN
top	Houston	NaN	NaN	NaN
freq	1433	NaN	NaN	NaN
mean	NaN	2013.621737	6.657516	187.187894
std	NaN	1.084081	3.328182	101.739060
min	NaN	2012.000000	1.000000	1.000000
50%	NaN	2014.000000	7.000000	192.000000
max	NaN	2015.000000	12.000000	366.000000

```
pd.plotting.scatter_matrix(pollution, alpha = 0.2);
```







markets.head()

The diagram illustrates the structure of the markets DataFrame with annotations:

- Location info**: Points to the columns `name`, `city`, `county`, and `state`.
- Duration of market**: Points to the columns `lat`, `lon`, and `months_open`.
- Goods sold at market**: Points to the columns `Bakedgoods`, `...`, `num_items_sold`, and `state_pop`.
- Market's state population**: Points to the column `state_pop`.

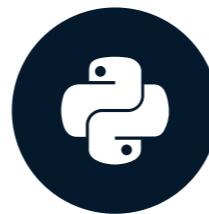
	name	city	county	state	lat	lon	months_open	Bakedgoods	...	num_items_sold	state_pop
	Island Market	Key Largo	Monroe	Florida	-80.427218	25.109214	6	1	...	18	19893297.0
	COFFO Harvest Farmers' Market	Florida City	Miami-Dade	Florida	-80.482299	25.449850	12	0	...	7	19893297.0
	COFFO Harvest Farmers' Market	Homestead	Miami-Dade	Florida	-80.483400	25.463500	12	0	...	7	19893297.0
	Verde Gardens Farmers Market	Homestead	Miami-Dade	Florida	-80.395607	25.506727	12	0	...	5	19893297.0
	Verde Community Farm and Market	Homestead	Miami-Dade	Florida	-80.395607	25.506727	9	0	...	5	19893297.0

Let's explore our data

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

Exploring the patterns

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON



Nick Strayer
Instructor

Digging in deeper

- Investigating correlations
- Are correlations driven by confounding?
- Anything surprising?

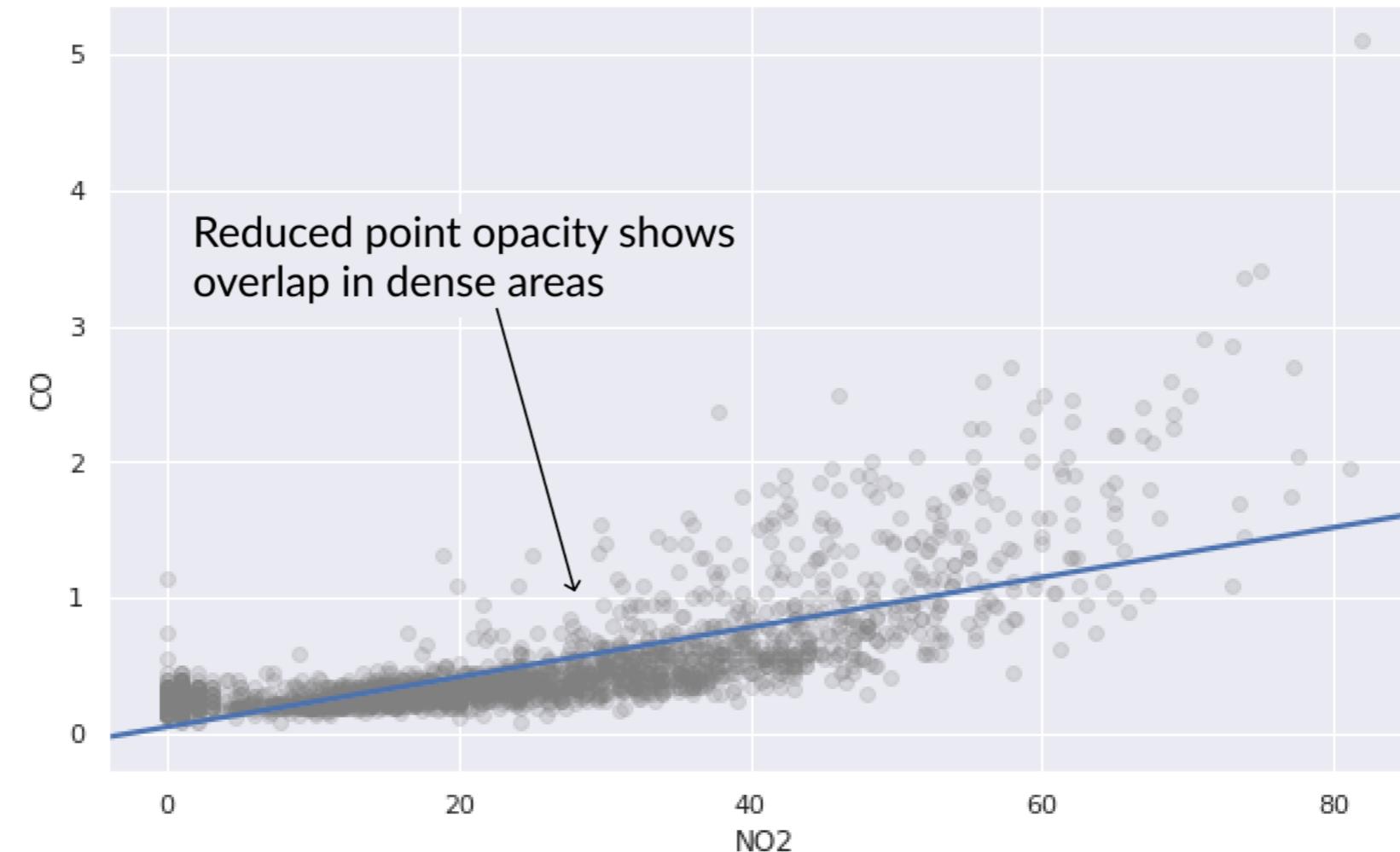


Target audiences

- Shared with peers
- Be smart about design decisions
- Remember they aren't as familiar with data

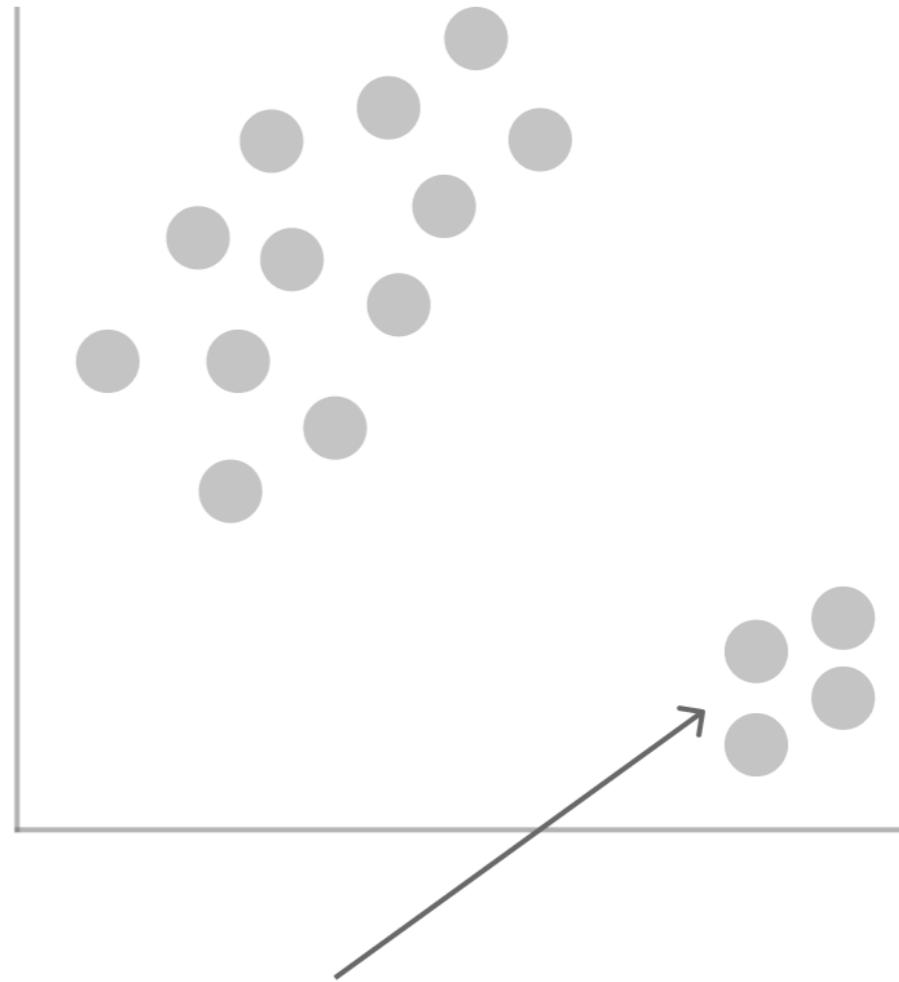


```
sns.regplot('NO2', 'CO', ci=False, data=pollution,  
            # Lower opacity of points  
            scatter_kws={'alpha':0.2, 'color':'grey'} )
```



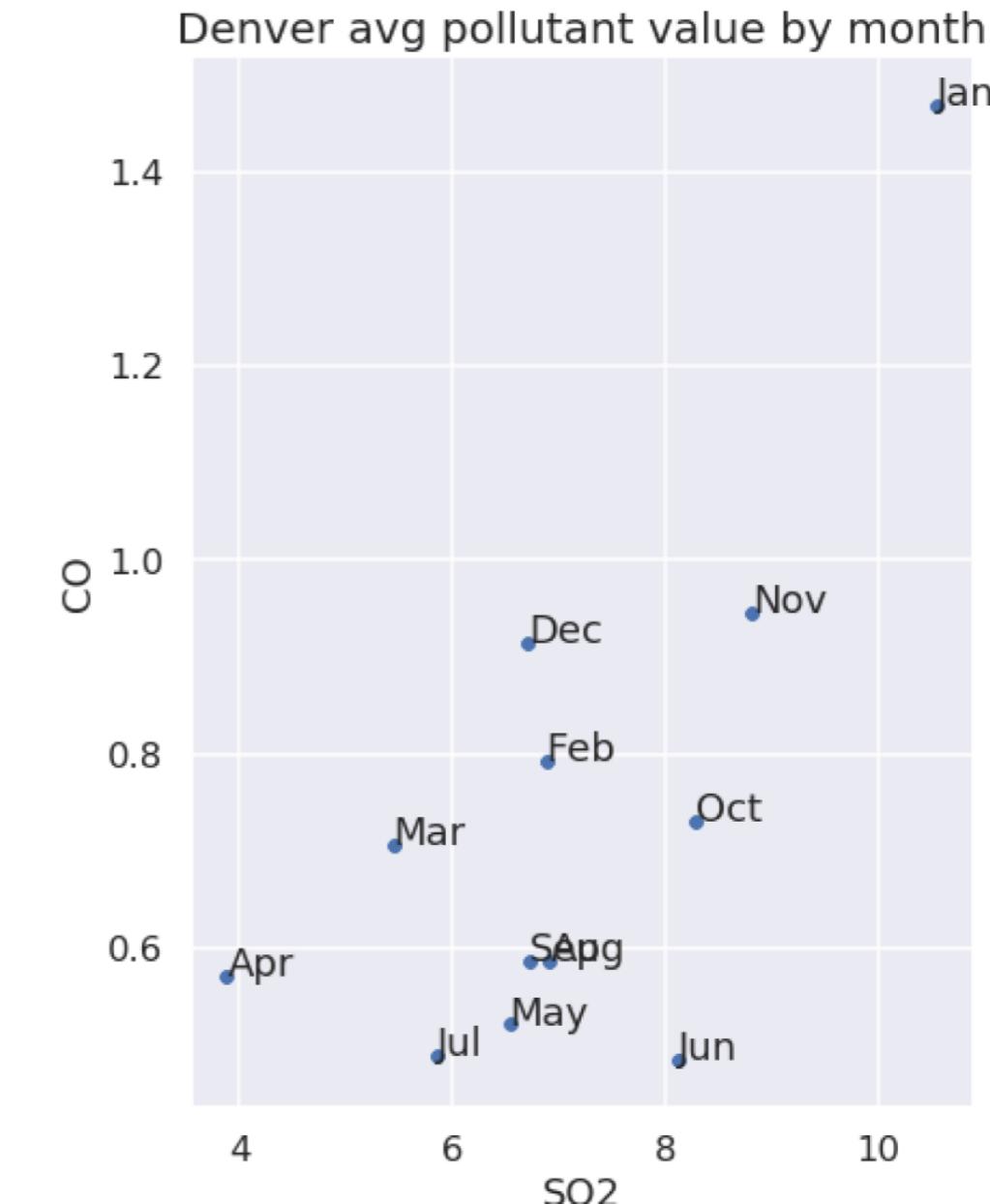
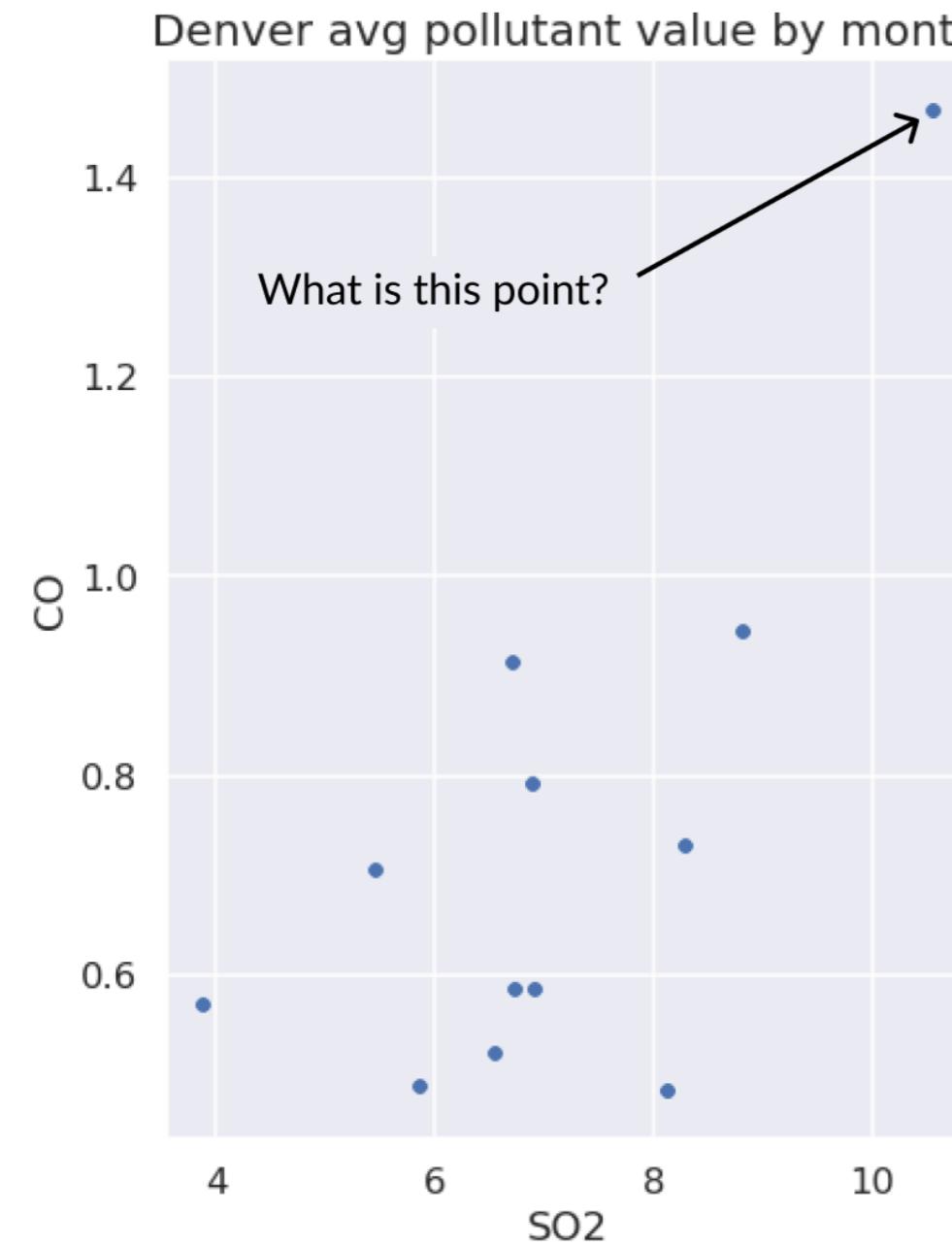
Profiling patterns

