Data Visualization 101

Data Science and Machine Learning Team

12/18/2020

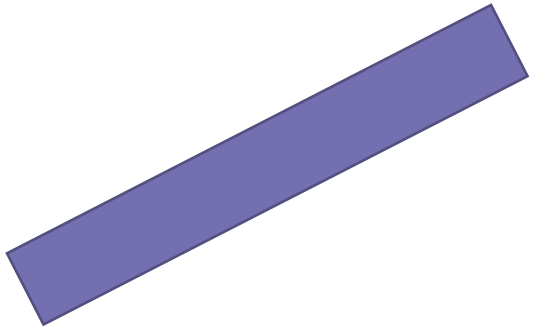
Andrew Wheeler, PhD

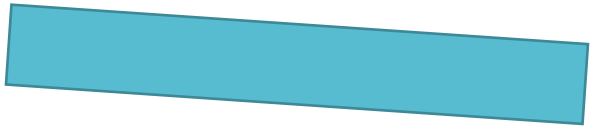
andrew.wheeler@hms.com

Agenda

- Perceptual Aspects of Data Viz
- Advice for Organizing Tables
- Color Advice for Tables/Graphs
- Example Making nice tables & graphs in Python

Which Rectangle is Longer?

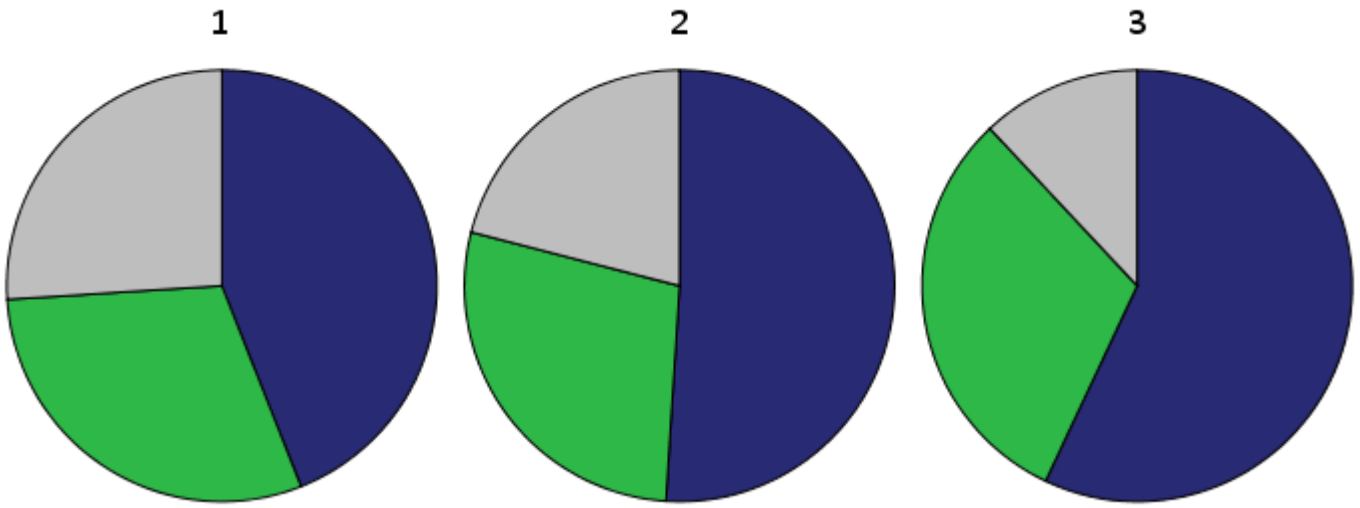




Easier to make comparisons when elements are *aligned*

Graph Types

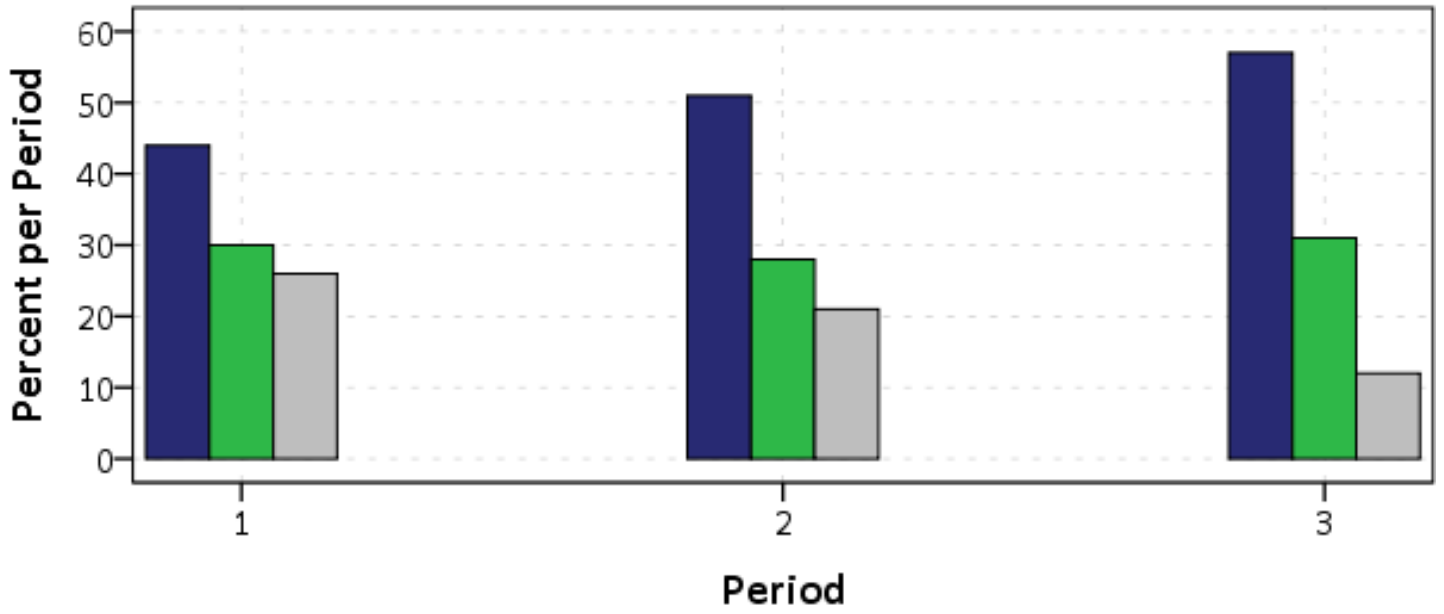
- Pie Charts are hard to make comparisons between



- Is green bigger in period 2 or period 1?
- How *much* bigger is blue than green in period 1?

