UNC GREENSBORO

CS 405/605 Data Science

Dr. Qianqian Tong

Course Schedule

Week 13: April 3 topic: Random Forest
April 5 topic: Validation
Project stage IV and V released, and the ddl will be April 28.

Week 14: April 10 topic: PCA
April 12 topic: Clustering-Kmeans

Week 15: April 17 topic: Visualization
April 19 topic: Visualization
HW3 is released, and the ddl will be April 28.

Week 16: April 24 Project Presentation (4 groups, each will have 15-20 min)
April 26 Project Presentation (5 groups, each will have 15-20 min)
All reports and homework must be submitted by April 28, and graded by the final week.

Visualization

What is data visualization?

What is the benefit of data visualization?

Different Types of Charts for Analyzing & Presenting Data

Data Visualization Tools: Matplotlib; seaborn; plotly

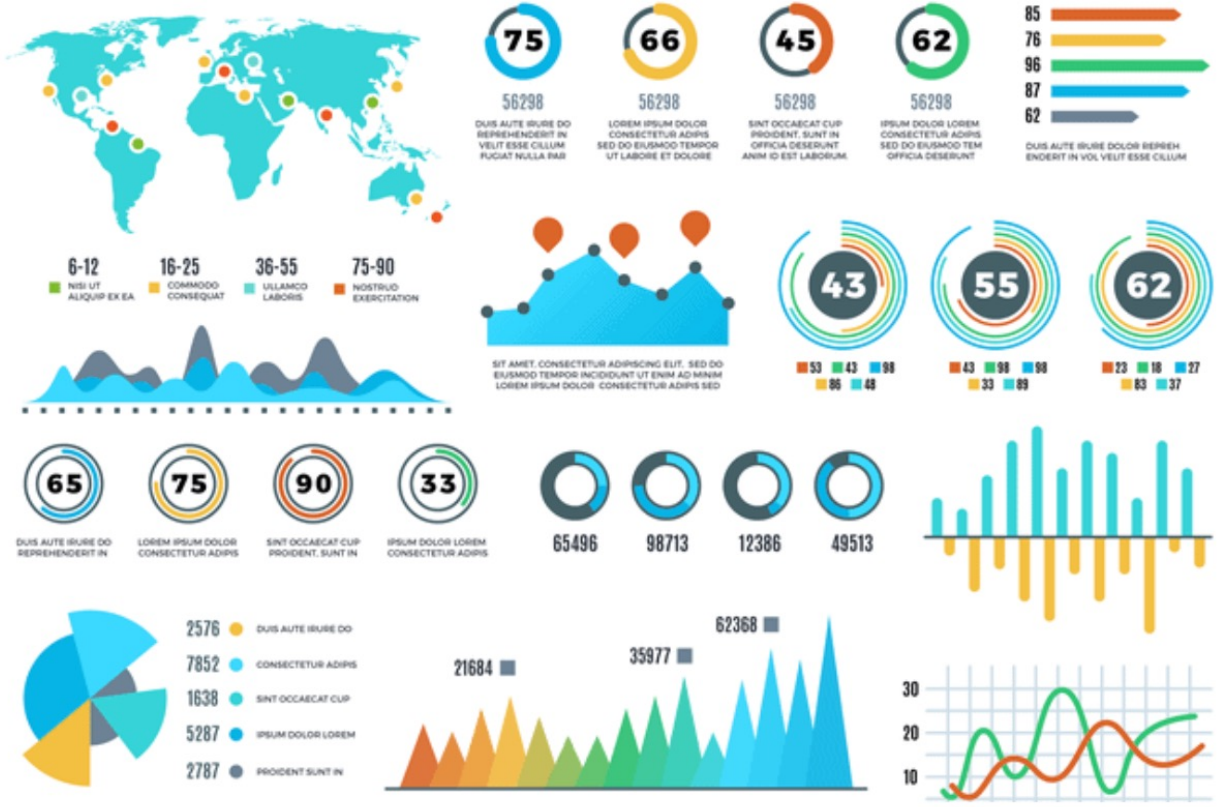
Visualization

What is data visualization?

In data science, visual presentation of the data is a first-class citizen.
We combine together various charts to better understand the data and the relationships it hides.

"A picture is worth a thousand words."

Data Visualization is the technique to represent the data/information in a pictorial or graphical format that Enables stakeholders and decision-makers to analyze and explore data visually and uncover deep insights.



UNC GREENSBORO

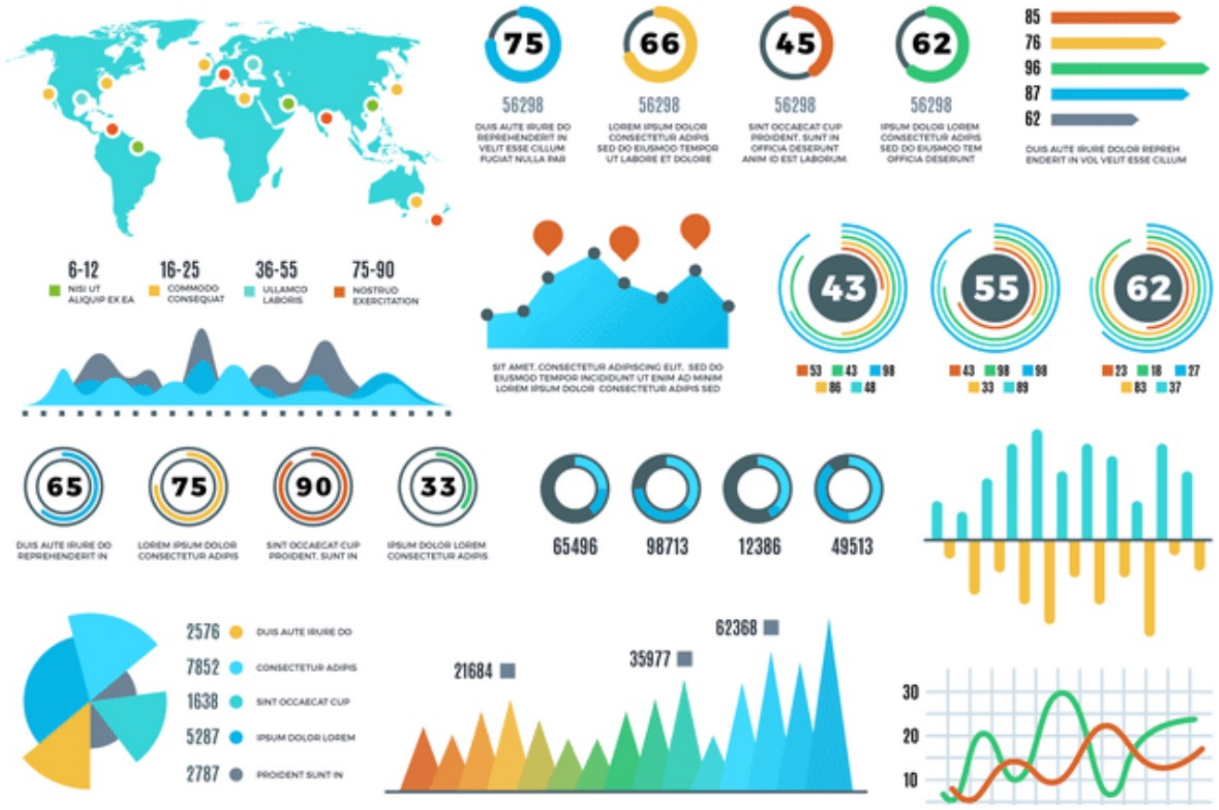
Visualization

What is data visualization?

Example. Consider a case where you are asked to illustrate crucial sales aspects (like sales performance, target, revenue, acquisition cost, etc.) from huge amounts of sales data, which one would you prefer:

1. Exploring the data using Excel (or spreadsheets) and keeping track of each sales aspect manually.

2. Exploring the data using different types of sales graphs and charts.



Obviously, you would prefer graphs and charts.
So, data visualization plays a key role in
data exploration and data analysis.

UNC GREENSBORO

Visualization

What is the benefit of data visualization?

