visualizations

Midterm Take-Home Assignment

- Your dataset is at: midterm/ext_lt_intratrd.tsv
- This is an **individual** assignment – no sharing
- Available now, submit to Camino by next Friday evening
- 5 questions (on next page)

Midterm Take-Home Questions:

1. Which **EU country** has the highest average intra-EU trade from 1999-2015 and what is this annual average?
2. Which **year** has the highest amount of intra-EU trade between 1999-2015 and what is the total amount?
3. Which EU country has **the largest increase** from 1999-2015 in terms of intra-EU trade? Macro-economically, what is your hypothesis behind this increase?
4. Which EU country has **the largest decrease** from 1999-2015 in terms of intra-EU trade? Macro-economically, what is your hypothesis behind this drop?
5. Analyze this dataset as a DataFrame and describe an **insight** that you are able to deduce.

4 visualizations

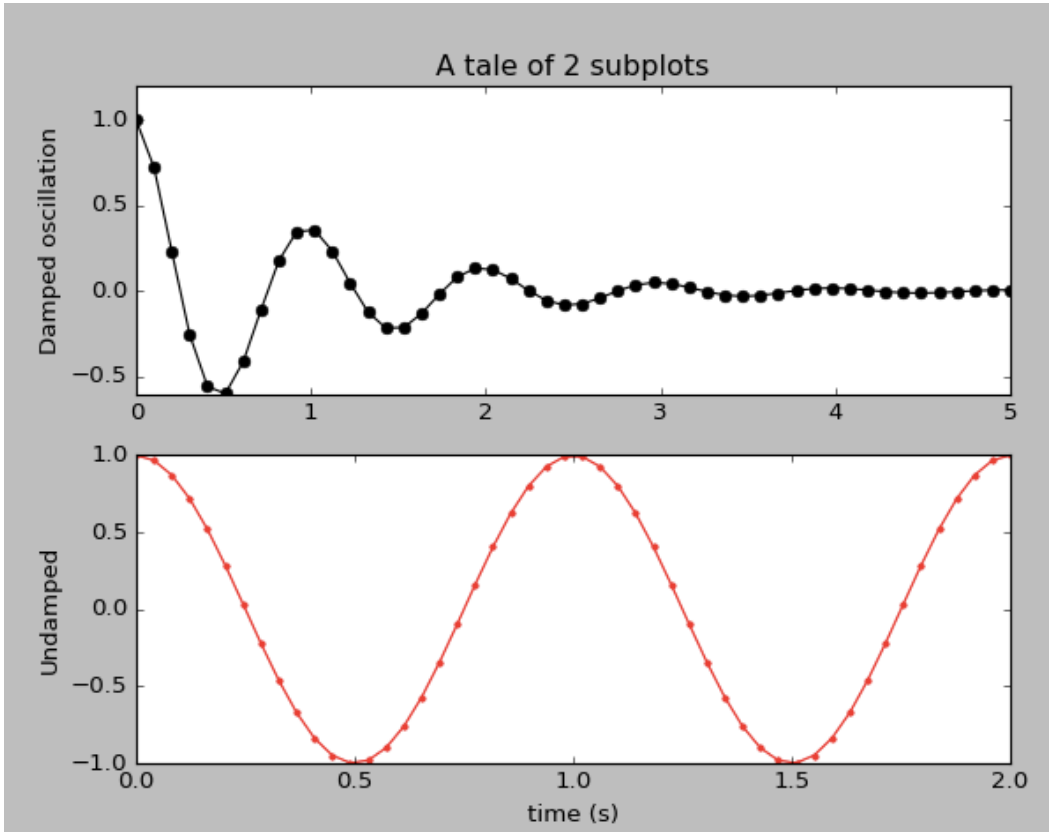
Choosing among visualization tools

- **Tableau** : no-code, drag and drop Excel type analysis
- **Matplotlib** : basic plotting -- bars, pies, lines, scatter plots, etc.
- **Seaborn** : statistical visualization -- use it if you're creating heatmaps or somehow summarizing your data and still want to show the distribution of your data
- **Bokeh** : interactive visualization -- if your data is so complex (or you haven't yet found the "message" in your data), then use Bokeh to create interactive visualizations that will allow your viewers to explore the data themselves

Agenda

- **Matplotlib**
- Seaborn
- Bokeh
- Tableau
- Final Project

Anatomy of a matplotlib plot



Anatomy of a matplotlib plot

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'ko-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
plt.show()
```

Import libraries

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'ko-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
plt.show()
```

Anatomy of a matplotlib plot

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'ko-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
plt.show()
```

Define 2 x-axis ranges in linear space

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'ko-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
plt.show()
```

Create 2 sinusoidal functions

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'ko-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
plt.show()
```

Plot subplot #1

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'ko-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
plt.show()
```