- Found interesting pattern in data
- How to quickly explore and explain the pattern?
- Use text!



What do these points have in common?

Using text scatters to id outliers

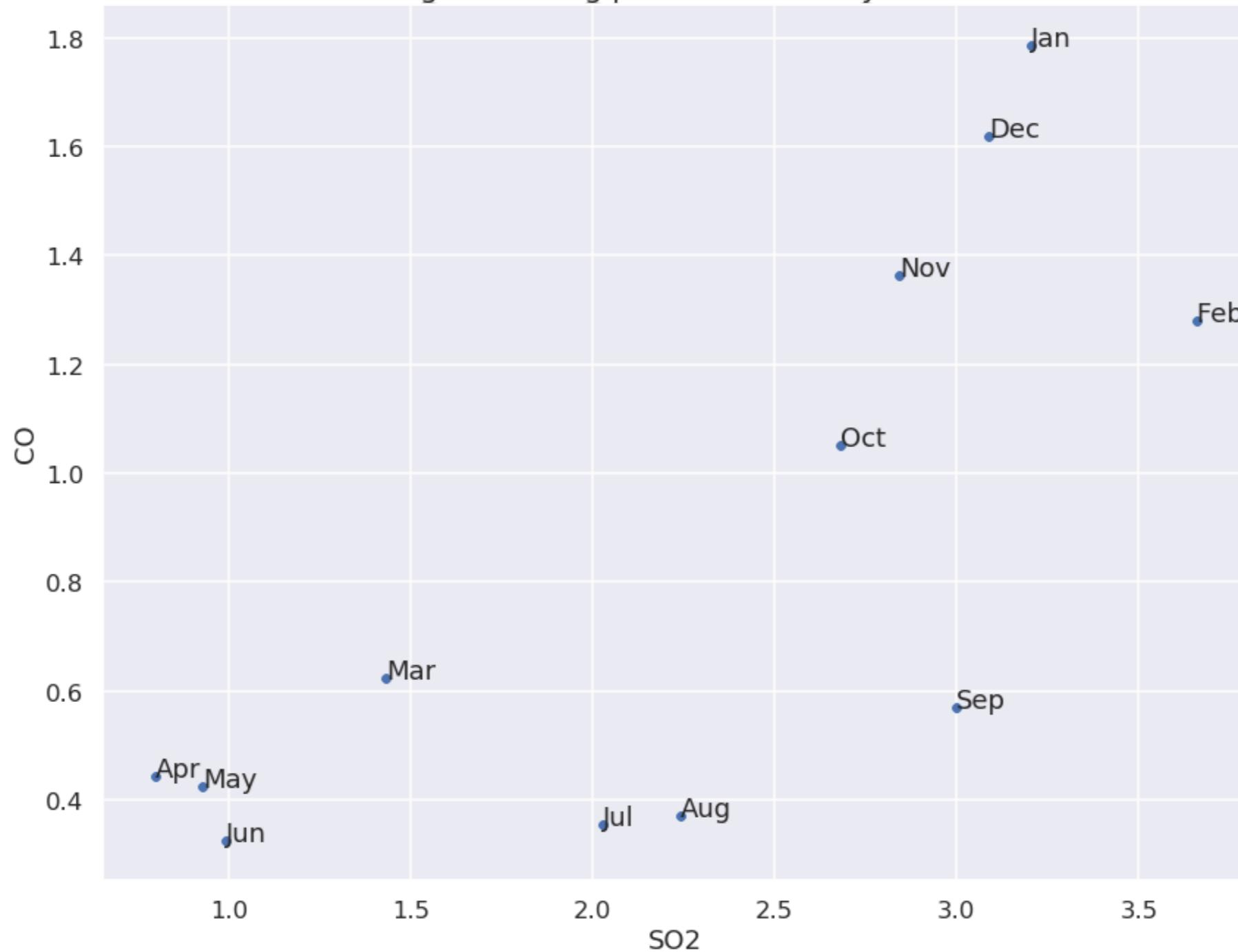


```
g = sns.scatterplot("S02", "CO", data=long_beach_avgs)

# Iterate over the rows of our data
for _, row in long_beach_avgs.iterrows():
    # Unpack columns from row
    month, S02, CO = row
    # Draw annotation in correct place
    g.annotate(month, (S02, CO))

plt.title('Long Beach avg S02 by CO')
```

Long Beach avg pollutant value by month



Let's dig in

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

Making your visualizations efficient

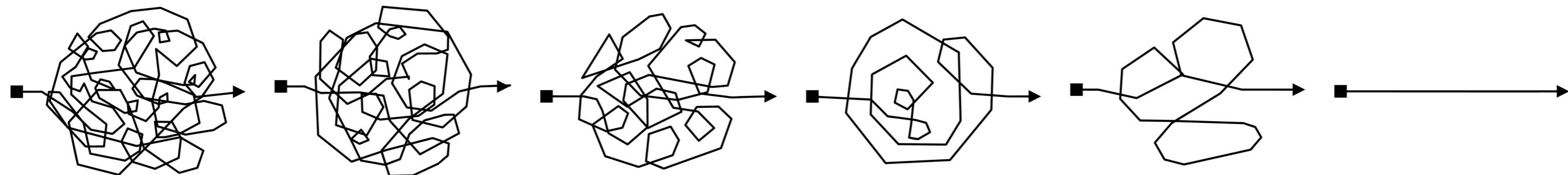
IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON



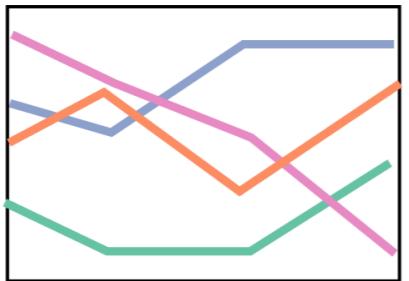
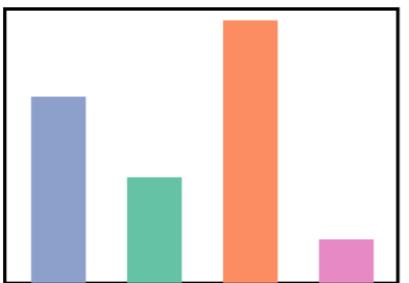
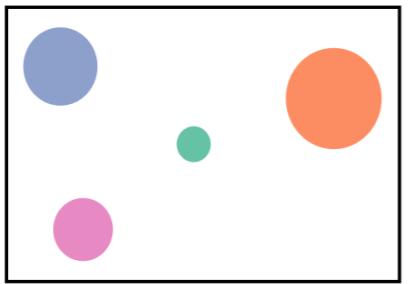
Nick Strayer
Instructor

What is efficient?

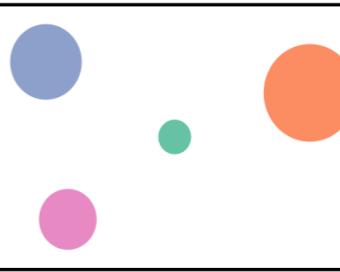
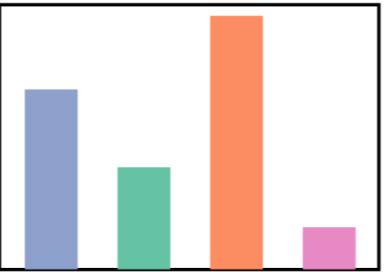
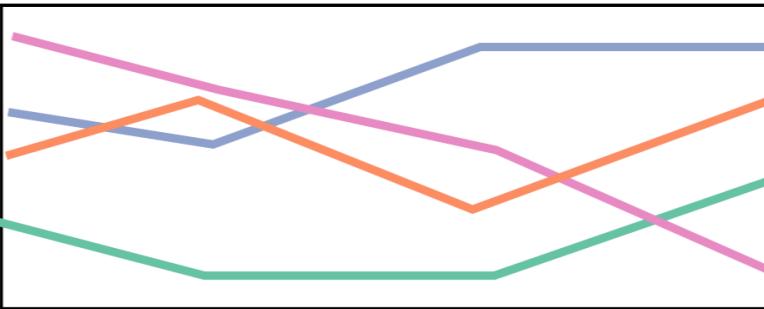
- Reduce the effort needed to see story
- Re-organize plots to keep focus
- Improve 'ink' to info ratio
- Don't compromise the message



