Highlighting data

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

Nick Strayer
Instructor

datacamp

About me

The New York Times

The Great Out-of-State Migration: Where Students Go

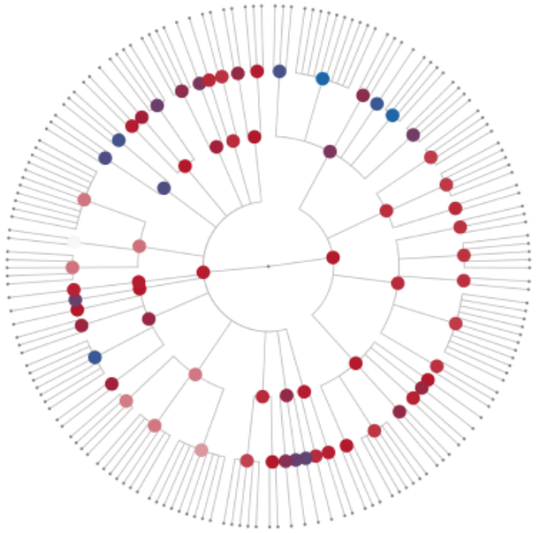
By NICK STRAYER AUG. 26, 2016

Public colleges and universities have historically served their own state residents, but the number of out-of-state freshmen attending them has nearly doubled since 1986, according to Department of Education data. RELATED ARTICLE

Exodus of Public University Students

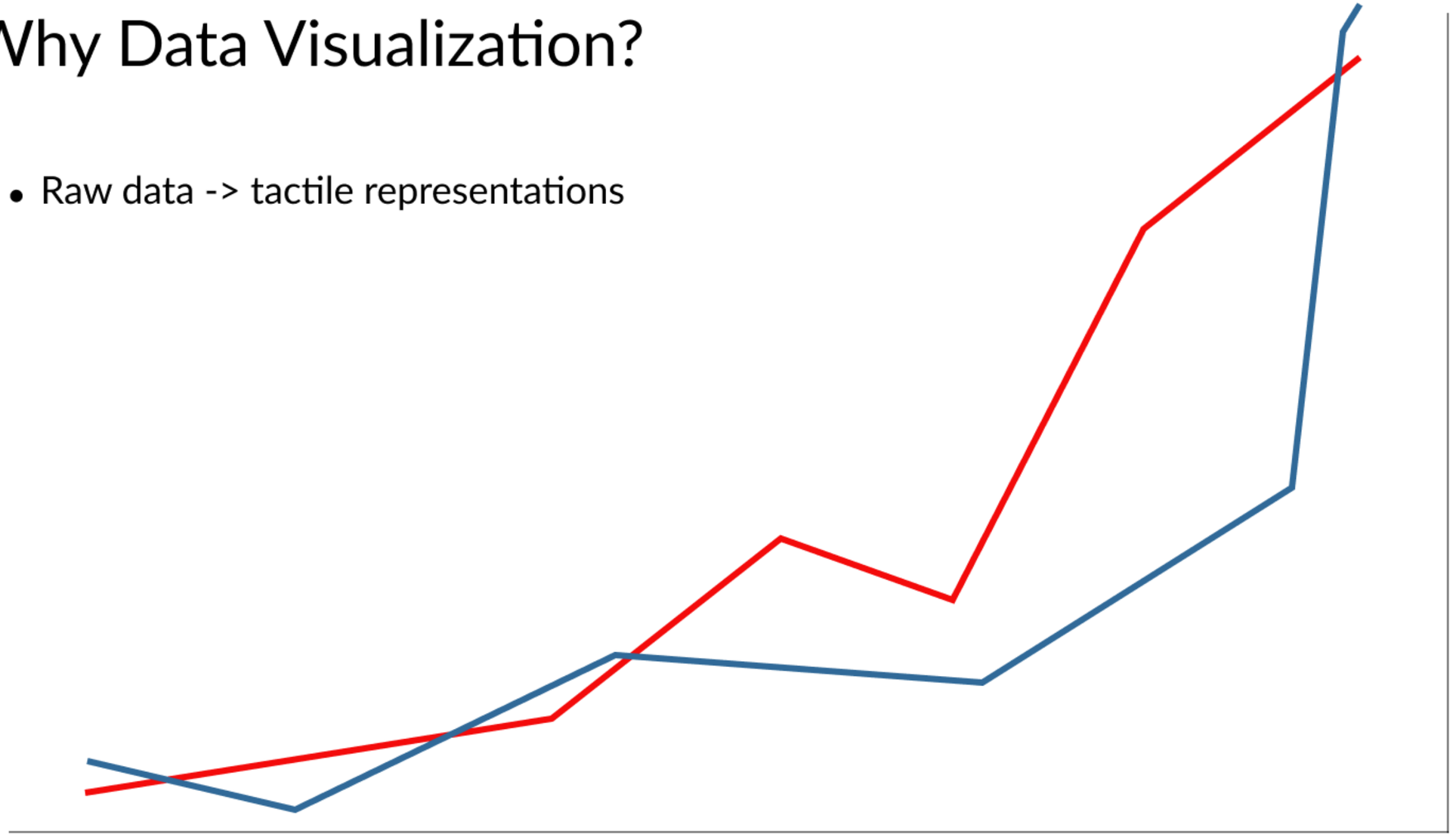
Arrows are in proportion to number of freshmen leaving their home state to attend public universities in other states.*

Source: U.S. Department of Education, 2014 data. *Note: Arrows show only movements of 500 or more students.



Why Data Visualization?

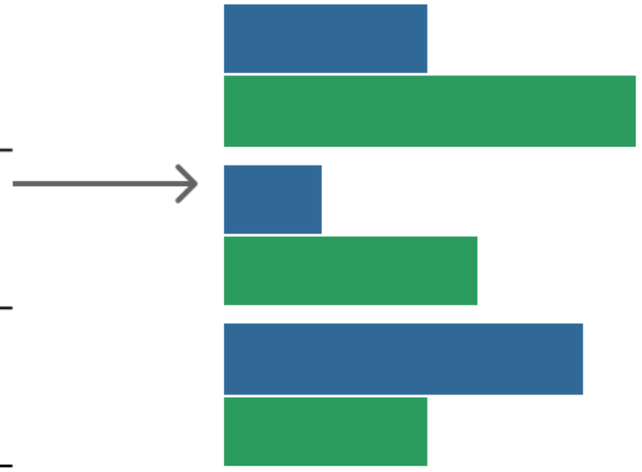
- Raw data -> tactile representations



Why Data Visualization?

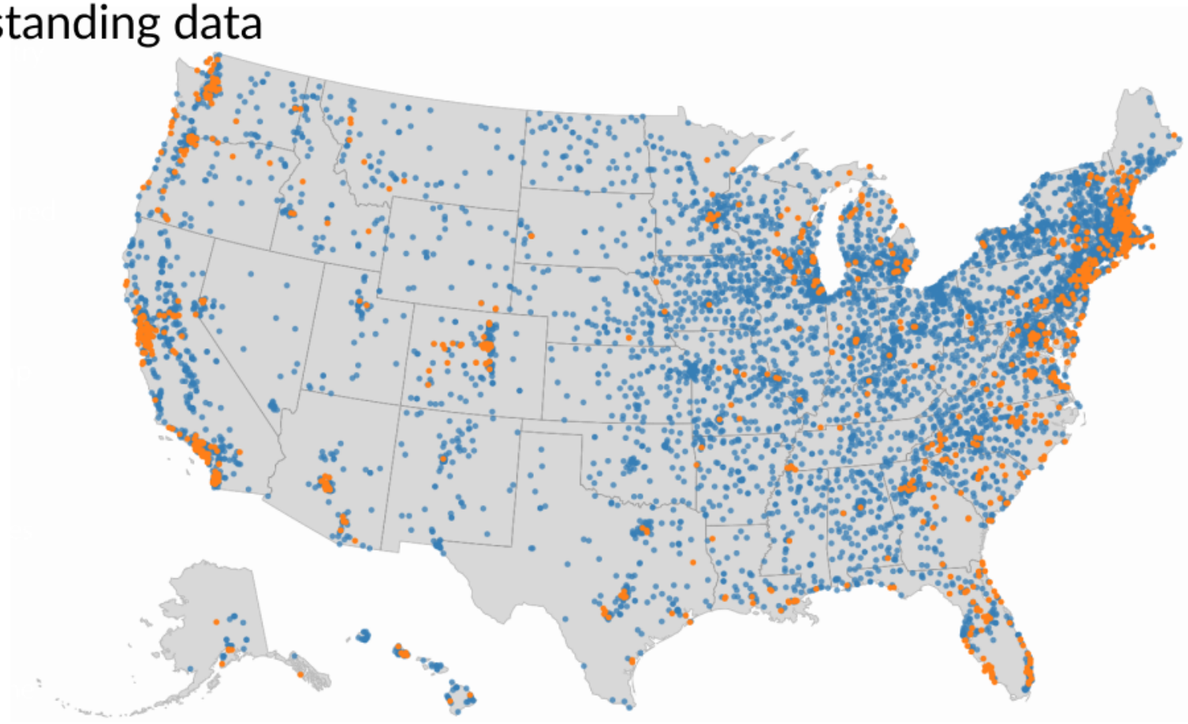
- Raw data -> tactile representations
- *Sometimes* purely cosmetic

a	b
4	8
2	5
7	4



Why Data Visualization?