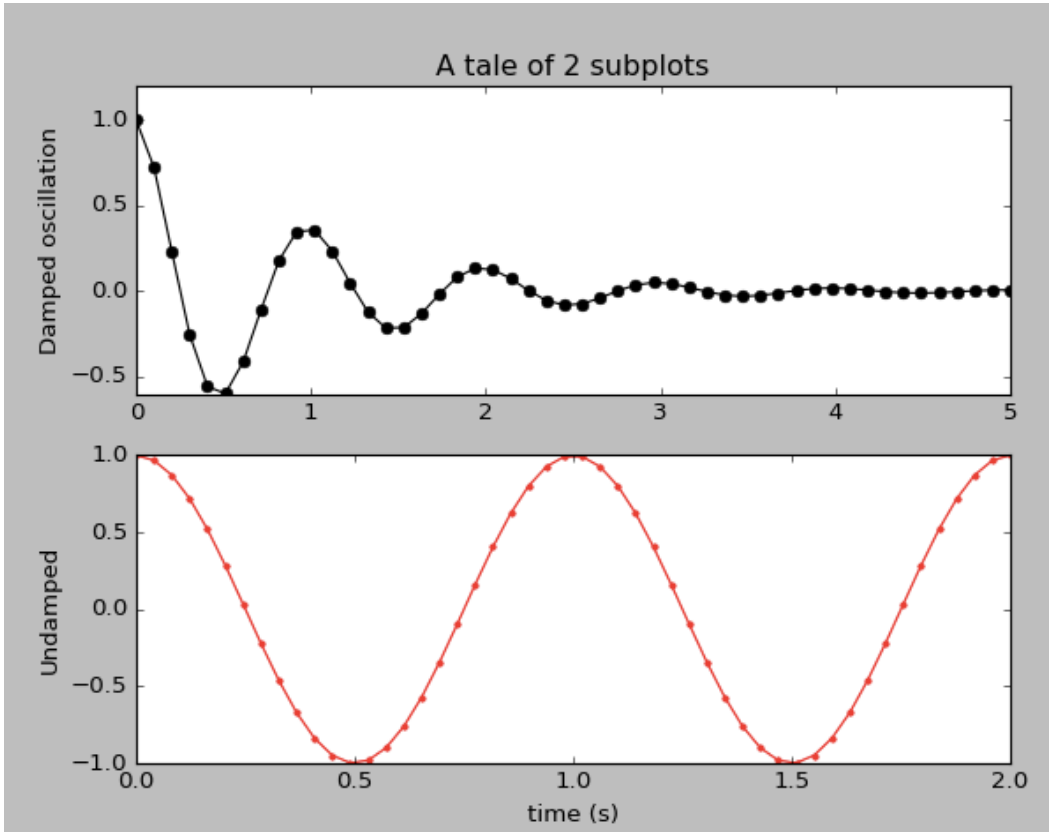
Plot subplot #2

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'ko-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
plt.show()
```

Finally: show it!

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'ko-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
plt.show()
```