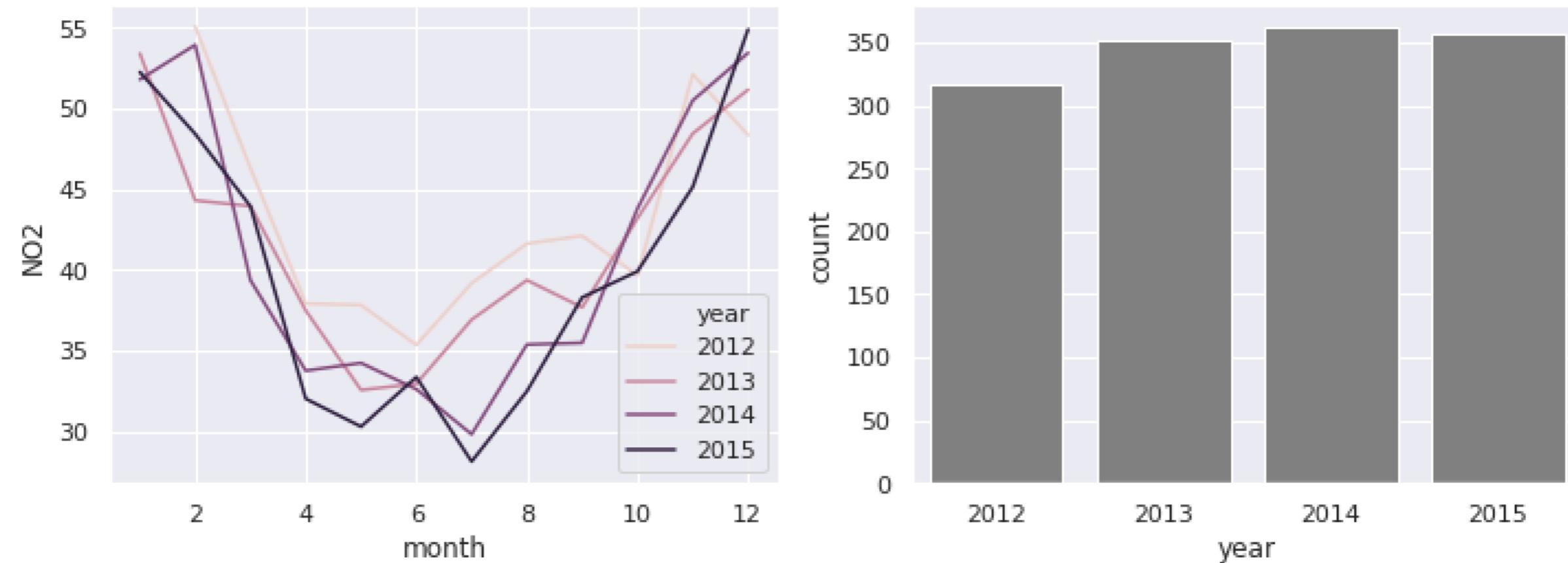
Combining plots



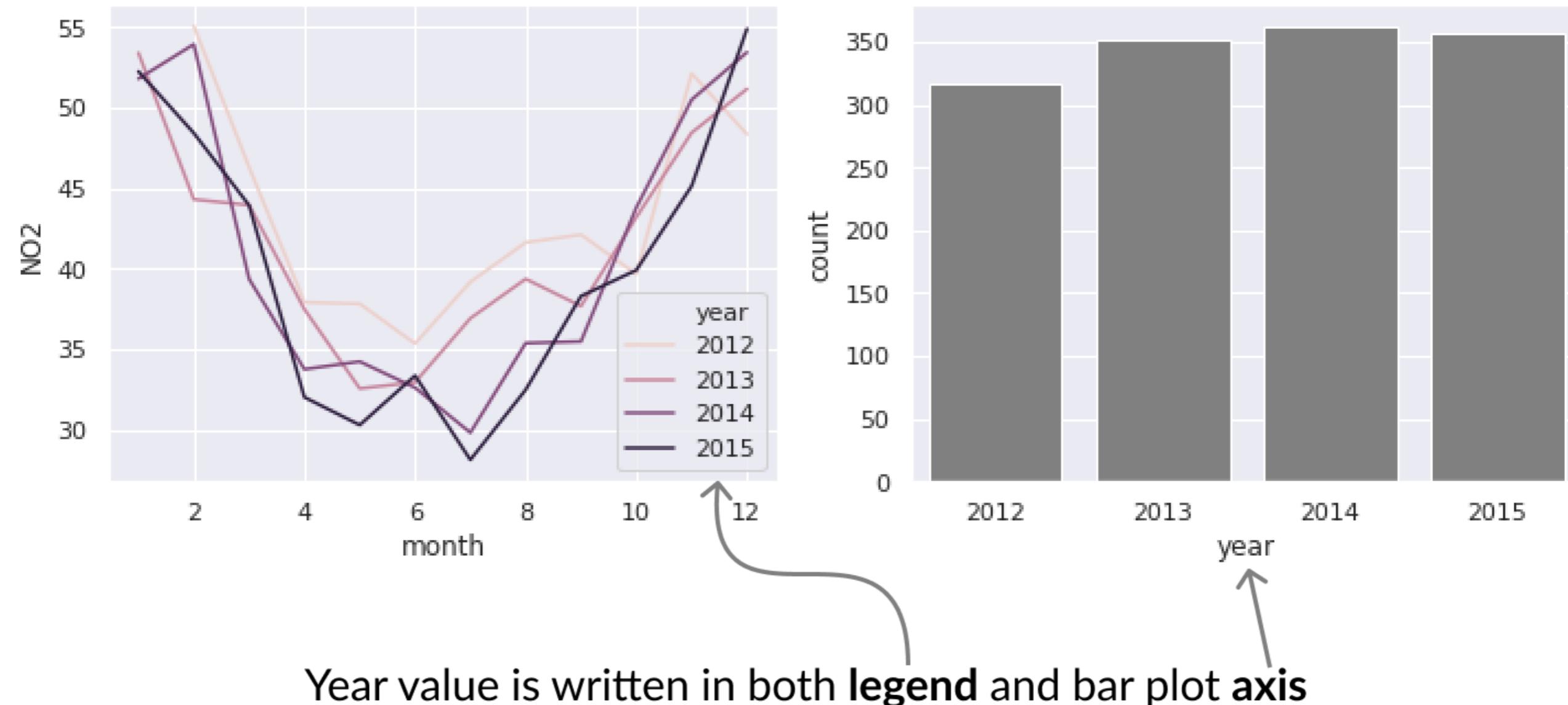
Allows viewer to keep context in mind without jumping around page



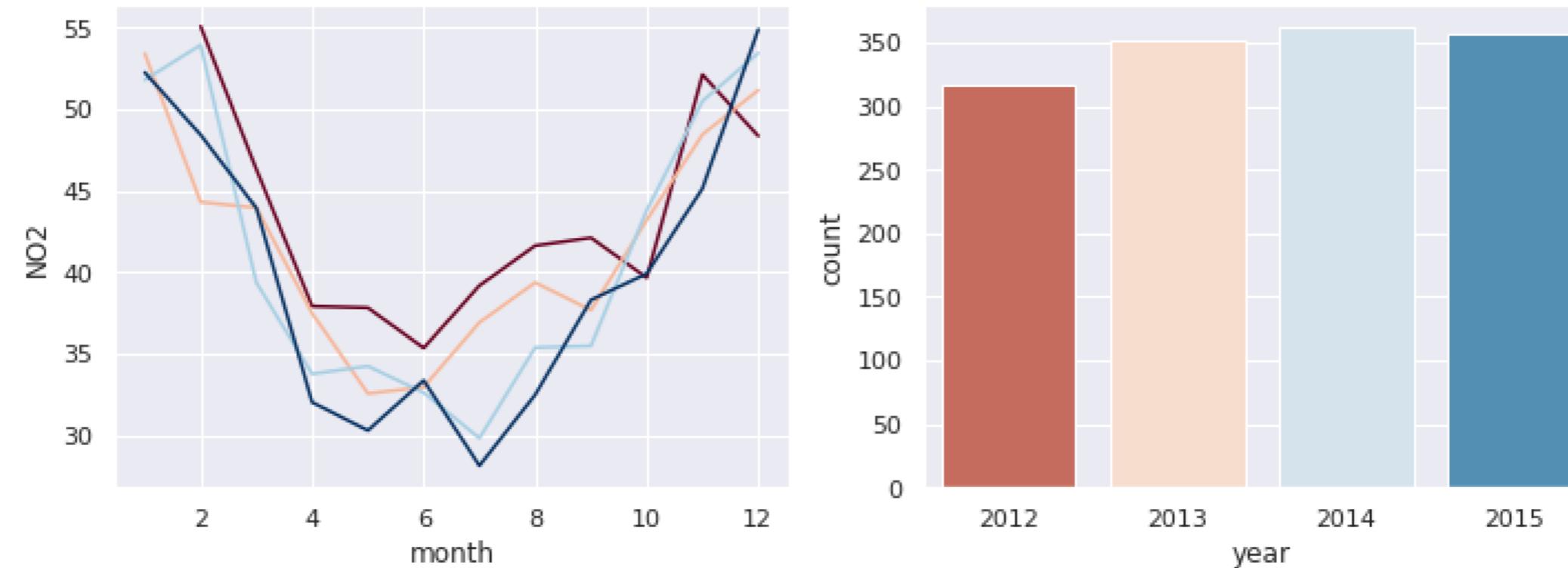
```
# Create a subplot w/ one row & two columns.  
f, (ax1, ax2) = plt.subplots(1, 2)  
# Pass each axes to respective plot  
sns.lineplot('month', 'NO2', 'year', ax=ax1, data=pol_by_month)  
sns.barplot('year', 'count', ax=ax2, data=obs_by_year)
```



Clear unnecessary legends



```
sns.lineplot('month', 'N02', 'year', ax=ax1, data=pol_by_month, palette='RdBu',)  
sns.barplot('year', 'count', 'year', ax=ax2, data=obs_by_year,  
            palette='RdBu', dodge=False)  
  
# Remove legends for both plots  
ax1.legend_.remove()  
ax2.legend_.remove()
```