- Helps in data analysis, data exploration and makes the data more understandable.
- Summaries the complex quantitative information in a small space.
- Helps in discovering the latest trends, hidden patterns in the data.
- Identifies the relationships/correlations between the variables.
- Helps in examining the areas that need attention or improvement

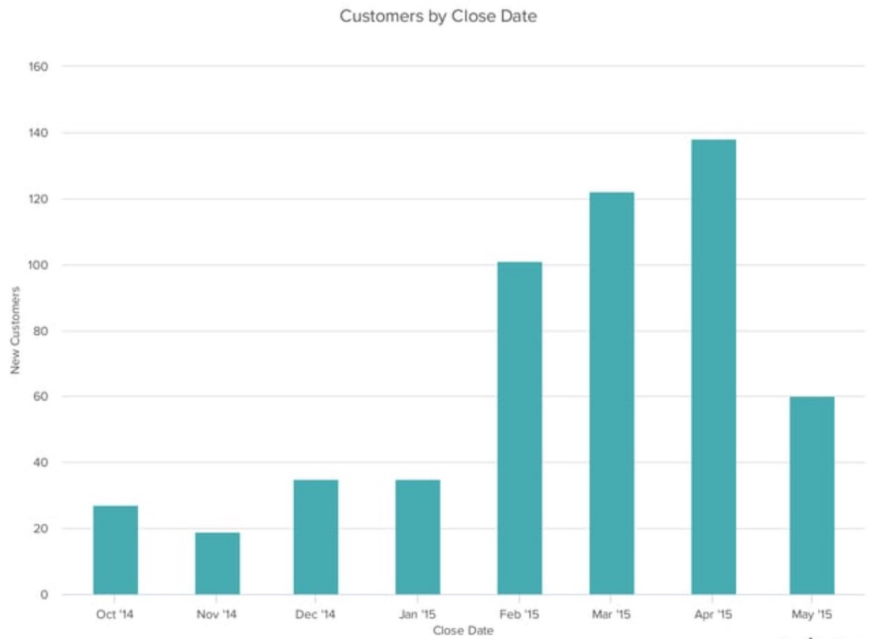
Different Types of Charts for Analyzing & Presenting Data

1) Column

A column chart is used to show a comparison among different items, or it can show a comparison of items over time. You could use this format to see the revenue per landing page or customers by close date.

- Use consistent colors throughout the chart, selecting accent colors to highlight meaningful data points or changes over time.

- Use labels to improve readability.



Visualization

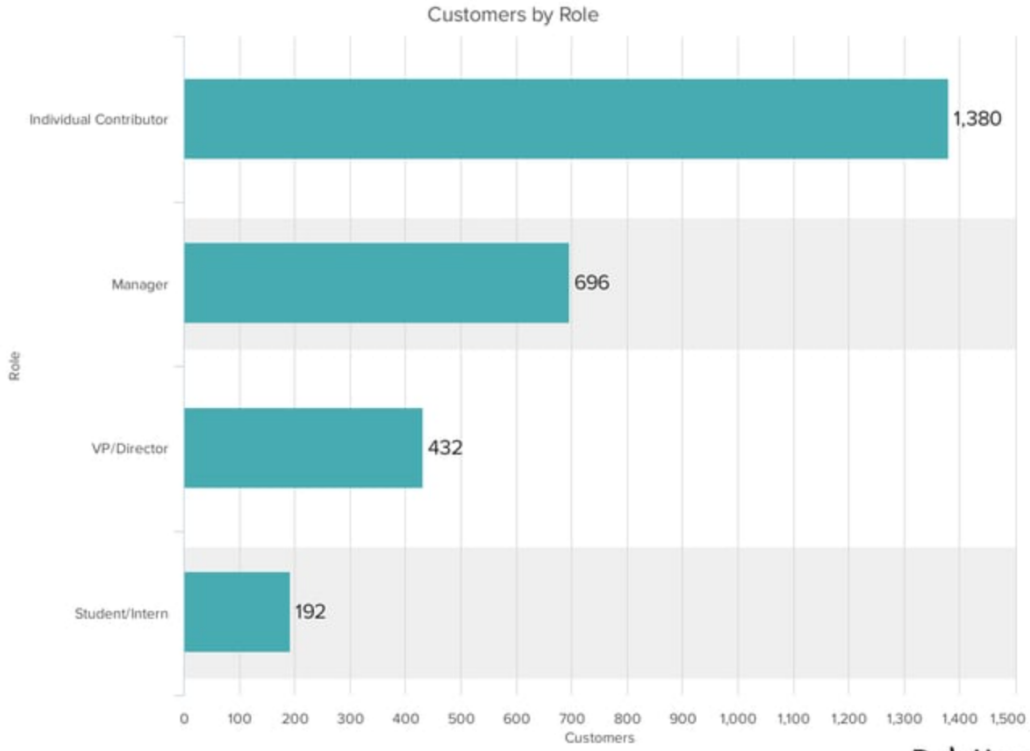
Different Types of Charts for Analyzing & Presenting Data

2) Bar

A bar chart (a horizontal column chart), should be used to avoid clutter when one data label is long or if you have more than 10 items to compare. This type of visualization can also be used to display negative numbers.

- Use consistent colors throughout the chart, selecting accent colors to highlight meaningful data points or changes over time

- Use labels to improve readability.



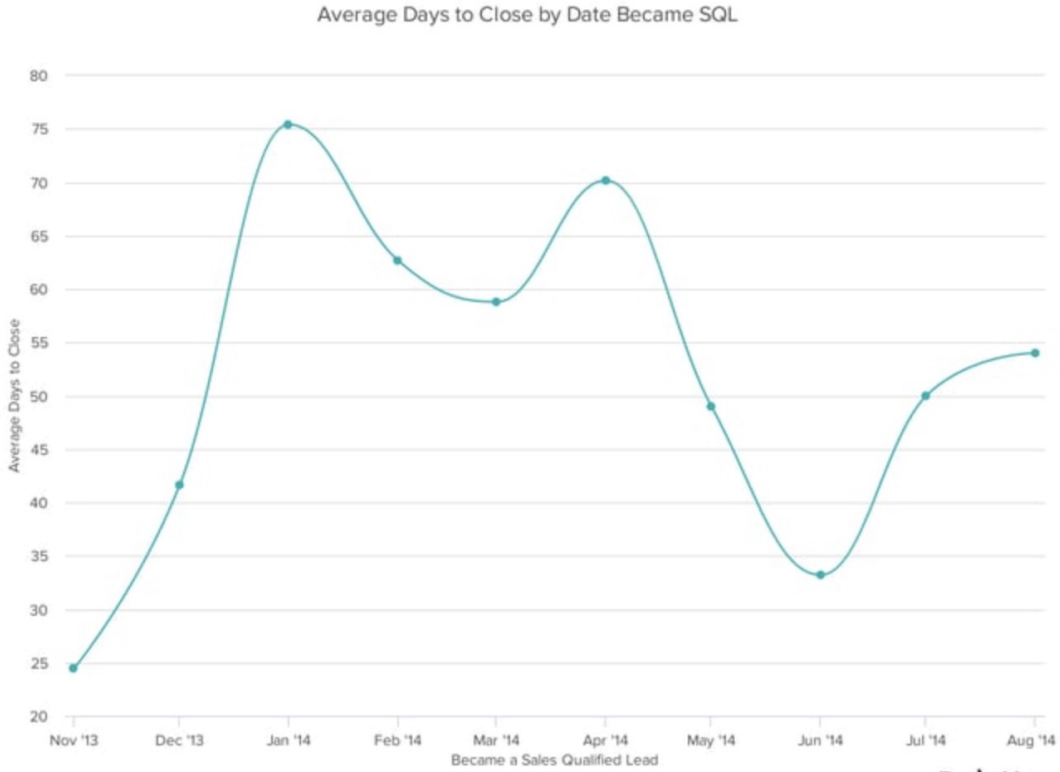
UNC GREENSBORO

Different Types of Charts for Analyzing & Presenting Data

3) Line

A line chart reveals trends or progress over time and can be used to show many different categories of data. You should use it when you chart a continuous data set.

- Use solid lines only.
- Don't plot more than four lines to avoid visual distractions.
- Use the right height so the lines take up roughly 2/3 of the y-axis' height.

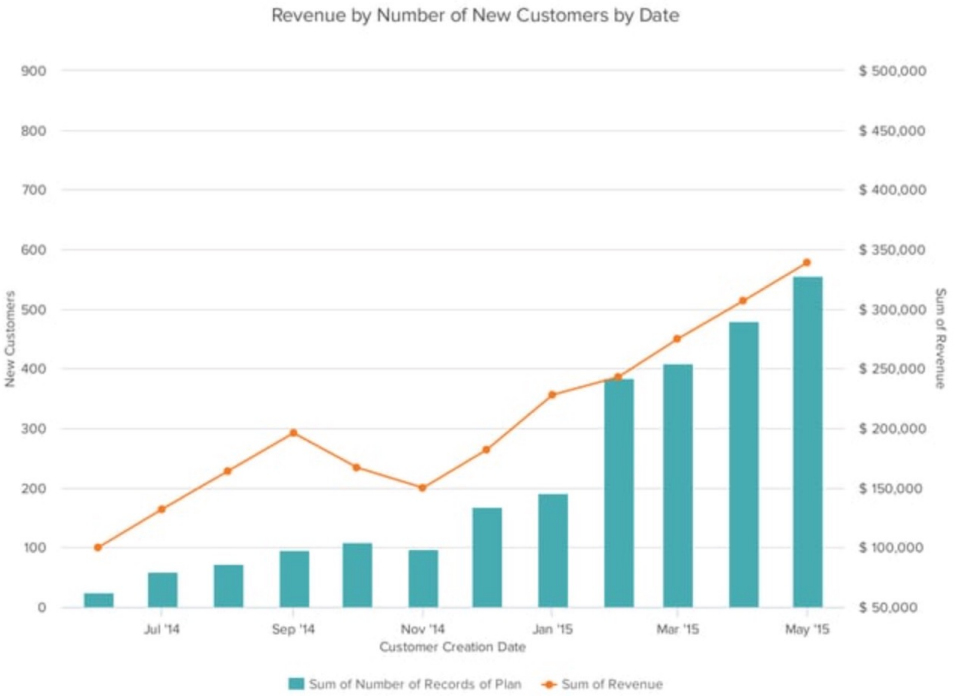


Different Types of Charts for Analyzing & Presenting Data

4) *Dual Axis*

A dual axis chart allows you to plot data using two y-axes and a shared x-axis. It's used with three data sets, one of which is based on a continuous set of data and another which is better suited to being grouped by category. This should be used to visualize a correlation or the lack thereof between these three data sets.

- Use the y-axis on the left side for the primary variable because brains are naturally inclined to look left first.
- Use different graphing styles to illustrate the two data sets, as illustrated above.
- Choose contrasting colors for the two data sets.

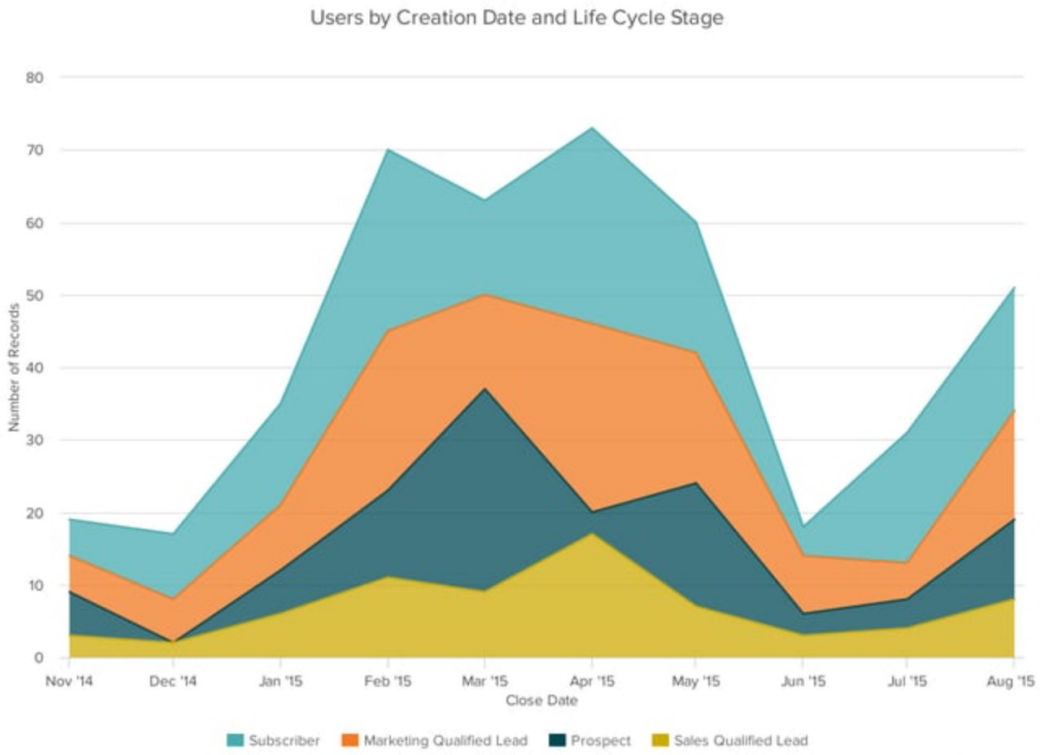


Different Types of Charts for Analyzing & Presenting Data

5) *Area*

basically a line chart, but the space between the x-axis and the line is filled with a color or pattern. It is useful for showing part-to-whole relations, such as showing individual sales reps' contribution to total sales for a year. It helps you analyze both overall and individual trend information

- Use transparent colors so information isn't obscured in the background.
- Don't display more than four categories to avoid clutter.
- Organize highly variable data at the top of the chart to make it easy to read

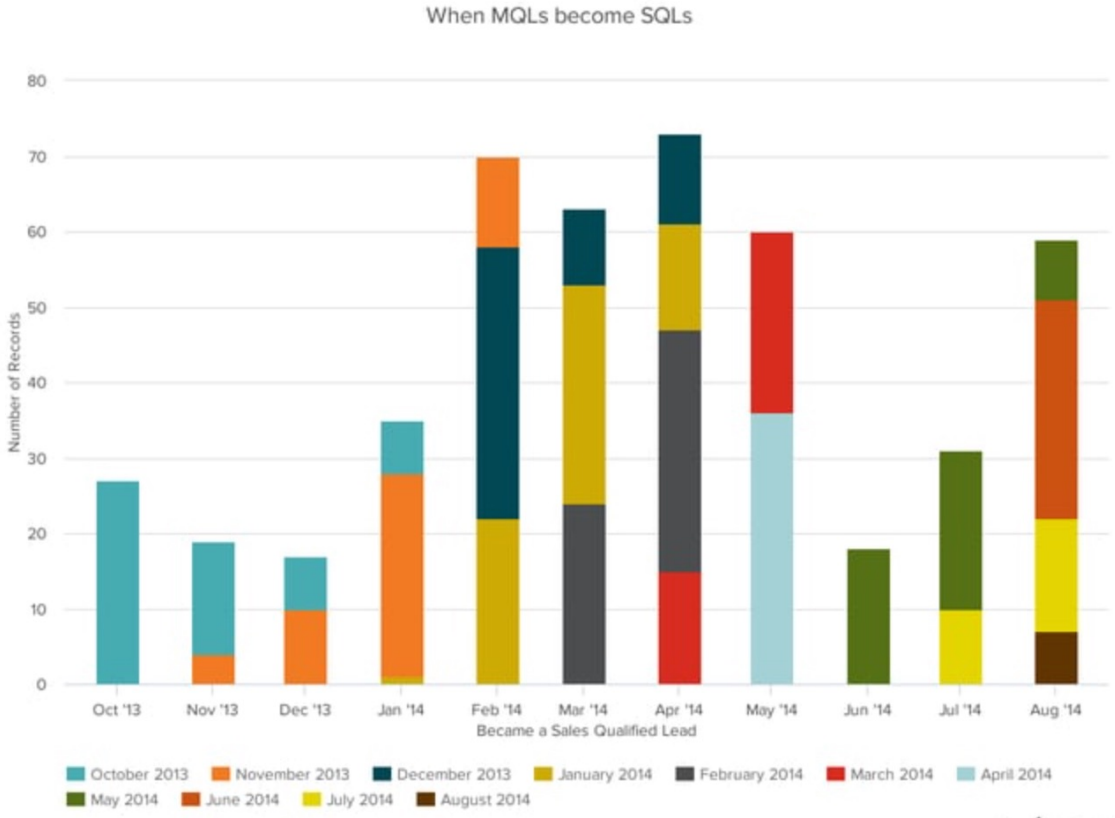


Different Types of Charts for Analyzing & Presenting Data

6) *Stacked Bar*

This should be used to compare many different items and show the composition of each item being compared.

- Best used to illustrate part-to-whole relationships.
- Use contrasting colors for greater clarity.
- Make chart scale large enough to view group sizes in relati one another.

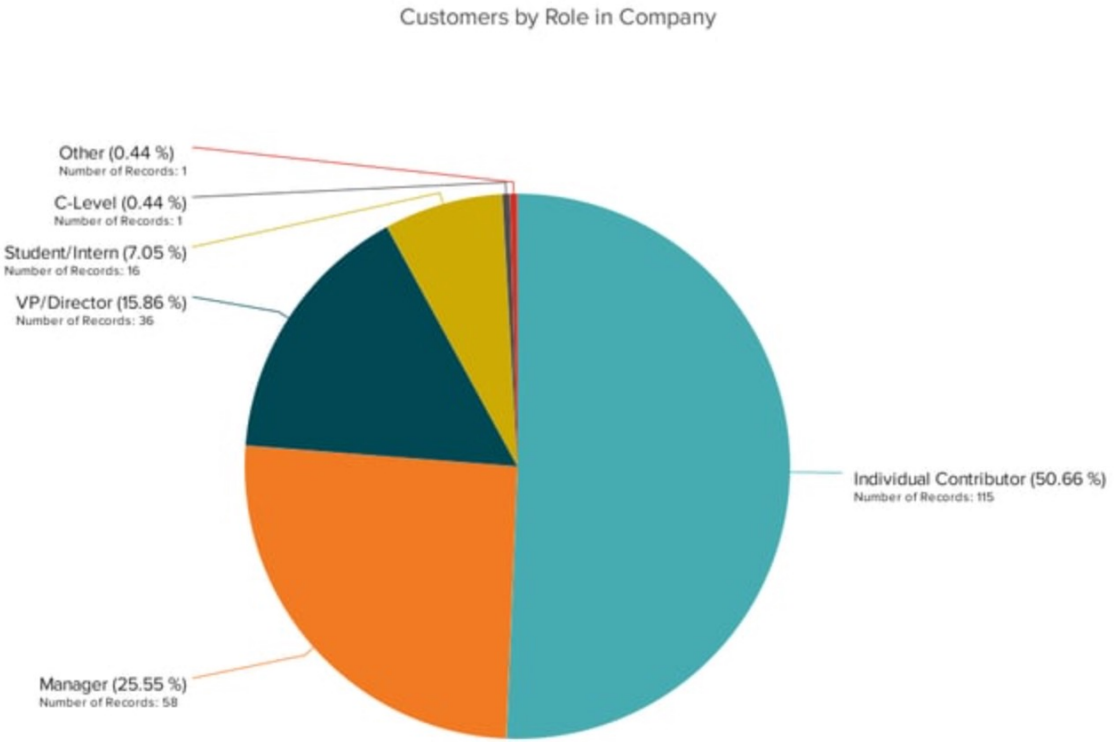


Different Types of Charts for Analyzing & Presenting Data

7) *Pie Chart*

A pie chart shows a static number and how categories represent part of a whole -- the composition of something. A pie chart represents numbers in percentages, and the total sum of all segments needs to equal 100%.

- Don't illustrate too many categories to ensure differentiation between slices.
- Ensure that the slice values add up to 100%.
- Order slices according to their size.



Different Types of Charts for Analyzing & Presenting Data

8) *Scatter Plot*

show the relationship between two different variables or it can reveal the distribution trends. It should be used when there are many different data points, and you want to highlight similarities in the data set. This is useful when looking for outliers or for understanding the distribution of your data.

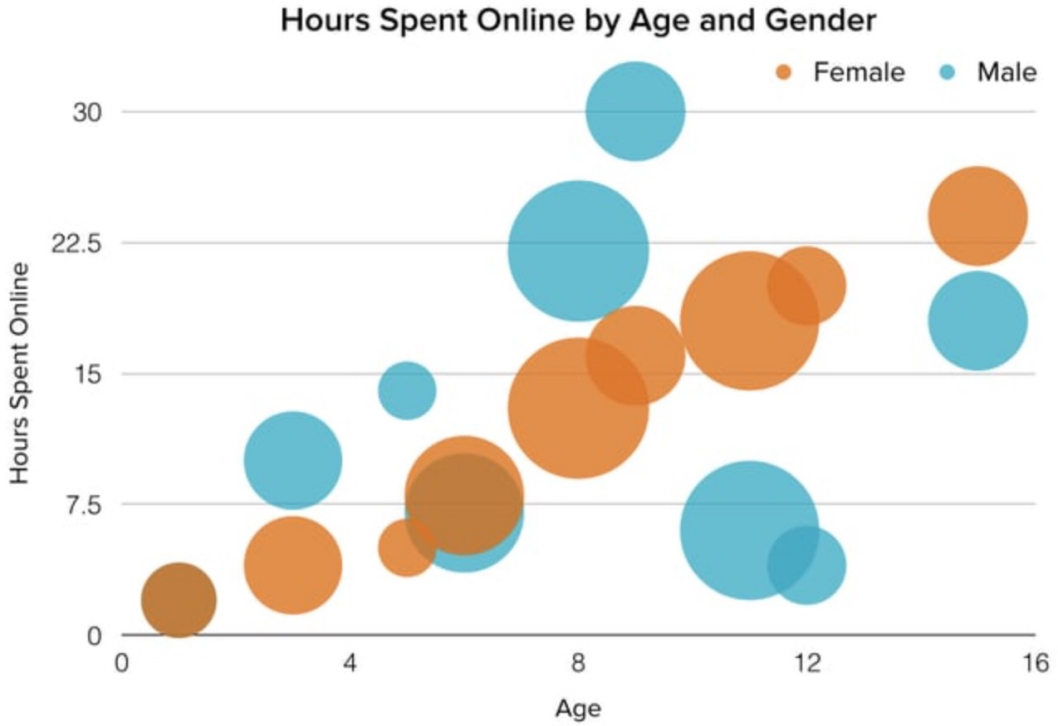
- Include more variables, such as different sizes, to incorporate more data.
- Start y-axis at 0 to represent data accurately.
- If you use trend lines, only use a maximum of two to make your plot easy to understand.

Different Types of Charts for Analyzing & Presenting Data

9) *Bubble*

A bubble chart is similar to a scatter plot in that it can show distribution or relationship. There is a third data set, which is indicated by the size of the bubble or circle.

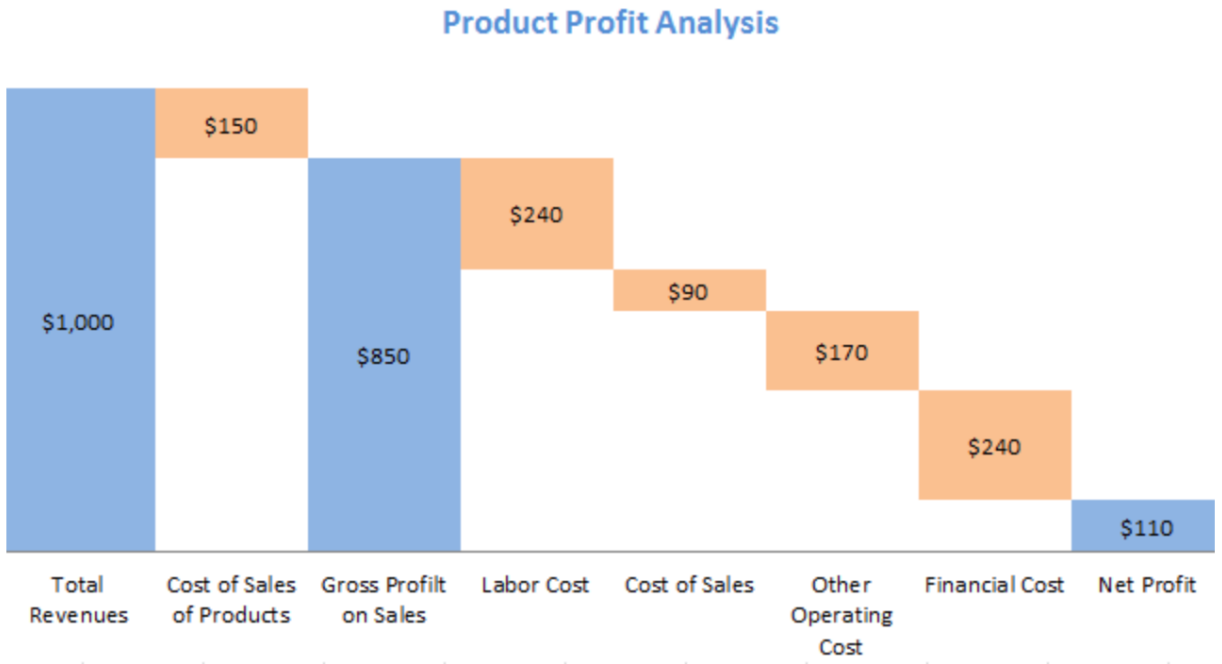
- Scale bubbles according to area, not diameter.
- Make sure labels are clear and visible.
- Use circular shapes only.



Different Types of Charts for Analyzing & Presenting Data

10) *Waterfall*

used to show how an initial value is affected by intermediate values -- either positive or negative -- and resulted in a final value. This should be used to reveal the composition of a number. An example of this would be to showcase how overall company revenue is influenced by different departments and leads to a specific profit number.



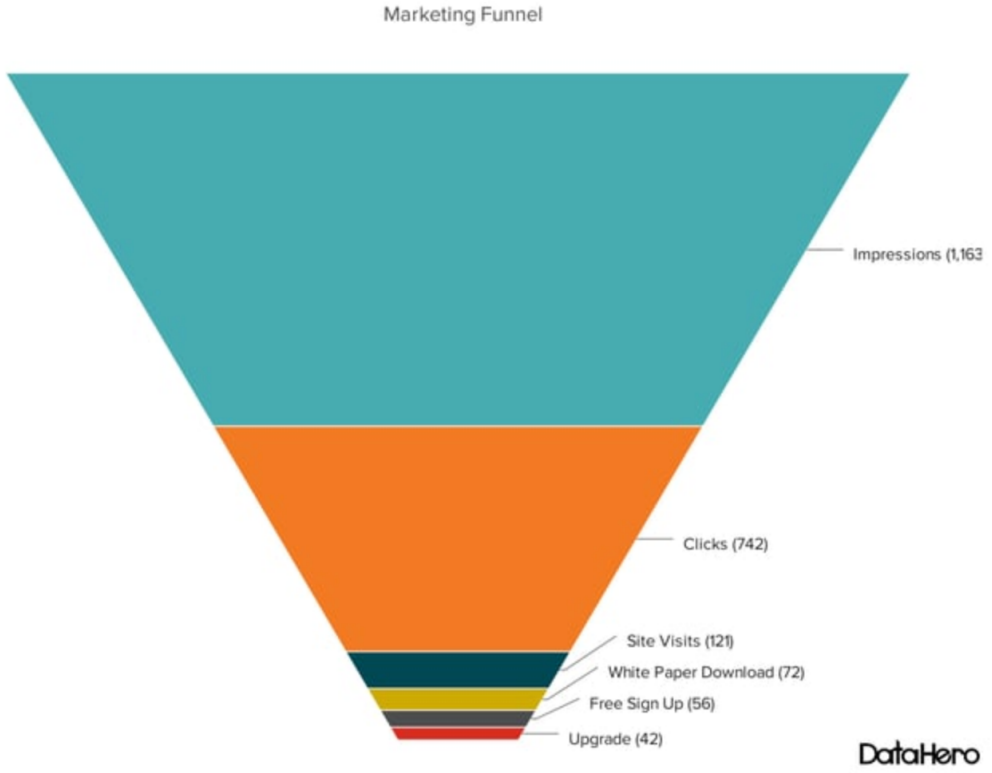
- Use contrasting colors to highlight differences
- Choose warm colors to indicate increases and cool colors to indicate decreases.

Different Types of Charts for Analyzing & Presenting Data

11) *Funnel*

A funnel chart shows a series of steps and the completion rate for each step. This can be used to track the sales process or the conversion rate across a series of pages or steps.

- Scale the size of each section to accurately reflect the size of the data set.
- Use contrasting colors or one color in gradating hues, from darkest to lightest as the size of the funnel decreases.



Different Types of Charts for Analyzing & Presenting Data

12) Bullet

A bullet graph reveals progress toward a goal, compares this to another measure, and provides context in the form of a rating or performance.

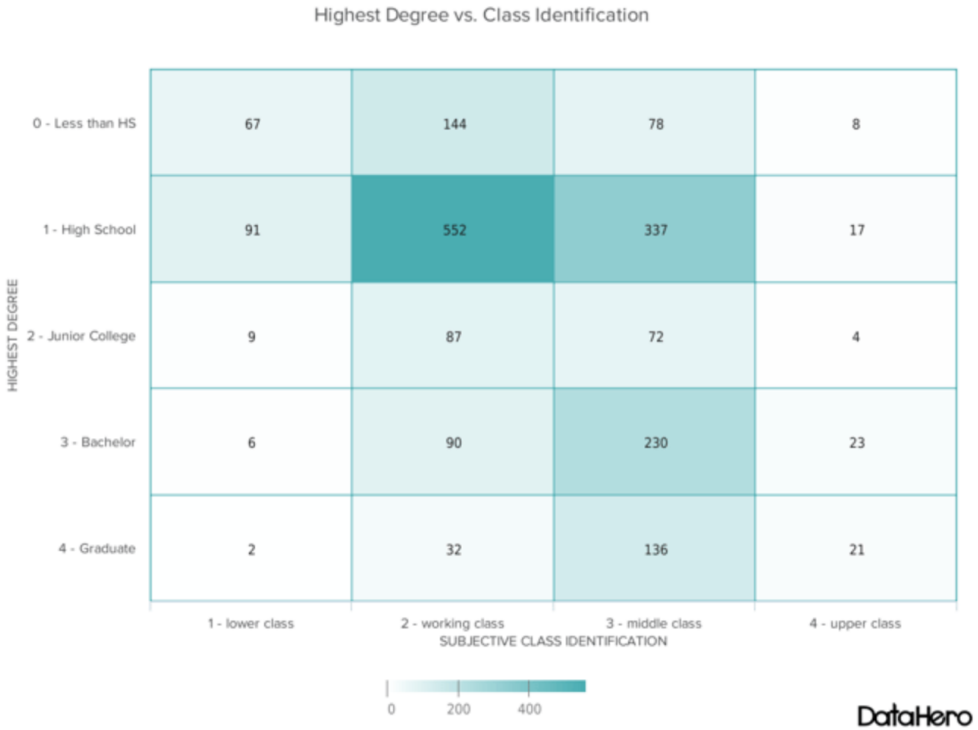
- Use contrasting colors to highlight how the data is progressing.
- Use one color in different shades to gauge progress.

Different Types of Charts for Analyzing & Presenting Data

13) Heat Map

A heat map shows the relationship between two items and provides rating information, such as high to low or poor to excellent. The rating information is displayed using varying colors or saturation.

- Use a basic and clear map outline to avoid distracting from the data.
- Use a single color in varying shades to show changes in data.
- Avoid using multiple patterns.

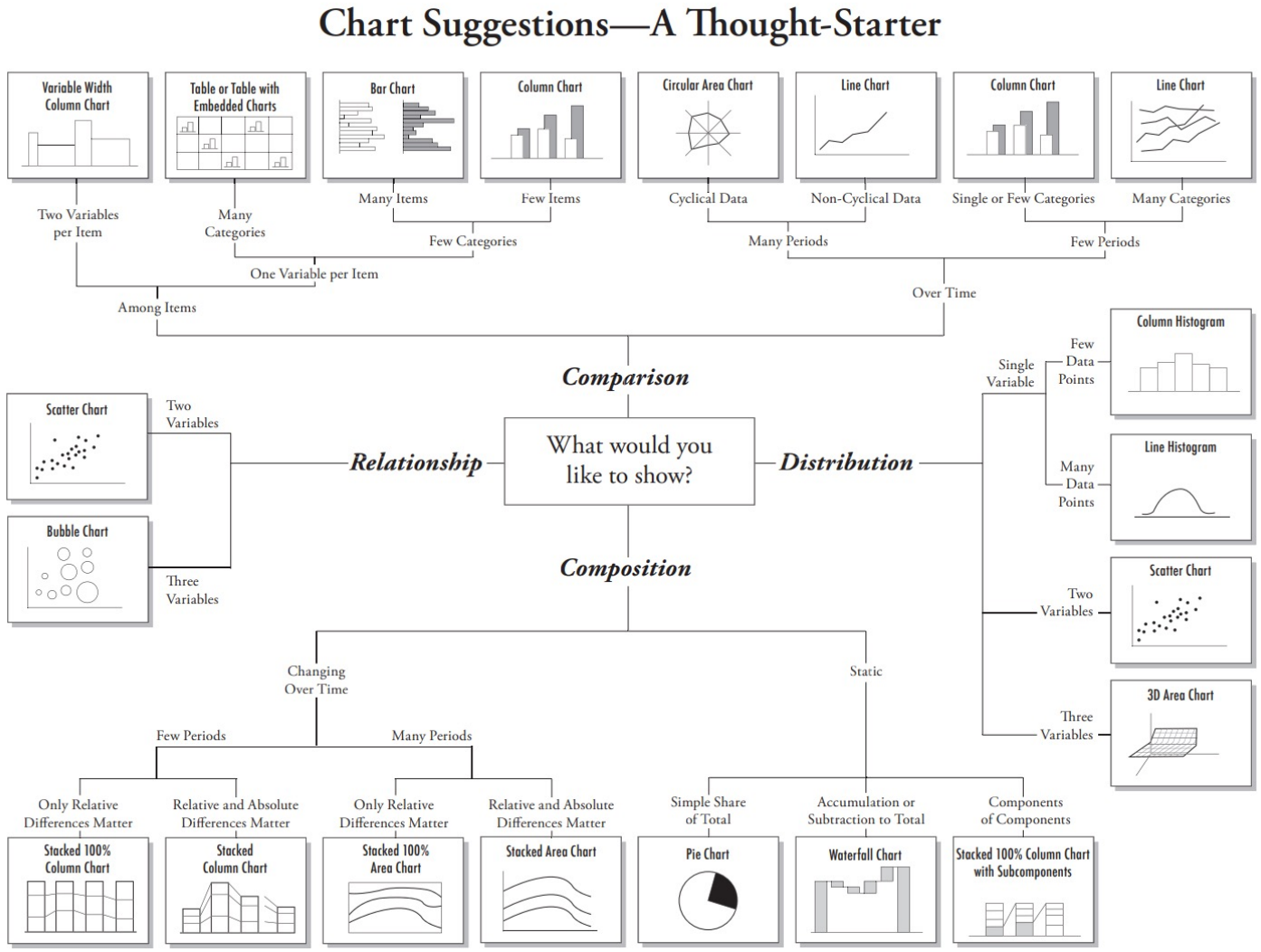


Visualization

Different Types of Charts for Analyzing & Presenting Data

There are four basic pres

1. Comparison
2. Composition
3. Distribution
4. Relationship



UNC GREENSBORO

Visualization

Data Visualization Tools:

- Matplotlib *2D and 3D plotting in Python*
- Seaborn
- Plotly
- ……

Visualization

Data Visualization Tools:

- Matplotlib

Installation quick-start

Install using pip:

```
pip install matplotlib
```

Install using conda:

```
conda install -c conda-forge matplotlib
```

To get started using Matplotlib in a Python program, include the symbols from the pylab module:

```
from pylab import *
```

OR

```
import matplotlib.pyplot as plt
```

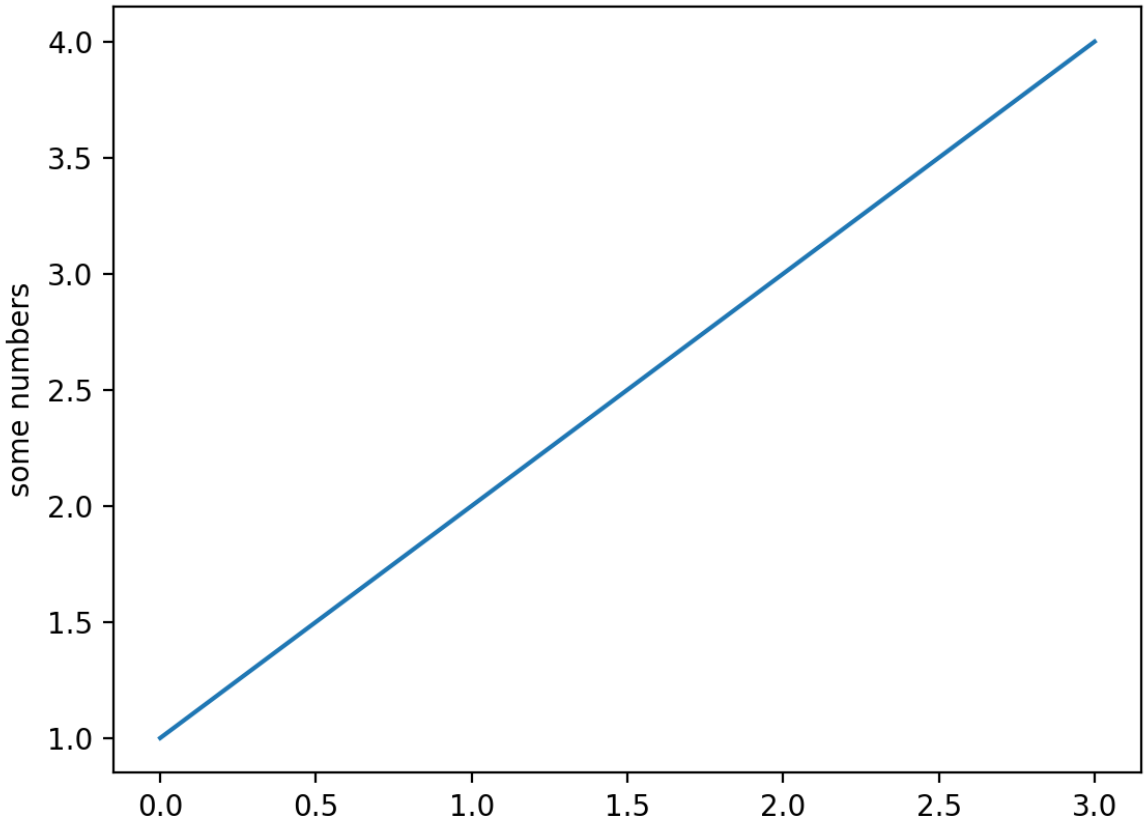
Visualization

Data Visualization Tools:

- Matplotlib

Generating visualizations with pyplot is very quick:

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```



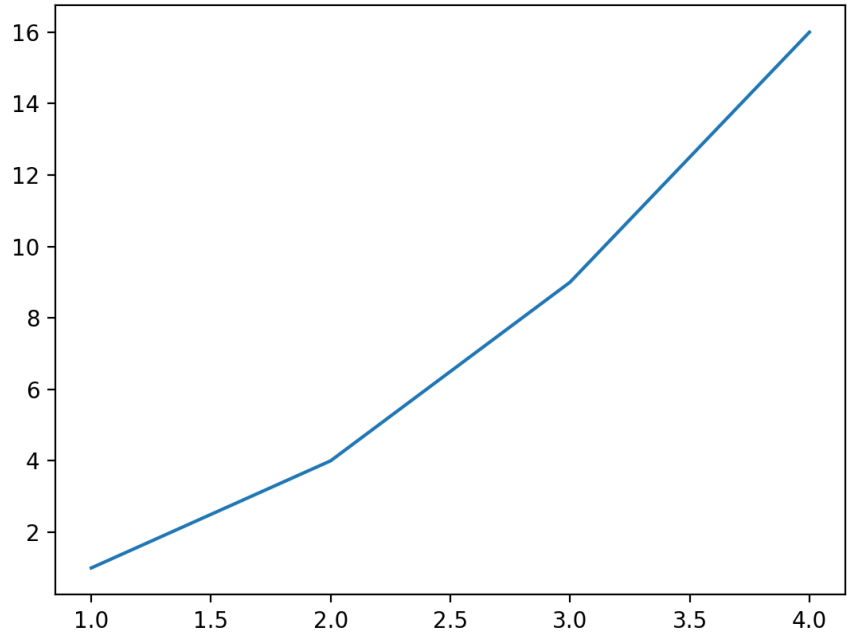
Visualization

Data Visualization Tools:

- Matplotlib

plot is a versatile function, and will take an arbitrary number of arguments.
For example, to plot x versus y, you can write:

```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
```



Visualization

Data Visualization Tools:

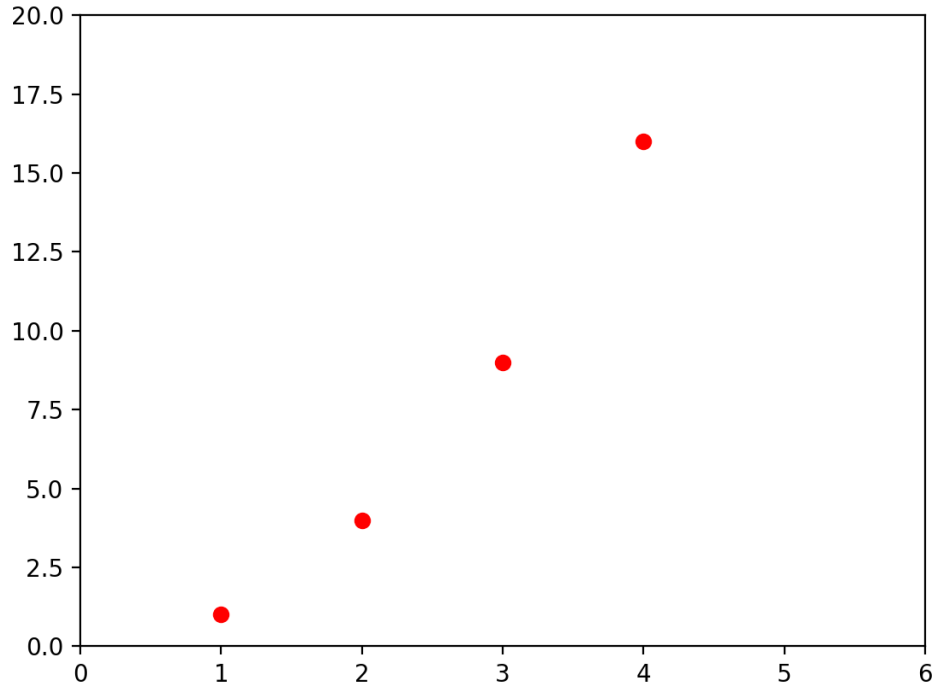
- Matplotlib

Formatting the style of your plot

For every x, y pair of arguments, there is an optional third a
that indicates the color and line type of the plot. The letters
from MATLAB, and you concatenate a color string with a li
string is 'b-', which is a solid blue line.

For example, to plot the above with red circles, you would issue

```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



UNC GREENSBORO

Visualization

Data Visualization Tools:

- Matplotlib
  - Controlling line properties
  - Plotting with keyword strings
  - Plotting with categorical variables
  - Working with multiple figures and axes
  - Working with text
  - ……

Let’s explore it:

http://localhost:8890/notebooks/Desktop/work/CS405-605-Data-Science_2/lecture/5.%20Matplotlib/01_VisualizationBasics_and_Matplotlib.ipynb

Visualization

Data Visualization Tools:

- Seaborn

Another library we really dig is seaborn, a library to maximize aesthetics of matplotlib plots. It's by by Michael Waskom. You'll need to install it with ”pip install seaborn” .

Seaborn provides an API on top of Matplotlib that offers sane choices for plot style and color defaults, defines simple high-level functions for common statistical plot types, and integrates with the functionality provided by Pandas DataFrames.

Let’s explore it:

http://localhost:8890/notebooks/Desktop/work/CS405-605-Data-Science_2/lecture/5.%20Matplotlib/02_More%20Plotting!.ipynb

Visualization

Data Visualization Tools:

- Plotly. https://plotly.com/python/

Plotly is an open-source library that provides a whole set of chart types as well as tools to create dynamic dashboards. You can think of Plotly as a suite of tools as it integrates or extends with libraries such as Dash or Chart Studio to provide interactive dashboards. Plotly's Python graphing library makes interactive, publication-quality graphs.

Plotly supports dynamic charts and animations as a first principle and this is the main difference between other visualization libraries like matplotlib or seaborn.

Visualization

Data Visualization Tools:

- Plotly.

Main Properties of Plotly:

- It can be used with other languages such as R, Python, Java.
- No JavaScript knowledge is required at all. You code Plotly in your choice of supported languages.
- Each Plotly visual is a JSON object. In this way, the visual can be accessed and used in different programming languages.
- With Plotly you can also build dynamic dashboards using **Dash** extension.
- Chart Studio allows you to create and update the graphics you want without any coding. It has a very simple and useful interface. It is especially useful in areas such as business intelligence.
- Plotly allows you to view the entire dataset in the same figure which is very important for the user experience.
- Transforming Matplotlib charts to Plotly charts is supported.
- Plotly has been added to the Pandas plotting backend with the new version of Pandas. So we can make plotting on Pandas without having to import Plotly Express.

Visualization

Data Visualization Tools:

- Plotly.

Getting Started with Data Visualization Using Plotly

To create interactive visualizations you first have to install the Plotly package in the working environment.

Install the package

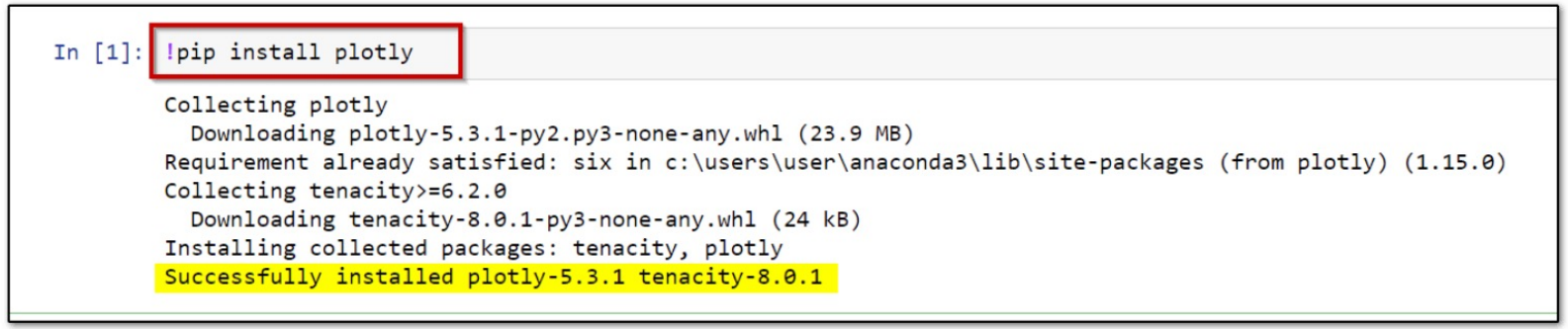
To install the package run the below command in the terminal or in the Jupyter notebook.

Step 1: Open Jupyter Notebook

Step 2: In the Jupyter Notebook, cre

Step 3: In the notebook run the below

```
!pip install plotly
```



Visualization

Data Visualization Tools:

- Plotly.

Basic Architecture of the Plotly Library

The Plotly library has the following modules:

1. **Graph_objs (plotly. graphs_objs):** It is the module that contains the objects or shape templates used to visualize.
2. **Plotly Express(plotly.express):** Plotly Express is the high-level API of Plotly and it's much easier to draw charts with this module.
3. **Subplots(make_subplots):** This module contains the helper functions for layouts of the multi-plot figures. Figures with predefined subplots configured in 'layout'.
4. **Figure Factories(plotly.figure_factory):** This module provides many special types of figures such that drawing these in Plotly or Plotly Express is quite difficult.
5. **I/O:** This module is the low-level interface for displaying, reading, and writing figures for static images, JSON, HTML and etc.

Visualization

Data Visualization Using Plotly Example

Let's take a sample dataset (taken from Open Source) and create a line chart, bar graph, histogram, etc from the data.

Step 1: Make Sure you have installed the Plotly package, if not then run the command to install the required library.