Anatomy of a matplotlib plot



Agenda

- Matplotlib
- **Seaborn**
- Bokeh
- Tableau
- Final Project

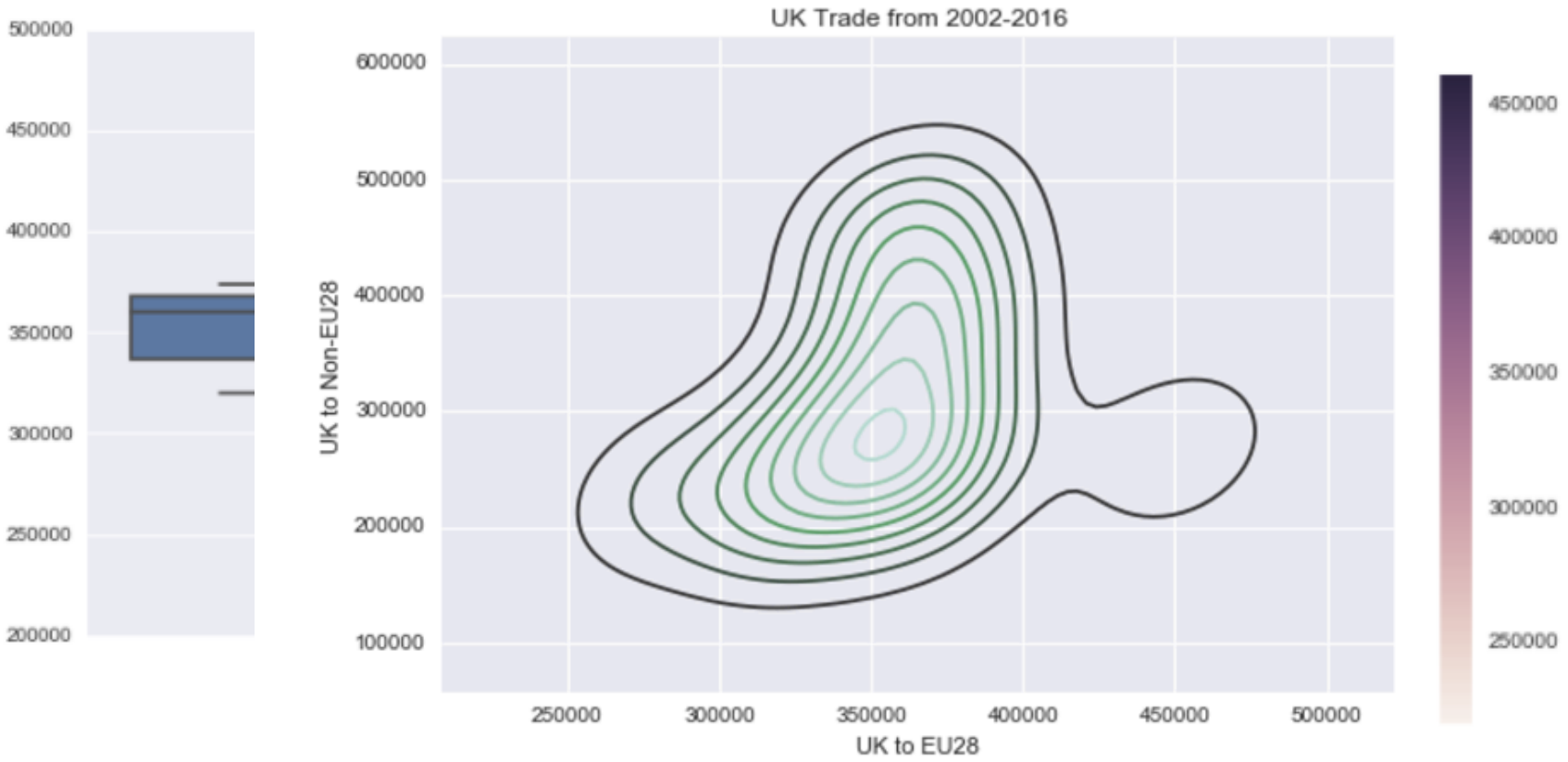
Seaborn

- Seaborn is a Python visualization library based on **matplotlib**.
- It provides a high-level interface for drawing **attractive statistical graphics**.

Seaborn : how attractive?

	trade_type	partner	2015	2014	2013	2012	2011	2010	2009
geo									
AT	Export	EU28	193043.0	187548.0	184865.0	181689.0	181889.0	166299.0	143052.0
AT	Export	EXT_EU28	82468.0	80797.0	78903.0	77668.0	73038.0	63858.0	53376.0
AT	Import	EU28	215023.0	210492.0	211444.0	212546.0	212734.0	187011.0	160875.0
AT	Import	EXT_EU28	65242.0	63511.0	64554.0	65337.0	62293.0	52877.0	44262.0
BE	Export	EU28	517836.0	502888.0	495145.0	486102.0	492083.0	449191.0	402557.0
BE	Export	EXT_EU28	201294.0	208167.0	210766.0	208074.0	191352.0	165868.0	129415.0
BE	Import	EU28	425974.0	444974.0	451816.0	461888.0	454022.0	407918.0	357932.0
BE	Import	EXT_EU28	251526.0	239457.0	228369.0	221686.0	216871.0	182226.0	150802.0
BG	Export	EU28	29752.0	27529.0	26702.0	24474.0	25378.0	19103.0	15329.0
BG	Export	EXT_EU28	16571.0	16558.0	17842.0	17067.0	15152.0	12019.0	8071.0

Seaborn : how attractive?



matplotlib_seaborn.ipynb

Agenda

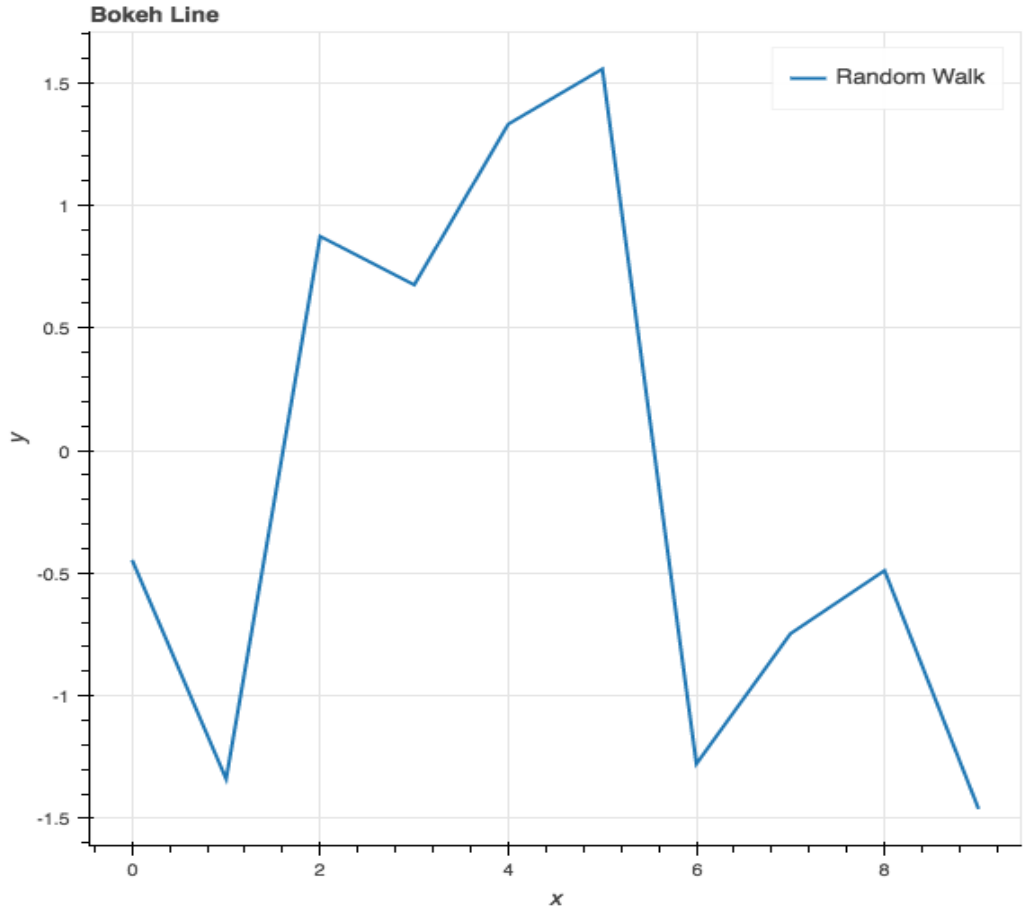
- Matplotlib
- Seaborn
- **Bokeh**
- Tableau
- Final Project

Bokeh: web interactive display

Creates interactive visualization for web presentation.