Let's practice

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

Tweaking your plots

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

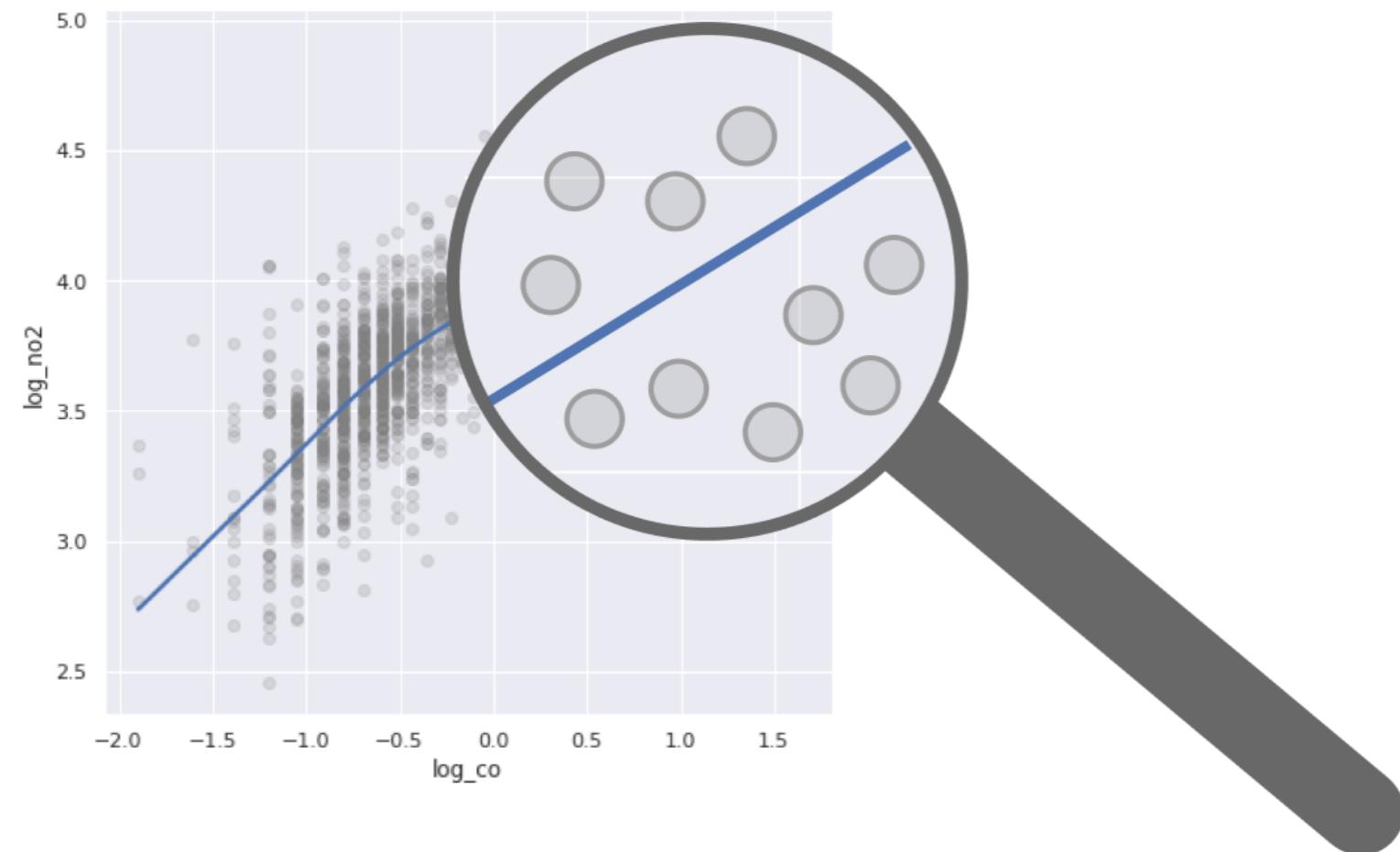


Nick Strayer

Instructor

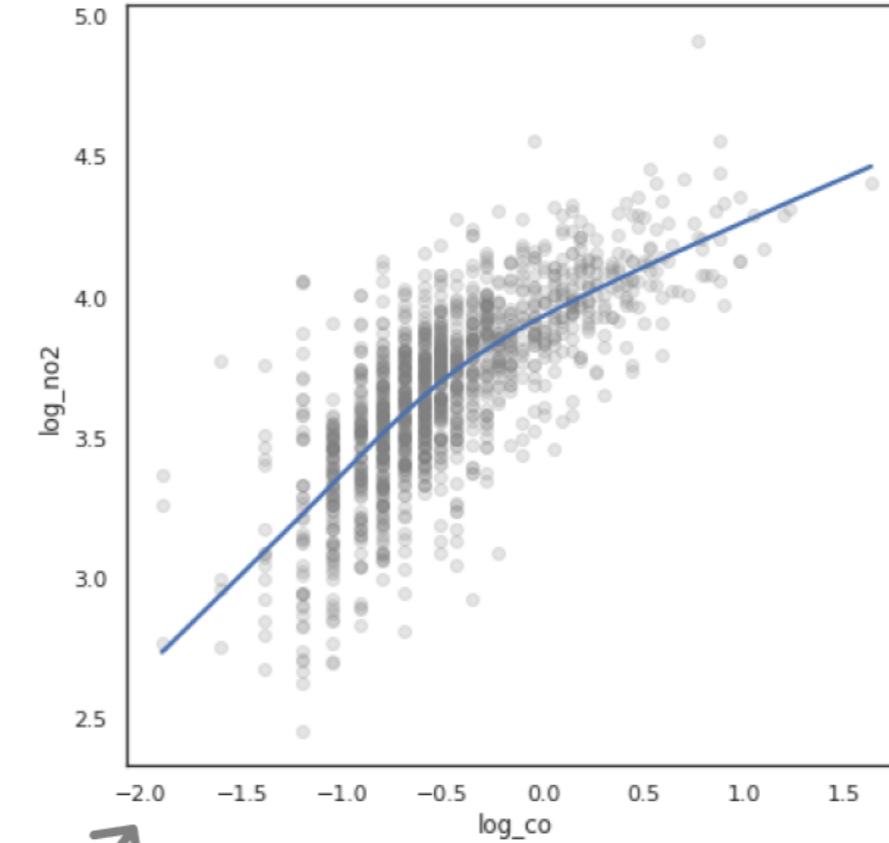
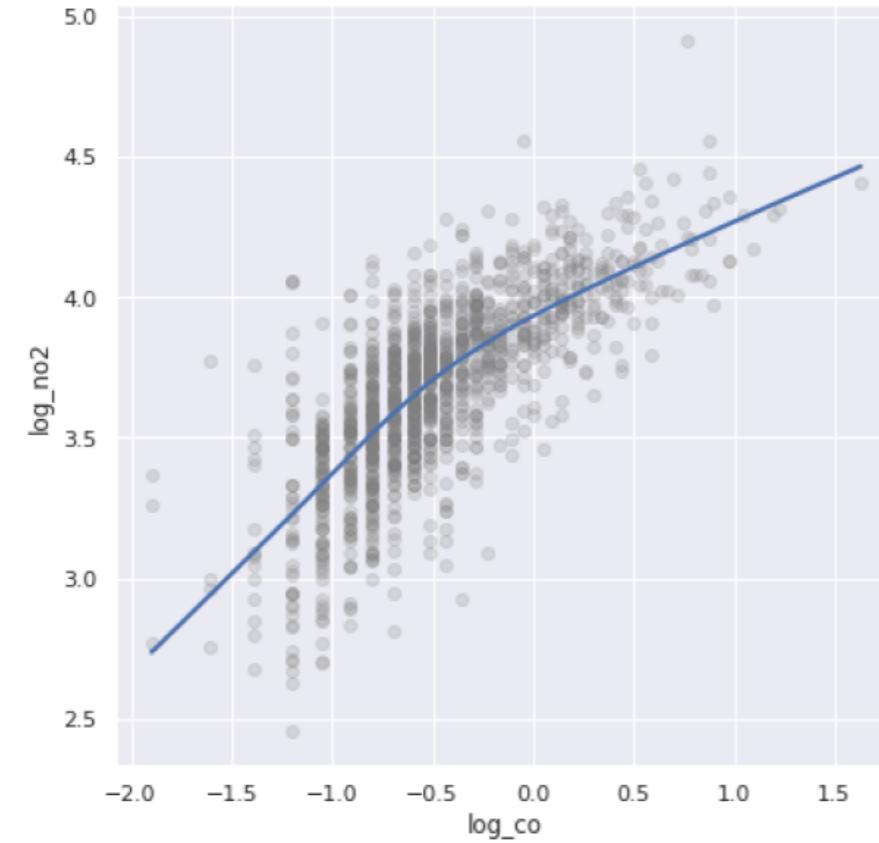
Looking at the small things

- Put yourself into the viewer's shoes



Is the aesthetic appropriate?

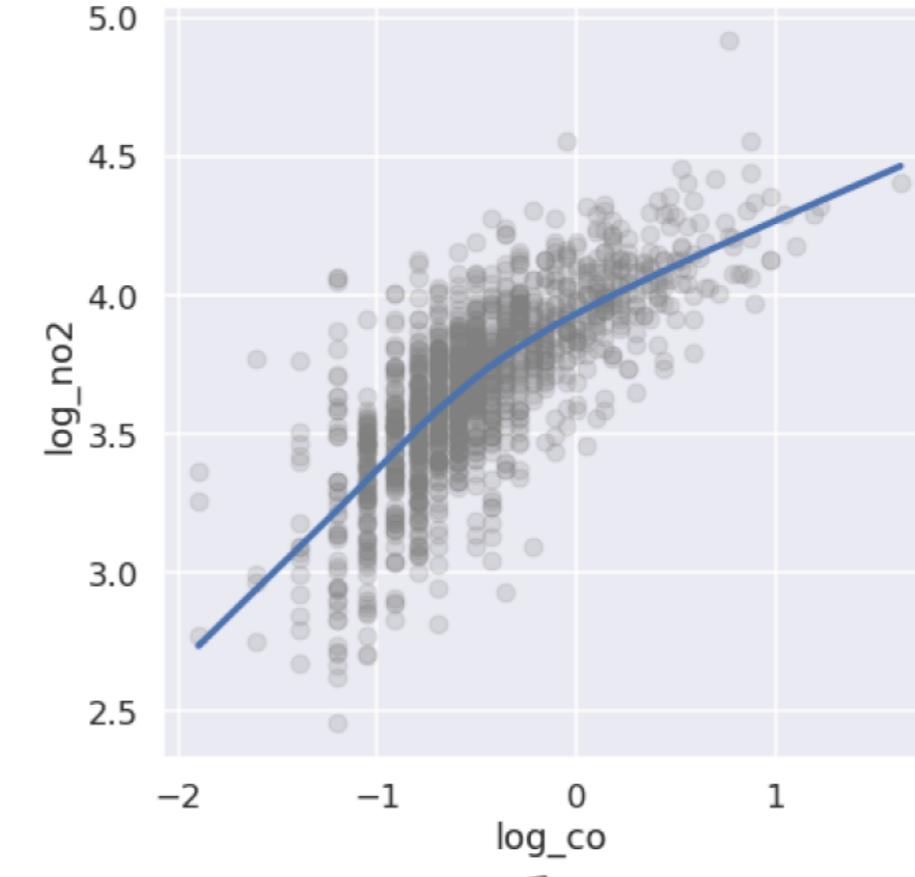
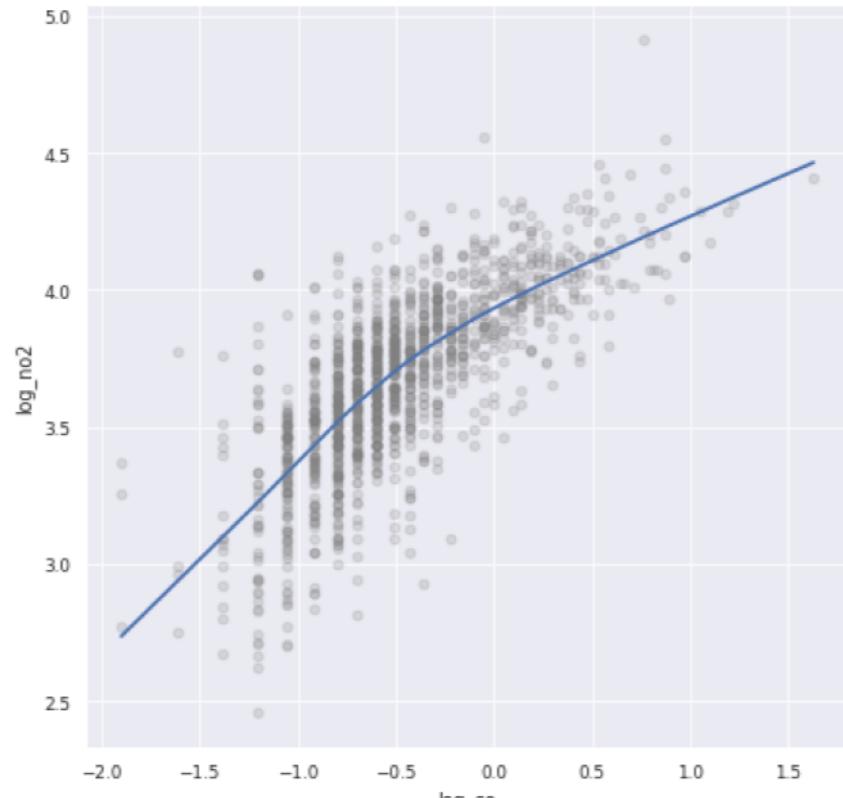
- Is the aesthetic appropriate for the context?



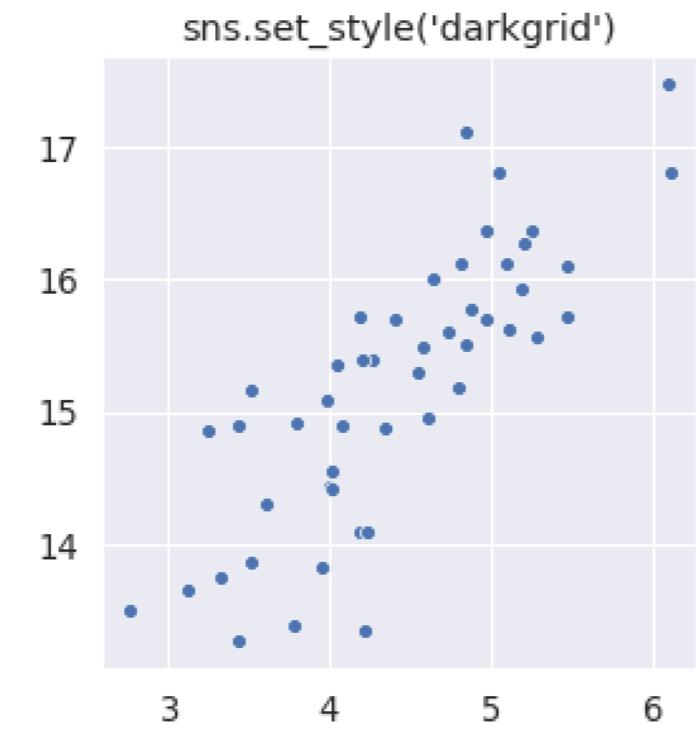
Grid not allowed?

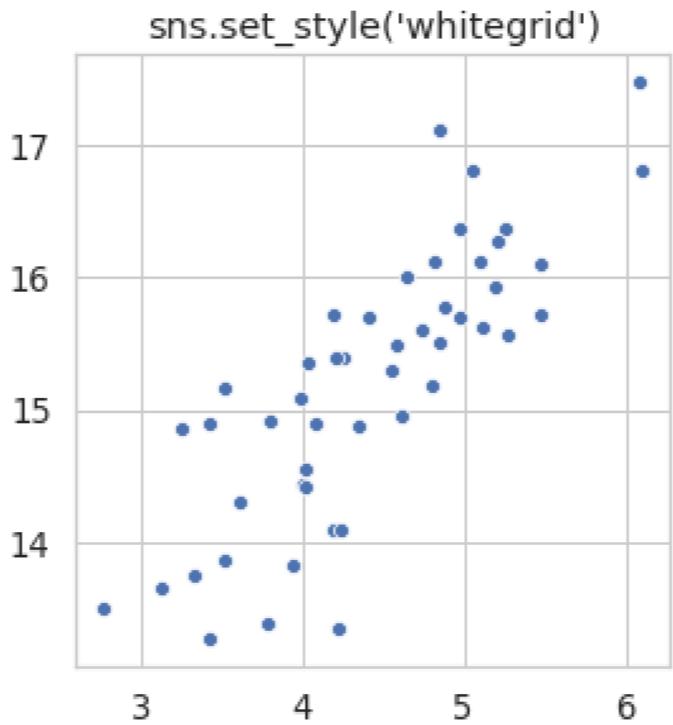
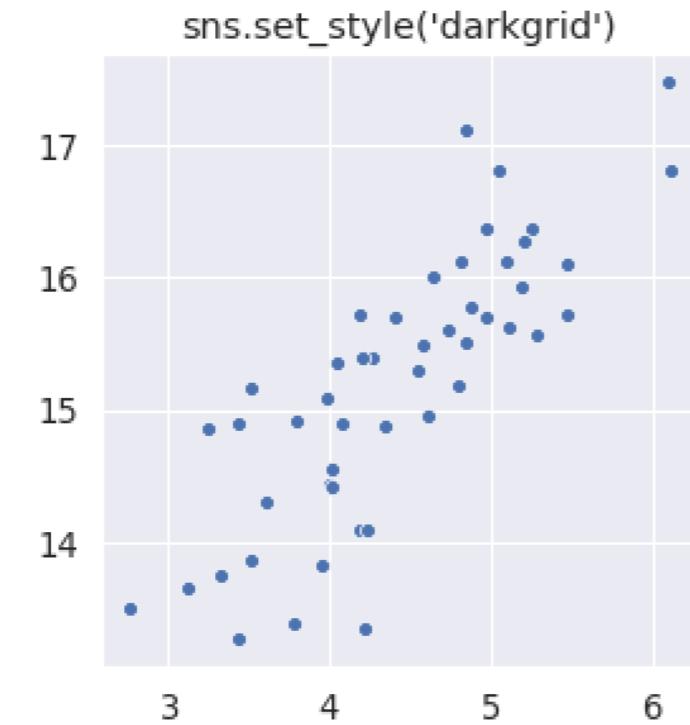
Font-sizes