- Raw data -> tactile representations
- *Sometimes* purely cosmetic
- *Sometimes* essential to understanding data




python


Seaborn


matplotlib

Prereqs

Introduction to Data Visualization in Python

Introduction to Data Visualization with Seaborn

Python Data Science Toolbox (Part 1)

Python Data Science Toolbox (Part 2)

Google Sheets

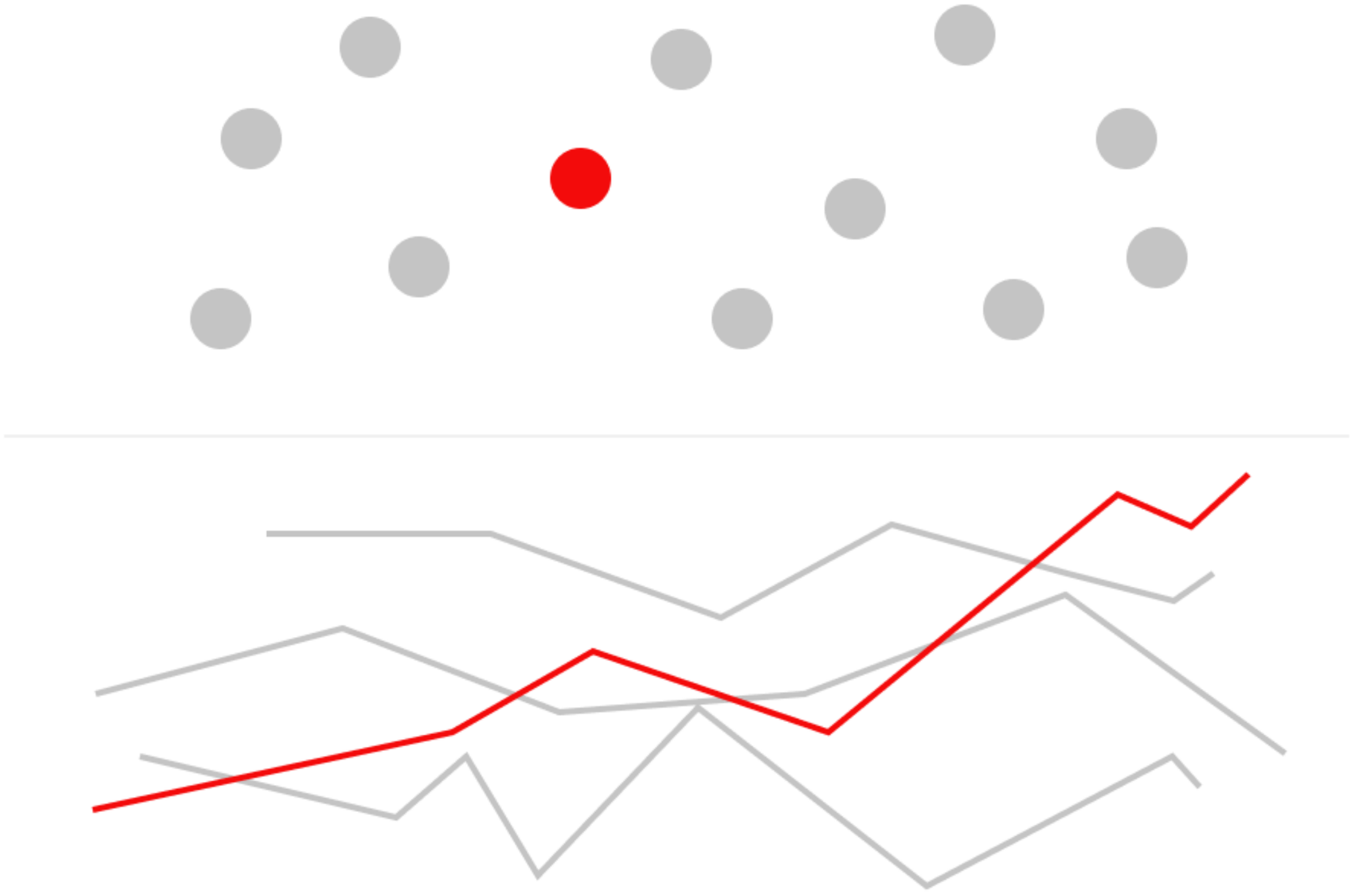
+ a b | e a u

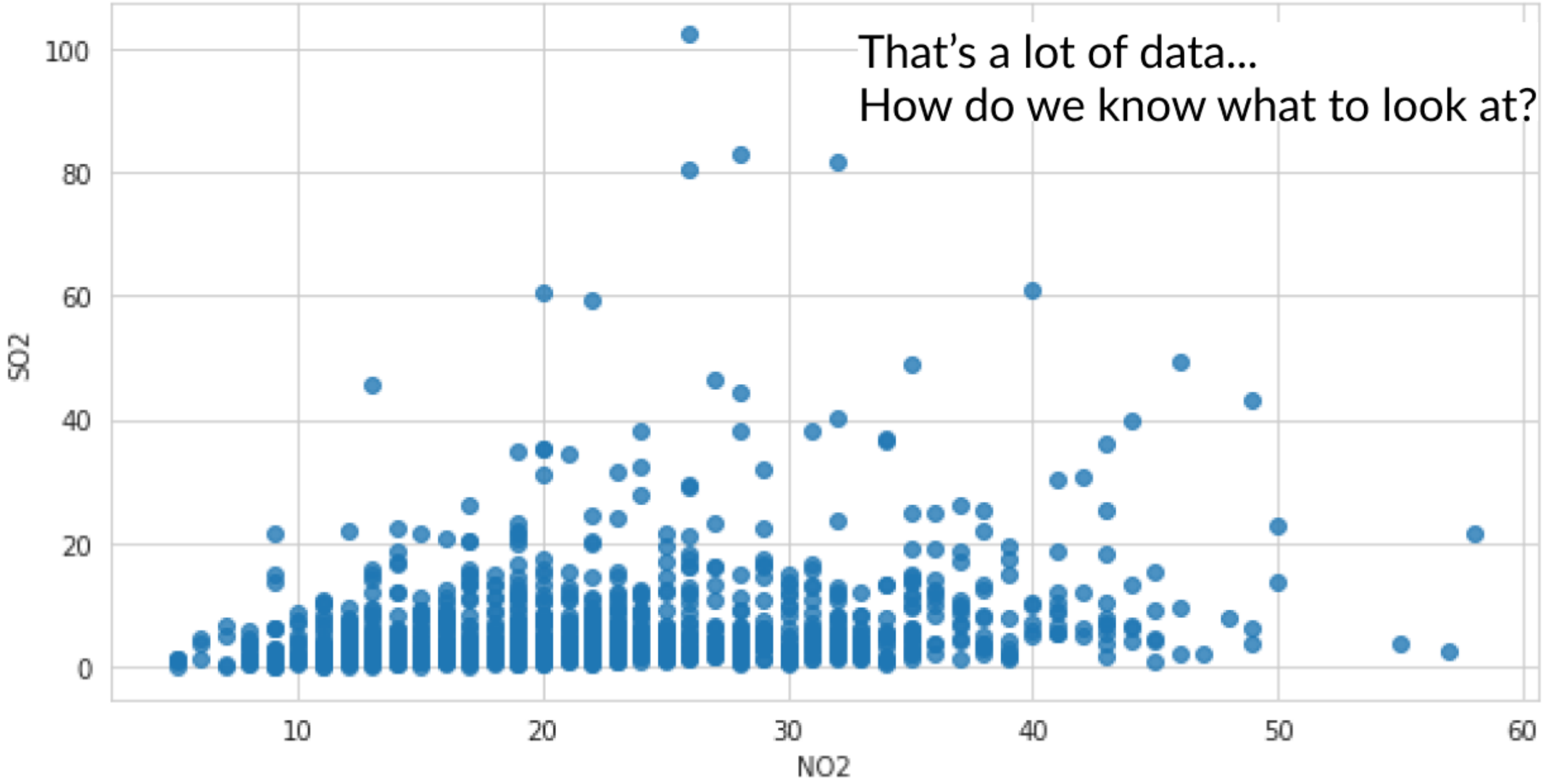
```
pollution.head()
```

```
          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
```