- In the style of D3.js
- High-performance interactivity over very large or streaming datasets
- Quick creation of plots, dashboards, data apps

Anatomy of a Simple Bokeh App



Anatomy of a Simple Bokeh App

```
import numpy as np
from bokeh.plotting import figure, output_file, show
a = np.arange(10)
b = np.random.randn(10)
output_file("lines.html")
p = figure(title="Bokeh Line", x_axis_label='x', y_axis_label='y')
p.line(a, b, legend="Random Walk", line_width=2)
show(p)
```

Import the libraries

```
import numpy as np
from bokeh.plotting import figure, output_file, show
a = np.arange(10)
b = np.random.randn(10)
output_file("lines.html")
p = figure(title="Bokeh Line", x_axis_label='x', y_axis_label='y')
p.line(a, b, legend="Random Walk", line_width=2)
show(p)
```

Define the x and y ranges

```
import numpy as np
from bokeh.plotting import figure, output_file, show
a = np.arange(10)
b = np.random.randn(10)
output_file("lines.html")
p = figure(title="Bokeh Line", x_axis_label='x', y_axis_label='y')
p.line(a, b, legend="Random Walk", line_width=2)
show(p)
```

Define name of the HTML file

```
import numpy as np
from bokeh.plotting import figure, output_file, show
a = np.arange(10)
b = np.random.randn(10)
output_file("lines.html")
p = figure(title="Bokeh Line", x_axis_label='x', y_axis_label='y')
p.line(a, b, legend="Random Walk", line_width=2)
show(p)
```

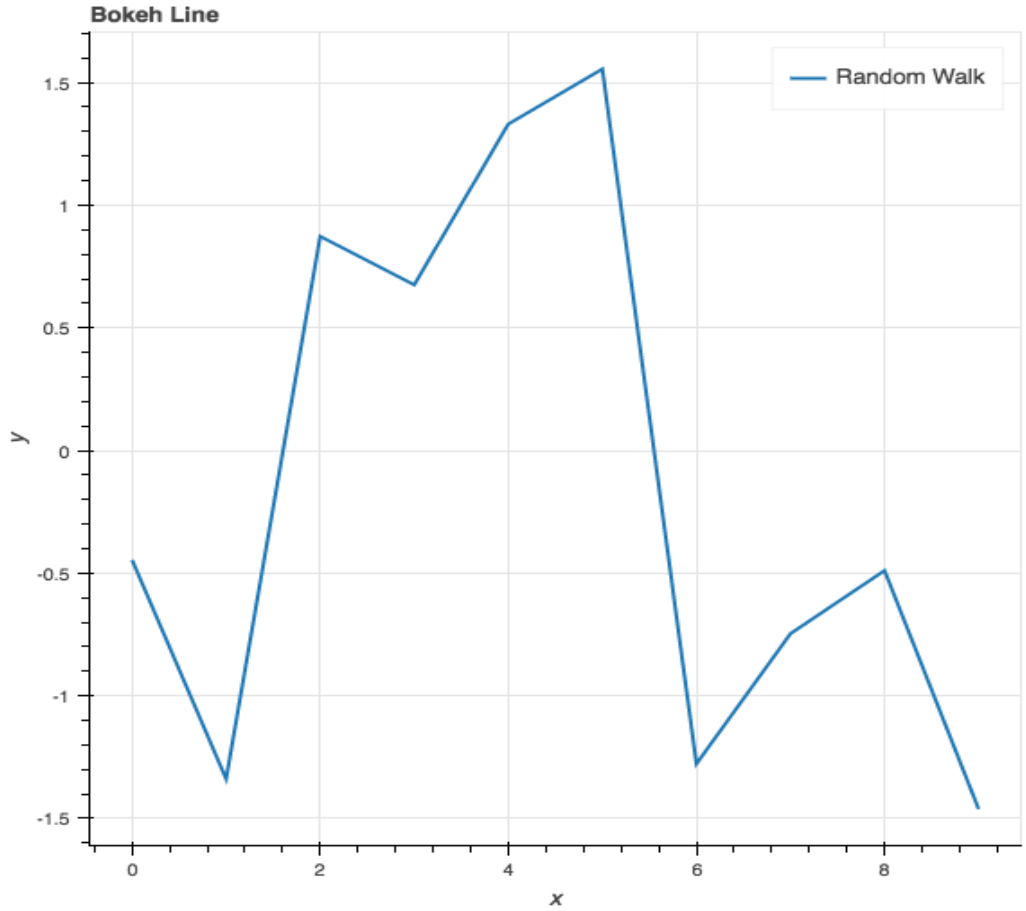
Create the actual figure

```
import numpy as np
from bokeh.plotting import figure, output_file, show
a = np.arange(10)
b = np.random.randn(10)
output_file("lines.html")
p = figure(title="Bokeh Line", x_axis_label='x', y_axis_label='y')
p.line(a, b, legend="Random Walk", line_width=2)
show(p)
```