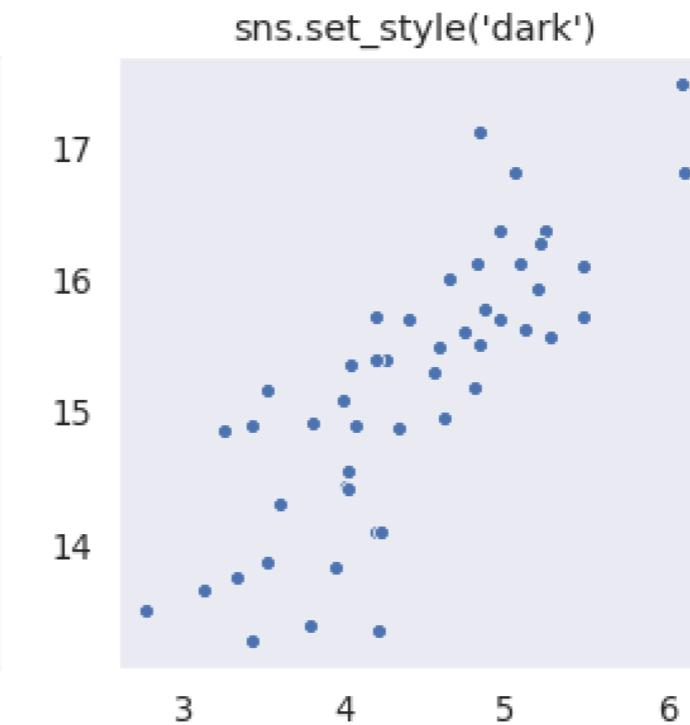
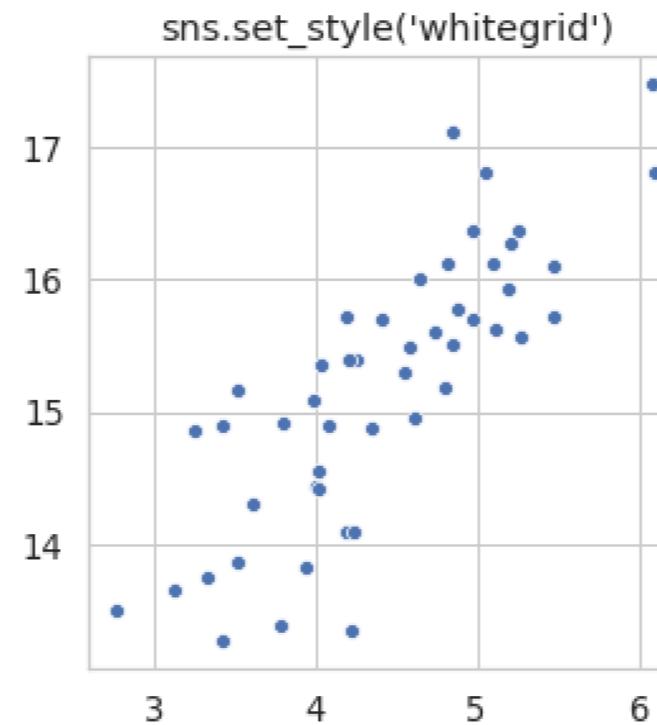
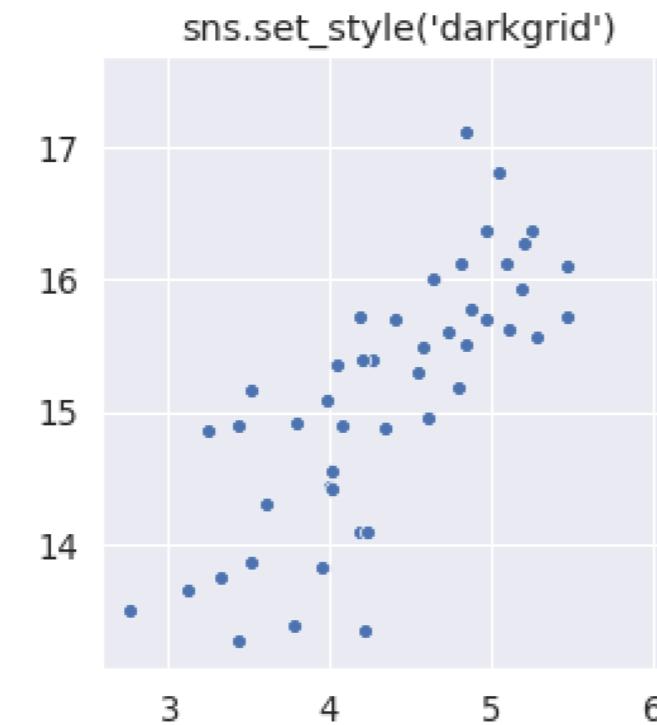
- Is everything legible?

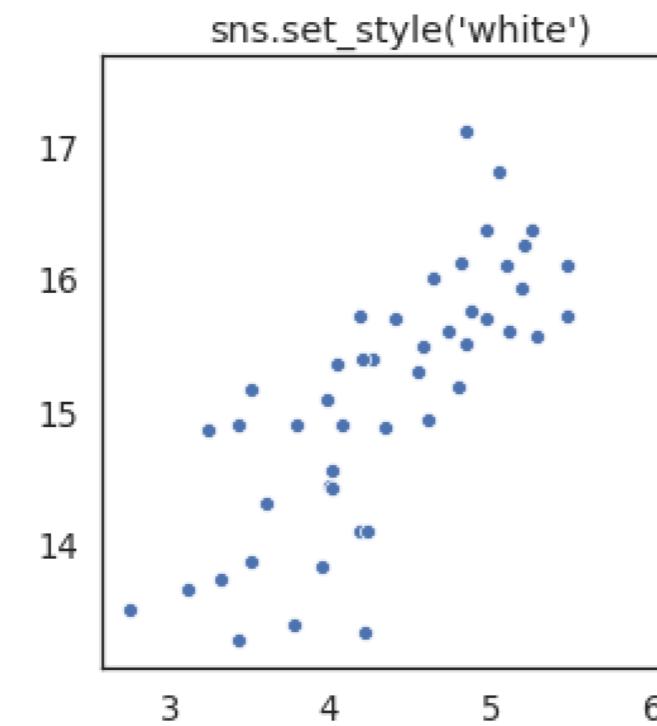
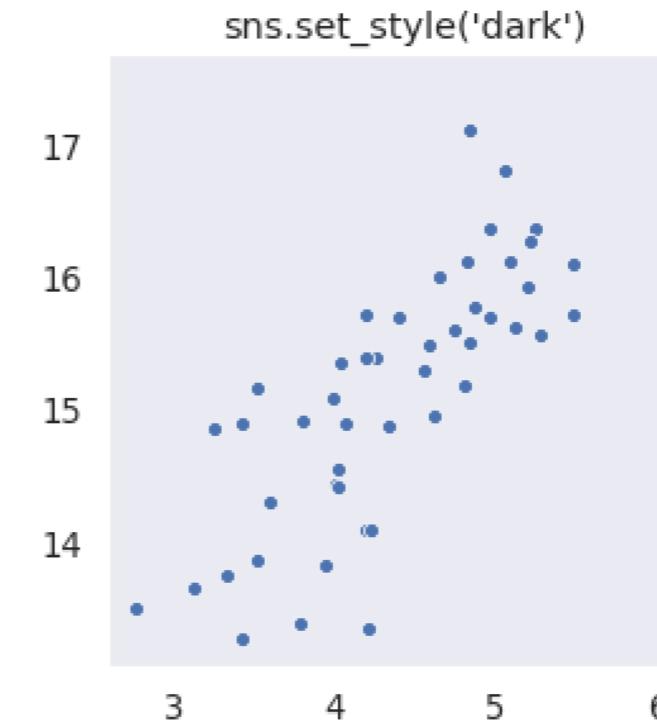
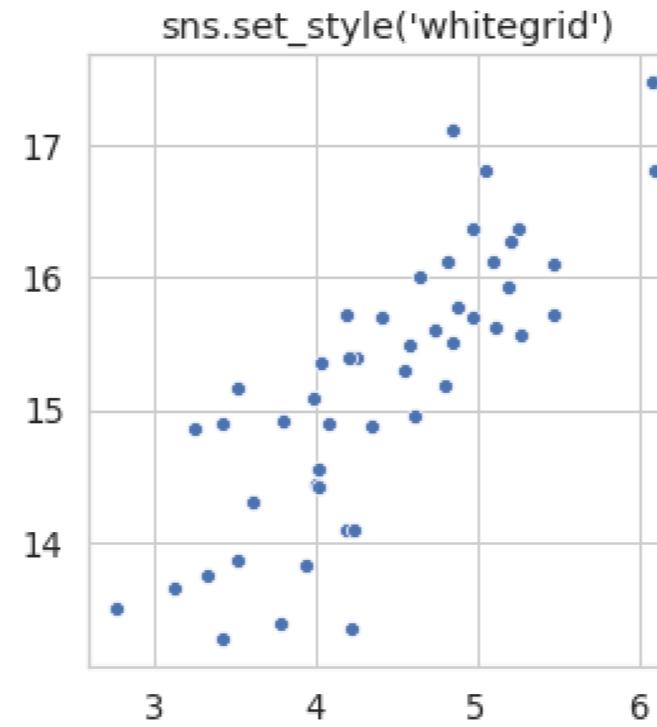
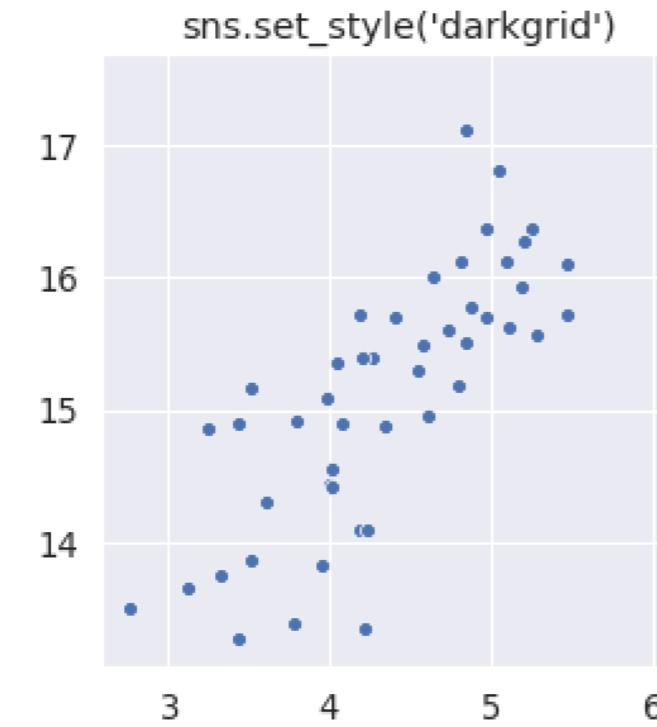