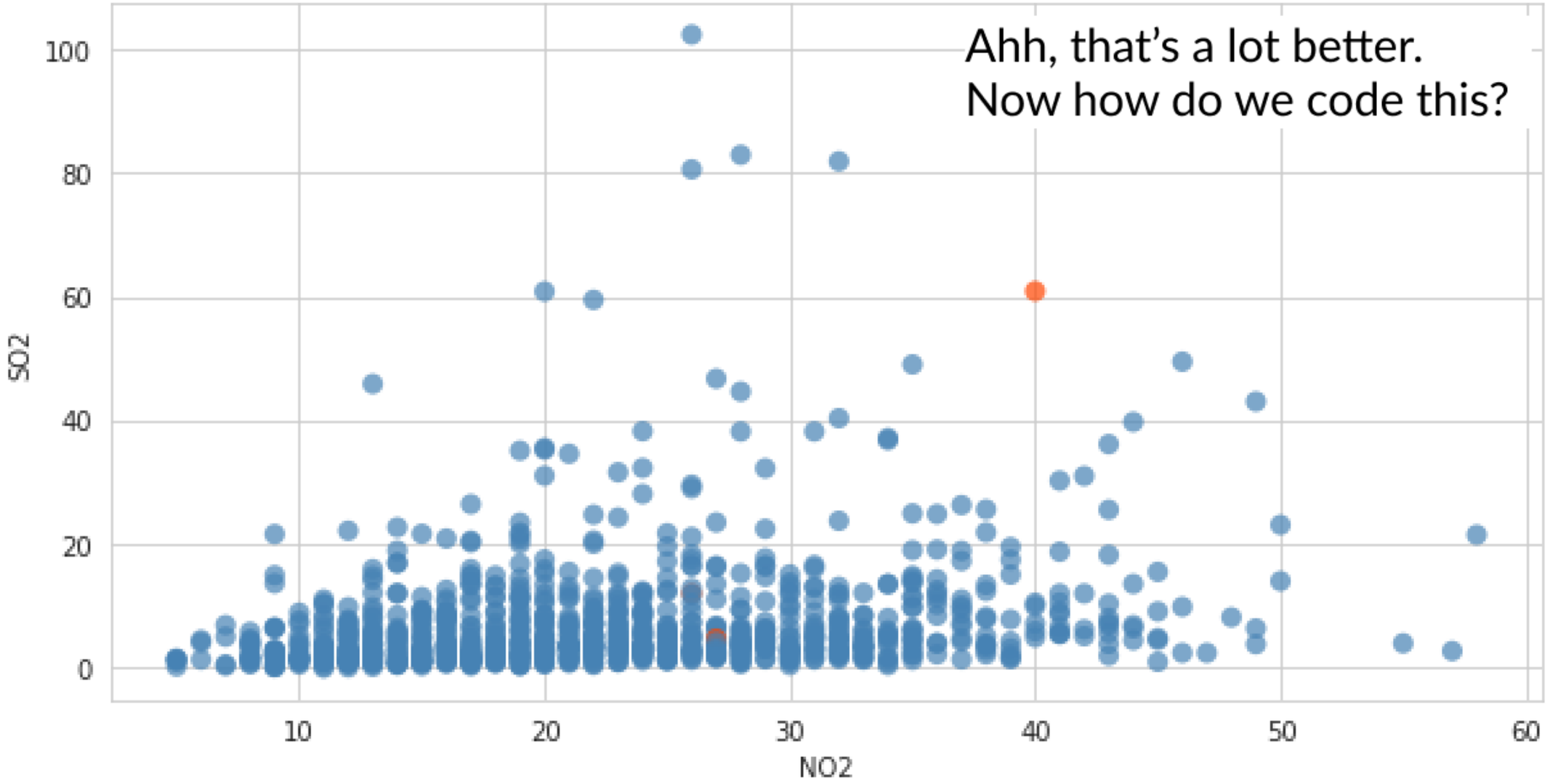
```
pollution.city.unique()
```

```
[ 'Boston',    'Cincinnati',     'Denver',       'Des Moines',
  'Fairbanks', 'Houston',        'Indianapolis', 'Long Beach',
  'New York',  'Salt Lake City', 'Vandenberg Air Force Base' ]
```




That's a lot of data...
How do we know what to look at?
100
80
60
40
20
0
SO2
10
20
30
40
50
60
NO2

```
cinci_pollution = pollution[pollution.city == 'Cincinnati']
# Make an array of colors based upon if a row is a given day
cinci_colors = ['orangered' if day == 38 else 'steelblue'
                for day in cinci_pollution.day]
# Plot with additional scatter plot argument facecolors
p = sns.regplot(x='NO2',
                y='SO2',
                data = cinci_pollution,
                fit_reg=False,
                scatter_kws={'facecolors': cinci_colors,'alpha': 0.7})
```


Ahh, that's a lot better.
Now how do we code this?
100
80
60
40
20
0
SO2
10
20
30
40
50
60
NO2

Let's make some highlights!

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

Comparing groups

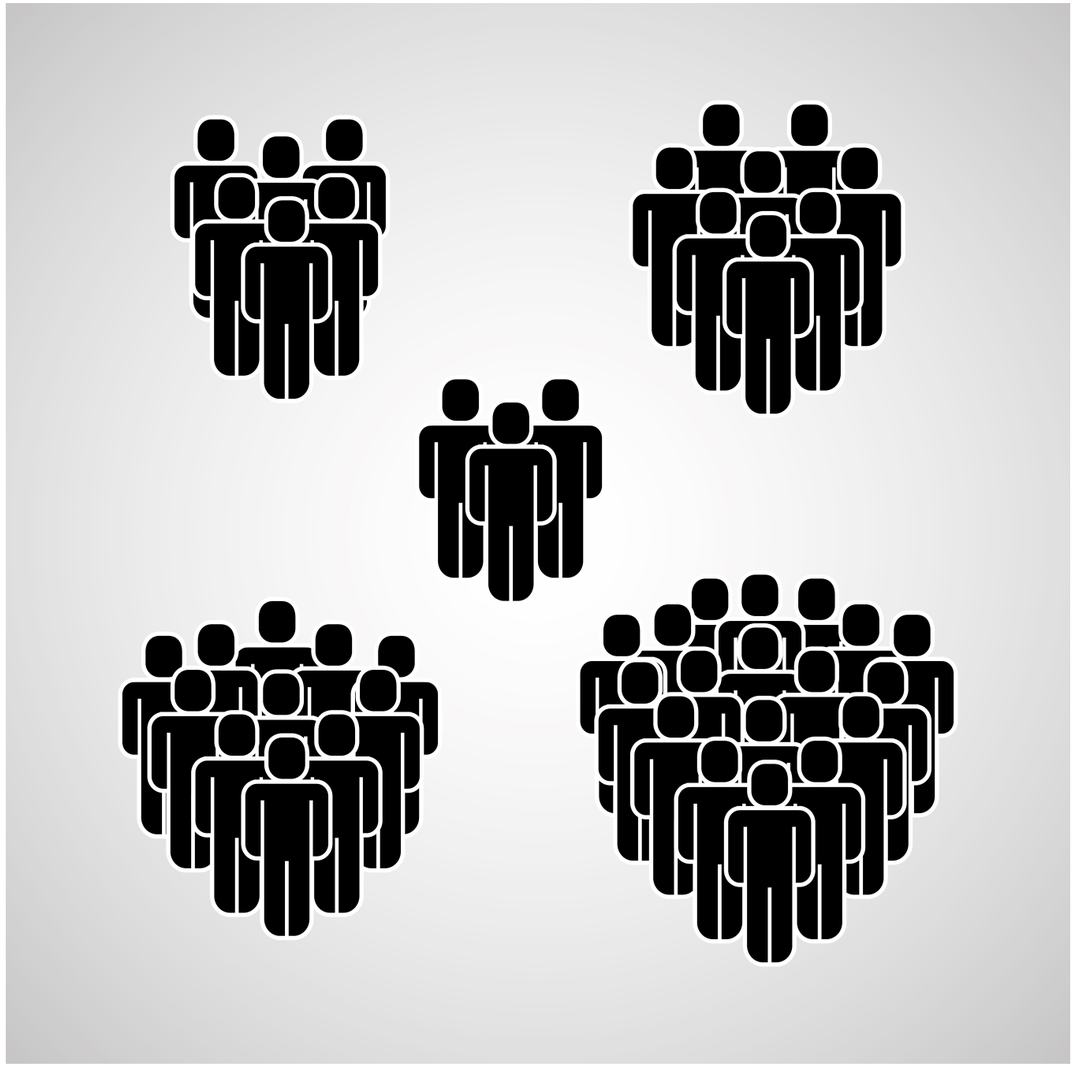
IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

