

Data Visualization

Course: <https://www.coursera.org/learn/python-for-data-visualization/lecture/IzX2d/welcome-to-the-course>

By: Vadhna Samedy Hun

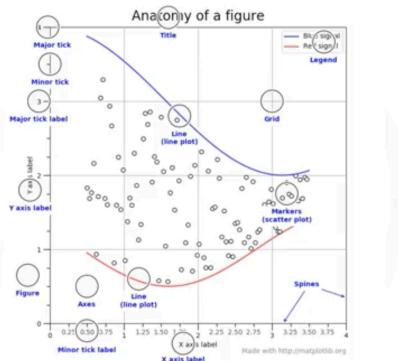
I. Introduction

- Data visualization is graphical representation of data and information.

Data visualization

- Represents data in visual formats
- Effectively communicates information, trends, and insights
- Uncovers patterns and identifies trends
- It presents complex information in an easier way to understand

Components of a plot



SOURCE: <https://matplotlib.org/stable/gallery/showcase/anatomy.html>

 Skills Network

IBM

1.1. Line Plots

- A line plot is a type of plot that displays information as a series of data points connected by straight lines.
 - Visualizing trends and changes over time.
 - Showing relationships.
 - Compare multiple data series.
 - Highlighting sudden changes or anomalies.

Your grade: 90%

Your latest: 90% • Your highest: 90% • To pass you need at least 70%. We keep your highest score.

Next item →

1. Matplotlib was created by:

- Daniel Johnson, a German physicist
- James Gosling, a Canadian computer scientist
- Cleve Moler, an American mathematician and computer programmer
- John Hunter, an American neurobiologist

 Correct

Correct! The creator of Matplotlib was John Hunter, an American neurobiologist.

1 / 1 point

2. Using the inline backend, at what point can you not modify a figure?

- After it is coded
- After it is rendered
- After it is created
- Before it is rendered

 Correct

Correct! One limitation of this backend is that you cannot modify a figure once it's rendered. So, after rendering the figure, we cannot add, for example, a figure title or labels to its axes.

1 / 1 point

3. Which two of the following are examples of Matplotlib magic functions?

- \$matplotlib outline
- #matplotlib inline
- %matplotlib notebook

 Correct

Correct! The command starts with "%matplotlib," and notebook is one of the Matplotlib backends.

- %matplotlib inline

 Correct

Correct! A sign of a magic function is that it starts with "%matplotlib."

1 / 1 point

4. True or False. A line plot displays information as a series of data points connected by straight lines.

1 / 1 point

- False
- True

 Correct

Correct! Line plots display information as a series of data points connected by straight lines.

5. Complete the following. Matplotlib's three main layers are:

1 point

- Backend, Artist, Scripting
- Line2D, Rectangle, Circle
- FigureCanvas, Renderer, Event
- Artist, Scripting, Histogram

 Incorrect

Incorrect. Please review the video: Introduction to Matplotlib.

6. What is Jupyter Notebook used for?

1 / 1 point

- A well-established data visualization library that can be integrated into different environments
- An open-source web application that allows you to create and share documents that contain live code, visualizations, and some explanatory text as well
- A Python library with a number of different backends available
- It is a tool used for creating conventional visualization tools using the plot function

 Correct

Correct! Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, visualizations, and some explanatory text as well.

7. Matplotlib was initially developed as what kind of tool?

1 / 1 point

- Visualization
- Chart creation
- Graphing
- Data analyzer

 Correct

Correct! Matplotlib was initially developed as an EEG and ECoG visualization tool.

8. Which of the following are the backend layers three built-in interface classes? Select three.

1 / 1 point

- Renderer

 Correct

Correct! Renderer is apart of the backend layers three built-in interface classes.

- Event

 Correct

Correct! Event is apart of the backend layers three built-in interface classes.

- Canvas

- FigureCanvas

 Correct

Correct! FigureCanvas is apart of the backend layers three built-in interface classes.

9. What is the purpose of line plots?

1 / 1 point

- Captures data from charts
- Displays data trends over time
- Line plots capture trends and changes over time, allowing us to see patterns and fluctuations.
- Line plots capture data and changes overtime

 **Correct**

Correct! Line plots can be misleading if the scales on the axes are not carefully chosen to reflect the data accurately. Line plots capture trends and changes over time, allowing us to see patterns and fluctuations.

10. True or False. Bar plots are ideal for comparing different categories or groups.

1 / 1 point

- False
- True

 **Correct**

Correct. Bar plots are ideal for comparing different categories or groups.

1.2. Area Plots

Area plot

- Also known as area chart or graph
- Displays magnitude and proportion of multiple variables
- Represents time or another ordered dimension
- Is based on the line plot

1.2.1. Histogram

Histogram

Represents the frequency distribution of a dataset

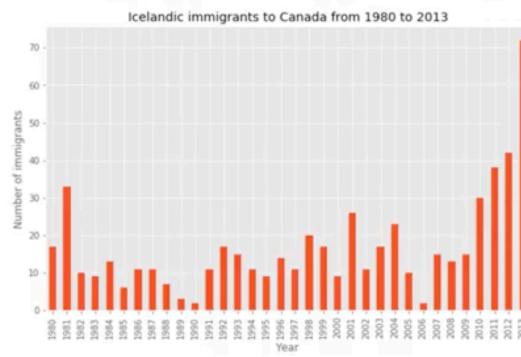
- Partitions numeric data into bins
- Assigns each data point in the dataset
- Counts the number of data points