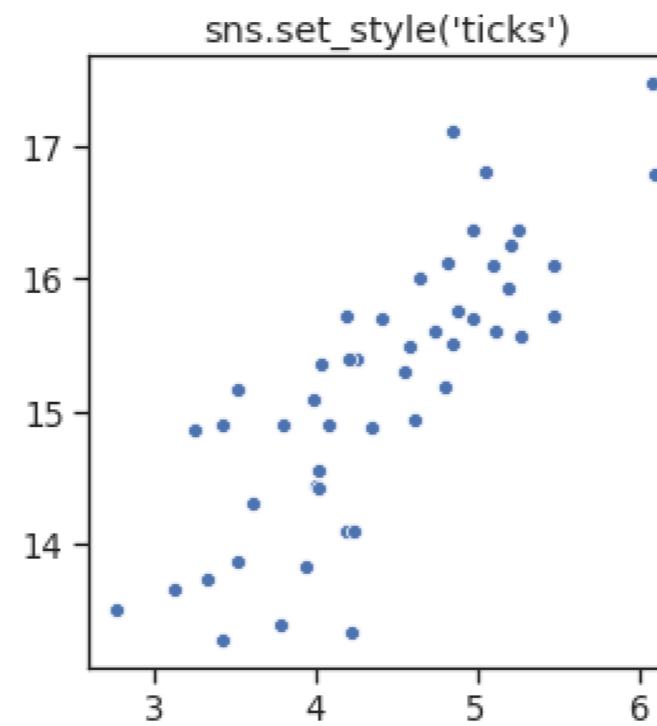
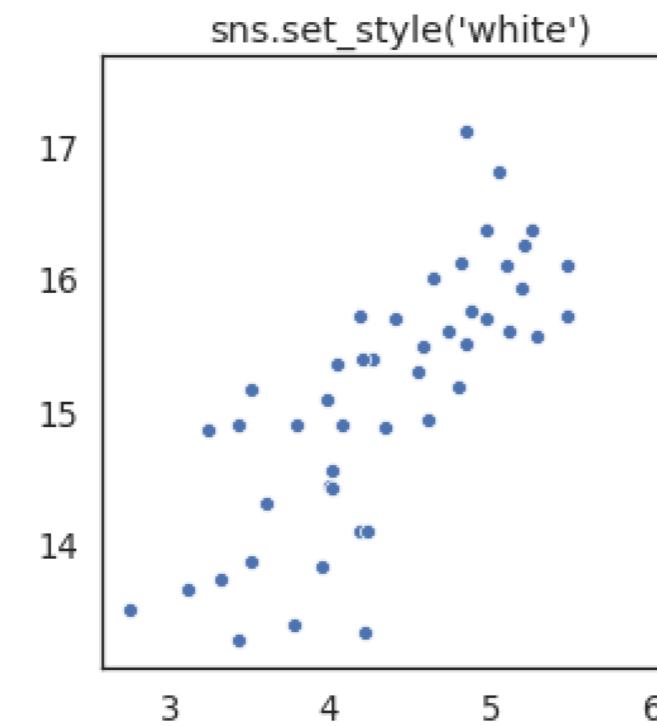
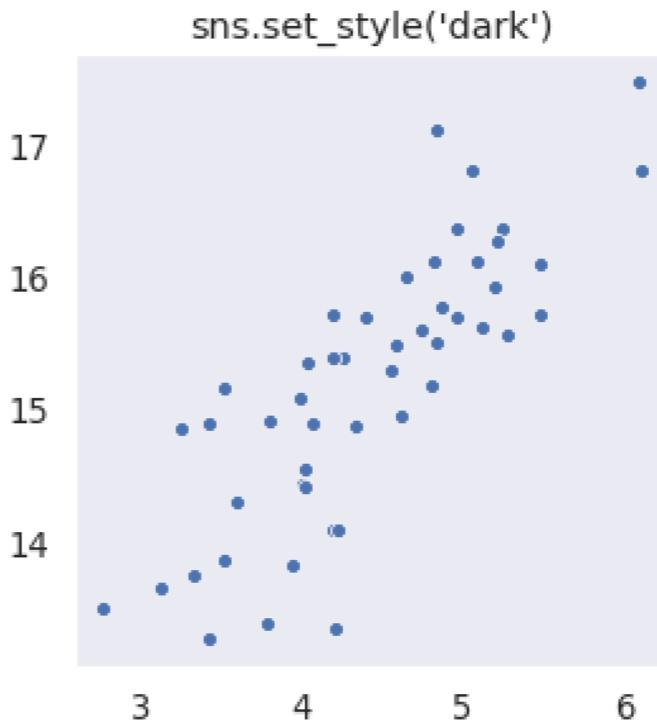
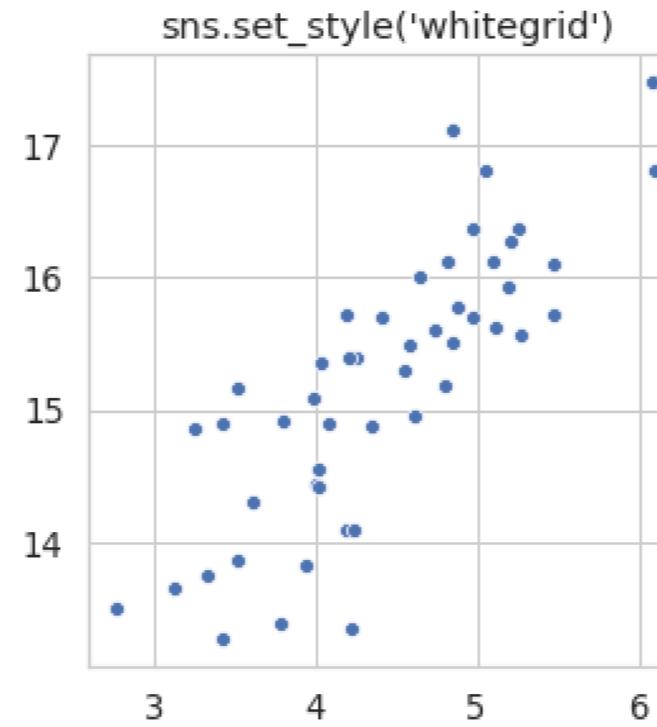
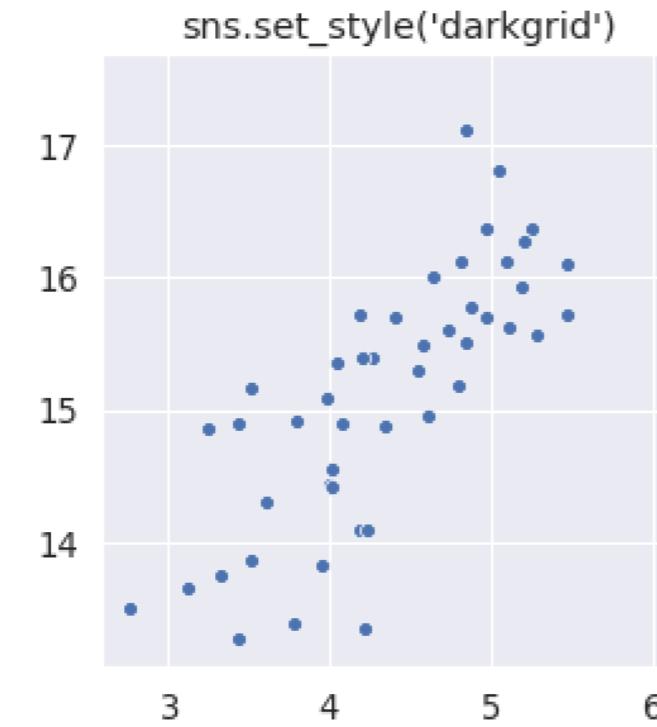
Text too small?