Nick Strayer
Instructor

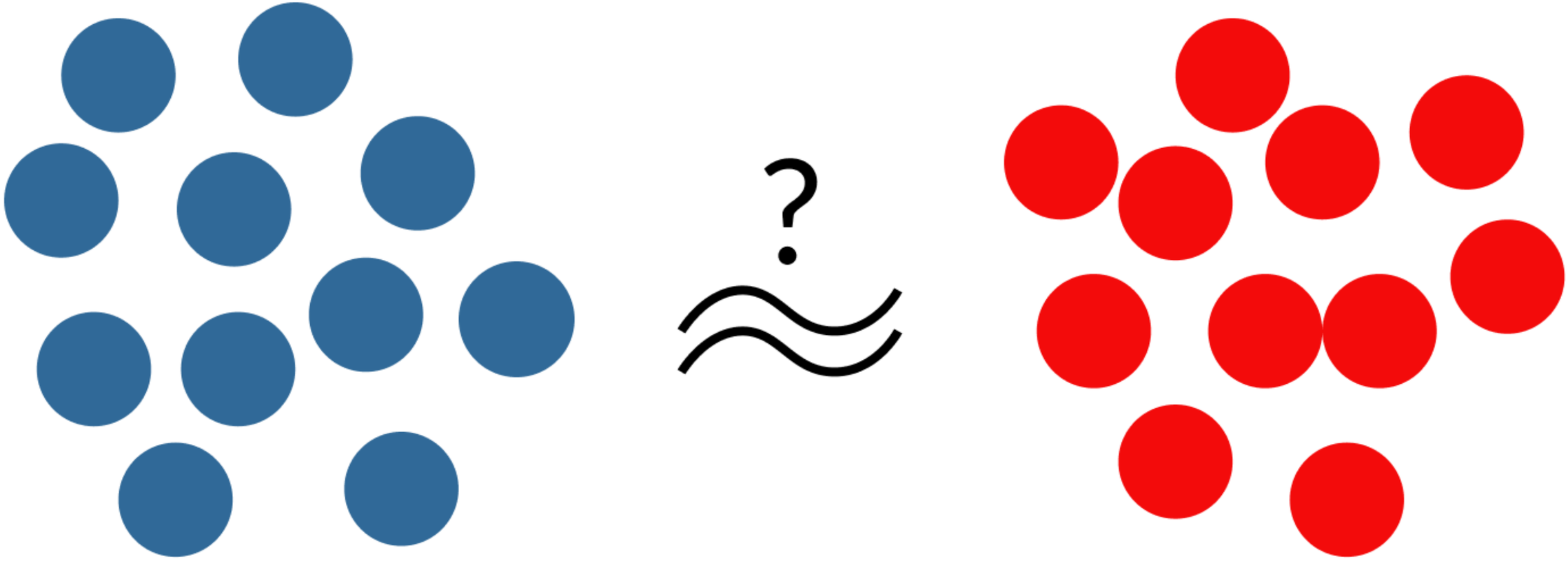
datacamp

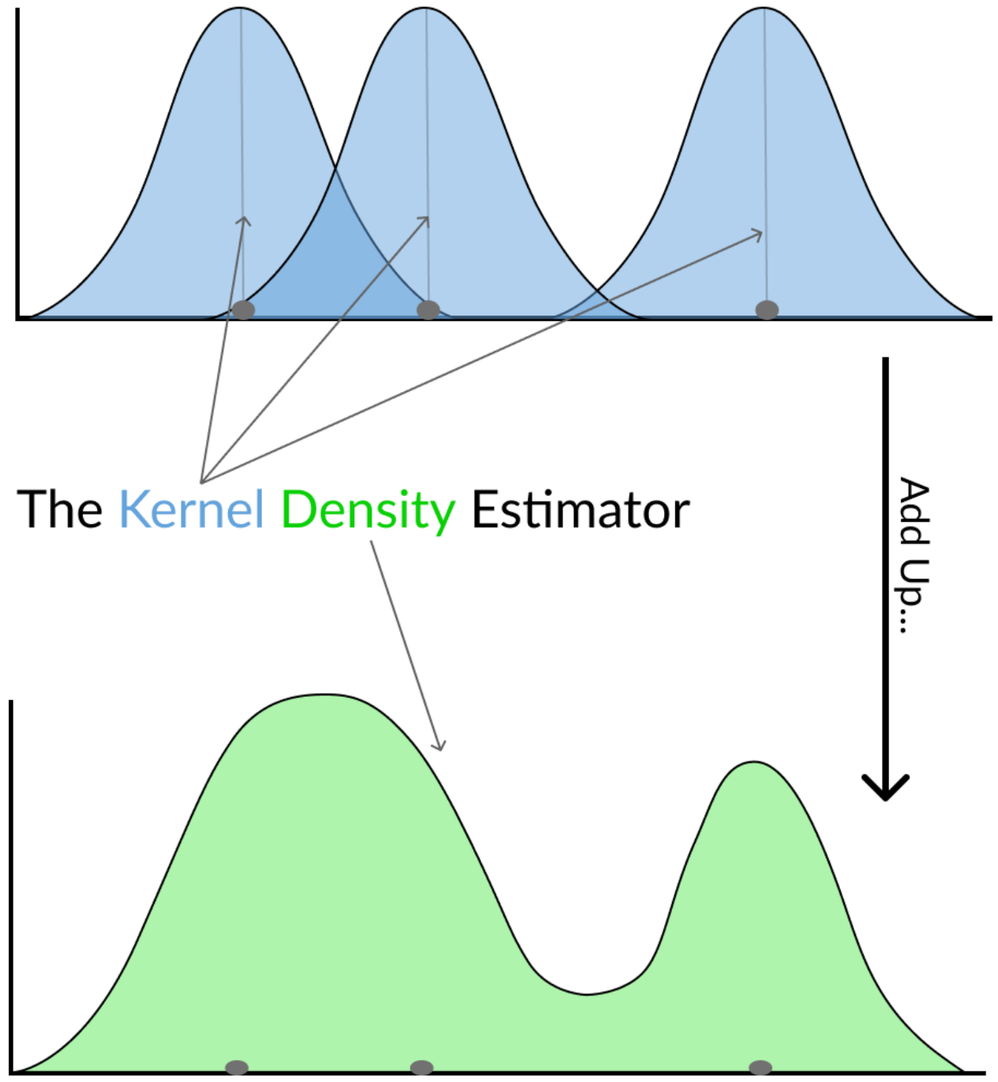
What does this mean?

- Values generally higher?
- Distribution of values wider? Narrower?
- Crucial for representing your data