1.2.2. Bar Chart

Bar chart

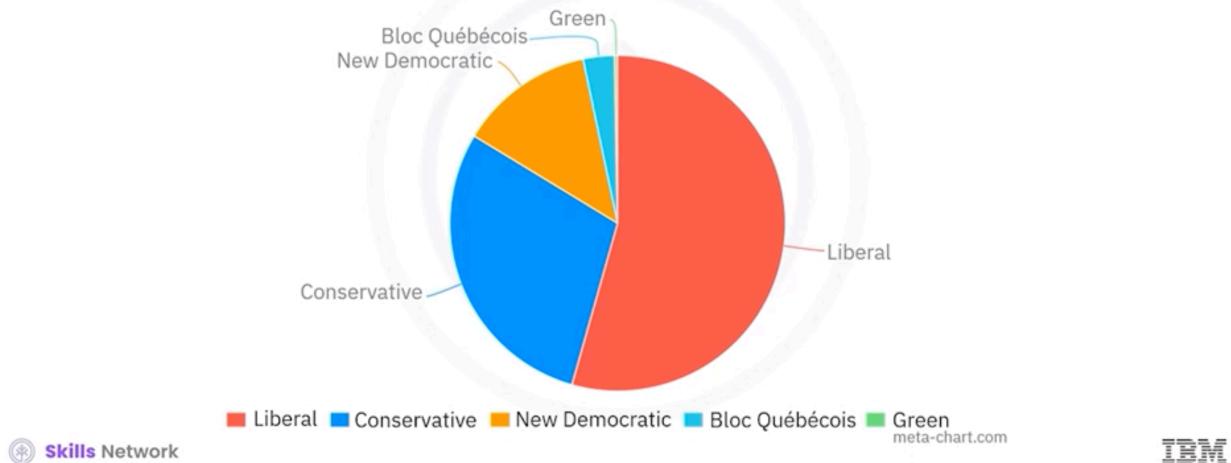
- Also known as a bar graph
- It compares the values of a variable



1.2.3. Pie Chart

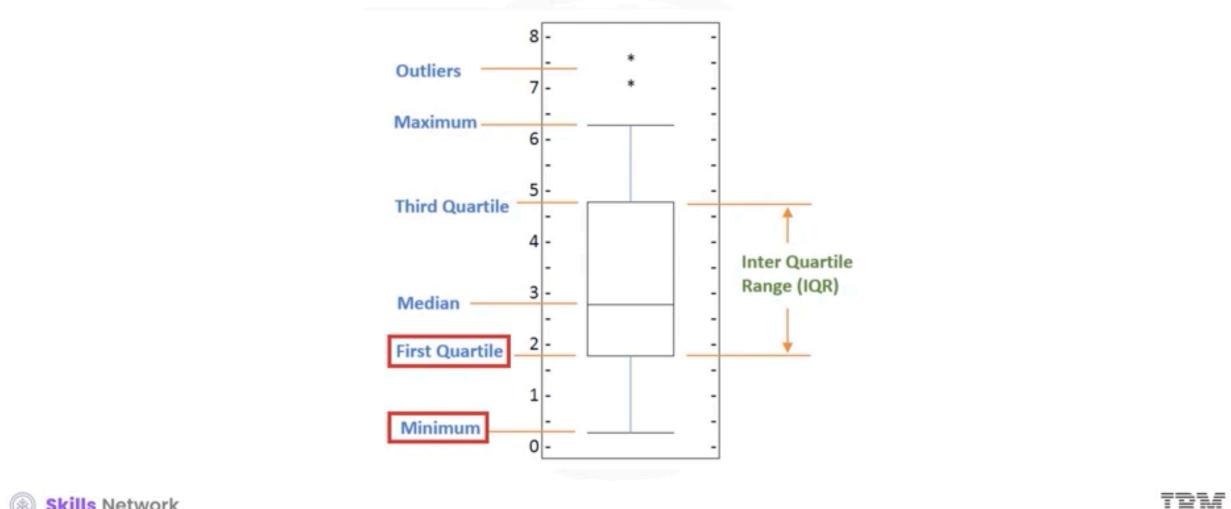
Pie chart

A circular statistical graphic, divided into segments



1.2.4. Box Plot

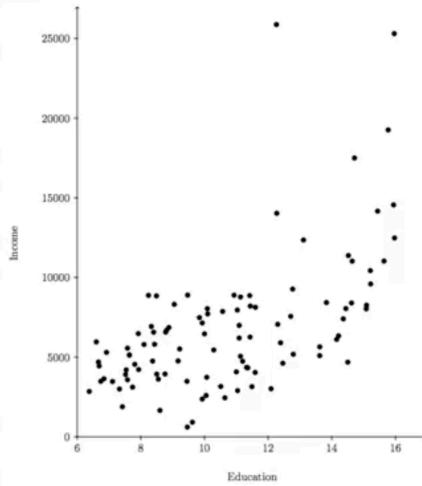
Box plot



1.2.5. Scatterplot

Scatter plot

- Displays values pertaining to two variables
- Determines the correlation between the two variables



```
explode_list = [0.0, 0, 0, 0.1, 0.1, 0.2]

df_continents['2013'].plot(
    kind='pie',
    figsize=(15, 6),
    autopct='%.1f%%',
    startangle=90,
    shadow=True,
    labels=None,
    pctdistance=1.12,
    explode=explode_list
)

# scale the title up by 12% to match pctdistance
plt.title('Immigration to Canada by Continent in 2013', y=1.12)
plt.axis('equal')

# add legend
```

```
plt.legend(labels=df_continents.index, loc='upper left')
```

```
# show plot  
plt.show()
```

Your grade: 90%

[Next item →](#)

Your latest: 90% • Your highest: 90% • To pass you need at least 70%. We keep your highest score.