Graph Types

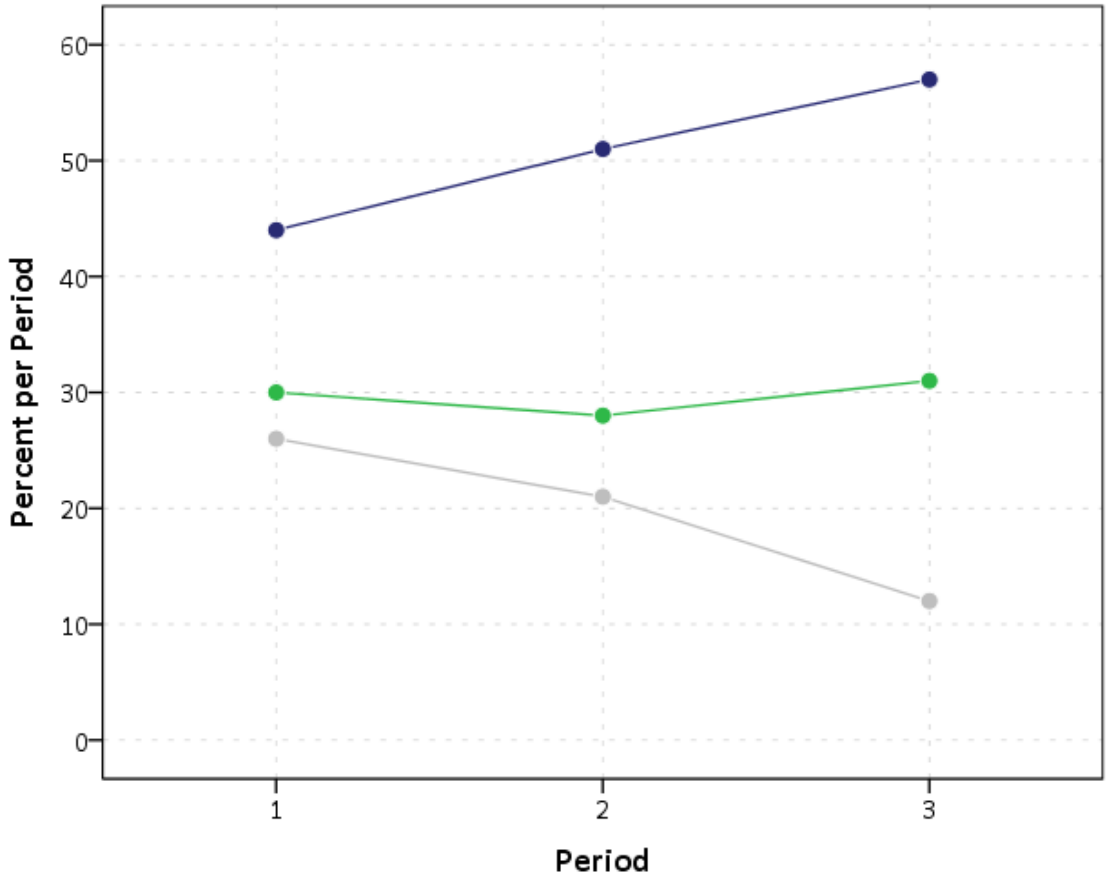
- Dodged bars are good for within groups, e.g. is grey bigger than green in period 1?



- But not as good for between groups, is green bigger in period 3 than period 1?

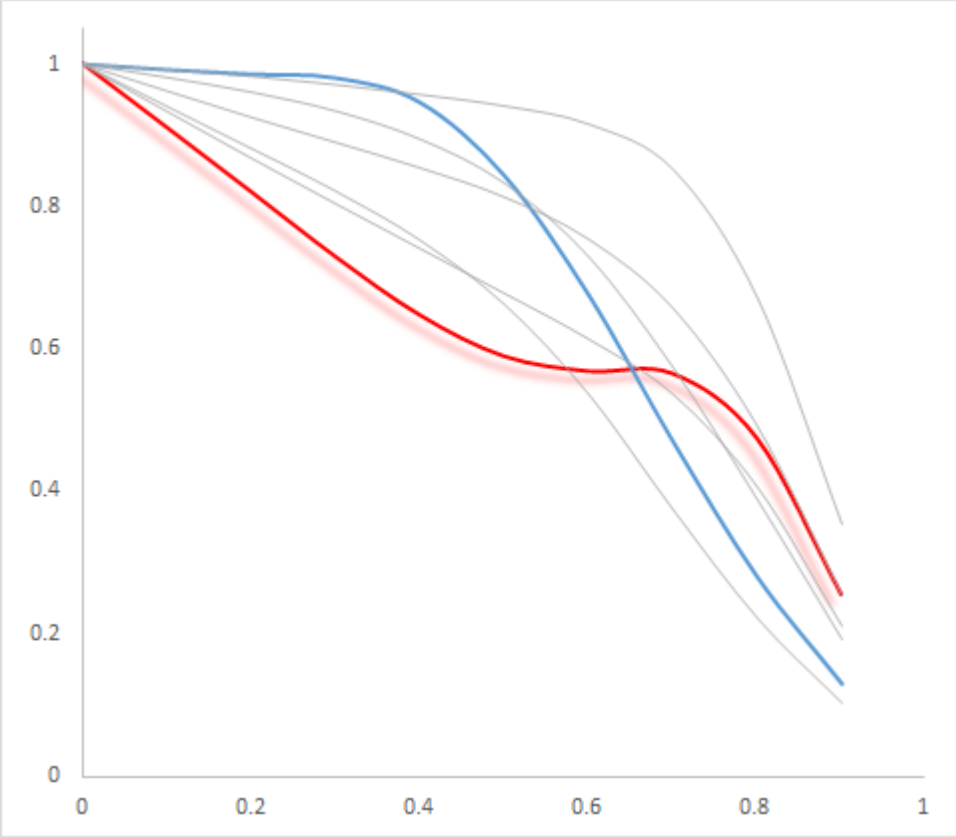
Graph Types

- I almost always like line plots over bar plots

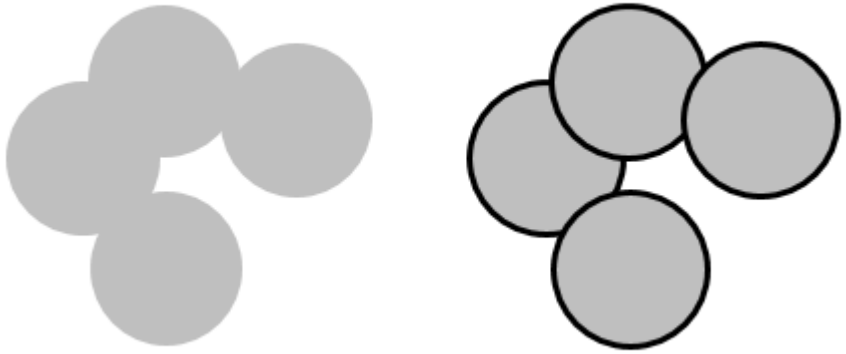


Highlighting and Object Detection

- Can use color and shading to highlight particular objects in a chart



Outlining objects makes it easier to distinguish between objects



Common Mistakes in Graphs

- Type is too small (especially for powerpoint presentations)
- Exported in low resolution format (should export in vector format or high resolution PNG – no JPEG!)
- Simple edits from the defaults – like nice variable names, decimal points for tick marks for integer data, choosing nice colors, etc.

Table Notes

- Which computation is easier?

```
1026015 - 67982
```

Or

```
  1,026,010
-     67,982
```

Easier to make comparisons within columns

Bad

cyclenum	0	1	2	3
TotalClaims	1,186	618	1,400	1,251
TotalSavings	$399,328	$158,565	$411,794	$632,563
Est. Finding	0.36	0.31	0.36	0.45

Better

cyclenum	*TotalClaims*	*TotalSavings*	*Est. Finding*
0	1,186	$ 399,328	0.36
1	618	$ 158,565	0.31
2	1,400	$ 411,794	0.36
3	1,251	$ 632,563	0.45

Sort Rows to Show Rankings

Bad

Client Name	*# Claims*	*Savings*
AETNAACAS	307	151,155
BCBSFL	999	548,166
BCBSMCOMM	900	379,336
BCBSNJ	522	254,866
BCBSTN	1,334	415,061
BCBSTNVSHP	1,146	330,493
BCN	291	63,422
BOSTONMEDICAL	2,755	1,220,836
COLORADO	354	105,299
CONNECTICUT	1,244	522,082
COVENTRY	994	288,123
HUMANA	1,470	636,774
MDWISE	93	30,794
MOLINA	75	22,602
TEXAS	999	366,418
UHS_COSMOS	2,999	937,328
UHS_NICE	417	107,760
UHS_UNET	1,998	1,214,266