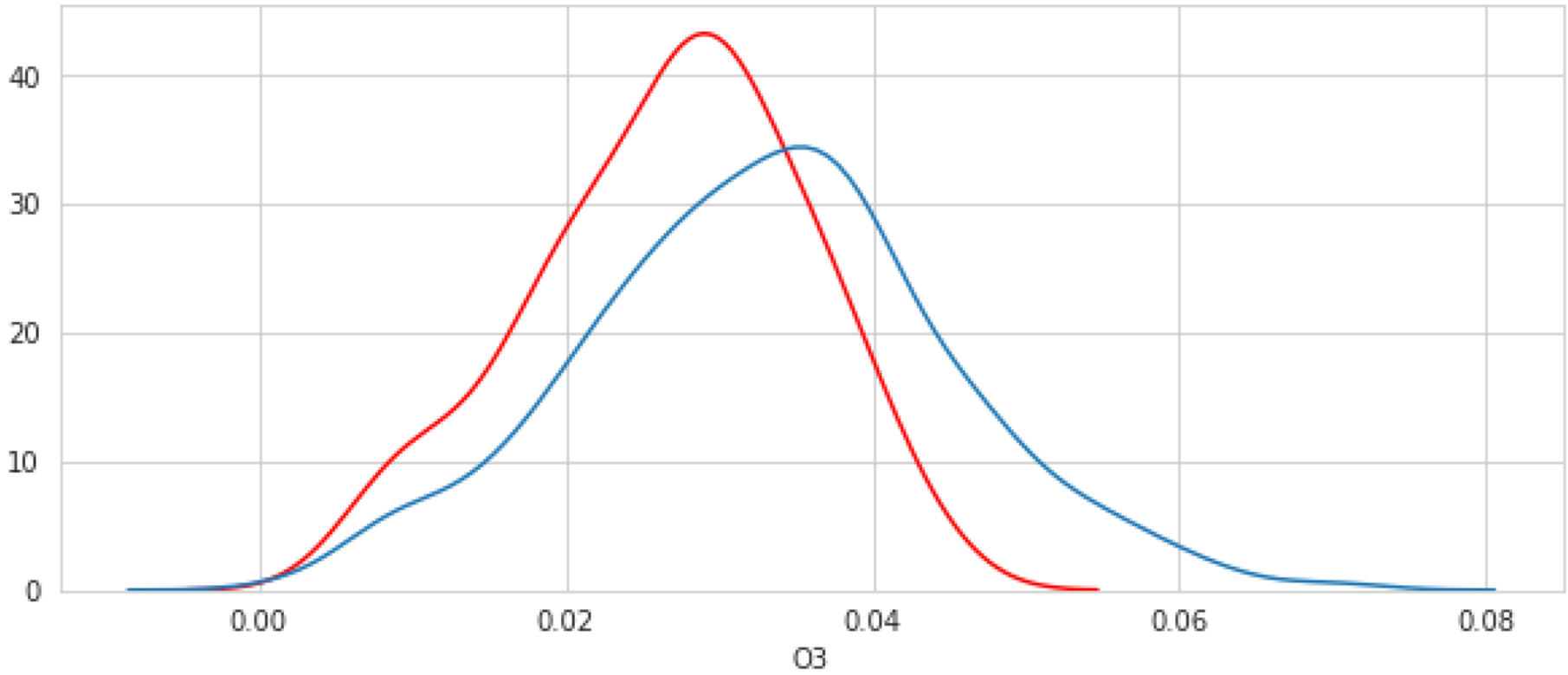


Comparing a couple classes

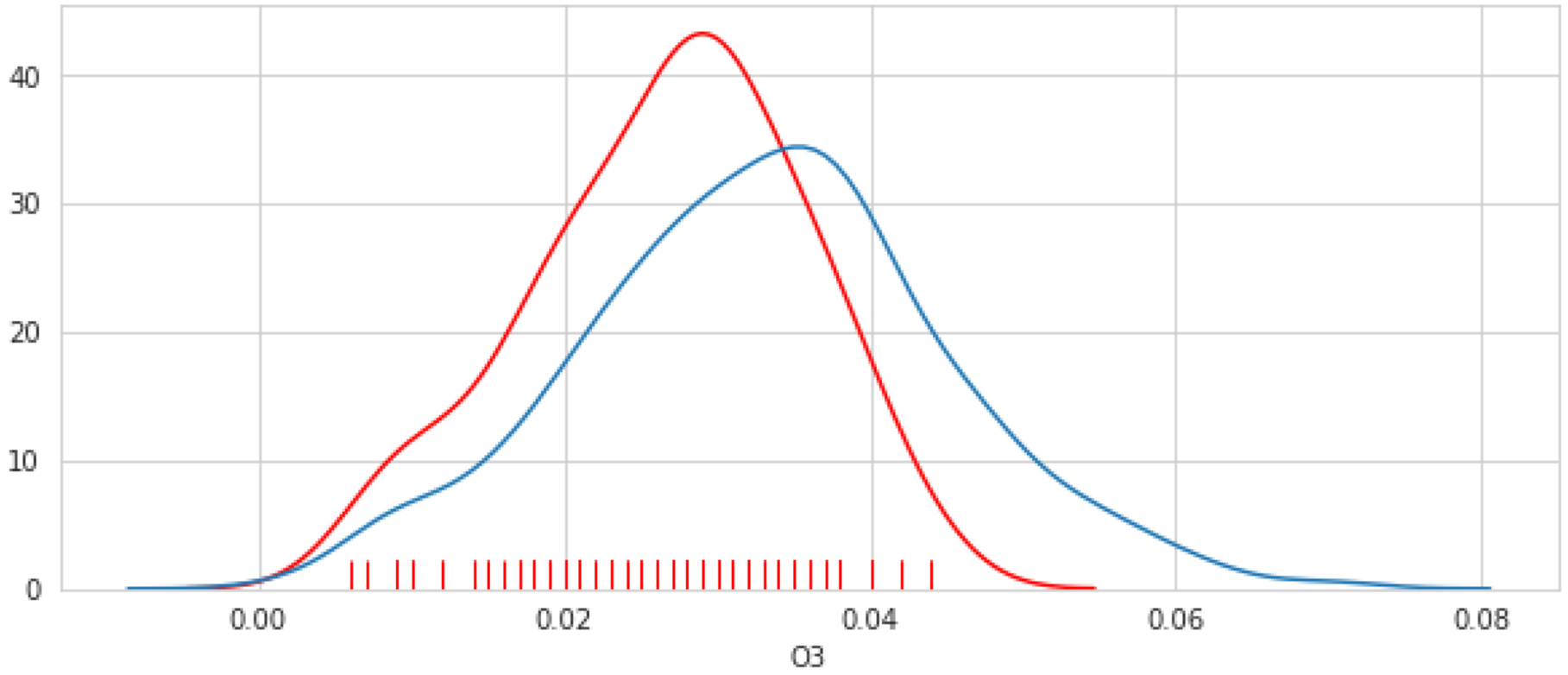



The Kernel Density Estimator
Add Up...

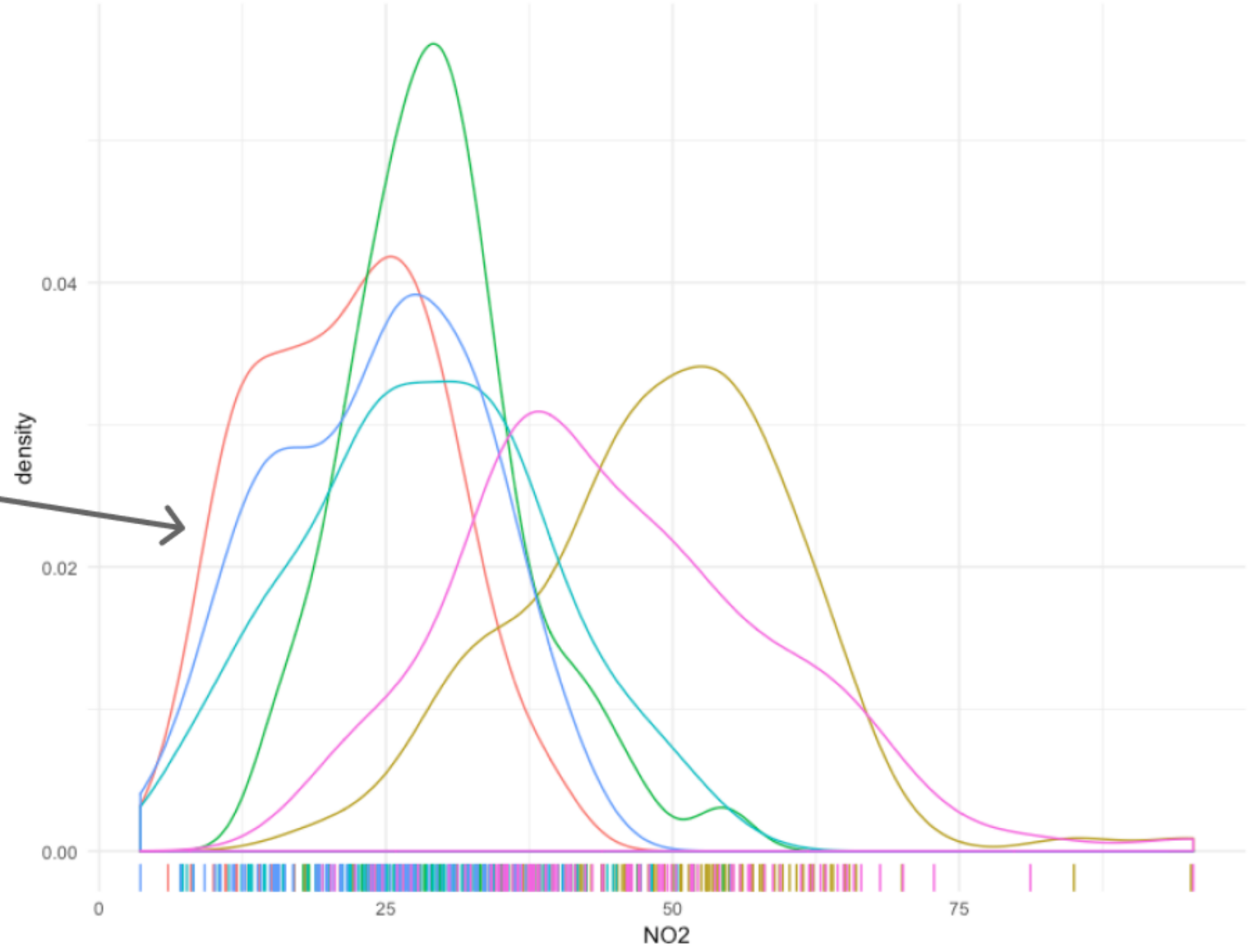
```
pollution_nov = pollution[pollution.month == 10]
sns.distplot(pollution_nov[pollution_nov.city == 'Denver'].O3, hist=False,
             color = 'red')
sns.distplot(pollution_nov[pollution_nov.city != 'Denver'].O3, hist=False)
```