1. Matplotlib was created by:

1 / 1 point

- Daniel Johnson, a German physicist
- James Gosling, a Canadian computer scientist
- Cleve Moler, an American mathematician and computer programmer
- John Hunter, an American neurobiologist

 **Correct**

Correct! The creator of Matplotlib was John Hunter, an American neurobiologist.

2. Using the inline backend, at what point can you not modify a figure?

1 / 1 point

- After it is coded
- After it is rendered
- After it is created
- Before it is rendered

 **Correct**

Correct! One limitation of this backend is that you cannot modify a figure once it's rendered. So, after rendering the figure, we cannot add, for example, a figure title or labels to its axes.

3. A pie chart is a _____ statistical graphic, divided into segments, to illustrate numerical proportions.

1 / 1 point

- Folium
- Bar chart
- Circular
- Line plot

 **Correct**

Correct! A pie chart is a circular statistical graphic, divided into segments, to illustrate numerical proportions.

4. How many primary dimensions does a box plot utilize to statistically represent the distribution of a given data?

1 point

- 4
- 1
- 3
- 5

 **Incorrect**

Incorrect. Please review the video Box Plots.

5. Area plots are similar to what other type of plot?

1 / 1 point

- Area
- Box
- Line
- Bar

 **Correct**

Correct! An area plot, also known as an area chart or graph, displays the magnitude and proportion of multiple variables over a continuous axis, typically representing time or another ordered dimension.

6. True or False. The letters in the box plot above represent: A = Median, B = Third Quartile, C = First Quartile, D = Inter Quartile Range, E = Minimum, and F = Outliers.

- True
- False

Correct

Correct! A = Mean, B = Third Quartile, C = First Quartile, D = Inter Quartile Range, E = Minimum, and F = Outliers.

7. What is the combination of function and parameter to create a box plot in Matplotlib?

1 point

- Function = plot, and Parameter = kind with value = "boxplot"
- Function = plot, and Parameter = kind with value = "box"
- Function = plot, and Parameter = type with value = "box"
- Function = boxplot, and Parameter = type with value = "plot"

Incorrect

Incorrect. Please review the video Plotting Directly with Matplotlib.

8. When creating a histogram in Matplotlib what is the first step?

1 / 1 point

- Import histogram as mpl and its scripting interface as plt.
- Import histogram as plt and its scripting interface as mpl.
- Import matplotlib as mpl and its scripting interface as plt.
- Import matplotlib as plt and its scripting interface as mpl.

Correct

Correct! The first step when creating a histogram in matplotlib is to import matplotlib as mpl and its scripting interface as plt.

9. True or False. The process of creating a scatter plot involves importing Matplotlib to visualize a large set of data.

1 / 1 point

True

False

 Correct

Correct! A scatter plot is a type of plot that displays values pertaining to typically two variables against each other. The process of creating a scatter plot involves importing Matplotlib to visualize a large set of data.

10. A bar chart is also known as a _____?

1 / 1 point

Bar graph

Pyplot

Histogram

Bar plot

 Correct

Correct! Unlike a histogram, a bar chart, also known as a bar graph, is a type of plot where the length of each bar is proportional to the value of the item that it represents.

II. Waffle Charts & Word Cloud

Waffle charts

- Represents categorical data in the form of:
 - Square tiles
 - Cells
- Displays proportion or percentage of different categories
- Simplifies data for all types of audiences



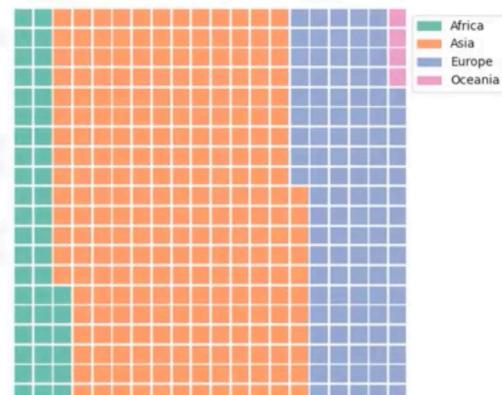
pywaffle library

```
import matplotlib.pyplot as plt
from pywaffle import Waffle

# Create data for the waffle chart
data = df_dsn[['Africa', 'Asia','Europe','Oceania']]

# Set up the waffle chart figure
fig = plt.figure(
    FigureClass=Waffle,
    rows=20,
    columns=20,
    values=data,
    legend={'labels':['Africa', 'Asia','Europe','Oceania'],
            'loc': 'upper left', 'bbox_to_anchor': (1, 1)})

# Display the chart
plt.show()
```



Word cloud

Bandusky 2

After leaving Reavis, I must attend college because it is definitely a requirement for becoming a veterinarian. In fact, a bachelor's degree is necessary in order to even enter a veterinarian program. One must also possess excellent communication, leadership, public speaking, and organizational skills. I have put a lot of thought and consideration into college, and I have decided that I would like to go to the University of Illinois. It is a wonderful school, and they even have a graduate program designed for students who want to become veterinarians.