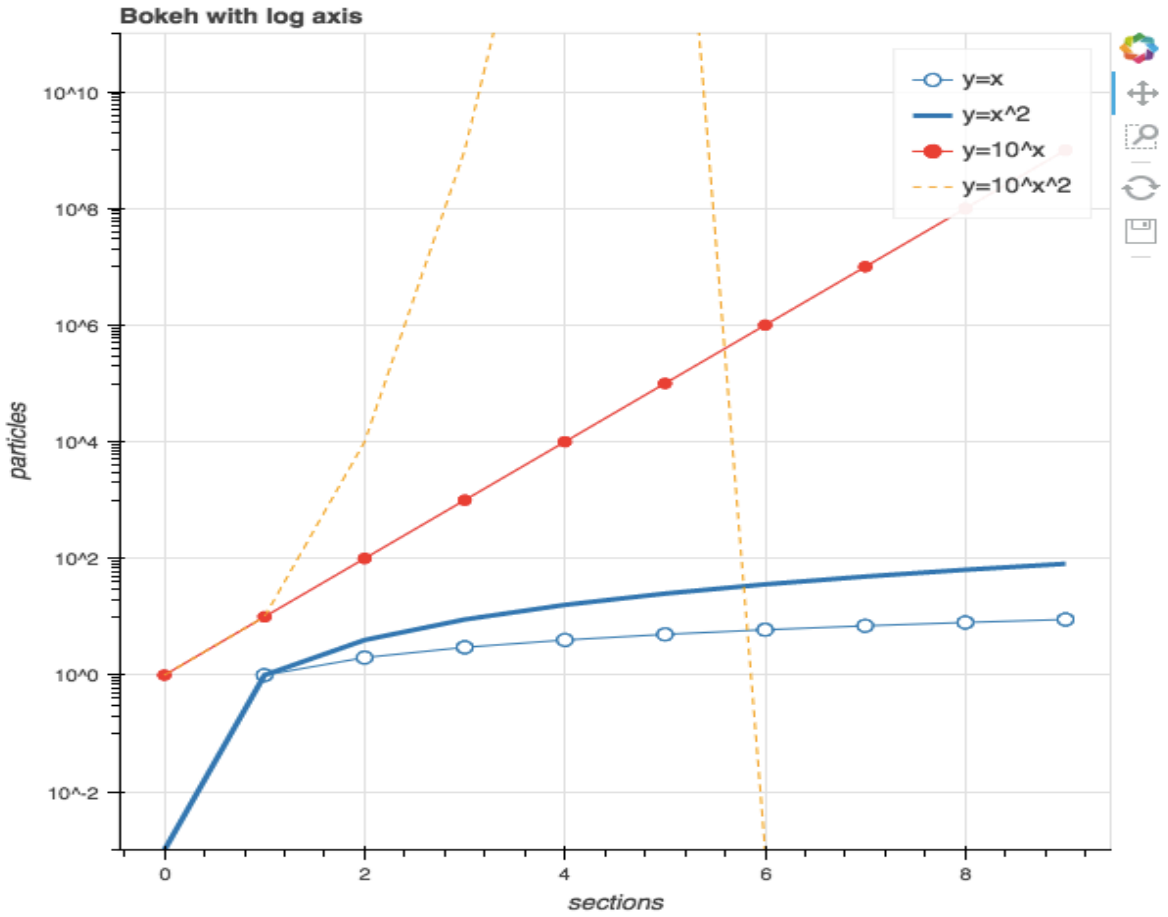
Finally, show it!

```
import numpy as np
from bokeh.plotting import figure, output_file, show
a = np.arange(10)
b = np.random.randn(10)
output_file("lines.html")
p = figure(title="Bokeh Line", x_axis_label='x', y_axis_label='y')
p.line(a, b, legend="Random Walk", line_width=2)
show(p)
```

Anatomy of a Simple Bokeh App



Anatomy of an Interactive Bokeh App



Anatomy of an Interactive Bokeh App

```
x = np.arange(10)
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]
output_file("loglines.html")
p = figure(tools="pan,box_zoom,reset,save", y_axis_type="log",
           y_range=[0.001, 10**11], title="Bokeh with log axis",
           x_axis_label='sections', y_axis_label='particles')
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x",
           fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2",
           line_color="orange", line_dash="4 4")
show(p)
```

Define x-range

```
x = np.arange(10)
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]
output_file("loglines.html")
p = figure(tools="pan,box_zoom,reset,save", y_axis_type="log",
           y_range=[0.001, 10**11], title="Bokeh with log axis",
           x_axis_label='sections', y_axis_label='particles')
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x",
           fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2",
           line_color="orange", line_dash="4 4")
show(p)
```

Define 3 separate functions in terms of x

```
x = np.arange(10)
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]
output_file("loglines.html")
p = figure(tools="pan,box_zoom,reset,save", y_axis_type="log",
           y_range=[0.001, 10**11], title="Bokeh with log axis",
           x_axis_label='sections', y_axis_label='particles')
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x",
           fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2",
           line_color="orange", line_dash="4 4")
show(p)
```

Specify name of output HTML file

```
x = np.arange(10)
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]
output_file("loglines.html")
p = figure(tools="pan,box_zoom,reset,save", y_axis_type="log",
           y_range=[0.001, 10**11], title="Bokeh with log axis",
           x_axis_label='sections', y_axis_label='particles')
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x",
           fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2",
           line_color="orange", line_dash="4 4")
show(p)
```

Specify the exact set of “tools” on graph

```
x = np.arange(10)
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]
output_file("loglines.html")
p = figure(tools="pan,box_zoom,reset,save", y_axis_type="log",
           y_range=[0.001, 10**11], title="Bokeh with log axis",
           x_axis_label='sections', y_axis_label='particles')
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x",
           fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2",
           line_color="orange", line_dash="4 4")
show(p)
```