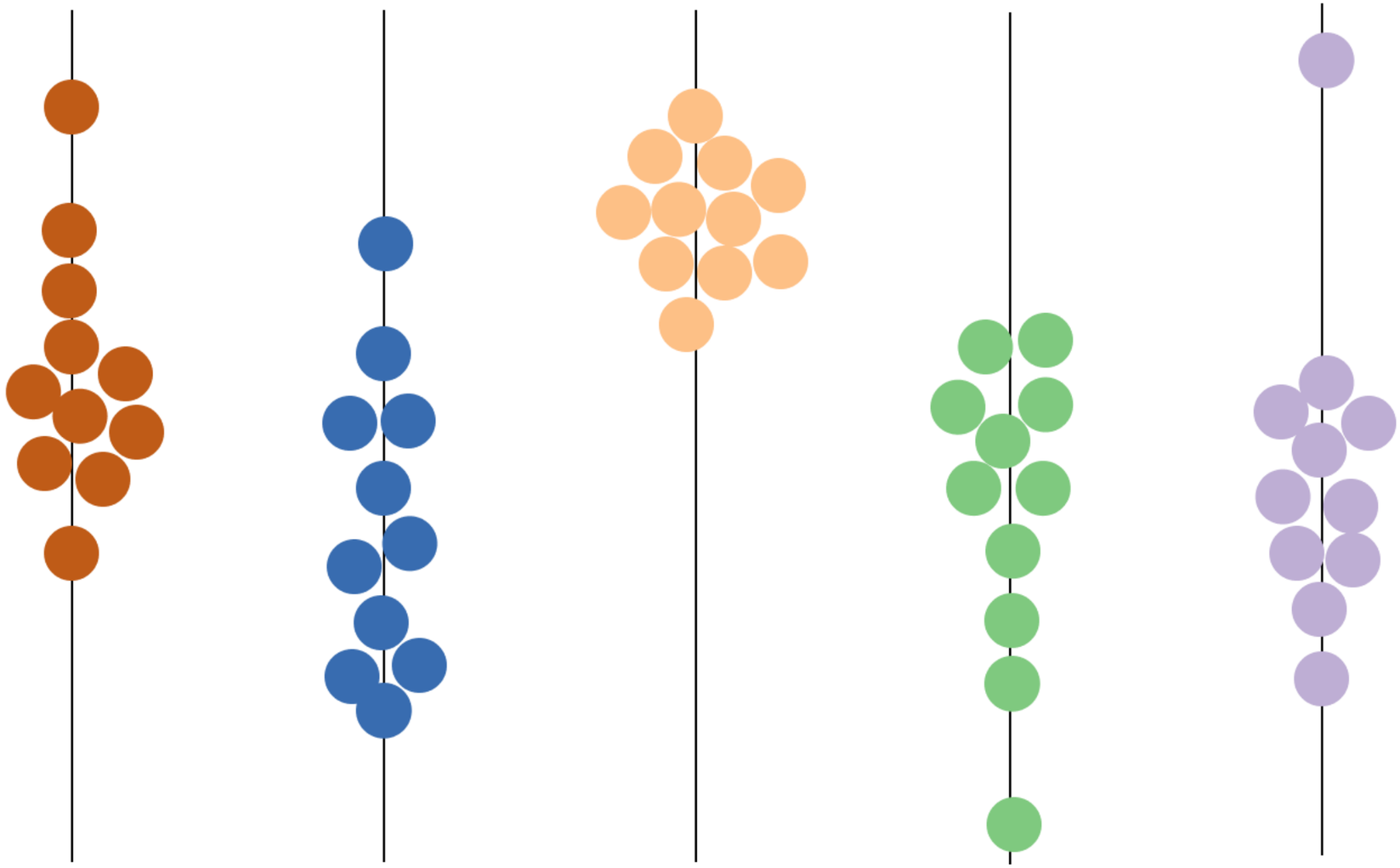
```
# Enable rugplot
sns.distplot(pollution_nov[pollution_nov.city == 'Denver'].O3,
             hist=False, color='red', rug=True )
sns.distplot(pollution_nov[pollution_nov.city != 'Denver'].O3, hist=False)
```



Hard to keep to keep track of lines

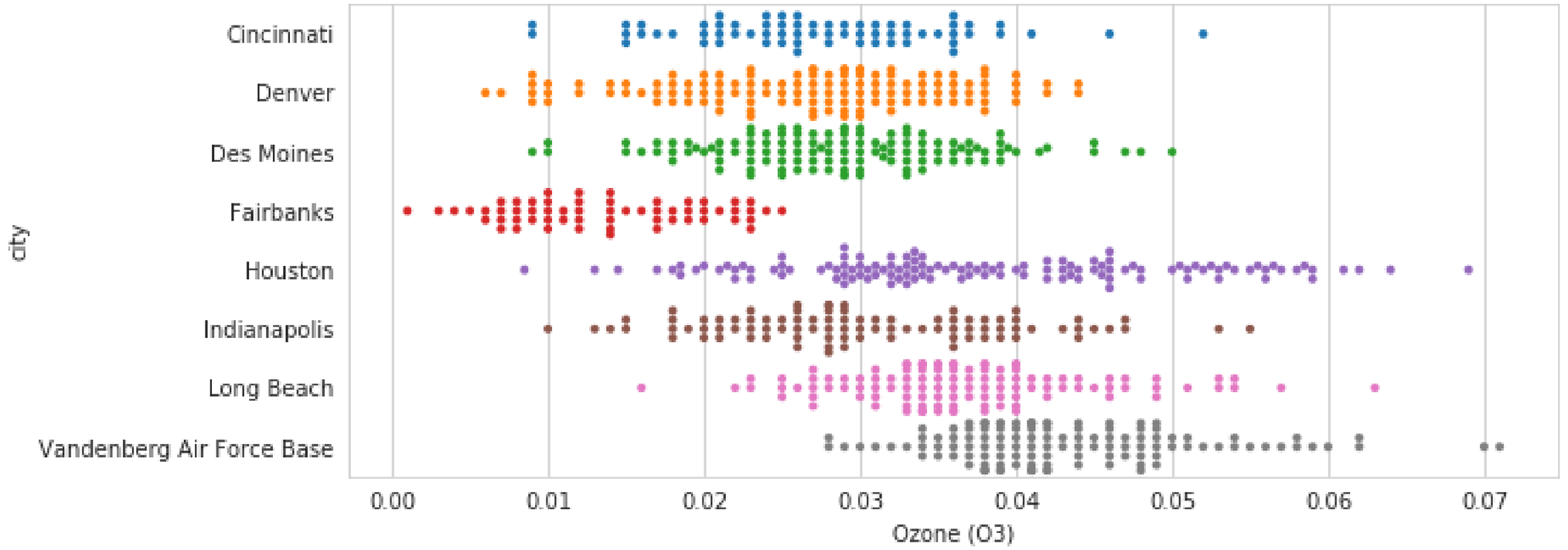


The beeswarm plot



```
pollution_nov = pollution[pollution.month == 10]


sns.swarmplot(y="city", x="O3", data=pollution_nov, size=4)
plt.xlabel("Ozone (O3)")
```



Let's compare!

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

Annotations

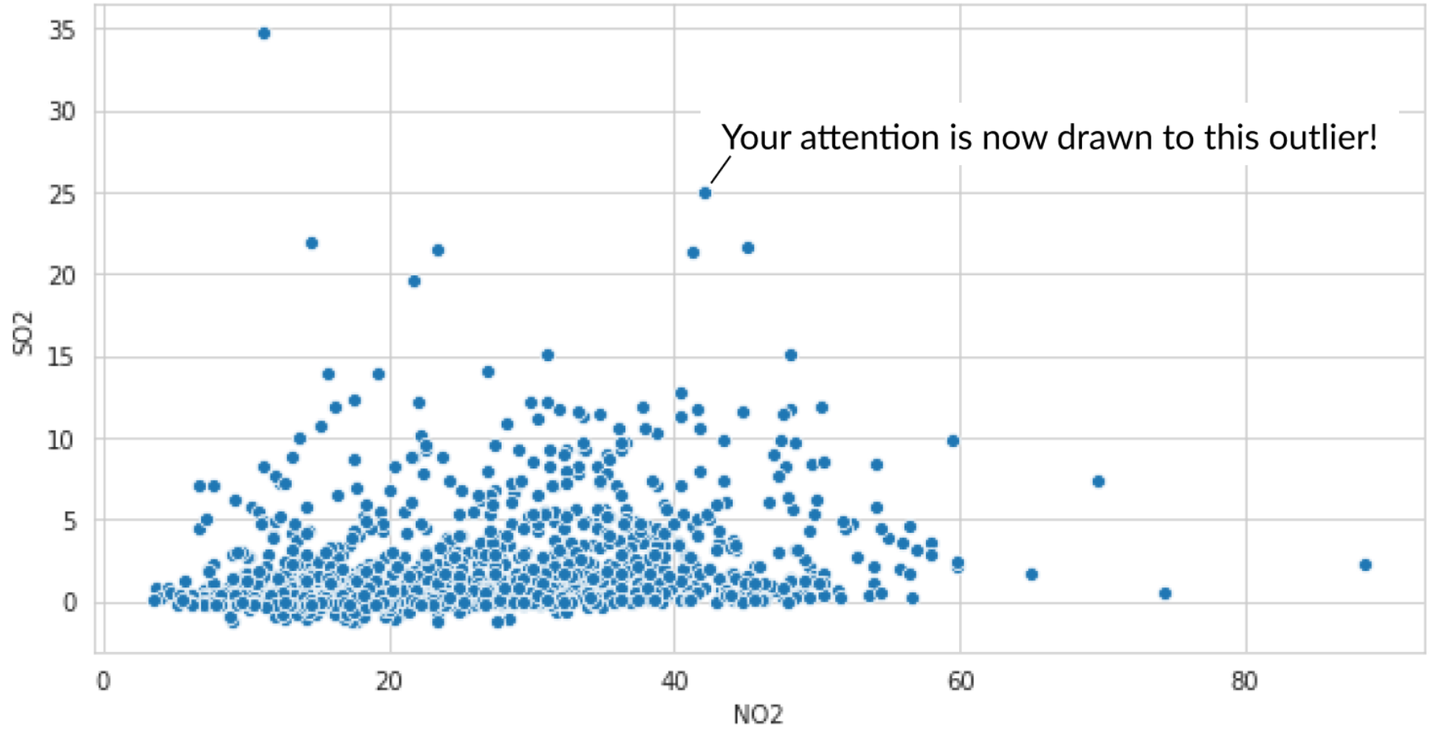
IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON

Nick Strayer
Instructor

datacamp

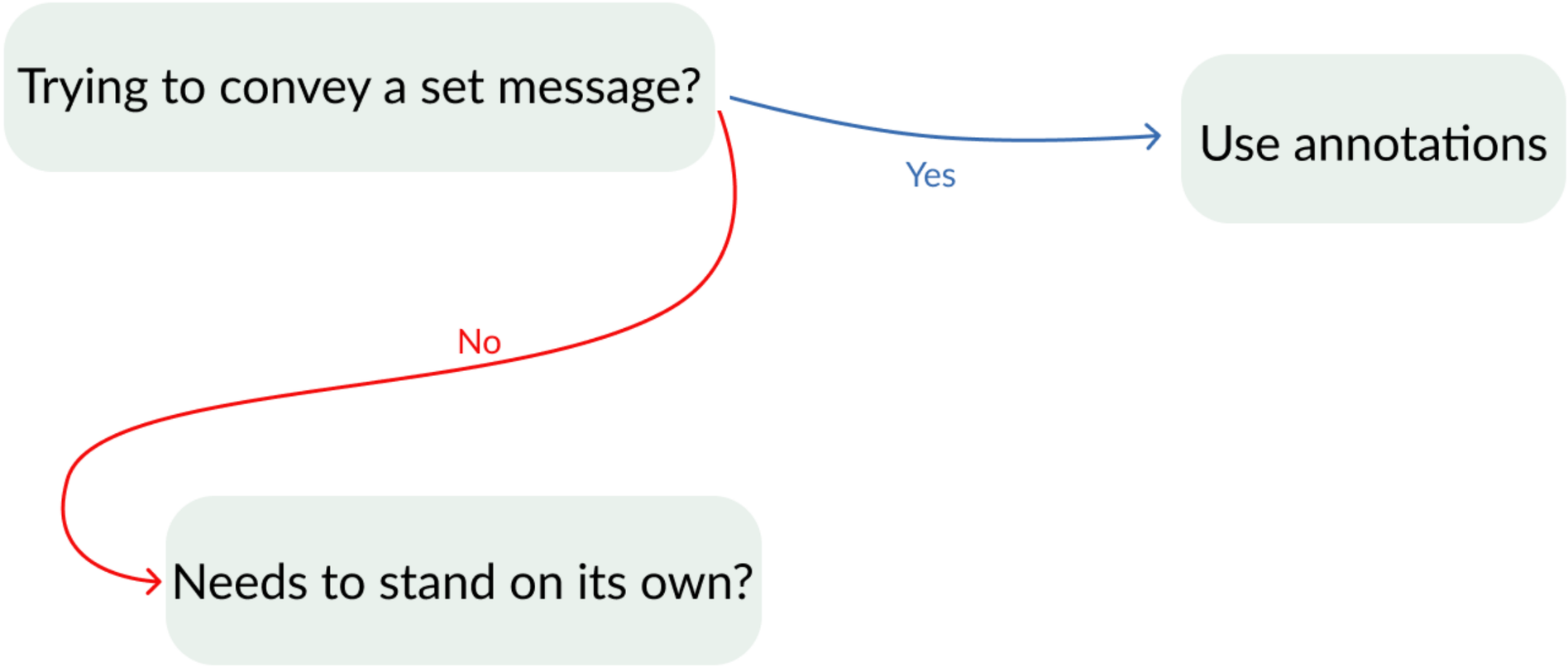
What annotations add

- Compact and efficient communication
- Opportunity to supply deeper insight to data