Better

Client Name	*# Claims*	*Savings*
UHS_COSMOS	2,999	937,328
BOSTONMEDICAL	2,755	1,220,836
UHS_UNET	1,998	1,214,266
HUMANA	1,470	636,774
BCBSTN	1,334	415,061
CONNECTICUT	1,244	522,082
BCBSTNVSHP	1,146	330,493
TEXAS	999	366,418
BCBSFL	999	548,166
COVENTRY	994	288,123
BCBSMCOMM	900	379,336
BCBSNJ	522	254,866
UHS_NICE	417	107,760
COLORADO	354	105,299
AETNAACAS	307	151,155
BCN	291	63,422
MDWISE	93	30,794
MOLINA	75	22,602

Avoid Unnecessary Precision

Bad

Cycle	RunDate	TotalClaims	TotalSavings	EstimatedFindingRate
0	4/1/20 6:02 PM	1186.00	399327.944	0.356004368
1	5/1/20 3:54 PM	618.00	158564.7465	0.309207461
2	5/11/20 4:57 PM	1400.00	411793.5833	0.355821802
3	6/1/20 7:05 PM	1251.00	632563.4828	0.45426193
4	6/18/20 7:57 PM	1082.00	351354.2047	0.315310381
5	6/29/20 6:25 PM	2571.00	1141862.5	0.337627265
6	7/6/20 6:20 PM	7905.00	2973832.755	0.325195951
7	7/13/20 1:45 PM	5649.00	2300207.784	0.359231712
8	7/20/20 3:05 PM	2883.00	1246279.378	0.313340165
9	7/27/20 1:30 PM	2623.00	894851.4529	0.333881844
10	8/3/20 7:28 PM	5574.00	1678049.65	0.322427939
11	8/10/20 3:34 PM	6083.00	3577184.387	0.408549487
12	8/17/20 3:28 PM	8387.00	3555219.521	0.33613977
13	8/24/20 1:38 PM	2402.00	921974.7194	0.353987871
14	8/31/20 1:49 PM	4746.00	2108498.003	0.36790395

Better

Cycle	RunDate	TotalClaims	Savings per $1,000	Est. Find Rate
0	4/1/20	1,186	399	36%
1	5/1/20	618	159	31%
2	5/11/20	1,400	412	36%
3	6/1/20	1,251	633	45%
4	6/18/20	1,082	351	32%
5	6/29/20	2,571	1,142	34%
6	7/6/20	7,905	2,974	33%
7	7/13/20	5,649	2,300	36%
8	7/20/20	2,883	1,246	31%
9	7/27/20	2,623	895	33%
10	8/3/20	5,574	1,678	32%
11	8/10/20	6,083	3,577	41%
12	8/17/20	8,387	3,555	34%
13	8/24/20	2,402	922	35%
14	8/31/20	4,746	2,108	37%

Color Notes

- Around 8% of the male population is red/green colorblind
- Avoid yellow for printing or projected presentations

Bad

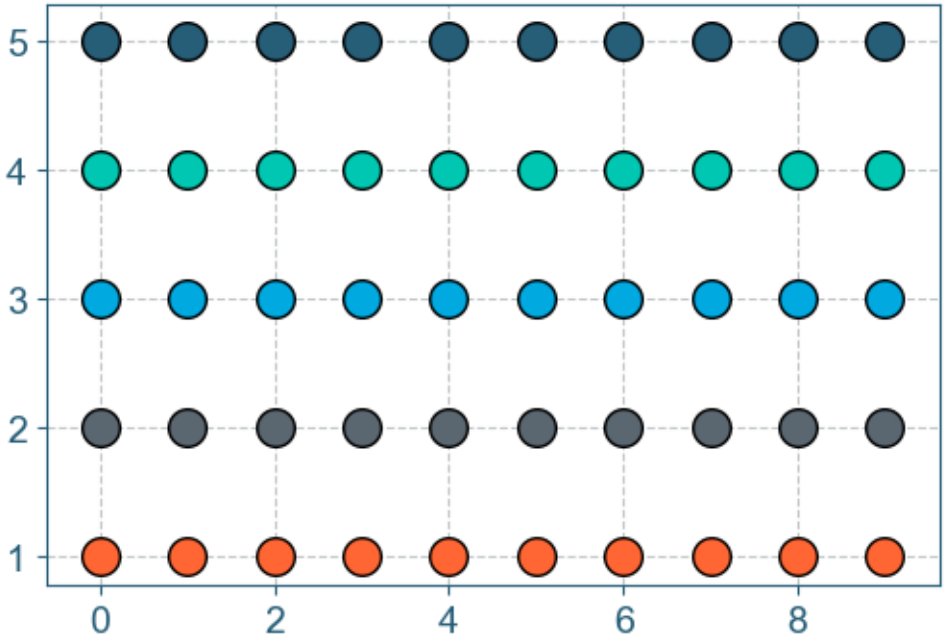
Ward	Month to Date			Year to Date		
	2012	2013	% Chg	2012	2013	% Chg
Ass-Agg	12	14	16.7%	93	94	1.1%
Auto Thft	4	1	-75.0%	26	25	-3.8%
Burg-Non	3	1	-66.7%	11	9	-18.2%
Burg-Res	36	30	-16.7%	191	189	-1.0%
Burg-Bus	7	2	-71.4%	23	16	-30.4%
Hom	0	0	0.0%	0	2	200.0%
Larceny	42	36	-14.3%	262	262	0.0%
Rape	1	0	-100.0%	5	4	-20.0%
Rob-Bus	0	1	100.0%	1	4	300.0%
Rob-In	3	3	0.0%	31	46	48.4%
Total	108	88	-18.5%	643	651	1.2%

Better

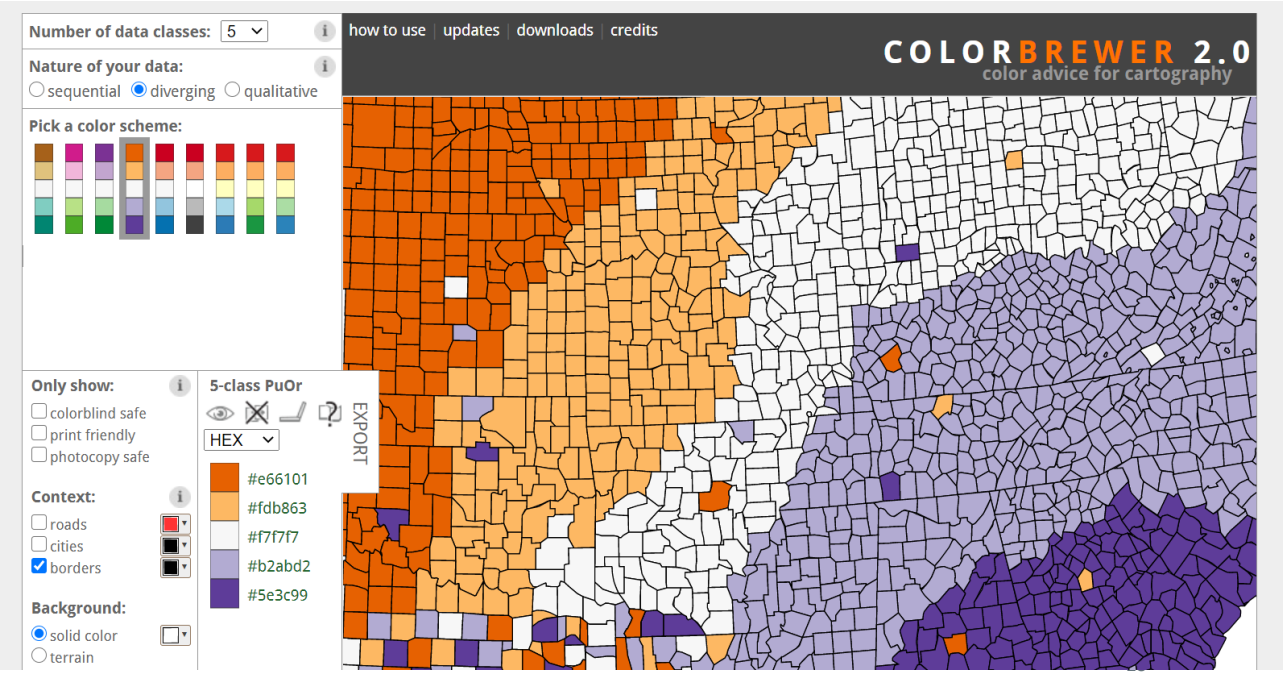
Ward	Month to Date			Year to Date		
	2012	2013	Z	2012	2013	Z
Hom	0	0	0	**0**	**2**	**3**
Rob-Bus	0	1	2	1	4	2
Rob-In	3	3	0	31	46	2
Rape	1	0	-2	5	4	0
Ass-Agg	12	14	1	93	94	0
Larceny	42	36	-1	262	262	0
Auto Thft	4	1	-2	26	25	0
Burg-Res	36	30	-1	191	189	0
Burg-Non	3	1	-1	11	9	-1
Burg-Bus	7	2	-2	23	16	-2
Total	108	88	-2	643	651	0

Choosing Colors

- Qualitative Color Palettes (HMS Marketing materials is good!)



- Continuous Color Palettes consult ColorBrewer.org or perceptual uniform (e.g. viridis, plasma)
- Avoid rainbow color palette



Example Using Python

- What we will be doing today

1) Load in data
2) Create basic line graph
3) Improve Style of Graph
4) Export graph to PNG file

- Simple aggregate table from my Payment Integrity Project
- Notebook available at https://github.com/hmsholdings/data-science-utils/tree/master/education/Intro_DataScience/DataViz_101

Loading in Data

This is some example aggregate data from my Payment Integrity DSML Project

In [1]:
```
# Import all the libraries
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import sys
import os

dat_dir = r'C:\Users\e009156\Documents\GitHub\data-science-utils\education\Intro_DataScience\DataViz_101'
os.chdir(dat_dir)
sys.path.append(dat_dir)
import hms_plotstyle #my person python plotting functions

time_dat = pd.read_excel('ExampleTables.xlsx', sheet_name='Original')
time_dat
```

Out[1]:

	cyclenum	RunDate	TotalClaims	TotalSavings	AverageSavings	EstimatedFindingRate
0	0	2020-04-01 18:02:38.054	1186	3.993279e+05	336.701471	0.356004
1	1	2020-05-01 15:54:17.446	618	1.585647e+05	256.577260	0.309207
2	2	2020-05-11 16:57:34.134	1400	4.117936e+05	294.138274	0.355822
3	3	2020-06-01 19:05:02.054	1251	6.325635e+05	505.646269	0.454262
4	4	2020-06-18 19:57:53.253	1082	3.513542e+05	324.726622	0.315310
5	5	2020-06-29 18:25:44.647	2571	1.141862e+06	444.131661	0.337627
6	6	2020-07-06 18:20:50.294	7905	2.973833e+06	376.196427	0.325196

A Nicer Printed Table Format

In [2]:
```
#Some nice print format functions for tables
format_dict = {'RunDate':'{:%m-%d-%Y}', 'TotalClaims': '{:,}', 'TotalSavings':
               'AverageSavings': '${:,.2f}', 'EstimatedFindingRate': '{:.2f}'}

time_dat.style.format(format_dict).hide_index()
```

Out[2]:

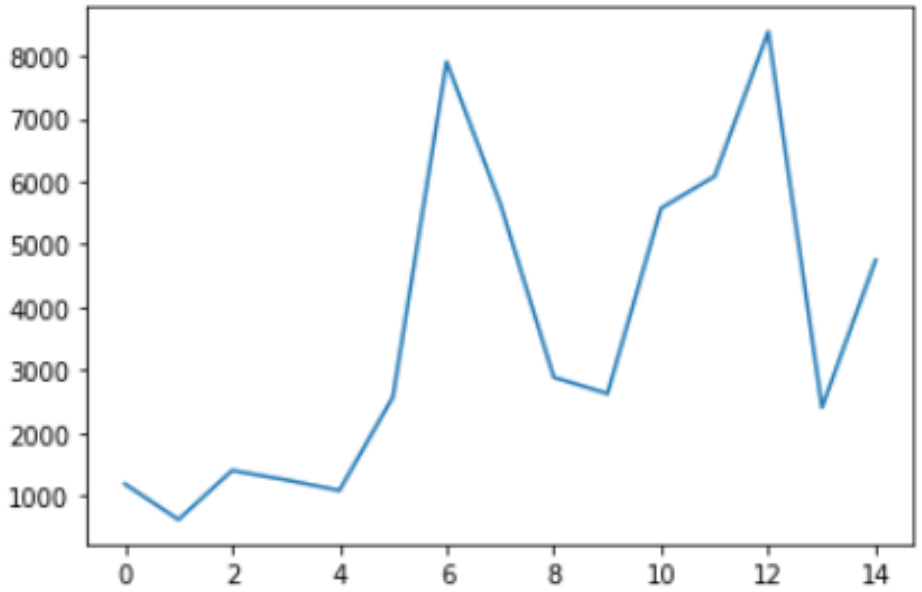
cyclenum	RunDate	TotalClaims	TotalSavings	AverageSavings	EstimatedFindingRate
0	04-01-2020	1,186	$399,328	$336.70	0.36
1	05-01-2020	618	$158,565	$256.58	0.31
2	05-11-2020	1,400	$411,794	$294.14	0.36
3	06-01-2020	1,251	$632,563	$505.65	0.45
4	06-18-2020	1,082	$351,354	$324.73	0.32
5	06-29-2020	2,571	$1,141,862	$444.13	0.34
6	07-06-2020	7,905	$2,973,833	$376.20	0.33
7	07-13-2020	5,649	$2,300,208	$407.19	0.36
8	07-20-2020	2,883	$1,246,279	$432.29	0.31
9	07-27-2020	2,623	$894,851	$341.16	0.33
10	08-03-2020	5,574	$1,678,050	$301.05	0.32
11	08-10-2020	6,083	$3,577,184	$588.06	0.41
12	08-17-2020	8,387	$3,555,220	$423.90	0.34
13	08-24-2020	2,402	$921,975	$383.84	0.35
14	08-31-2020	4,746	$2,108,498	$444.27	0.37

You can also limit the columns shown by doing:

```
time_dat[['cyclenum','RunDate']]
```

Making a Simple Line Graph

```
In [3]: #Making a simple line graph
        fig, ax = plt.subplots()
        ax.plot(time_dat['cyclenum'], time_dat['TotalClaims'])
        plt.show()
```



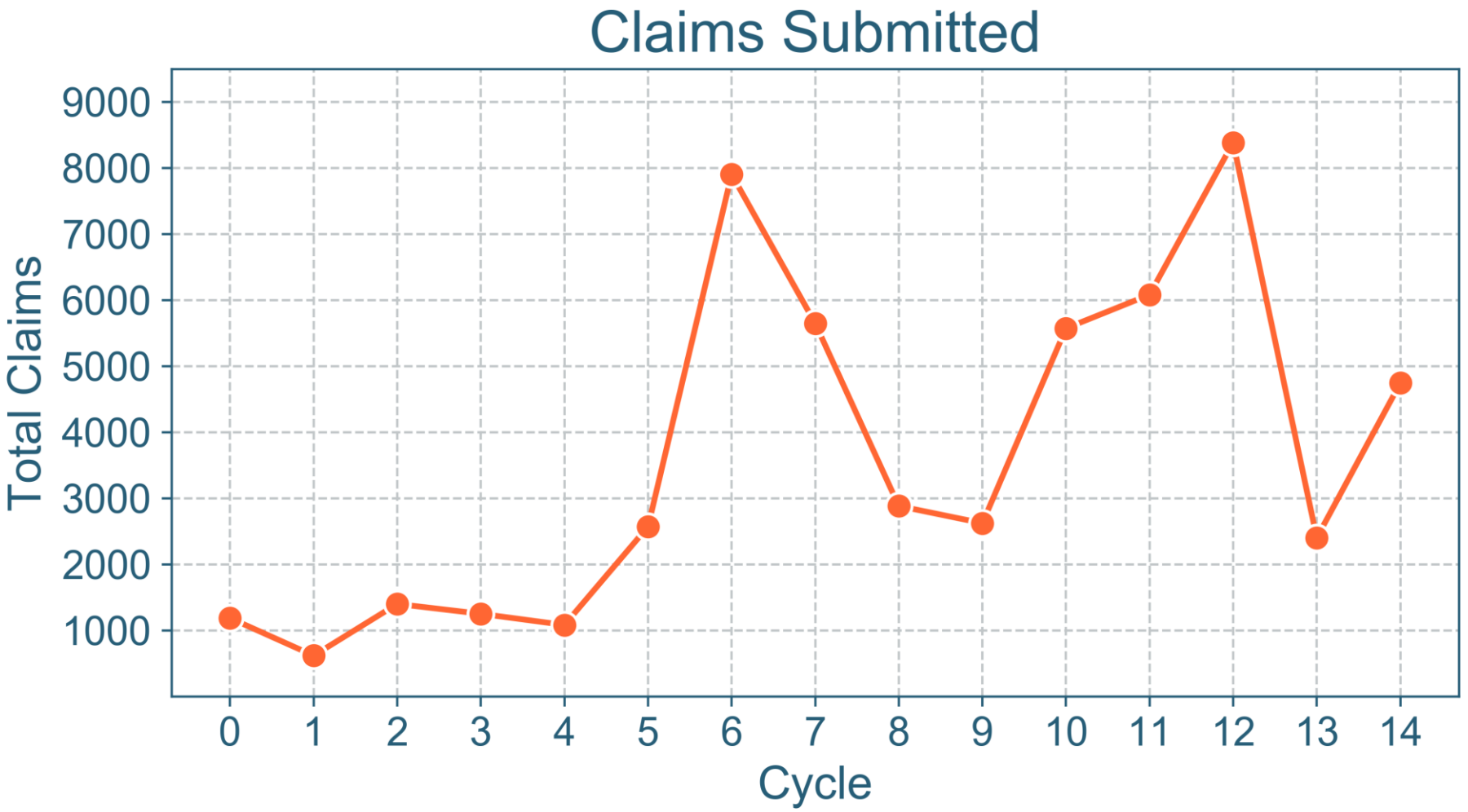

```
Matplotlib default line graph

plt.show() #shows image in
jupyter notebook or spyder
```

Changing the Default Matplotlib Settings

In [4]:
```
#Updating the HMS plot style
matplotlib.rcParams.update(hms_plotstyle.hms_style)

#Redoing the same chart, updating nicer names and styles
fig, ax = plt.subplots(figsize=(8,4)) #making the dimensions bigger and wider

#Setting the style for the line and marker
ax.plot(time_dat['cyclenum'], time_dat['TotalClaims'],
        marker='o', markeredgecolor='w',
        linewidth=2, markersize=9)

#X, Y axis and title
ax.set_ylabel('Total Claims')
ax.set_xlabel('Cycle')
plt.title('Claims Submitted')

#Setting y limit to include 0
ax.set_ylim(0, 9500)

#Setting the X and Y ticks
plt.xticks(range(time_dat['cyclenum'].max()+1))
plt.yticks(np.linspace(1000,9000,9))

#Saving the file to a high res PNG file
plt.savefig('LineChart.png', dpi=500, bbox_inches='tight')
plt.show()
```

Plot on the next page!

Changing the Default Matplotlib Settings



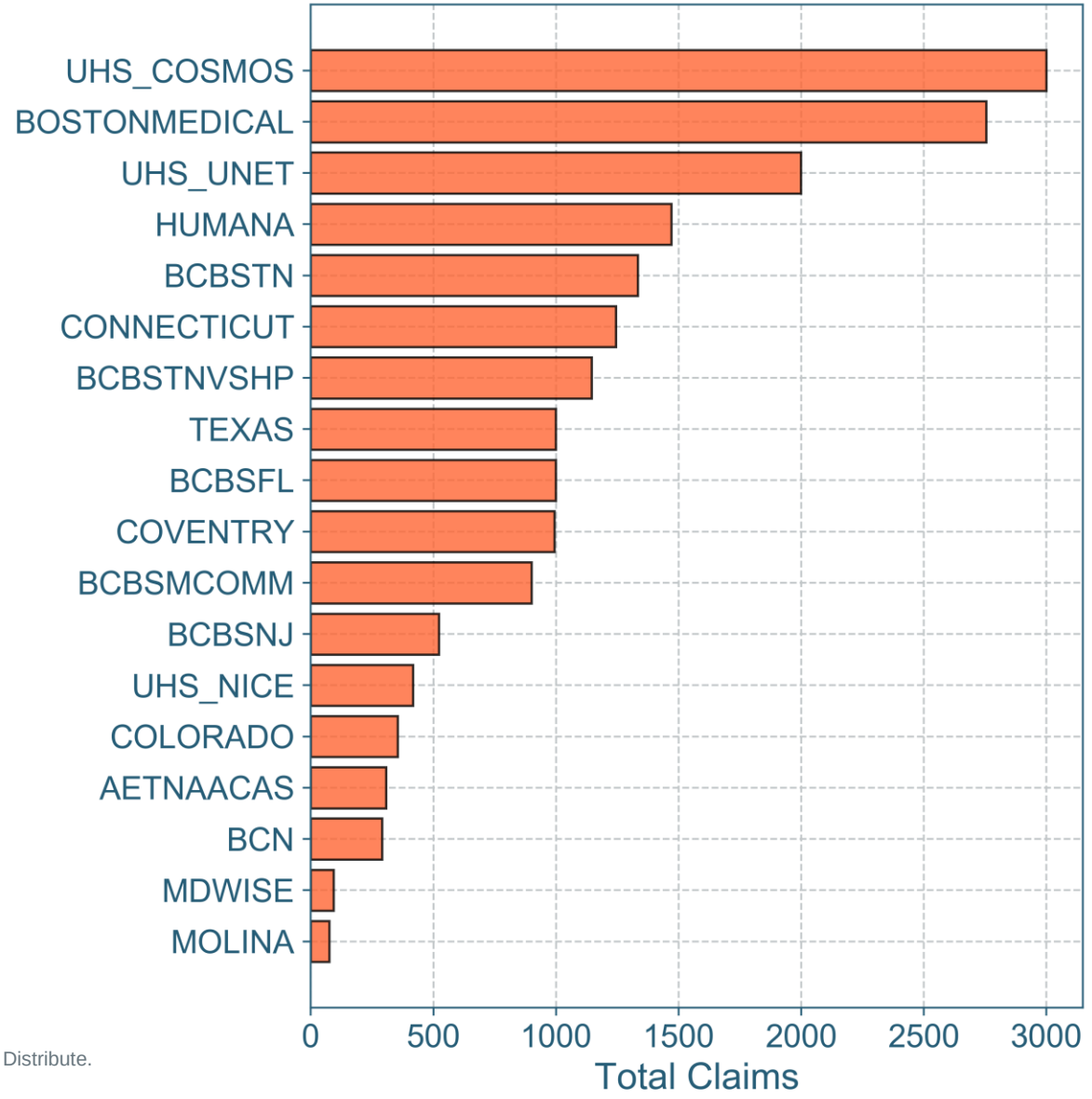
Making the line thicker and adding in points makes the trend line stand out more.