Once I have completed a veterinarian program, I will be able to pursue my dream career. This career provides numerous benefits, the first of which is salary. The average veterinarian salary is \$60,000 a year, a salary that would definitely allow me to live a comfortable life. Secondly, it is a rewarding job. This job would provide me with the satisfaction of knowing that I am helping or saving an animal's life. Finally, becoming a veterinarian would assure me a lifetime of happiness. I know I would love going to my job every day, because I would be working with what I love most: animals.

In summary, when I graduate from Reavis, I plan to go to college to become a veterinarian. I love animals and I want to do anything that I can to help them. I know I am only a freshman, but I also know that I am growing up quickly. As Ferris Bueller quotes, "Life moves pretty fast. If you don't stop and look around once in a while, you could miss it!"



 Skills Network



```
def create_waffle_chart(categories, values, height, width, colormap, value_sign=

    # compute the proportion of each category with respect to the total
    total_values = sum(values)
    category_proportions = [(float(value) / total_values) for value in values]

    # compute the total number of tiles
    total_num_tiles = width * height # total number of tiles
    print ('Total number of tiles is', total_num_tiles)

    # compute the number of tiles for each category
    tiles_per_category = [round(proportion * total_num_tiles) for proportion in categories]

    # print out number of tiles per category
    for i, tiles in enumerate(tiles_per_category):
        print (df_dsn.index.values[i] + ': ' + str(tiles))

    # initialize the waffle chart as an empty matrix
```

```

waffle_chart = np.zeros((height, width))

# define indices to loop through waffle chart
category_index = 0
tile_index = 0

# populate the waffle chart
for col in range(width):
    for row in range(height):
        tile_index += 1

        # if the number of tiles populated for the current category
        # is equal to its corresponding allocated tiles...
        if tile_index > sum(tiles_per_category[0:category_index]):
            # ...proceed to the next category
            category_index += 1

        # set the class value to an integer, which increases with class
        waffle_chart[row, col] = category_index

# instantiate a new figure object
fig = plt.figure()

# use matshow to display the waffle chart
colormap = plt.cm.coolwarm
plt.matshow(waffle_chart, cmap=colormap)
plt.colorbar()

# get the axis
ax = plt.gca()

# set minor ticks
ax.set_xticks(np.arange(-.5, (width), 1), minor=True)
ax.set_yticks(np.arange(-.5, (height), 1), minor=True)

# add gridlines based on minor ticks

```

```

ax.grid(which='minor', color='w', linestyle='-', linewidth=2)

plt.xticks([])
plt.yticks([])

# compute cumulative sum of individual categories to match color schemes better
values_cumsum = np.cumsum(values)
total_values = values_cumsum[len(values_cumsum) - 1]

# create legend
legend_handles = []
for i, category in enumerate(categories):
    if value_sign == '%':
        label_str = category + ' (' + str(values[i]) + value_sign + ')'
    else:
        label_str = category + ' (' + value_sign + str(values[i]) + ')'

    color_val = colormap(float(values_cumsum[i])/total_values)
    legend_handles.append(mpatches.Patch(color=color_val, label=label_str))

# add legend to chart
plt.legend(
    handles=legend_handles,
    loc='lower center',
    ncol=len(categories),
    bbox_to_anchor=(0., -0.2, 0.95, .1)
)
plt.show()

```

```

df_CI = df_can[(df_can.index == 'China') | (df_can.index=='India')]

fig = plt.figure(FigureClass = Waffle,
                  rows = 20, columns = 30, #pass the number of rows and columns for the waffle
                  values = df_CI['Total'], #pass the data to be used for display
                  cmap_name = 'tab20', #color scheme

```

```
legend = {'labels': [f'{k} ({v})' for k, v in zip(df_CI.index.values,df_CI.Total)],  
          'loc': 'lower left', 'bbox_to_anchor':(0,-0.1),'ncol': 2}  
#notice the use of list comprehension for creating labels  
#from index and total of the dataset  
)  
  
#Display the waffle chart  
plt.show()
```

III. Folium

What is Folium?

- Folium is a powerful data visualization library in Python that was built primarily to help people visualize geospatial data.
- With Folium, you can create a map of any location in the world using latitude and longitude values. You can also create a map and superimpose markers and clusters on top of the map for interesting visualizations.
- You can also create maps of different styles, such as street-level maps, stamen maps, and a couple of others.

Label the marker

```
# generate map of Canada
canada_map = folium.Map(
    location=[56.130, -106.35],
    zoom_start=4
)

## add a red marker to Ontario

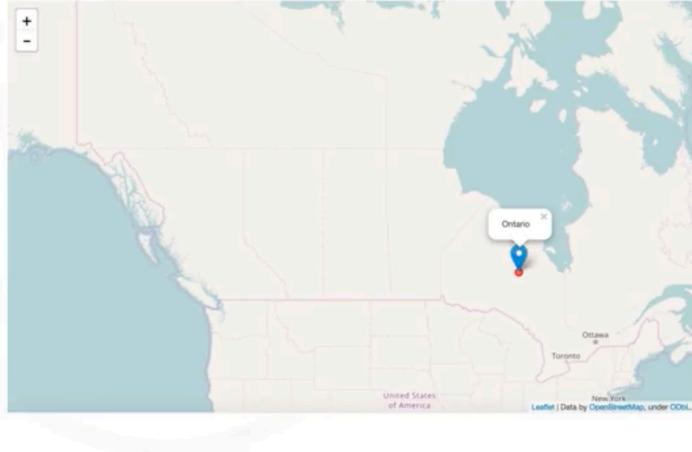
# create a feature group
ontario = folium.map.FeatureGroup()

# style the feature group
ontario.add_child(
    folium.features.CircleMarker(
        [51.25, -85.32], radius = 5,
        color = "red", fill_color = "Red"
    )
)

# add the feature group to the map
canada_map.add_child(ontario)

# label the marker
folium.Marker([51.25, -85.32],
    popup='Ontario').add_to(canada_map)

# display map
canada_map
```