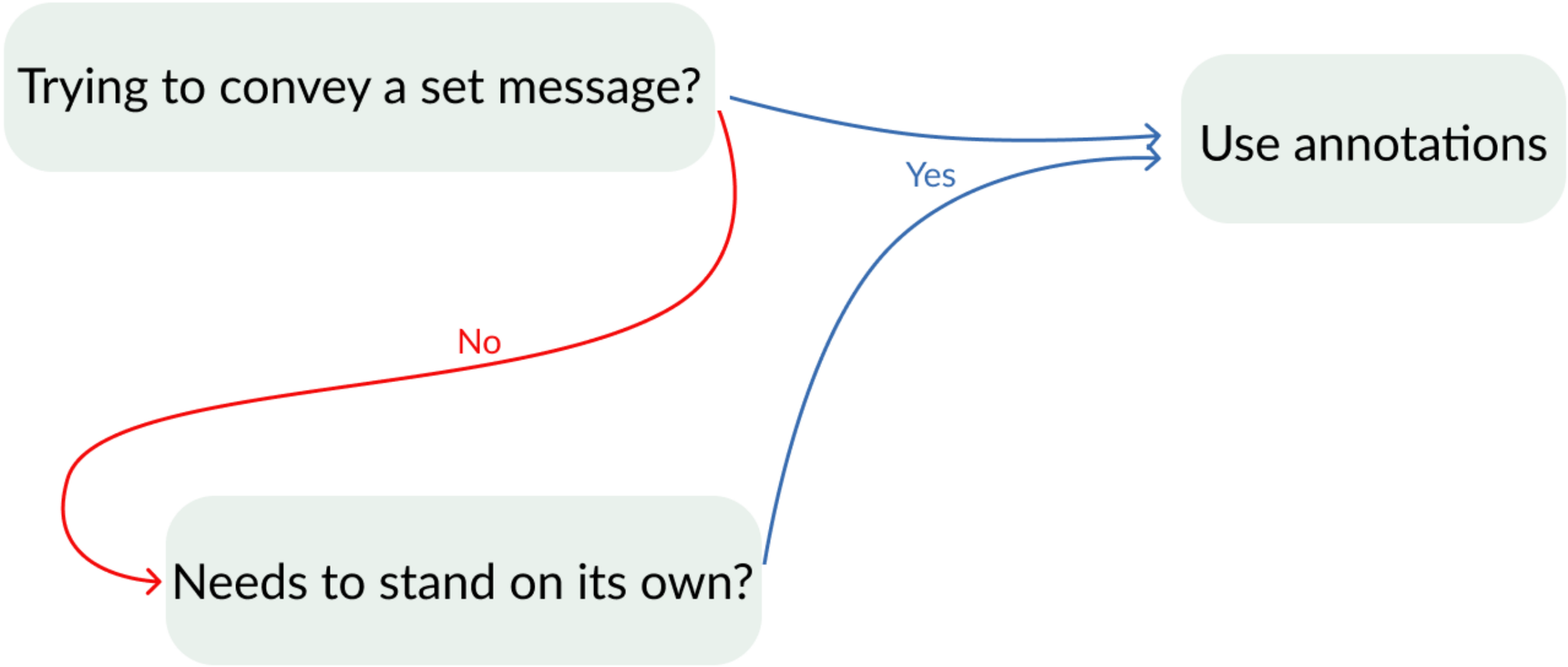


When to use annotations

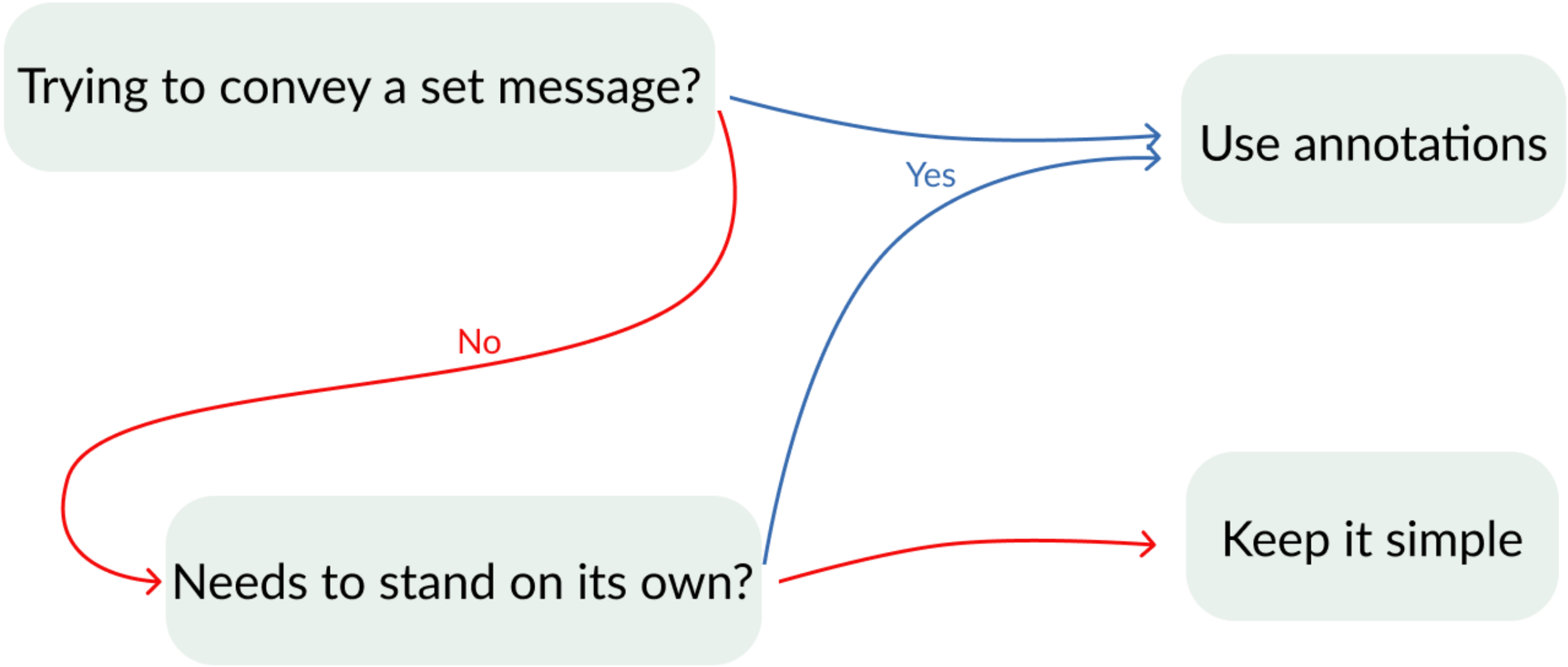
When to use annotations



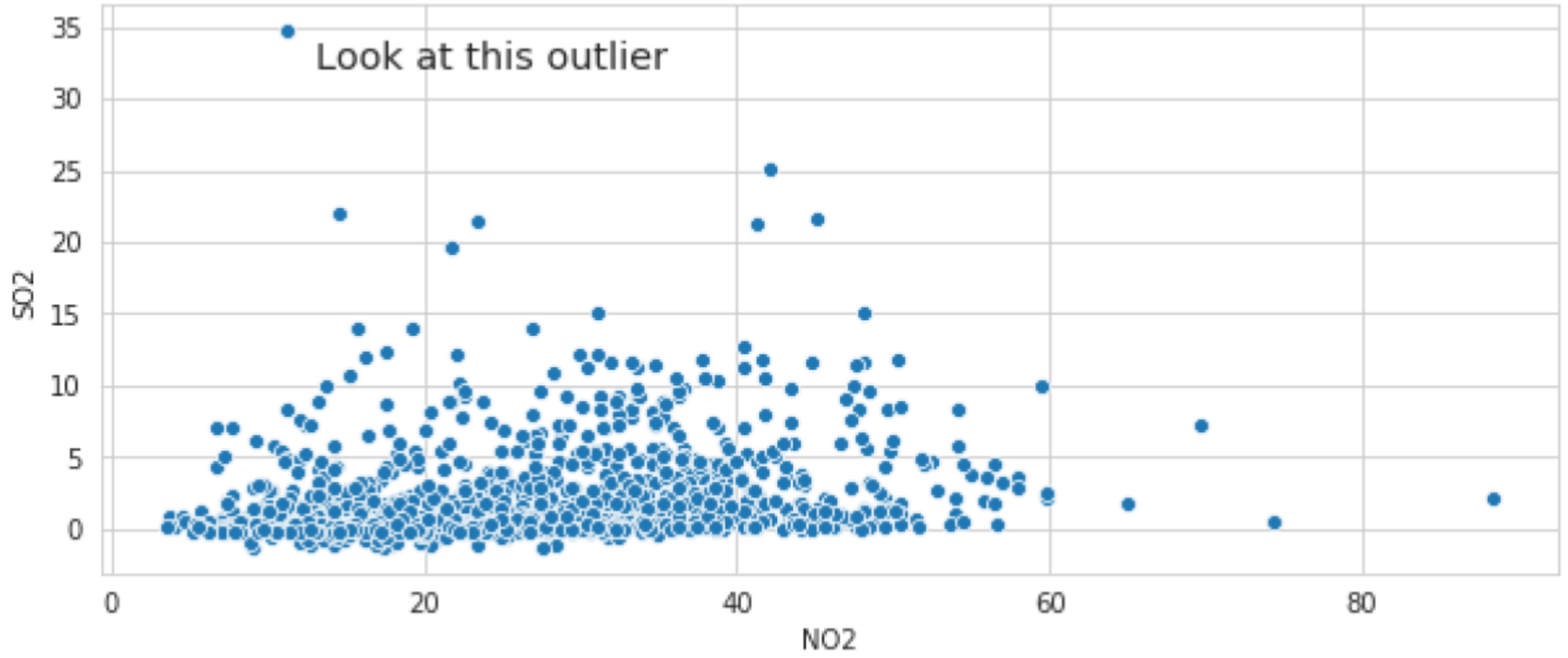
When to use annotations



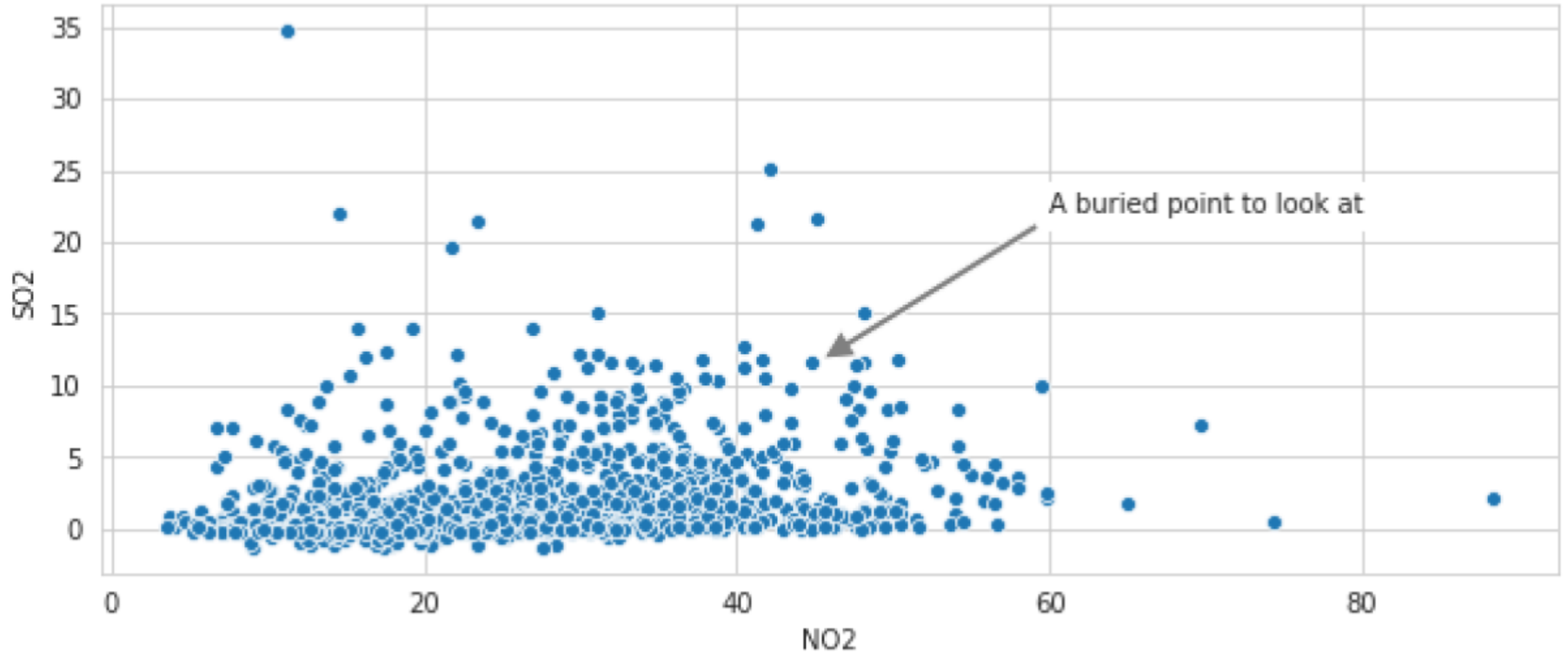
When to use annotations



```
sns.scatterplot(x='NO2', y='SO2', data = houston_pollution)
# X and Y location of outlier and text
plt.text(13,33,'Look at this outlier',
         # Text properties for alignment and size.
         fontdict = {'ha': 'left', 'size': 'x-large'})
```



```
sns.scatterplot(x='NO2', y='SO2', data = houston_pollution)
# Arrow start and annotation location
plt.annotate('A buried point to look at', xy=(45.5,11.8), xytext=(60,22),
  # Arrow configuration and background box
  arrowprops={'facecolor':'grey', 'width': 3}, backgroundcolor = 'white' )
```



Let's annotate

IMPROVING YOUR DATA VISUALIZATIONS IN PYTHON