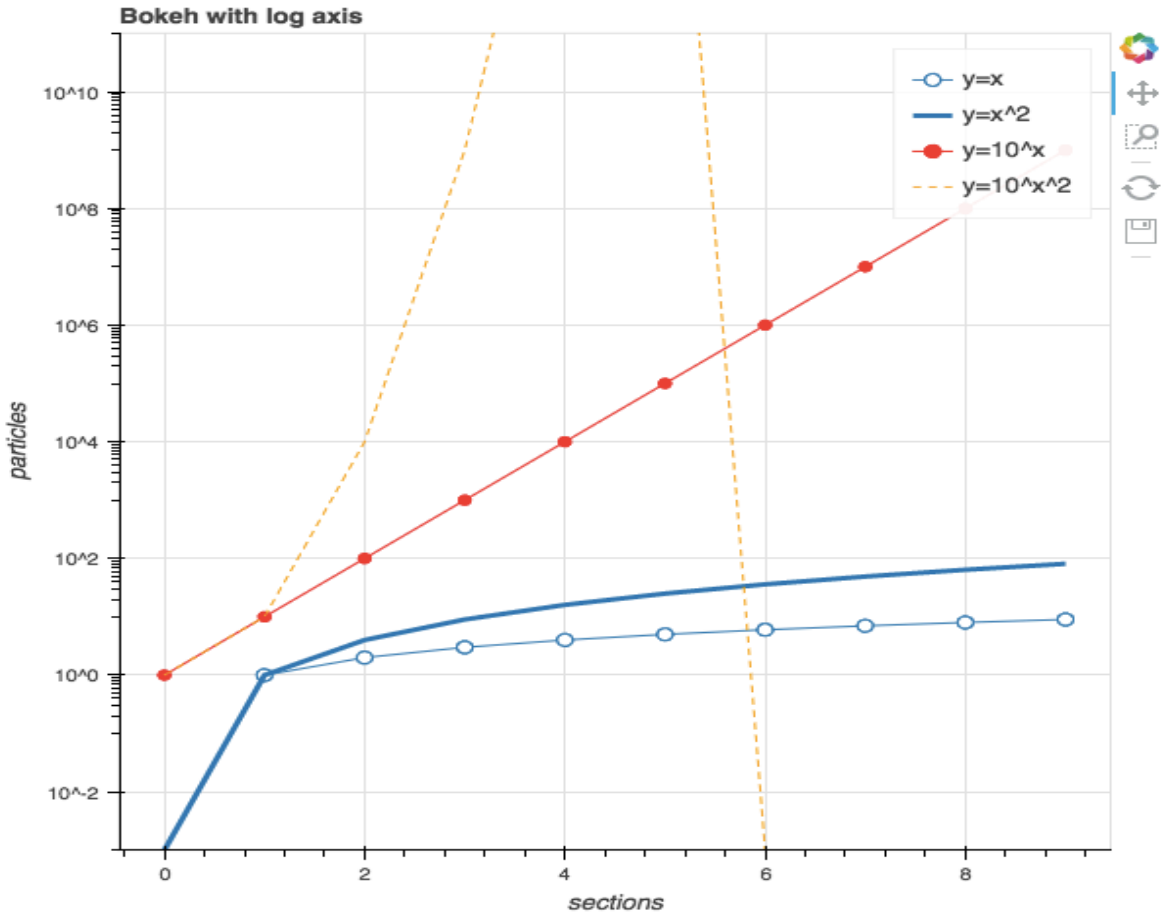
Actually creating the 3 separate graphs

```
x = np.arange(10)
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]
output_file("loglines.html")
p = figure(tools="pan,box_zoom,reset,save", y_axis_type="log",
           y_range=[0.001, 10**11], title="Bokeh with log axis",
           x_axis_label='sections', y_axis_label='particles')
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x",
           fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2",
           line_color="orange", line_dash="4 4")
show(p)
```

Finally, show it!

```
x = np.arange(10)
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]
output_file("loglines.html")
p = figure(tools="pan,box_zoom,reset,save", y_axis_type="log",
           y_range=[0.001, 10**11], title="Bokeh with log axis",
           x_axis_label='sections', y_axis_label='particles')
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x",
           fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2",
           line_color="orange", line_dash="4 4")
show(p)
```

Anatomy of an Interactive Bokeh App



Agenda

- Matplotlib
- Bokeh
- **Tableau**
- Final Project

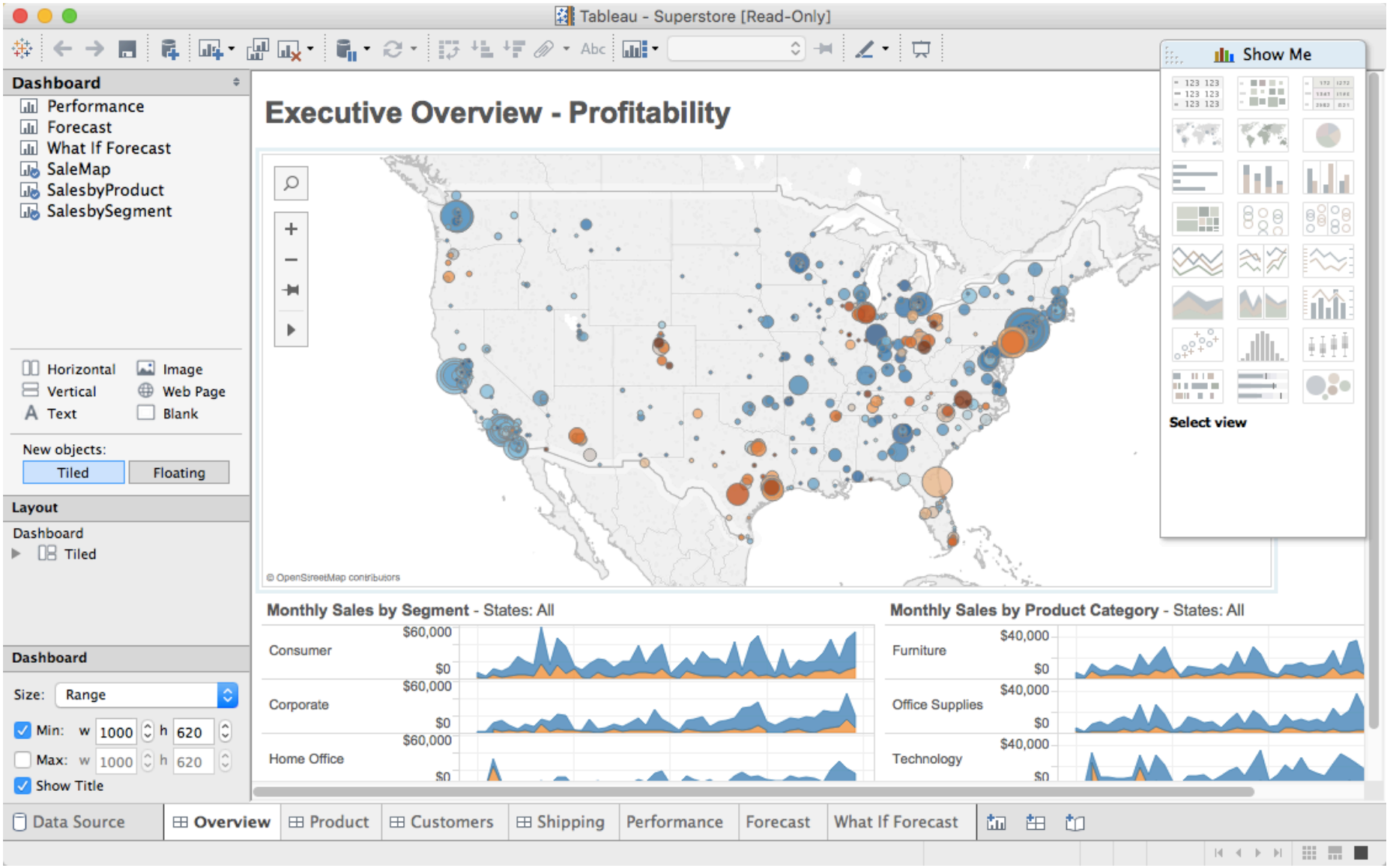


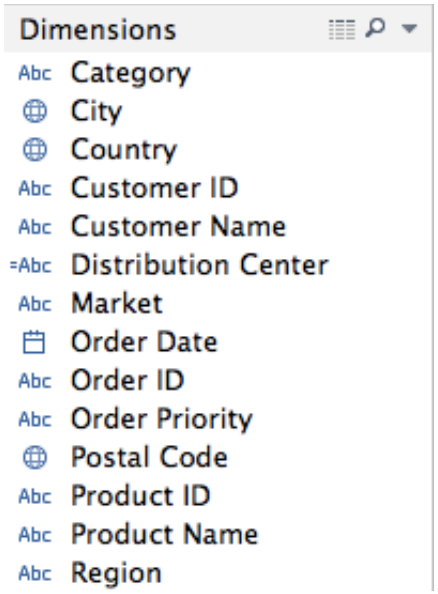
Tableau - Superstore [Read-Only]
Dashboard
Performance
Forecast
What If Forecast
SaleMap
SalesbyProduct
SalesbySegment
Horizontal
Vertical
Text
Image
Web Page
Blank
New objects:
Tiled
Floating
Layout
Dashboard
Tiled
Dashboard
Size:
Range
Min: w 1000 h 620
Max: w 1000 h 620
Show Title
Executive Overview - Profitability
© OpenStreetMap contributors
Show Me
Select view
Monthly Sales by Segment - States: All
Consumer
Corporate
Home Office
$60,000
$0
Monthly Sales by Product Category - States: All
Furniture
Office Supplies
Technology
$40,000
$0
Data Source
Overview
Product
Customers
Shipping
Performance
Forecast
What If Forecast

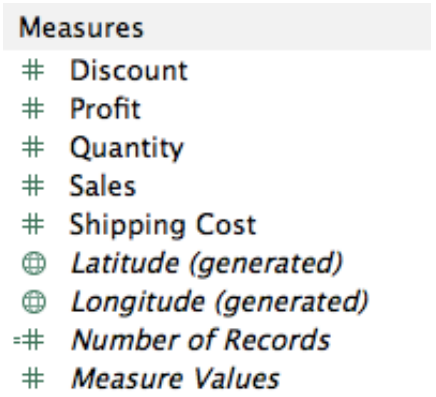
Data Ingestion

- Joins: inner, left, right, full
- Extract – Transform – Load (ETL)
- Field Transformation
- Live / Extract
- Filtering
- Large dataset & role of Tableau