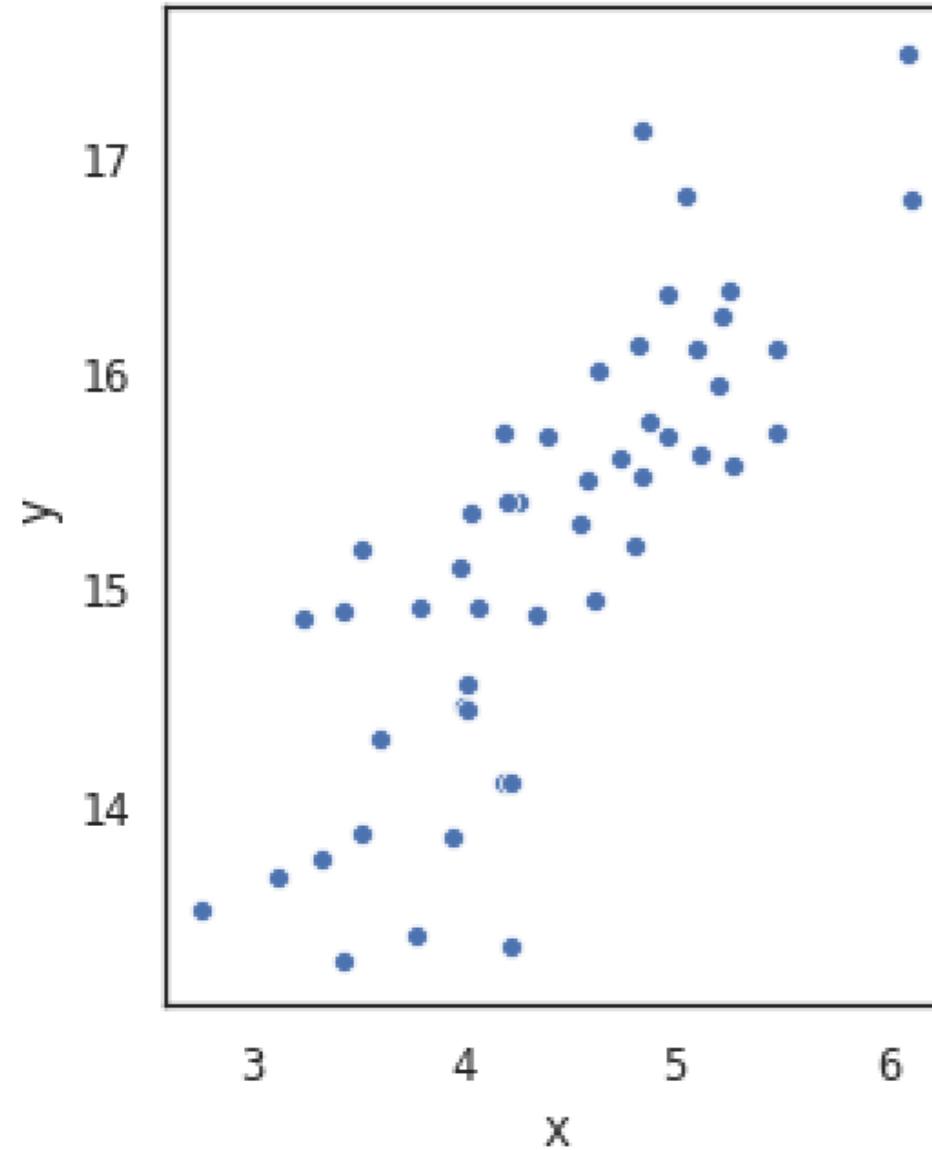




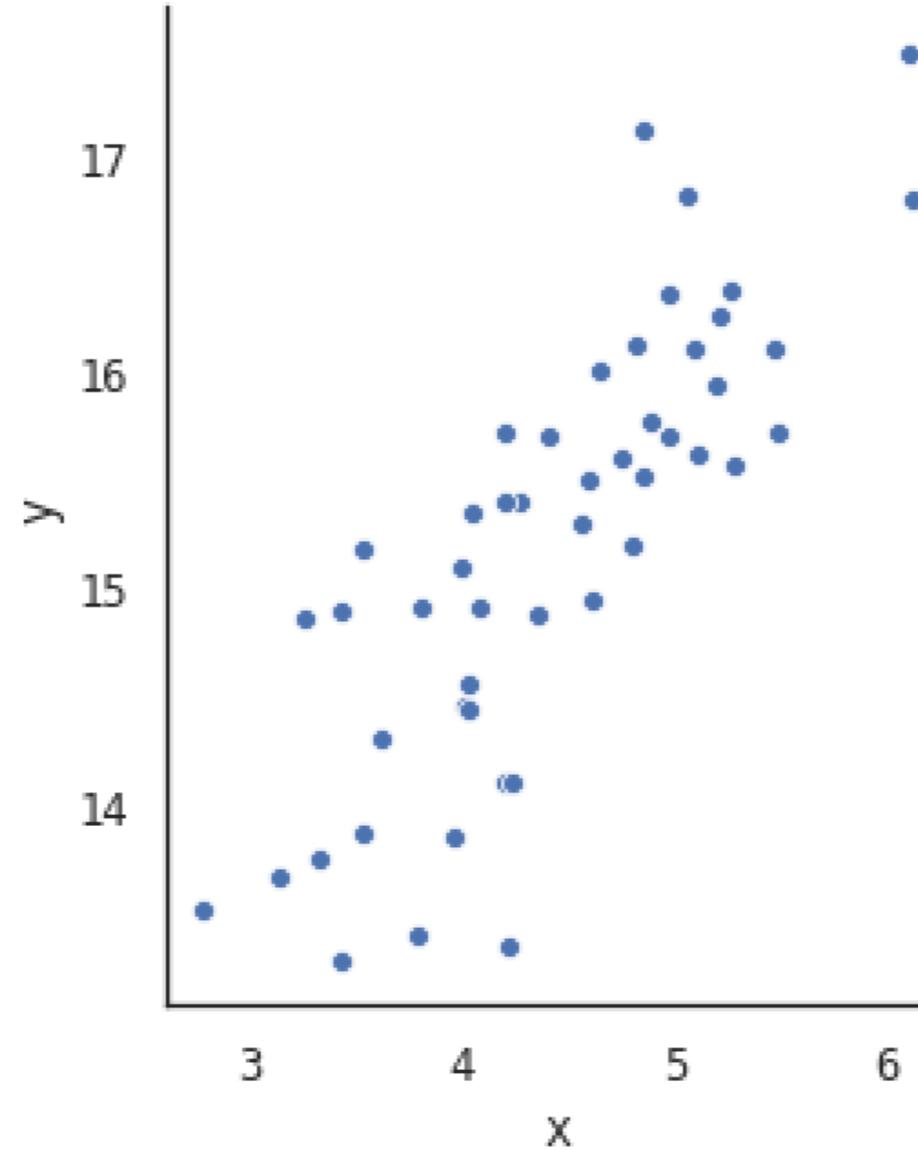




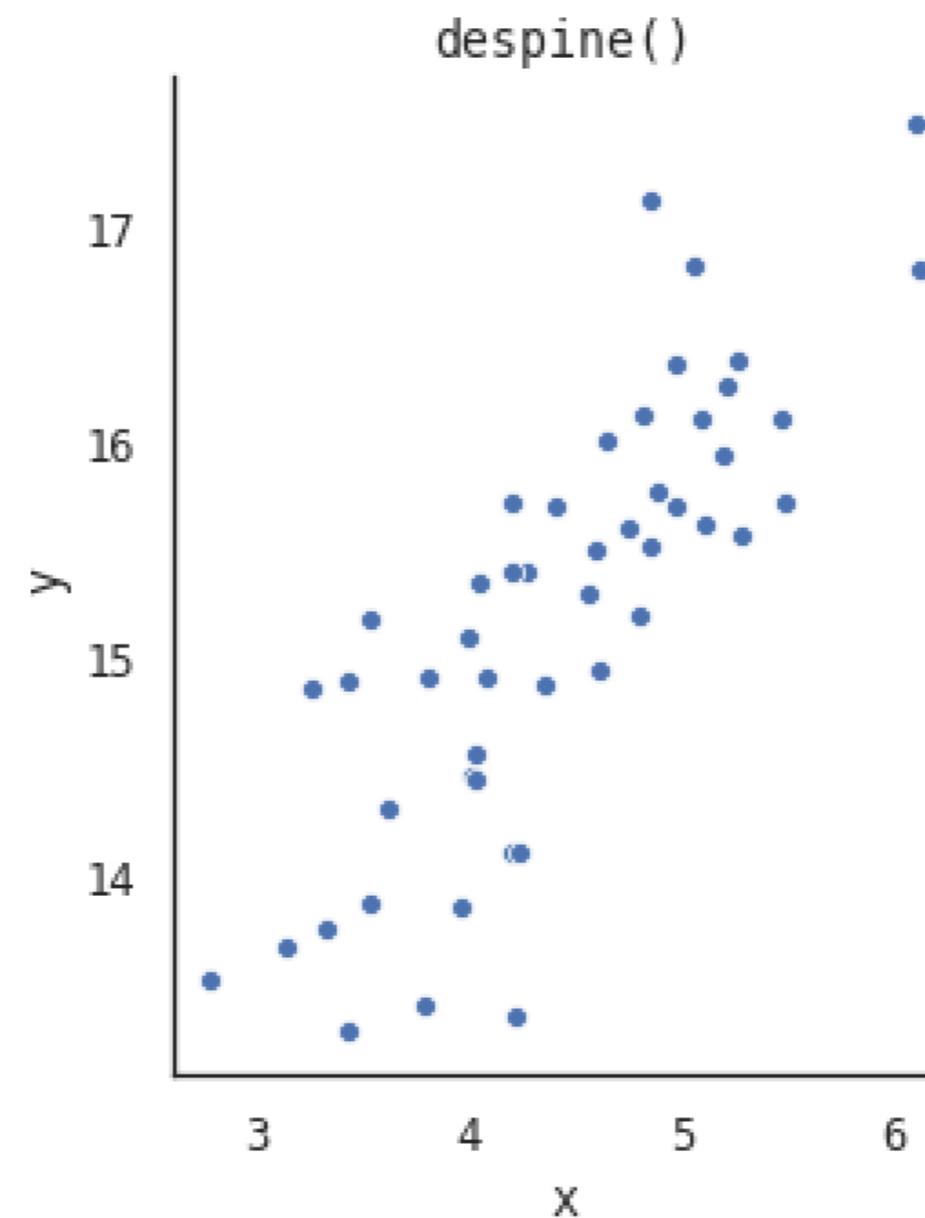
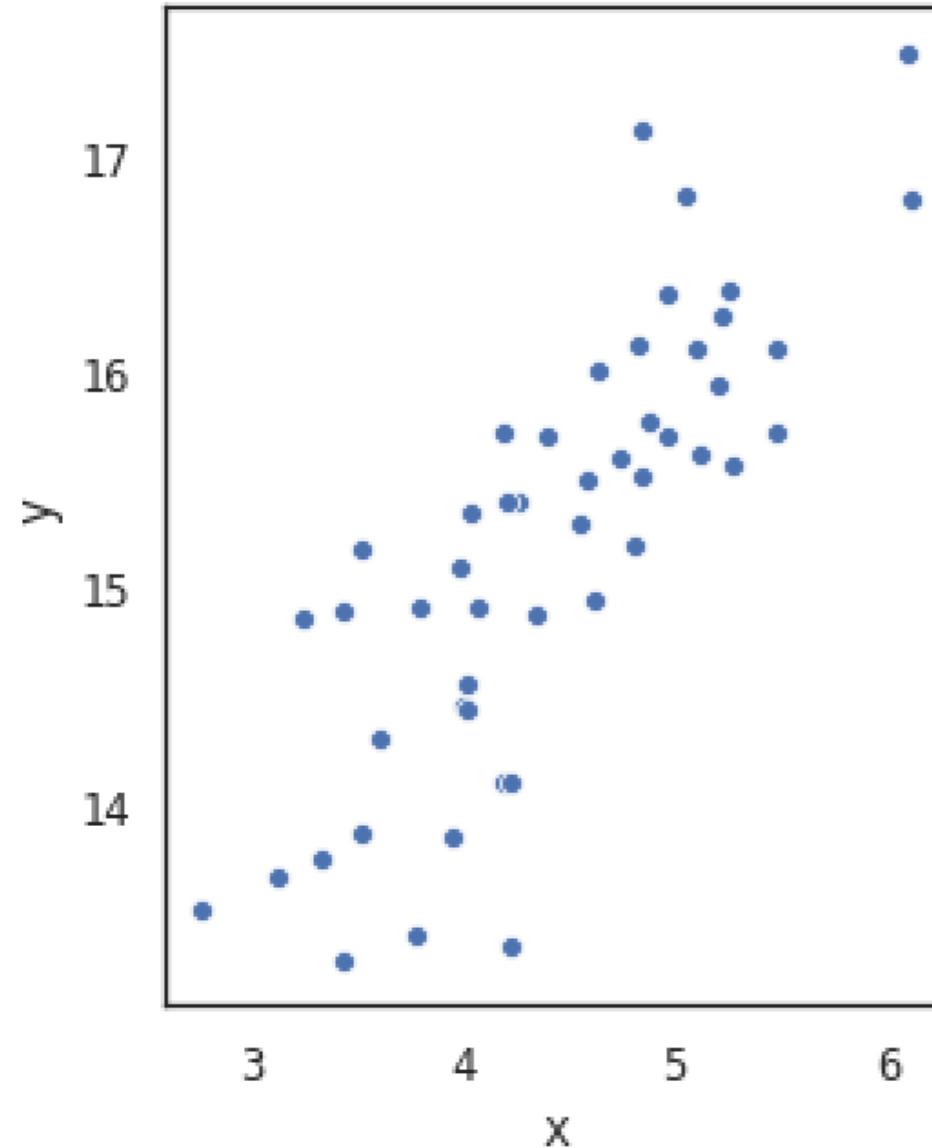
Removing spines from plots



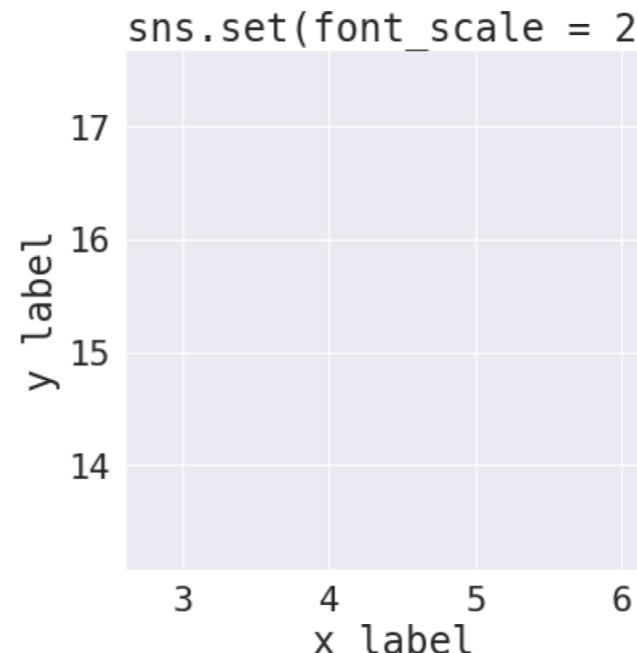
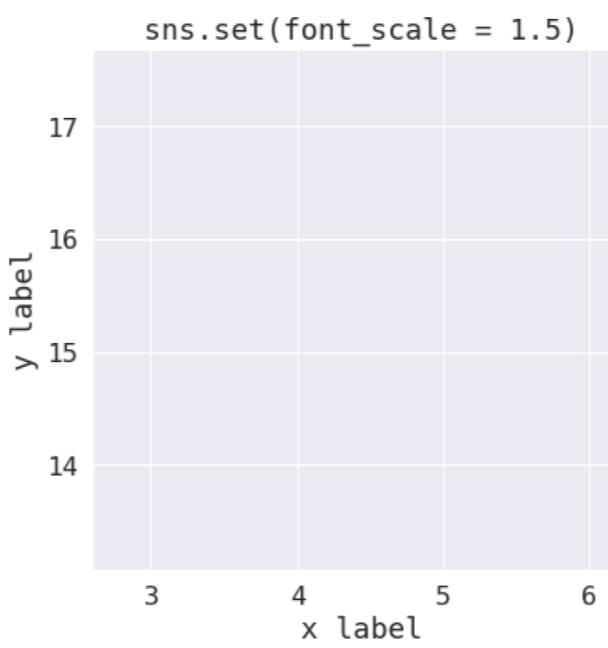
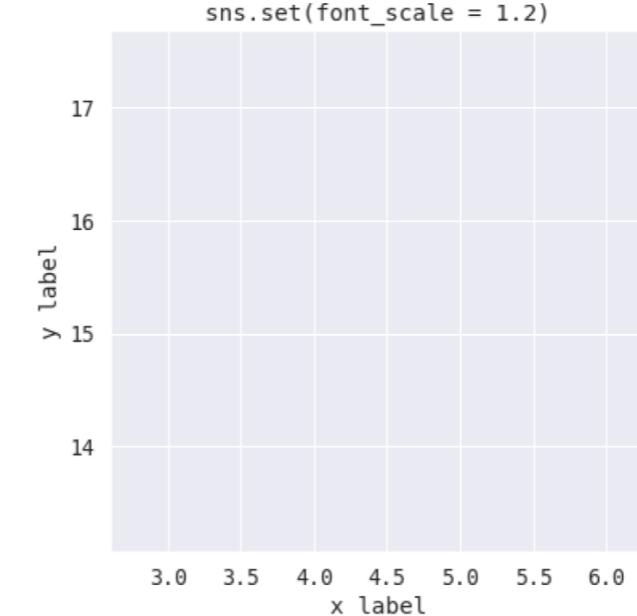
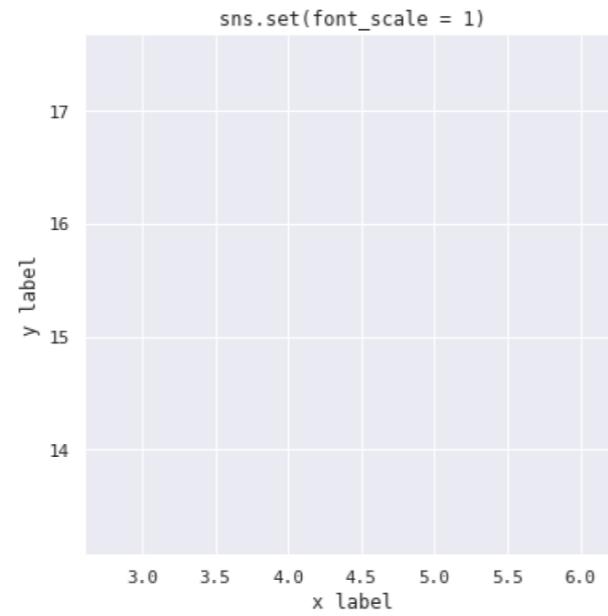
`despine()`