Example Bar Chart

In [5]:
```
#This is another set of aggregate data, but by client instead of time period
bar_dat = pd.read_excel('ExampleTables.xlsx', sheet_name='ClientName_Orig')

#Ordering the data
bar_dat.sort_values(by='# Claims', inplace=True)

#Plot the finding rate per client
fig, ax = plt.subplots(figsize=(6,8))
ax.barh(bar_dat['Client Name'], bar_dat['# Claims'],
        edgecolor='black', alpha=0.8)
ax.set_xlabel('Total Claims')
plt.savefig('BarChart.png', dpi=500, bbox_inches='tight')
plt.show()
```

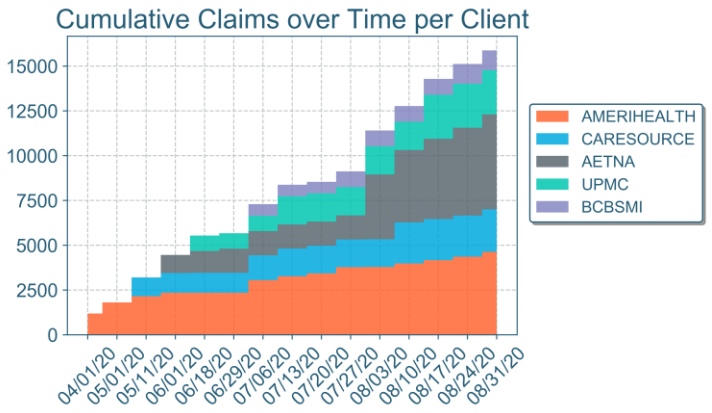
Ordering the bar chart works similar to tables. It highlights the top/bottom clients.

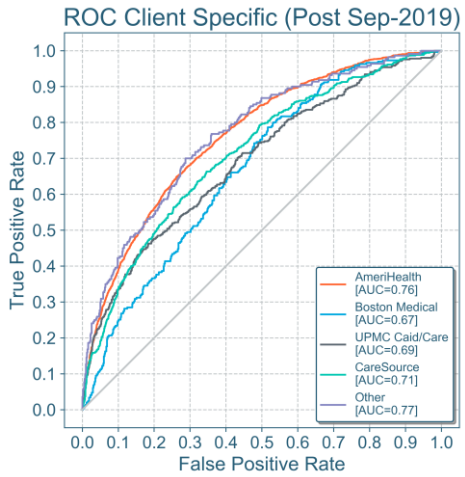
Example Bar Chart

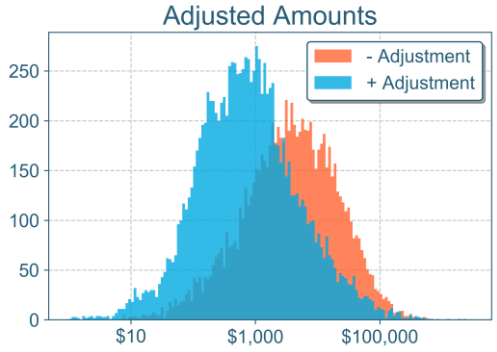


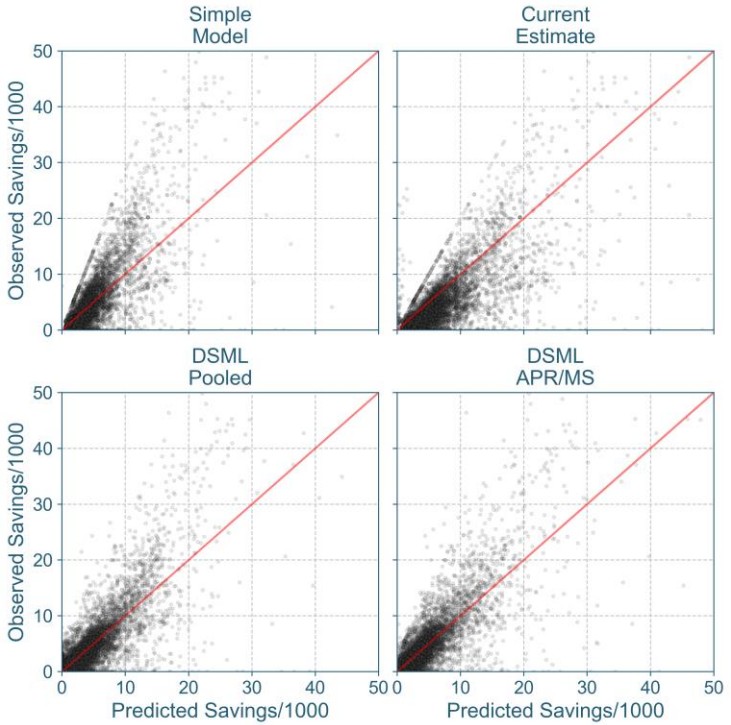
Horizontal bars allow you to read the names of the clients easier than vertical bars.