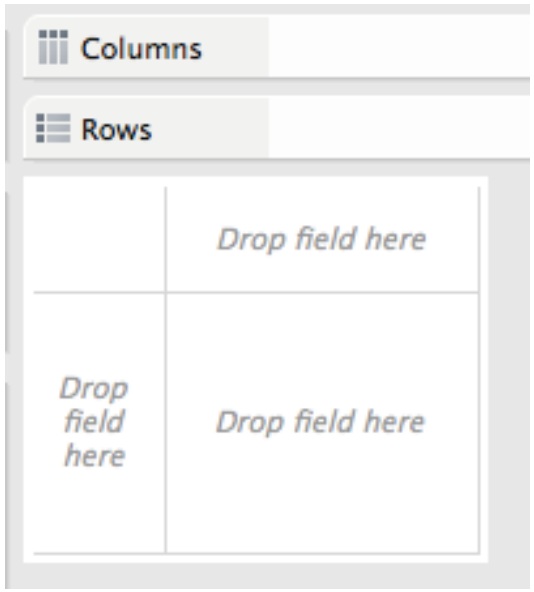
Dimensions & Measures

- Dimensions : categorical
- Measures : numerical

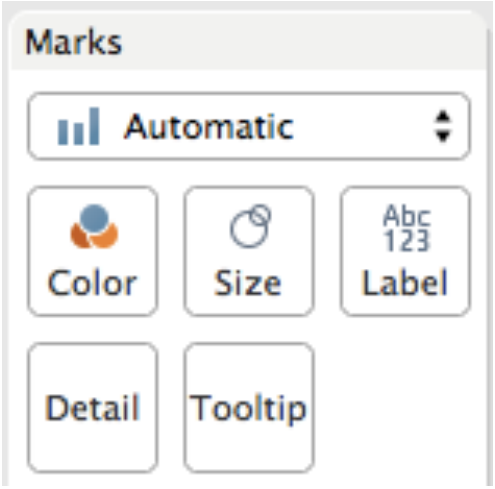




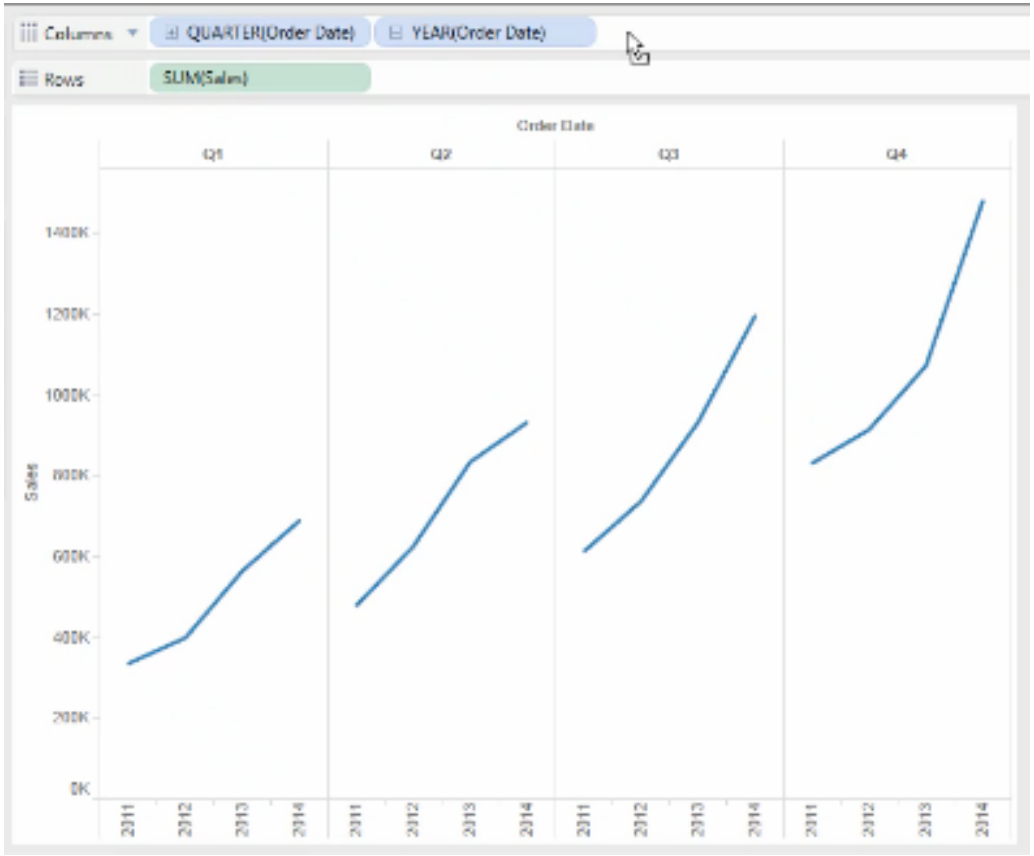
Rows & Columns



Marks



Multi-level Analysis



Group Exercise

- Load the “office_supplies.xls” dataset into Tableau
- Answer the following questions:
    1.

Agenda

- Matplotlib
- Bokeh
- Tableau
- **Final Project**

Your final project : guidelines

- Goal: **apply** what you have learned in this class to a realistic data science challenge + exercise your creativity + have fun!
- This is meant to be a significant **individual effort** to learn by practicing what you are learning to a real-world data science problem.
- The **writeup** of your final project is in the form of a Jupyter notebook and associated data – to be uploaded to the final project assignment in Camino.
- You are to submit your final notebook by September 3 @ 11:59pm.

Your final project : topic selection

- Goal: **apply** what you have learned in this class to a realistic data science challenge + exercise your creativity + have fun!
- You can choose any ”significant” data set via downloadable sites, APIs, or use any of the datasets from the class.
- You need to propose an interesting data insight investigation that you would like to explore, analyze the data, visualize the data, and finally write up your conclusion on what insights you have reached.
- Grading of your final project will be based on the following rubric.

Your final project : grading rubric

Area	Details	Grading %
Topic Selection	Did you create a reasonably interesting data insight hypothesis for your investigation?	10%
Packaging	Did you create a Jupyter project packaging that looks professional and understandable?	10%
Analysis Competence	Does your notebook show competence in using the data science tools we learned in class?	40%
Insight	Does your project show useful or interesting insights from the data analysis you have done?	40%