Removing spines from plots



Setting font-size



Let's tweak some plots

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

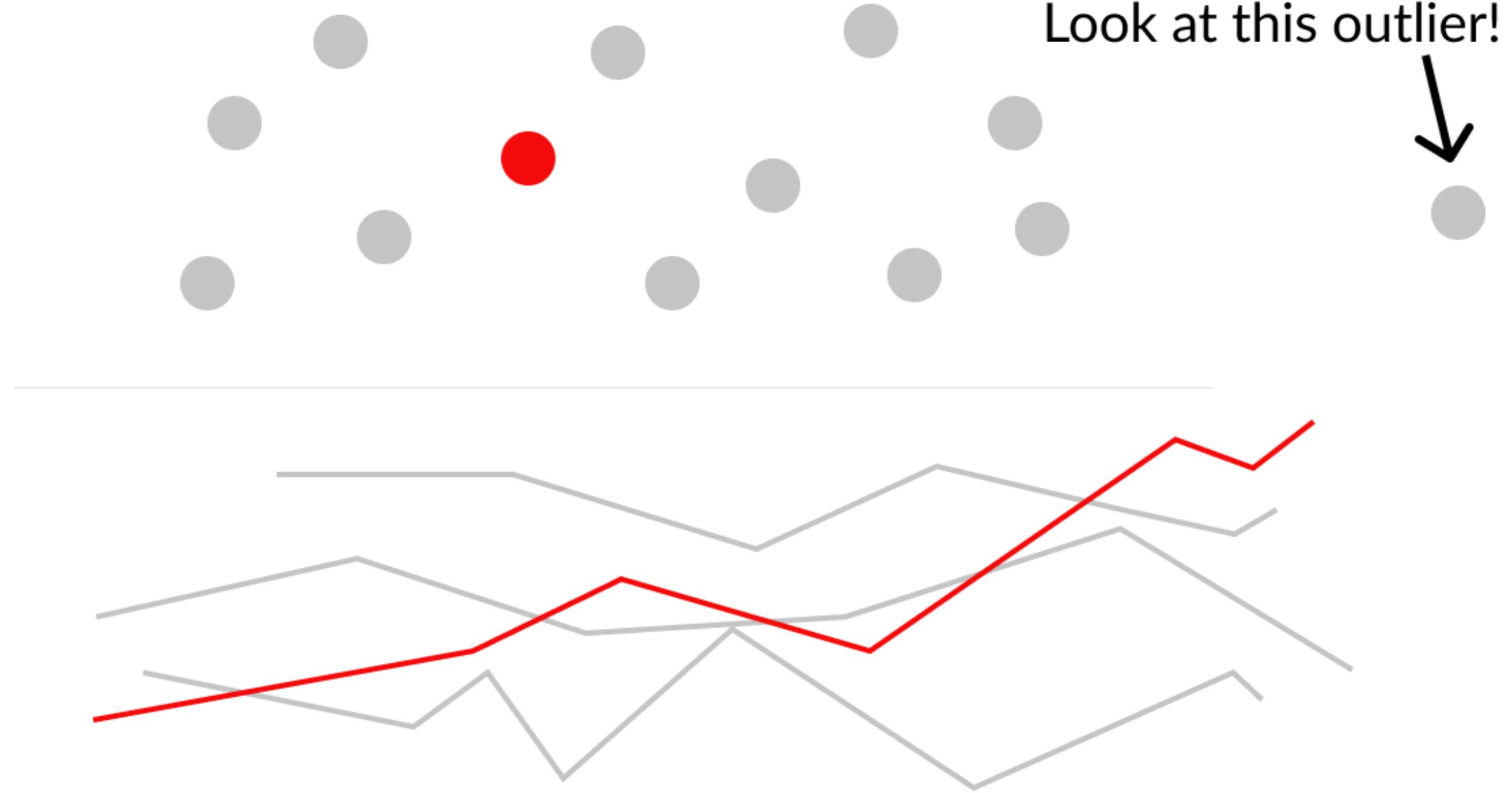
Wrap-Up

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

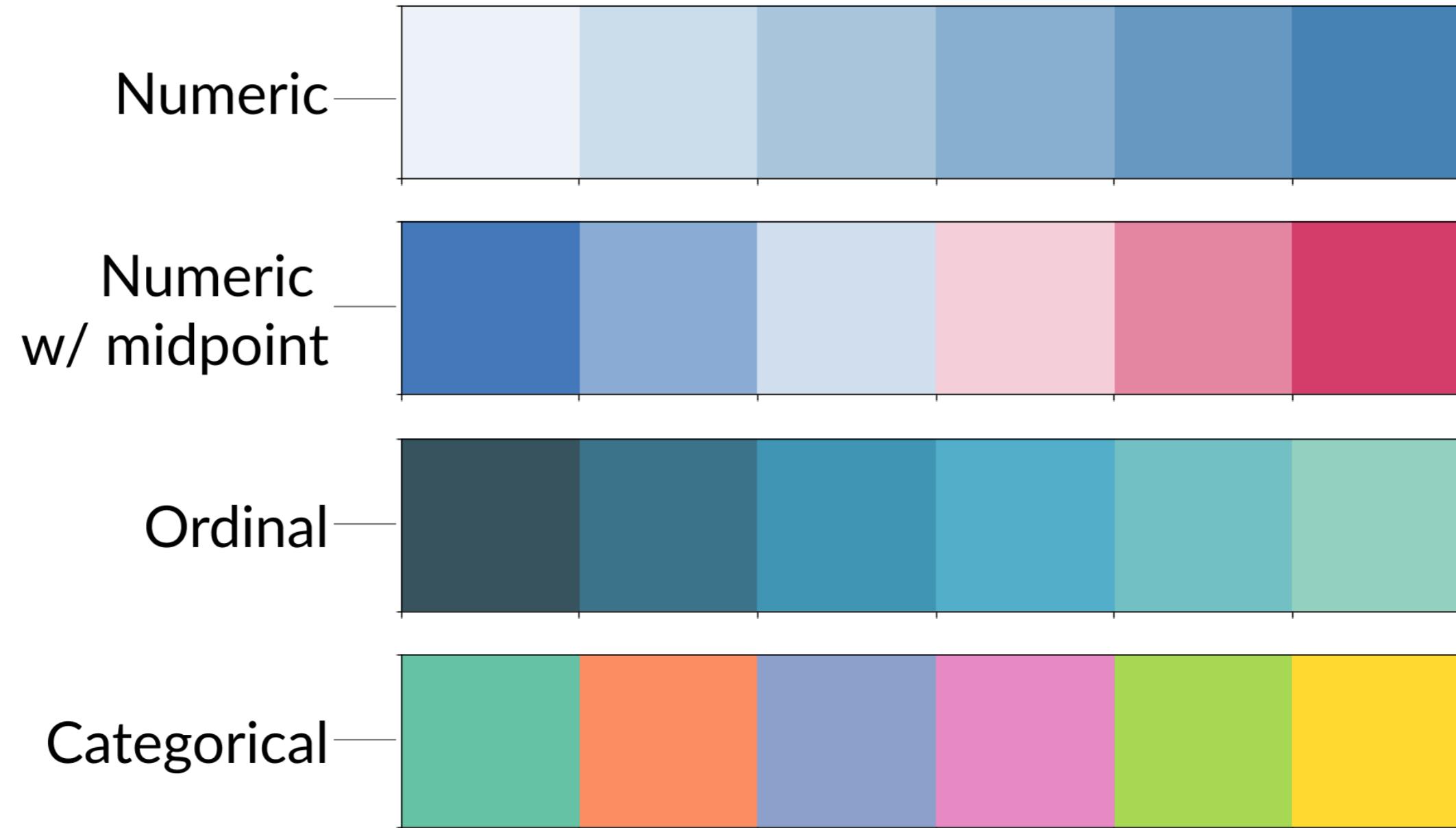


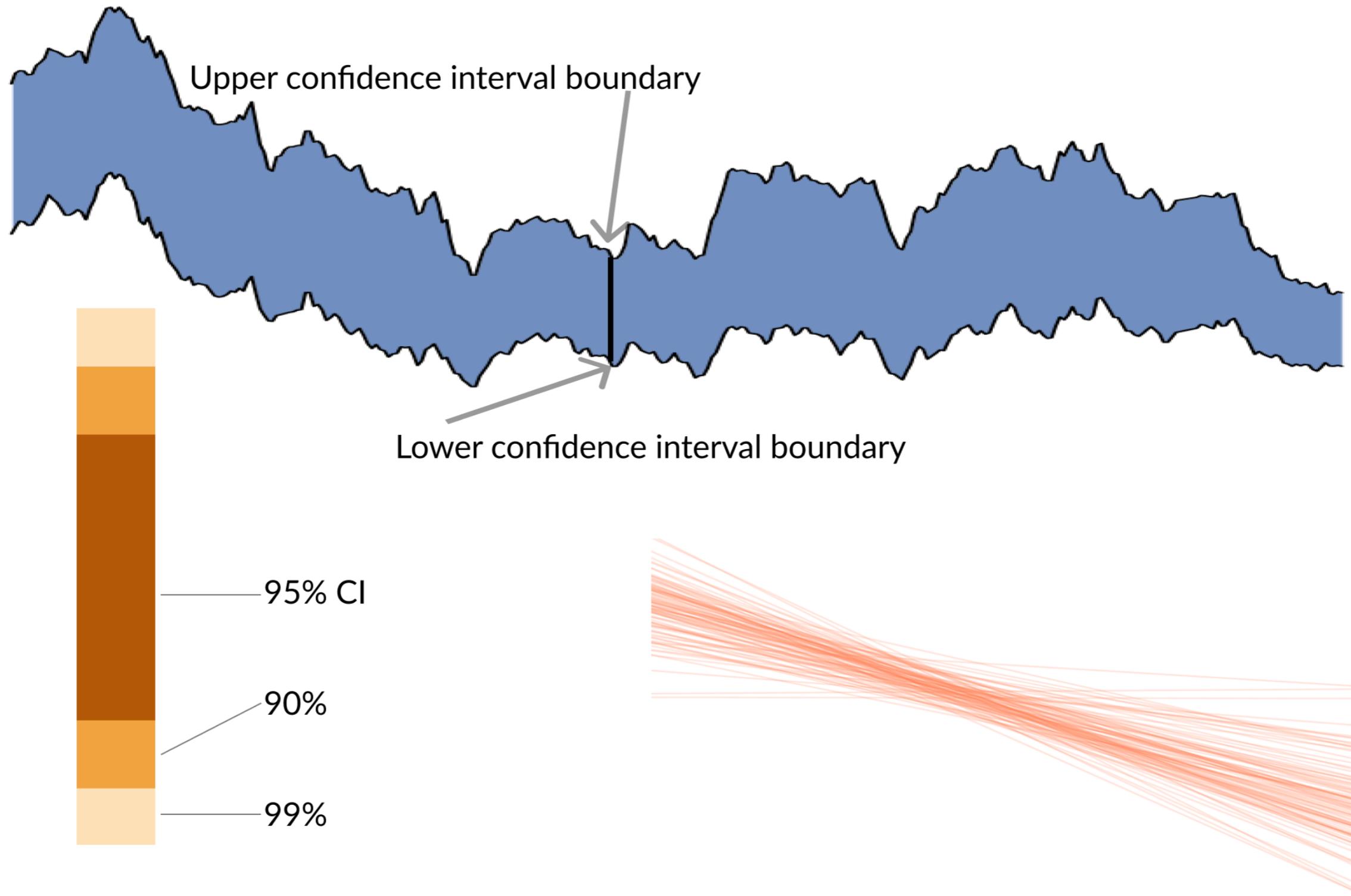
Nick Strayer

Instructor

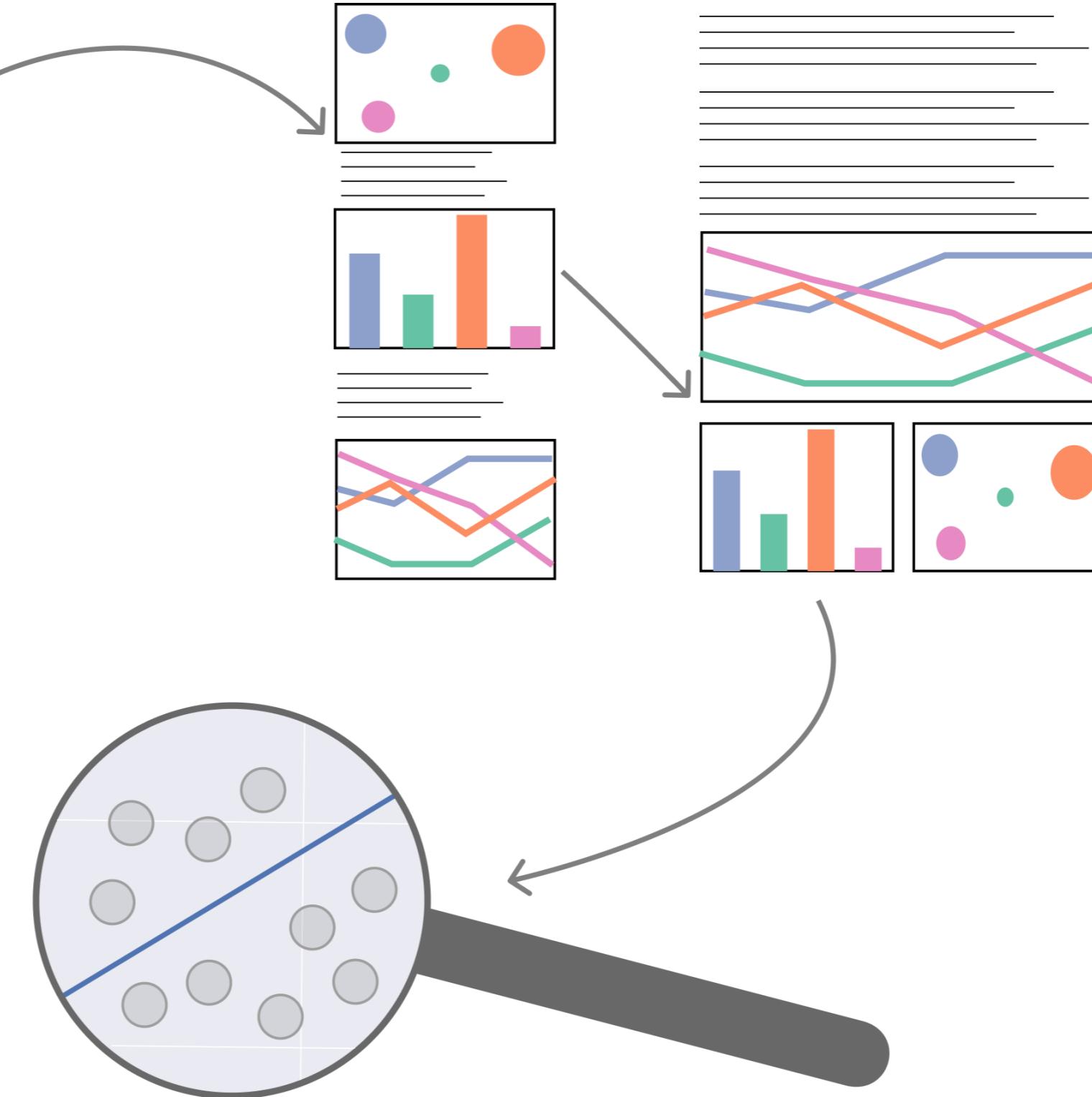


Using color responsibly





Rhode Island
District of Columbia
Washington
New Hampshire
Vermont
Connecticut
Maine
Wisconsin
Ohio
Massachusetts
New York
Minnesota
Indiana
West Virginia
Pennsylvania
Florida
Kentucky
Illinois
Tennessee
Alabama
Mississippi
Arkansas
Louisiana
Texas
New Mexico
Wyoming
North Dakota
Arizona
Rhode Island
District of Columbia
Washington
New Hampshire
Vermont
Connecticut
Maine
Wisconsin
Ohio
Massachusetts
New York
Minnesota
Indiana
West Virginia
Pennsylvania
Florida
Kentucky
Illinois
Tennessee
Alabama
Mississippi
Arkansas
Louisiana
Texas
New Mexico
Wyoming
North Dakota
Arizona



Going further

Blogs

- [Flowing data](#)
 - Curated list of data visualizations.
- [Datawrapper Blog](#)
 - Articles that dig deep into visualization techniques and mistakes.

Twitter

- [#datavis](#)
 - An ongoing stream of cool projects and inspiration.

Thank you!

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON