



Instructor Materials

Chapter 5: Storytelling with Data



Big Data & Analytics

Cisco | Networking Academy®
Mind Wide Open™



Chapter 5 - Sections & Objectives

- 5.1 Building a Data Story
 - Explain the fundamentals of creating an argument from data.
- 5.2 The Power of Visualization
 - Explain how to use Python libraries to create the appropriate visualizations for a communicative purpose.
- 5.3 Preparation for Chapter 5 Labs
 - Describe the sources of Big Data.



5.1 Building a Data Story

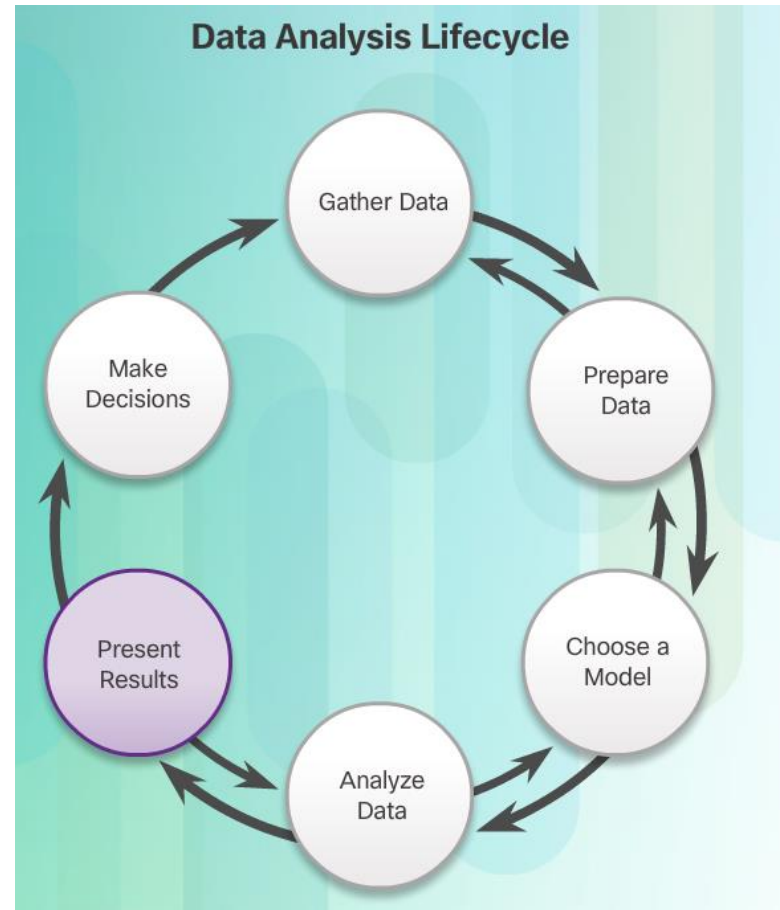


Cisco | Networking Academy®
Mind Wide Open™

Building a Data Story

Telling a Story

- Results of data analysis are shown during the *Present Results* part of the data analysis lifecycle.
- Results drive changes made by decision makers.
- Do not spend too much time on the data. Give enough to explain your point.





Building a Data Story

Audience

- Who is your audience?
 - Who will hear the story?
 - What is the listener's motivation?
 - What is the listener's level of knowledge and familiarity with the business problem?
 - What are possible reactions?
- Where is your audience?
 - Online
 - Audio only
 - Face-to-face
 - Will the presentation be shared?
- When is your audience available?
 - What to do if someone cannot attend
 - Record the presentation?
 - Confidentiality/Security concerns





Building a Data Story

Business Value and Goal

- Business value means different things to different audiences so be clear on why someone should care about the story being told
- What do you want members of the audience to take away?
- What is the call to action, if any?





Building a Data Story

Using Evidence

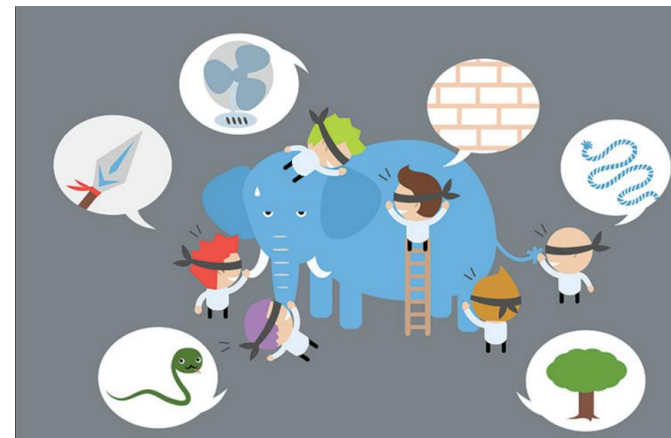
- Should be critical to the end goal
- If a piece of evidence does not support concluding remarks or is secondary to the primary focus, consider leaving the evidence out of the presentation.



Building a Data Story

Deductive Reasoning

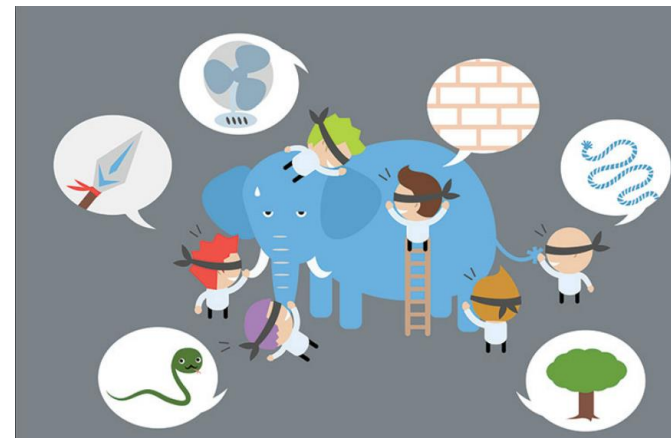
- Uses facts or premises to arrive at a conclusion
- Considered “top-down” because it moves from a general premise to specific facts derived from the general premise
- Sound deductive reasoning leads to conclusions that are true.
- Example: syllogism All mammals have eyes. Humans are mammals. Therefore, humans have eyes.



Building a Data Story

Inductive Reasoning

- Moves from specific to general
- Create a conclusion based on observations, patterns, and hypotheses
- We sample a population, study the sample, and then make inferences that we believe will be true for the entire population.
- Be sure the sample represents the population to which the conclusion is being applied

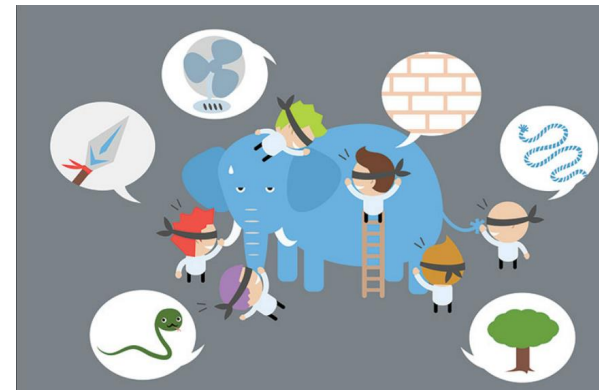




Building a Data Story

Fallacies

- Argument might not apply a rule of logic
- Argument might leave out or misinterpret a crucial premise
- Conclusion might not follow logically from the premise(s)
- Formal Fallacy
 - One or more premises shown to be false
 - If milk is kept in the refrigerator, it will not spoil. The milk is spoiled. Therefore, the milk was not kept in the refrigerator.
- Informal Fallacy
 - One or more premises do not adequately support the conclusion
 - Some people have psychic powers. Can you prove it? No one has been able to disprove it.





5.2 The Power of Visualization



Cisco | Networking Academy®
Mind Wide Open™

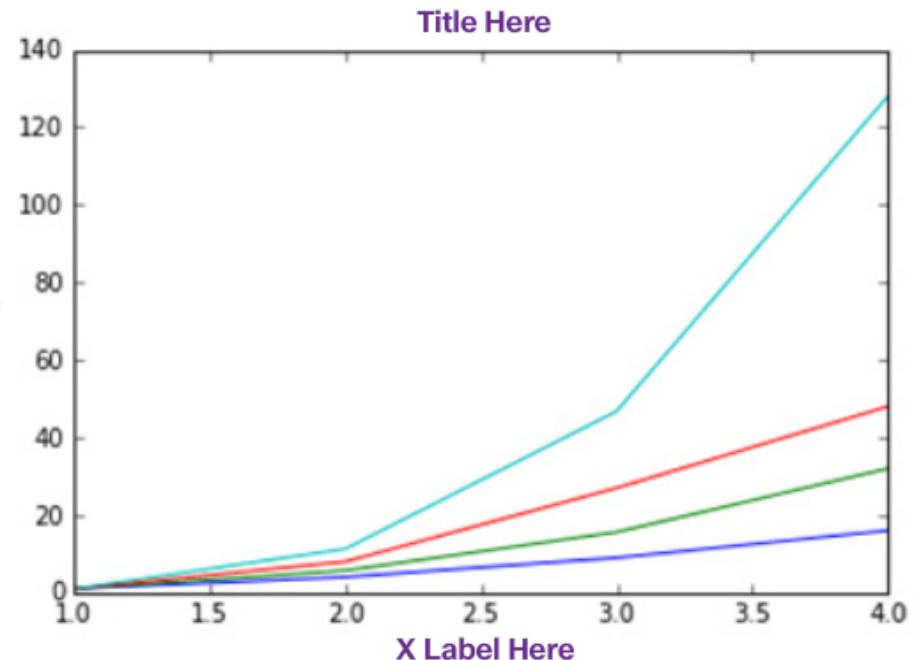


The Power of Visualization

Pyplot

- Pyplot is a Matplotlib module.
- Pyplot includes a collection of style functions you can use to create and customize a plot

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot([1,2,3,4], [1, 4, 9, 16])
plt.plot([1,2,3,4], [1, 5.7, 15.6, 32])
plt.plot([1,2,3,4], [1, 8, 27, 48])
plt.plot([1,2,3,4], [1, 11.3, 46.8, 128])
plt.xlabel('X Label Here')
plt.ylabel('Y Label Here')
plt.title('Title Here')
plt.show()
```





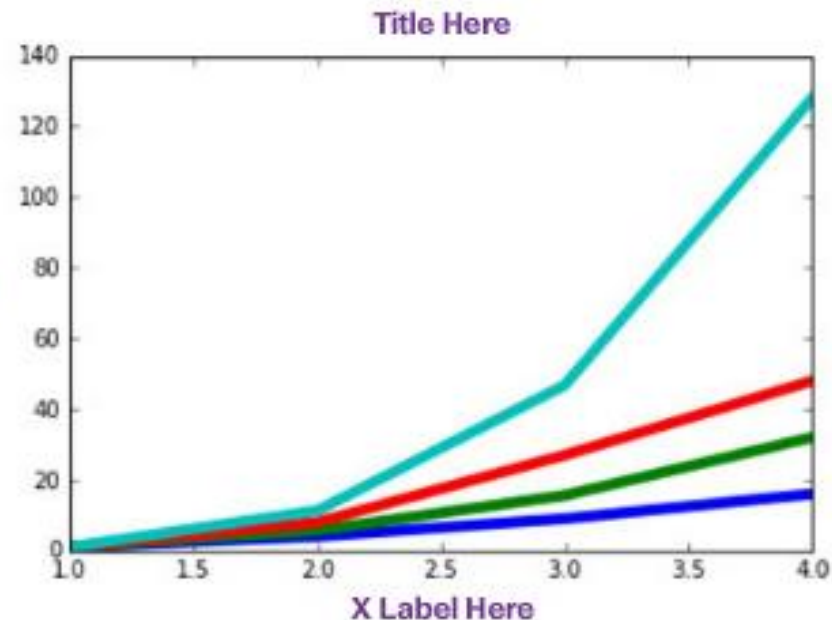
The Power of Visualization

Pyplot Custom Style Sheet

- In pyplot you can create a custom style sheet so all plots have the same style feature and you avoid making minor errors to the inline code.
- If you store the style sheet in a non-default location, you must provide path information when you reference it.

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
plt.style.use('mystyle.mplstyle')
plt.plot([1,2,3,4], [1, 4, 9, 16])
plt.plot([1,2,3,4], [1, 5.7, 15.6, 32])
plt.plot([1,2,3,4], [1, 8, 27, 48])
plt.plot([1,2,3,4], [1, 11.3, 46.8, 128])
plt.xlabel('X Label Here')
plt.ylabel('Y Label Here')
plt.title('Title Here')
plt.show()
```

Y Label Here

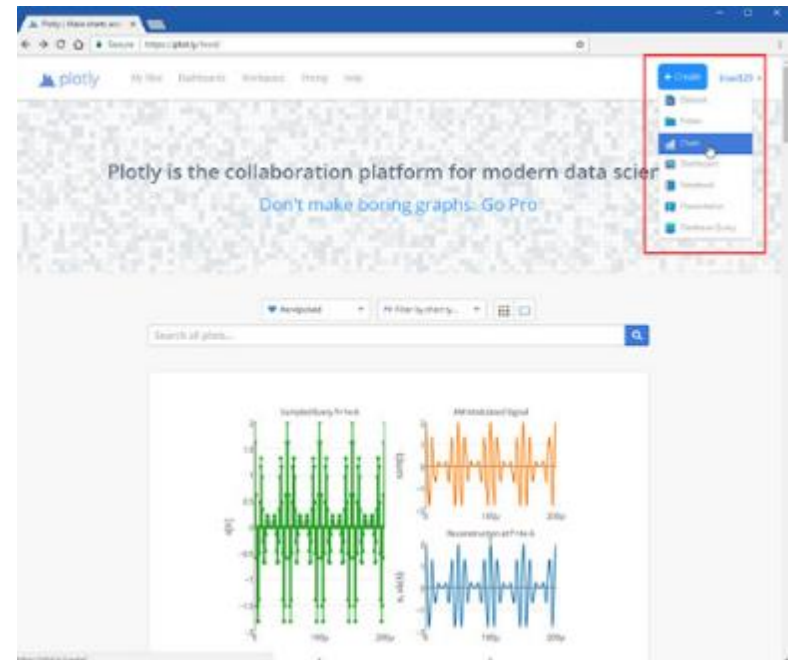
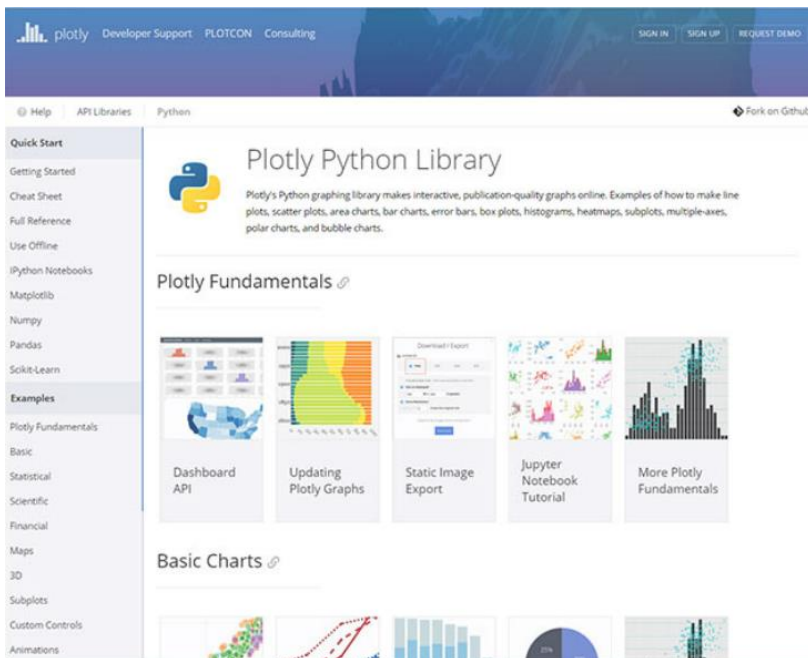




The Power of Visualization

Plotly

- Plotly is an online tool to generate data visualizations.
- Has resources including free content, API libraries, figure converters, apps for Google Chrome, and an open source JavaScript library
- Code, images, and data can be exported

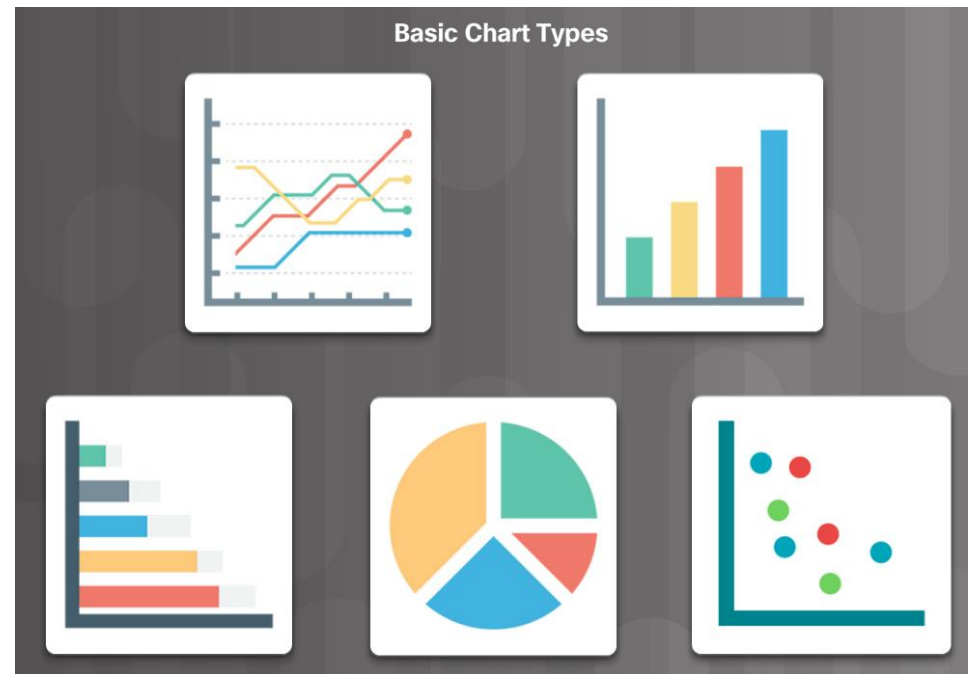




The Power of Visualization

Common Types of Data Visualizations

- How many variables?
- How many data points are in each variable?
- Is the data over time or comparing items?
- Most popular charts: line, column, bar, pie, and scatter

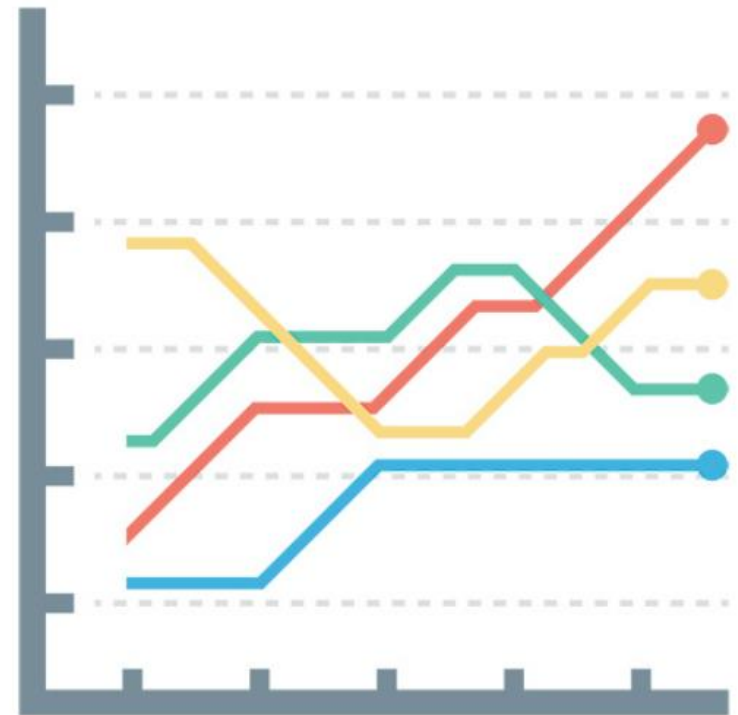




The Power of Visualization

Line Charts

- One of the most commonly used
 - Used when there is a continuous set of data, the number of data points is high, and you would like to show a trend in data over time
- Examples
 - Quarterly sales for past five years
 - Number of customers per week in the year
- Best practices
 - Label axes.
 - Plot time on the x-axis (horizontal).
 - Plot data values on the y-axis (vertical).
 - Keep data sets to a minimum.
 - Minimize gridlines.
 - Modify the axis starting point if necessary.

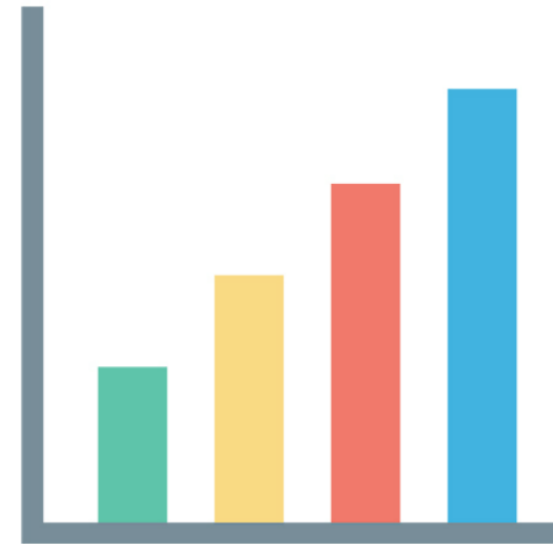




The Power of Visualization

Column Chart

- Positioned Vertically
 - Most commonly used when you want to display the value of a specific data point compared across similar categories
- Examples
 - Population of five nations
 - Yearly sales for four companies
- Best practices
 - Label axes.
 - If time is used, plot on the x-axis (horizontal).
 - Use solid colors for columns.
 - Avoid using more than 7 categories on the horizontal axis.
 - Start the y-axis value at zero.
 - Spacing between columns should be about half the width of a column.





The Power of Visualization

Bar Chart

■ Positioned Horizontally

- Most commonly used when you want to display the value of a specific data point compared across similar categories and the names for each data point is long.

■ Examples

- GDP of 25 nations
- Car sales by salesman

■ Best practices

- Label axes.
- Order bars from longest to shortest.
- Use solid colors for bars.
- Avoid using more than 7 categories on the horizontal axis.
- Start the x-axis value at zero.
- Spacing between rows should be about half the width of a column.





The Power of Visualization

Pie Charts

- Pie chart
 - Used to show the composition of a static number.
 - Segments show a percentage of that number
 - Segments total 100%
- Examples
 - Annual expenses by type
 - Energy sources by type used
- Best practices
 - Keep categories to a minimum. Consolidate when necessary.
 - Use different colors for different segments and order by size.
 - Ensure segment values total 100%.

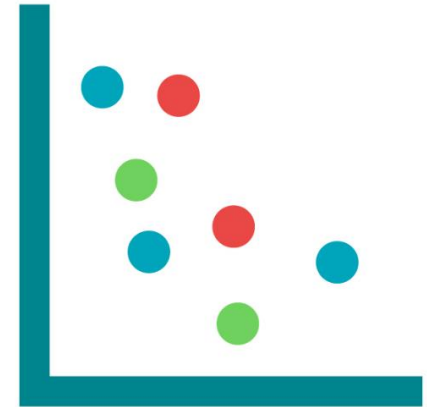




The Power of Visualization

Scatter Plot

- Clustering and Correlation Visualizations
 - Used to show correlation or distribution of data points
 - Useful in showing clustering or identifying data outliers
- Examples
 - Comparing life expectancy to GDP
 - Comparing daily sales of ice cream to average temperature
- Best practices
 - Label axes.
 - Ensure data set is large enough.
 - Start y-axis at zero. X-axis start value depends on data.
 - Consider adding a trend line, but don't use more than two.





5.3 Chapter 5 Labs



Cisco | Networking Academy®
Mind Wide Open™



Chapter 5 Preparation for Chapter 5 Labs

Folium Library

- Combines the strength of Python scripts with the mapping abilities of the Leaflet.js library
- Allows Python data frames to be displayed within an interactive Leaflet map
- Tileset – collection of raster or vector data that can display a map on mobile devices and within a browser

```
import folium
svmap = folium.Map(location=[37.3861, -122.0839], tiles="Stamen Terrain", zoom_start=11)
folium.CircleMarker([37.409006, -121.954078], popup='Cisco Systems', color='#0F8ABE', fill_color='#0F8ABE')
folium.CircleMarker([37.331697, -122.030628], popup='Apple Computer', color='#E46924', fill_color='#E46924')
folium.CircleMarker([37.422392, -122.084208], popup='Google', color='#E46924', fill_color='#E46924')
folium.CircleMarker([37.484610, -122.147930], popup='Facebook', color='#E46924', fill_color='#E46924')
folium.CircleMarker([37.387583, -121.963523], popup='Intel', color='#E46924', fill_color='#E46924')
svmap
```



5.4 Summary



Cisco | Networking Academy®
Mind Wide Open™



Chapter Summary

Summary

- Data can be summarized using visualizations to help others understand the data.
- Must know who your audience is, where they are, and when the audience is available?
- Evidence presented can be derived from deductive reasoning or inductive reasoning and should not suffer from a logical fallacy (formal or informal).
- Deductive reasoning uses facts, propositions, or other statements of truth to arrive at a conclusion.
- Inductive reasoning creates a conclusion based on observations, patterns, and hypotheses.
- Types of charts used in visualizations are line, column, bar, pie, and scatter.
- Pyplot is a matplotlib extension that includes style functions used to create and customize a plot.
- Plotly is an online tool used to create a visualization.