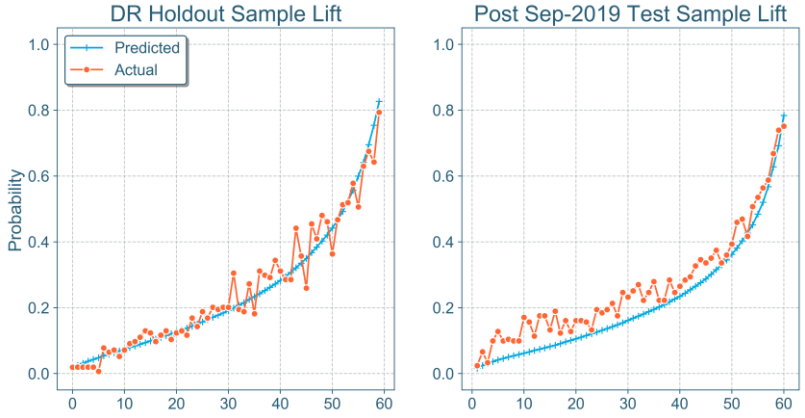
Example Gallery of Python Graphs

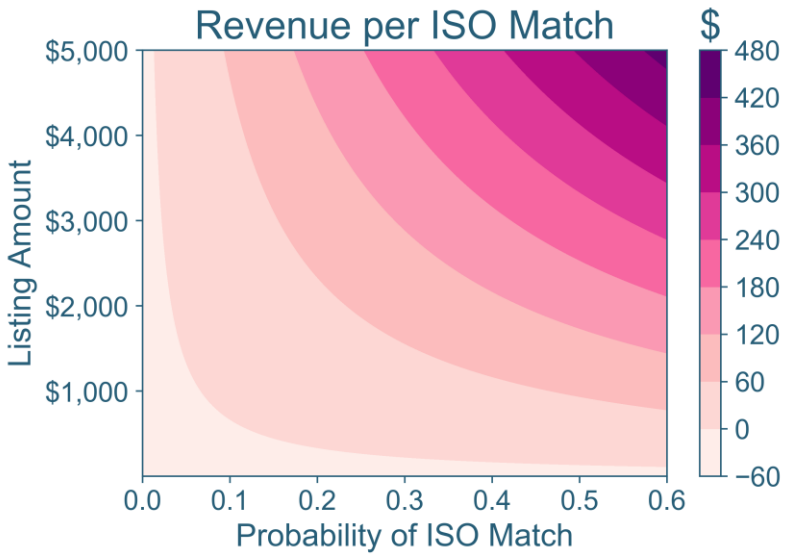




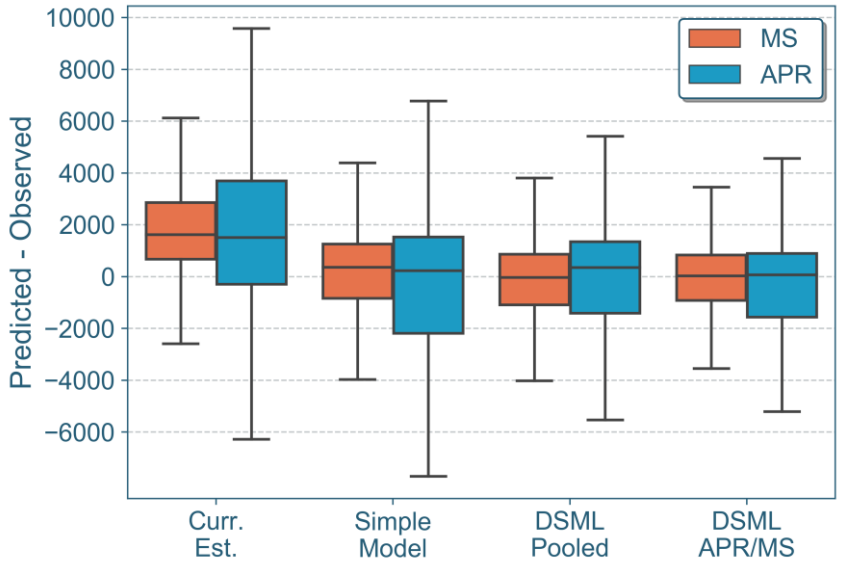








Revenue is 15% of listing minus $10



More Resources

- My blog posts on matplotlib
  - Notes on matplotlib and seaborn charts
  - Histogram notes
  - Notes on making scatterplots in Matplotlib
- More examples of HMS Plot style I created, on github
- My favorite introductory Data Viz. book is Albert Cairo's *The Functional Art*.

Questions?

Future Topics

Have requests?

Let me know!

Introduction to Data Science Course Outline

Andrew Wheeler, PhD, andrew.wheeler@hms.com

- Lesson 01: Data Science 101
- Lesson 02: Machine Learning 101
- Lesson 03: Evaluating Predictions
- Lesson 04: Intro Data Transformation in Python
- Lesson 05: Data Visualization 101
- Lesson 06: Feature Engineering
- Lesson 07: Missing Data
- Lesson 08: Big Data and Parallel Computing Intro
- Lesson 09: Dimension Reduction and Unsupervised Learning
- Lesson 10: High Cardinality (Many Categories)
- Lesson 11: Intro to Forecasting
- Lesson 12: Conducting Experiments

Data Visualization 101

Data Science and Machine Learning Team

12/18/2020

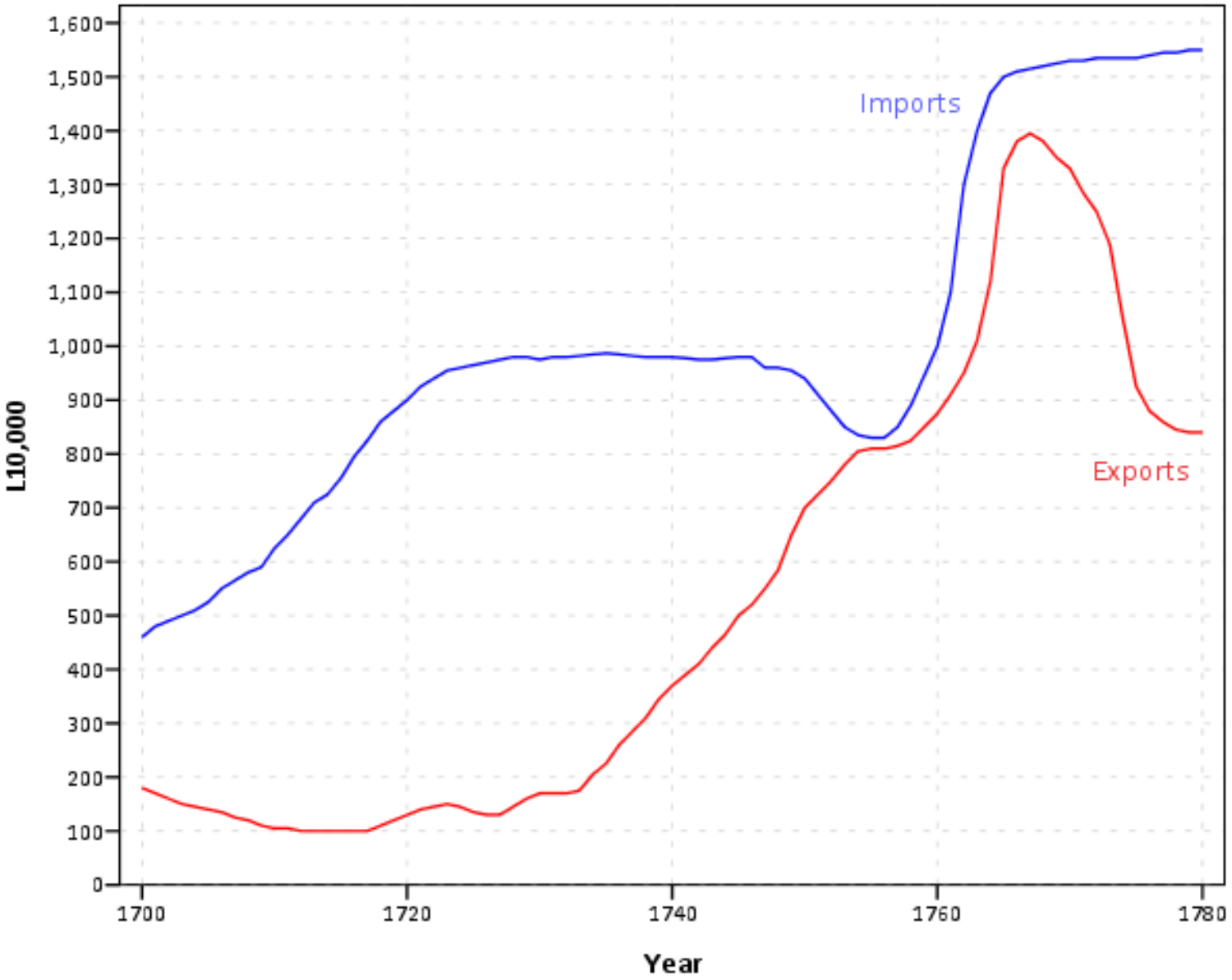
Andrew Wheeler, PhD

andrew.wheeler@hms.com

Graph Types

- Differences between lines are hard to tell

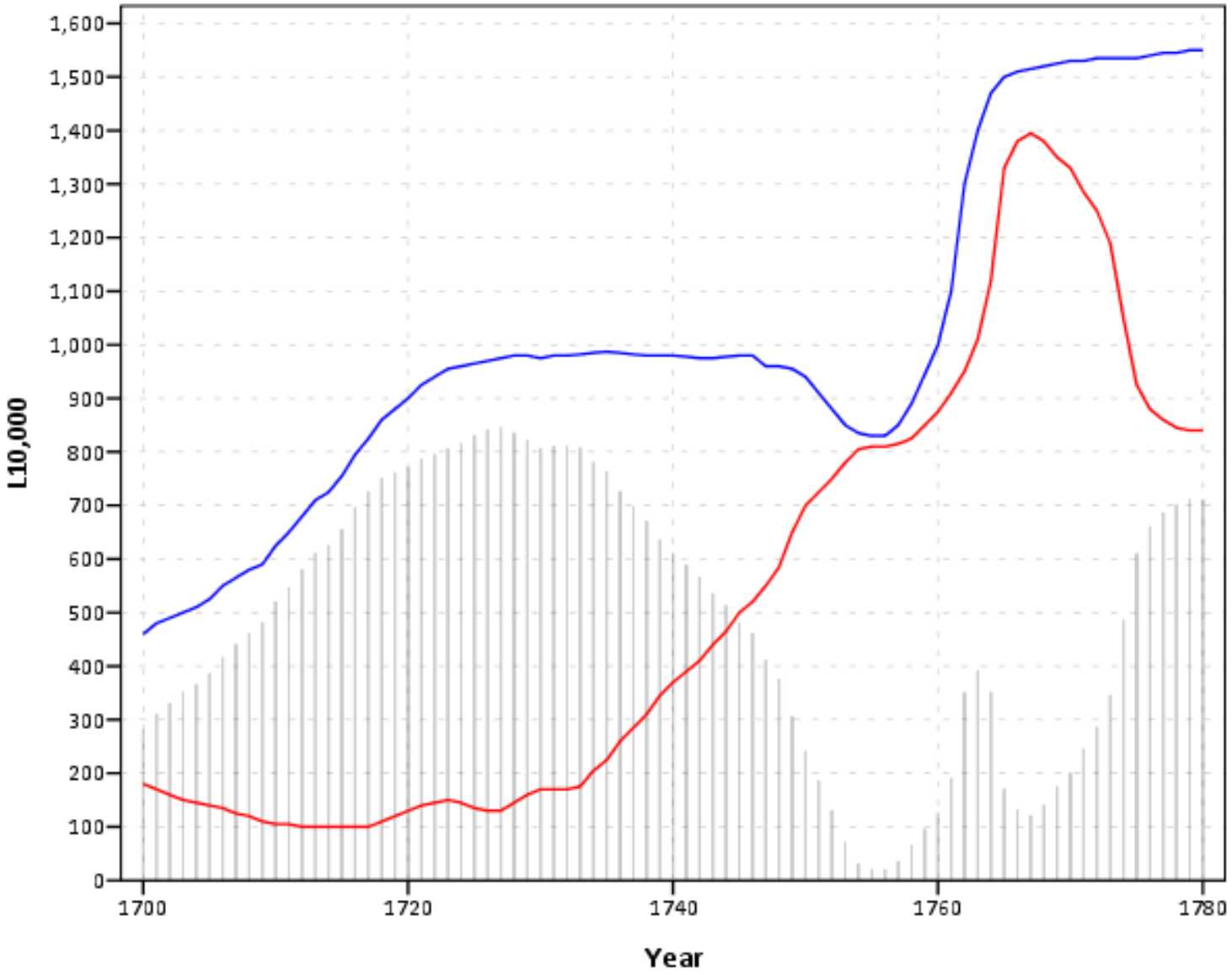
- The differences in imports/exports around 1760 are small correct?



Graph Types

- Differences between lines are hard to tell

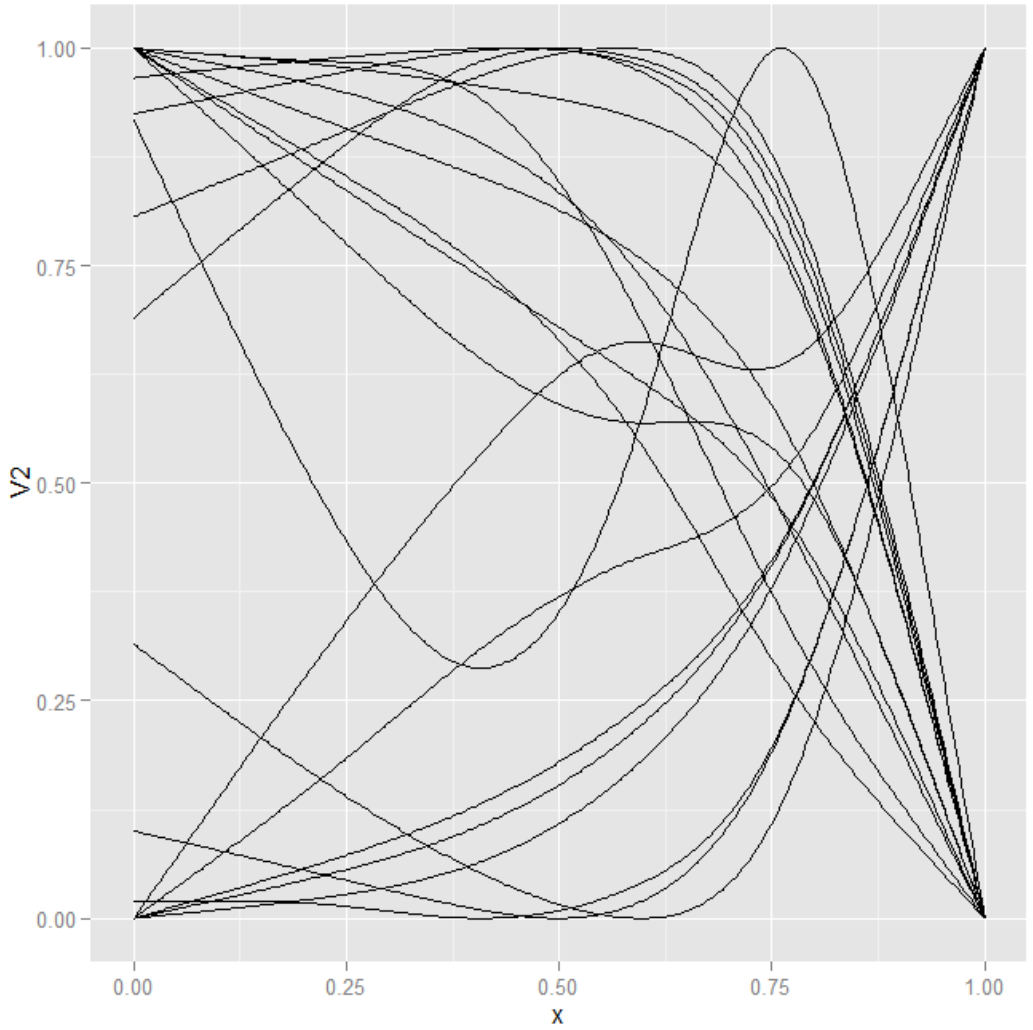
- The differences in imports/exports around 1760 are small correct?



Red line is exports, Blue line is imports, grey bars are Imports minus exports.

Making Complicated Plots More Readable

- Many lines are difficult to disentangle



Small Multiples

- Consider making a small multiple graph