```
In [1]: !pip install plotly

        Collecting plotly
          Downloading plotly-5.3.1-py2.py3-none-any.whl (23.9 MB)
        Requirement already satisfied: six in c:\users\user\anaconda3\lib\site-packages (from plotly) (1.15.0)
        Collecting tenacity>=6.2.0
          Downloading tenacity-8.0.1-py3-none-any.whl (24 kB)
        Installing collected packages: tenacity, plotly
        Successfully installed plotly-5.3.1 tenacity-8.0.1
```

Step 2: Import the required packages and dataset.

Note: In this demo, the Cereal dataset is being used. You can download the dataset from Kaggle on your laptop.

```
import pandas as pd
import plotly.express as px

df = pd.read_csv("C:\\Users\\user\\Desktop\\Sample Dataset\\cereal.csv")
```

Give Your File Path For the Dataset

Visualization

Step 3: You can view the dataset headers (column names) by running the following command.

```
In [10]: df.info()
```

Note: df is the variable where we have stored the data in the previous step

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 77 entries, 0 to 76
Data columns (total 16 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   name      77 non-null     object
 1   mfr       77 non-null     object
 2   type      77 non-null     object
 3   calories  77 non-null     int64
 4   protein   77 non-null     int64
 5   fat       77 non-null     int64
 6   sodium    77 non-null     int64
 7   fiber     77 non-null     float64
 8   carbo     77 non-null     float64
 9   sugars    77 non-null     int64
 10  potass    77 non-null     int64
 11  vitamins  77 non-null     int64
 12  shelf     77 non-null     int64
 13  weight    77 non-null     float64
 14  cups      77 non-null     float64
 15  rating    77 non-null     float64
dtypes: float64(5), int64(8), object(3)
memory usage: 9.8+ KB
```

Visualization

Step 4: To view the entries of the dataset run the df.head() command.

memory usage: 9.8+ KB

```
In [11]: df.head()
```

Out[11]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.0	0.33	68.402973
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.0	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.0	0.33	59.425505
3	All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	8.0	0	330	25	3	1.0	0.50	93.704912
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	25	3	1.0	0.75	34.384843

Visualization

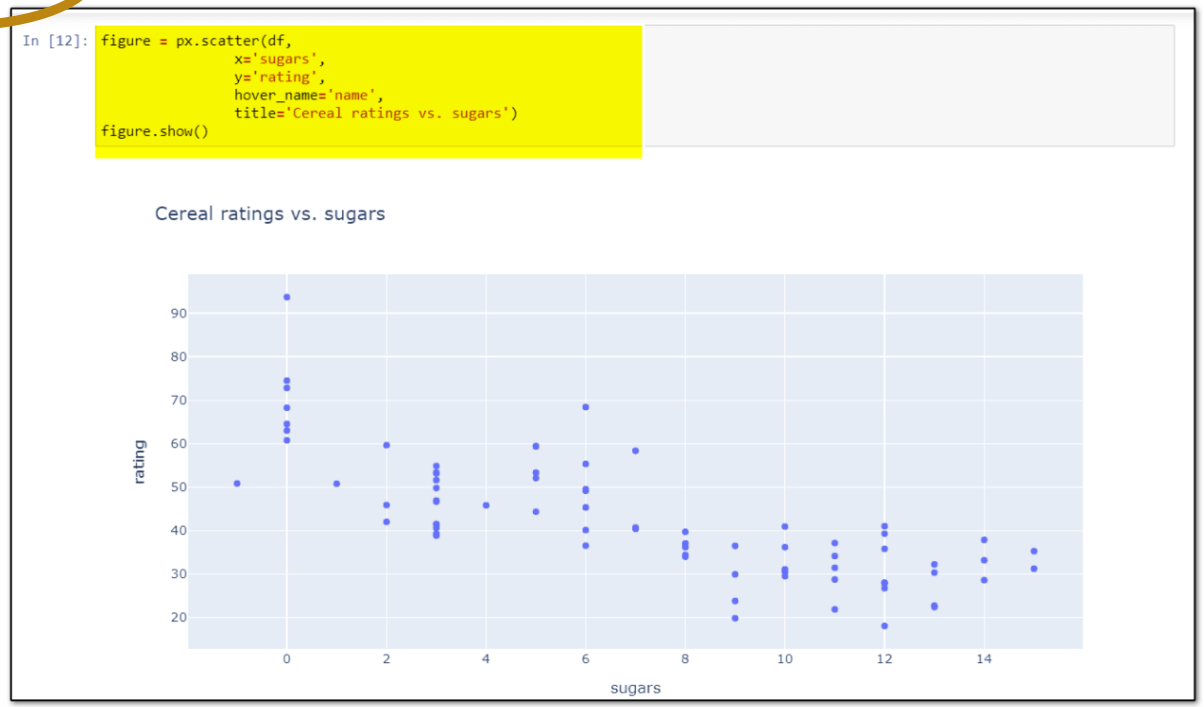
Step 5: Let's look at the relationships between the **rating** and **sugars** and include the cereal name as a hover label. Run the below command to do so

```
figure = px.scatter(df,
                    x='sugars',
                    y='rating',
                    hover_name='name',
                    title='Cereal ratings vs. sugars')

figure.show()
```

Display figures in python

In most situations, you can omit the call to .show() and allow the figure to display itself.



UNC GREENSBORO

Visualization

Step 6: Create a static histogram image for the rating distribution.

```
fig = px.histogram(df, x='rating', title='Rating distribution')
fig.show()
```

UNC GREENSBORO

Visualization

Step 7: In addition to the different chart types, most types support the same basic function signature so you can easily facet the data or change colors or sizes based on the values in your DataFrame using the below code.

```
In [15]: example = px.scatter(df,
                              x='sugars',
                              y='rating',
                              color='mfr',
                              size='calories',
                              facet_row='shelf',
                              facet_col='type',
                              hover_name='name',
                              category_orders={'shelf': ['Top', 'Middle', 'Bottom']})
         example.show()
```

UNC GREENSBORO

Visualization

Data Visualization Tools:

- Plotly.

  This was just an example of how you can create scatter plots, histograms, etc using simple python commands and Plotly for data visualization purposes. Similarly, you can create some advanced and visual funnel charts, treemaps, geographical maps, etc to perform data visualization using Plotly.

Let’s explore it:

http://localhost:8890/notebooks/Desktop/work/CS405-605-Data-Science_2/lecture/6.Plotly/Plotly.ipynb

Visualization

Data Visualization Tools:

- Plotly.

For best results, you can copy and paste this Notebook and key. Run `$ pip install plotly` inside a terminal then start up a Notebook. We'll also be using ggplot, seaborn, and prettyplotlib, which you can also all install form pip. Let's get started.

```
%matplotlib inline
import matplotlib.pyplot as plt # side-stepping mpl backend
import matplotlib.gridspec as gridspec # subplots
import numpy as np
```

You can sign up for an account on Plotly. Plotly is free for public use, you own your data, and you control the privacy.

```
!pip install plotly
!pip install chart_studio
!pip install credentials
import chart_studio
import chart_studio.plotly as py

import plotly.tools as tls
from plotly.graph_objs import *
from credentials import *


#https://chart-studio.plotly.com/settings/api#/
#creat your own account
username =      # type in your username
key =          # type in your key
py.sign_in(username, key)
```

For HW and project, use your own account!

UNC GREENSBORO