 Skills Network

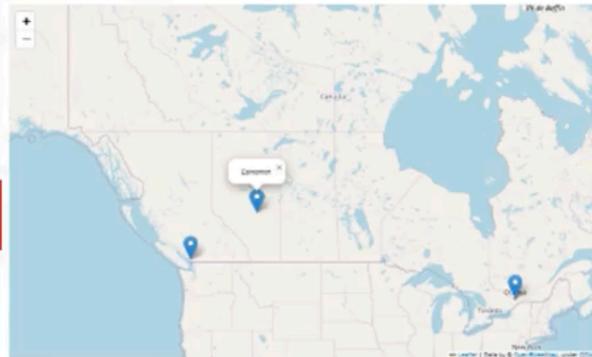
IBM

Multiple markers

```
# Define a list of locations and their corresponding popups
locations = [
    {"location": [45.4215, -75.6989], "popup": "Ottawa"},
    {"location": [53.5461, -113.4938], "popup": "Edmonton"},
    {"location": [49.2827, -123.1207], "popup": "Vancouver"},
    # Add more locations and their popups here
]

# Add markers for each location in the list
for loc in locations:
    folium.Marker(location=loc["location"],
        popup=loc["popup"]).add_to(map)

# Display the map with the markers
map
```



 Skills Network

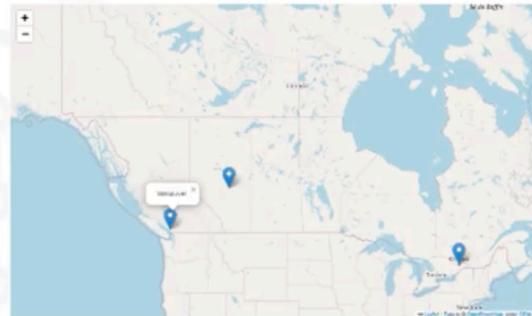
IBM

Multiple markers

```
#import MarkerCluster
from folium.plugins import MarkerCluster

# Create a MarkerCluster object
marker_cluster = MarkerCluster().add_to(map)

# Add markers for each location in the list to the MarkerCluster
for loc in locations:
    folium.Marker(location=loc["location"],
                  popup=loc["popup"]).add_to(marker_cluster)
```



3.1. Choropleth Maps

GeoJson File

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "name": "Brunei"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              [114.204017, 4.525874], [114.599961, 4.900011], [115.45071, 5.44773],
              [115.4057, 4.955228], [115.347461, 4.316636], [114.869557, 4.348314],
              [114.659596, 4.007637], [114.204017, 4.525874]
            ]
          ]
        ],
        "id": "BRN"
      }
    }
  ]
}
```



```

# instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

# loop through the 100 crimes and add each to the incidents feature group
for lat, lng, in zip(df_incidents.Y, df_incidents.X):
    incidents.add_child(
        folium.vector_layers.CircleMarker(
            [lat, lng],
            radius=5, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )

# add incidents to map
sanfran_map.add_child(incidents)

# instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

# loop through the 100 crimes and add each to the incidents feature group
for lat, lng, in zip(df_incidents.Y, df_incidents.X):
    incidents.add_child(
        folium.vector_layers.CircleMarker(
            [lat, lng],
            radius=5, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )

```

```
# add pop-up text to each marker on the map
latitudes = list(df_incidents.Y)
longitudes = list(df_incidents.X)
labels = list(df_incidents.Category)

for lat, lng, label in zip(latitudes, longitudes, labels):
    folium.Marker([lat, lng], popup=label).add_to(sanfran_map)

# add incidents to map
sanfran_map.add_child(incidents)
```

Your grade: 100%

[Next item →](#)

Your latest: 100% • Your highest: 100% • To pass you need at least 70%. We keep your highest score.

1. True or False. Seaborn is based on Matplotlib.

1 / 1 point

- True
 False

Correct

Correct! Although Seaborn is another data visualization library, it is based on Matplotlib

2. The default map style in Folium is the _____.

1 / 1 point

- Stamen Terrain
 Stamen Toner
 Open Street Map
 Arial

Correct

Correct! The default map style in Folium is the Open Street Map. It shows a street view of an area when you are zoomed in and the borders of the world countries when you are zoomed out all the way.

3. The code for setting the initial zoom level in Folium is `zoom_start=parameter`.

1 / 1 point

- True
- False

 **Correct**

Correct! With the initial zoom, you can easily change the zoom level after the map is rendered by zooming in or out. You can play with this parameter to determine the initial zoom level for different values.

4. What parameter specifies the latitude and longitude coordinates of the map's center point?

1 / 1 point

- Location
- Navigation
- Destination
- Geographic

 **Correct**

Correct! The 'location' parameter specifies the latitude and longitude coordinates of the map's center point.

5. True or False. Markers represent specific locations or points of interest, providing additional information when clicked.

1 / 1 point

- True
- False

 **Correct**

Correct! Markers represent specific locations or points of interest, providing additional information when clicked. Markers are like signposts that guide us through the map, highlighting essential elements.

6. What type of map does a Choropleth map represent?

1 / 1 point

- Thematic
- Open Street Map
- Stamen Terrain
- Stamen Toner

 **Correct**

Correct! A choropleth map is a thematic map in which areas are shaded or patterned in proportion to the measurement of the statistical variable displayed on the map.

7. For what is Folium primarily used?

1 / 1 point

- Visualize geospatial data
- Depict the importance of different words in the body of the text
- Highlight progress against a given threshold
- Provide a high-level interface for drawing attractive statistical graphics

Correct

Correct! Folium is a Python library used for creating interactive maps and visualizations. It provides a simple and intuitive way to generate maps using data from various sources, including GeoJSON, Pandas DataFrames, and NumPy arrays.

8. To create a choropleth map of a region of interest, Folium requires a _____ file that includes geospatial data of the region.

1 / 1 point

- Geo
- HTML
- Json
- GeoJson

Correct

Correct! To create a choropleth map of a region of interest, Folium requires a GeoJson file that includes geospatial data of the region.

9. _____ provides specialized plot types such as regression, distribution, and categorical plots that are particularly useful for analyzing data and modeling relationships.

1 / 1 point

- Seaborn
- Bar plot
- Line plot
- Pie plot

 **Correct**

Correct! Seaborn provides specialized plot types such as regression, distribution, and categorical plots that are particularly useful for analyzing data and modeling relationships.

10. What type of data do Waffle Charts represent?

1 / 1 point

- Statistical
- Numerical
- Sequential
- Categorical

 **Correct**

Correct! Waffle charts are a visualization technique that represents categorical data in the form of square tiles or cells.

IV. Dashboard

Dashboard

- Real-time visuals simplify business moving parts
- Display Key Performance Indicators (KPI) for analysis
- Help businesses by providing the big picture

Best dashboards answer important business questions



Using `plotly.graph_objects`

```
# Import required packages
import plotly.graph_objects as go
import plotly.express as px
import numpy as np

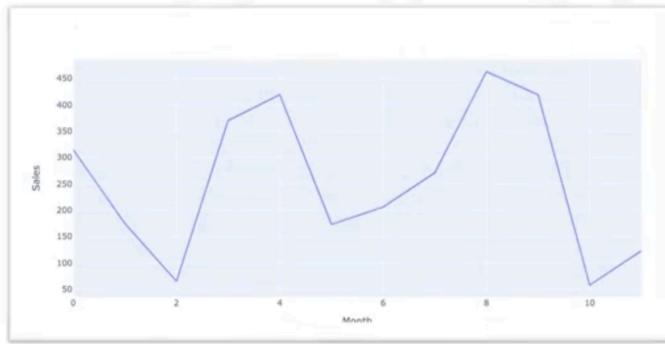
# Set random seed for reproducibility
np.random.seed(10)
x = np.arange(12)

# Create random y values
y = np.random.randint(50, 500, size=12)
```



Using plotly.graph_objects

```
fig = go.Figure(data=go.Scatter(x=x, y=y))  
fig.update_layout(title='Simple Line Plot', xaxis_title='Month', yaxis_title='Sales')  
fig.show()
```



```
fig = go.Figure()  
fig.add_trace(go.Scatter(x=data['Distance'], y=data['DepTime'], mode="markers")  
fig.update_layout(title='Distance vs Departure Time', xaxis_title='Distance', yaxis  
fig.show()
```

4.1. Dash

Dash Components

- Core components

```
import dash_core_components as dcc
```

- HTML Components

```
import dash_html_components as html
```

[dash-basic-html-and-core-components.pdf](#)

4.1.1. Callbacks

Dash: Callbacks

- The Callback function is a python function
- It is automatically called by Dash

Decorator

`@app.callback`

Dash: Callback function

```
def callback_function:  
    ....  
    ....  
    ....  
    return some_result
```

```
@app.callback(Output, Input)
```

Callback with one input

```
# Import required packages
import pandas as pd
import plotly.express as px
import dash
import dash_html_components as html
import dash_core_components as dcc
from dash.dependencies import Input, Output

# Read the data
airline_data = pd.read_csv('airline_2m.csv',
                           encoding = "ISO-8859-1",
                           dtype={'Div1Airport': str,
                                  'Div1TailNum': str,
                                  'Div2Airport': str,
                                  'Div2TailNum': str})
```



Callback with one input

```
app = dash.Dash()
# Design dash app layout
app.layout = html.Div(children=[ html.H1('Airline Dashboard',
                                         style={'textAlign': 'center', 'color': colors['text'],
                                                'font-size': 40}),
                                   html.Div(["Input: ", dcc.Input(id='input-yr', value='2010',
                                                               type='number', style={'height':'50px', 'font-size': 35}],),
                                   style={'font-size': 40},
                                   html.Br(),
                                   html.Br(),
                                   html.Div(dcc.Graph(id='bar-plot')),
                               ])
```



Callback with one input

```
@app.callback( Output(component_id='bar-plot', component_property='figure'),
                Input(component_id='input-yr', component_property='value'))

def get_graph(entered_year):
    # Select data
    df = airline_data[airline_data['Year']==int(entered_year)]
    # Top 10 airline carrier in terms of number of flights
    g1 = df.groupby(['Reporting_Airline'])['Flights'].sum().nlargest(10).reset_index()
    # Plot the graph
    fig1 = px.bar(g1, x='Reporting_Airline', y='Flights', title='Top 10 airline carrier in
                                                year ' + str(entered_year) + ' in terms of number of flights')
    fig1.update_layout()
    return fig1

if __name__ == '__main__':
    app.run_server(port=8002, host='127.0.0.1', debug=True)
```



[userinput-and-callbacks.pdf](#)

[creating-dash-application.pdf](#)

Function	Description	Syntax	Example
Plotly Express			
scatter	Create a scatter plot	<code>px.scatter(dataframe, x=x_column, y=y_column)</code>	<code>px.scatter(df, x=age_array, y=income_array)</code>
line	Create a line plot	<code>px.line(x=x_column, y=y_column, title='title')</code>	<code>px.line(x=months_array, y=no_bicycle_sold_array, title='Pass Percentage')</code>
bar	Create a bar plot	<code>px.bar(x=x_column, y=y_column, title='title')</code>	<code>px.bar(x=grade_array, y=score_array, title='Pass Percentage')</code>
sunburst	Create a sunburst plot	<code>px.sunburst(dataframe, path=[col1,col2..], values='column', title='title')</code>	<code>px.sunburst(data, path=['Month', 'DestStateName'], values='Flights', title='Flight Distribution Hierarchy')</code>
histogram	Create a histogram	<code>px.histogram(x=x, title='title')</code>	<code>px.histogram(x=heights_array, title='Distribution of Heights')</code>
bubble	Create a bubble chart	<code>px.scatter(dataframe, x=x,y=y,size=size,title=title")</code>	<code>px.scatter(bub_data, x="City", y="Numberofcrimes", size="Numberofcrimes", hover_name="City", title="Crime Statistics")</code>
pie	Create a pie chart	<code>px.pie(values=x, names=y, title=title")</code>	<code>px.pie(values=exp_percent, names=householdcategories, title="Household Expenditure")</code>
Plotly Graph Objects			
Scatter	Create a scatter	<code>go.Scatter(x=x, y=y, mode='markers')</code>	<code>go.Scatter(x=age_array, y=income_array, mode='markers')</code>
	Create a line plot	<code>go.Scatter(x=x, y=y, mode='lines')</code>	<code>go.Bar(x=months_array, y=no_bicycle_sold_array, mode='lines')</code>
add_trace	Add additional traces to an existing figure	<code>fig.add_trace(trace_object)</code>	<code>fig.add_trace(go.Scatter(x=months_array, y=no_bicycle_sold_array))</code>
update_layout	Update the layout of a figure, such as title, axis labels, and annotations.	<code>fig.update_layout(layout_object)</code>	<code>fig.update_layout(title='Bicycle Sales', xaxis_title='Months', yaxis_title='Number of Bicycles Sold')</code>
Dash			
dash_core_components.Input	Create an input component	<code>dcc.Input(value='', type='text')</code>	<code>dcc.Input(value='Hello', type='text')</code>
dash_core_components.Graph	Create a graph component	<code>dcc.Graph(figure=fig)</code>	<code>dcc.Graph(figure=fig)</code>
dash_html_components.Div	Create a div element	<code>html.Div(children=component_list)</code>	<code>html.Div(children=[html.H1('Hello Dash'), html.P('Welcome to Dash'))]</code>
dash_core_components.Dropdown	Create a dropdown component	<code>dcc.Dropdown(options=options_list, value=default_value)</code>	<code>dcc.Dropdown(options=[{'label': 'Option 1', 'value': '1'}, {'label': 'Option 2', 'value': '2'}], value='1')</code>

Your grade: 80%

Next item →

Your latest: 80% • Your highest: 80% • To pass you need at least 70%. We keep your highest score.

1. True or False. Web-based visualizations created using Plotly Python can be displayed in Jupyter Notebook, saved to standalone HTML files, or served as part of pure Python-built web applications using Dash. 1 / 1 point

True
 False

✓ **Correct**

Correct! Web-based visualizations created using Plotly Python can be displayed in Jupyter Notebook, saved to standalone HTML files, or served as part of pure Python-built web applications using Dash.

2. True or False. The Plotly graph objects module provides an automatically generated hierarchy of classes. It is the low-level interface to figures, traces, and layouts. 1 / 1 point

True
 False

✓ **Correct**

Correct! The Plotly graph objects module provides an automatically generated hierarchy of classes. It is the low-level interface to figures, traces, and layouts.

3. True or False. A Plotly.graph contains a JSON object which has a dictionary structure. 1 / 1 point

True
 False

✓ **Correct**

Correct! A Plotly.graph contains a JSON object which has a dictionary structure.

4. Fill in the blank. Dashboards offer _____ in real-time.

1 / 1 point

- Videos
- Messages
- Chat
- Visuals

 **Correct**

Correct! Real-time visuals on the dashboard simplify the comprehension of various aspects of the business. Also, getting the big picture in one place can help businesses make informed decisions, thereby improving performance.

5. Fill in the blank. Matplotlib is a comprehensive library for creating static, animated, and interactive _____ in Python.

1 / 1 point

- Graphs
- Charts
- Visualizations
- Videos

 **Correct**

Correct! Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

6. Data can be presented by using different types of what?

1 point

- Plots
- Dashboards
- Matplotlib
- Visuals

 Incorrect

Incorrect. Please review the Dashboarding Overview video.

7. True or False. Plotly Express is a high-level wrapper for Plotly.

1 / 1 point

- False
- True

 Correct

Correct! Plotly Express is a high-level wrapper for Plotly. It is a recommended starting point for creating the most common figures provided by Plotly because of its simple syntax. It uses graph objects internally.

8. A callback function is a _____.

1 / 1 point

- An extension
- A file
- Decorator
- A line of Code

 Correct

Correct! Whenever there is a change in the input component value, the Callback function wrapped by the decorator is called, followed by the update to the output component children in the application layout.

9. Dash is an Open-Source User Interface Python library for creating reactive, _____.

1 point

- Code
- Graphs
- Data
- Web-based applications

 Incorrect

Incorrect. Please review the video Make Dashboards Interactive.

10. True or False. Output sets results returned from the callback function to a component id.

1 / 1 point

- False
- True

 Correct

Correct! Output sets results returned from the callback function to a component id. The set input is provided to the callback function to a component id.

FINAL PROJECT:

Your grade: 93.33%

Next item →

Your latest: 93.33% • Your highest: 93.33% • To pass you need at least 73%. We keep your highest score.

1. Which best practice should you follow to ensure effective data visualization?

1 / 1 point

- Avoid legends and titles to keep the visualization uncluttered.
- Use multiple colors and fonts to make the visualization more vibrant.
- Label the axes clearly, provide context, and use only necessary data and labels.
- Include all available data to provide a comprehensive view, regardless of relevance.

 Correct

Correct! Label the axes clearly, provide context, and use only necessary data and labels.

2. Which layer within the Matplotlib architecture generates graphics and plots quickly and easily?

1 / 1 point

- Backend Layer
- FigureCanvas
- Figure Layer
- Scripting Layer

 Correct

Correct! The scripting layer is the appropriate layer for everyday purposes. It is considered a lighter scripting interface to simplify common tasks and for a quick and easy generation of graphics and plots.

3. Which plot library is an excellent choice for geospatial data visualization and helps to create interactive and customizable maps? 1 / 1 point

- Folium
- PyWaffle
- Seaborn
- Matplotlib

Correct

Correct! Folium is an excellent choice for geospatial data visualization and helps to create interactive and customizable maps.

4. What is the code for the Matplotlib magic function? 1 / 1 point

- \$matplotlib
- +matplotlib
- %matplotlib
- #matplotlib

Correct

Correct! The magic function code is %matplotlib. If the plot gets generated in a new pop window, you can enforce generating plots into a browser using this code.

5. Area plots are particularly effective in depicting data of what nature? 1 / 1 point

- Continuous dataset
- Population demographics
- Numerical data
- Metadata

Correct

Correct! Area plots are particularly effective in depicting data with a cumulative nature, such as visualizing population demographics.

6. What is a circular graphic that displays numeric proportions by dividing a circle into proportional slices? 1 / 1 point

- Pie chart
- Bar chart
- Radial column chart
- Table chart

Correct

Correct! A pie chart is a circular statistical graphic divided into segments to illustrate numerical proportions.

7. Fill in the blank. A _____ displays two variables against each other.

1 / 1 point

- Box plot
- Bar chart
- Scatter plot
- Pie chart

 **Correct**

Correct! A scatter plot is a type of plot that displays values about typically two variables against each other. Usually, the plotting involves a dependent variable against an independent variable.

8. Where can Waffle Charts be applied?

1 / 1 point

- Market research
- Customer feedback analysis
- Social media analysis
- Budget allocation

 **Correct**

Correct! You can use waffle charts to demonstrate the allocation of budgetary resources across categories or departments within an organization.

9. Which tool depicts the importance of different words in the body of text?

1 / 1 point

- A Word Cloud
- A Regression Plot
- A Box Plot
- A Waffle Chart

 **Correct**

Correct! A Word Cloud is an advanced visualization tool that depicts the importance of different words in the body of text. The more a specific word appears in a source of textual data, the bigger and bolder it appears in the word cloud.

10. What type of file does Folium require to create a Choropleth map of a specific region?

1 / 1 point

- Json
- Geo
- GeoJson
- HTML

 **Correct**

Correct! To create a choropleth map of a region of interest, Folium requires a GeoJson file that includes geospatial data of the region.

11. What feature enhances interactivity and provides contextual information to maps?

1 / 1 point

- Longitude
- Latitude
- Markers
- Zoom function

 **Correct**

Correct! Markers represent specific locations or points of interest, providing additional information when clicked. Markers are like signposts that guide us through the map, highlighting essential elements.

12. Which tool is the Python framework for building web analytic applications?

1 / 1 point

- Streamlit
- Dash
- Panel
- Voilà

 **Correct**

Correct! The Python framework Dash creates web analytics applications. It is designed for creating data visualization apps with highly customized user interfaces. It builds on top of Flask, Plotly.js, and React.js.

13. What command creates a line chart using Plotly Graph Objects?

1 point

- fig.show()
- px.line()
- go.Scatter()
- go.Figure()

 **Incorrect**

Incorrect. Please review the video introduction to Plotly.

14. What does the @app.callback decorator tell Dash when used in a Callback function?

1 / 1 point

- It tells that the function is automatically called by Dash whenever an input component's property changes.
- It tells you to connect the input and output components to the desired properties.
- It tells you to provide the title to the dash app.
- It tells output component children in the application layout.

 **Correct**

Correct! A Callback function is automatically called by Dash whenever an input component's property changes.

15. Which parameter is needed to decorate a callback function?

1 / 1 point

- Dash core
- Dash
- Input
- HTML

 **Correct**

Correct! Output sets results returned from the callback function to a component id. The callback function receives this set input as a component id.

