
NITTE
EDUCATION TRUST
Data Science using Python
Dr. Vani Vasudevan

Outline

- Introduction to Data Science
- Data Preprocessing
- Data Visualization
- Introduction to Machine Learning
- Foundation of Neural Networks and Deep Learning

Unit III - Data Visualization

The greatest value of a picture is when it forces us to notice what we never expected to see.

- John Tukey

NITTE EDUCATION TRUST

Data Visualization – Definition & Importance

- Visual representation reveals patterns and trends
- A picture worth thousand words true
- Essential for exploring and presenting data
- Provides insights beyond summary statistics alone
- Simplifies complex datasets for better understanding
- Supports decision making with clearer communication

NITTE EDUCATION TRUST

Data Visualization using Matplotlib

- Matplotlib multiplatform library for visualization tasks
- Built on NumPy arrays efficient processing
- Designed to integrate with SciPy stack
- Conceived by John Hunter in 2002
- Works seamlessly across multiple different platforms
- Widely used tool for scientific visualization

NITTE EDUCATION TRUST

Data Visualization using Matplotlib

Overview: Matplotlib is the fundamental library for static visualizations.

- It's a tool for creating publication-ready charts.
- It is a very large library with the varied capabilities
- The recommended interface for matplotlib is pyplot

- **Key Features:** Line plots, scatter plots, bar charts, etc.

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

NITTE EDUCATION TRUST

Data Visualization using Matplotlib

- Figures are containers for graphical data
- Axes define regions with coordinate points
- One figure can contain multiple axes
- Figures axes created explicit or implicit
- Implicit uses plt.plot plt.hist functions
- plt.plot generates line plot with values

NITTE EDUCATION TRUST

Style Plots using Matplotlib

- **Customization:** Modify colors, styles, marker, and grids for enhanced aesthetics.

```
x= np.range(10)
y =np.range(10,110,10)
plt.plot(x, y, color='red', linestyle='--', linewidth=2, marker='s')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Styled Line Plot')
plt.grid(True)
plt.show()
```

NITTE EDUCATION TRUST

Line Chart in Matplotlib

Visualizing trends over time.

```
plt.plot([1, 2, 3, 4], [10, 20, 25, 30])
plt.title('Simple Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

NITTE EDUCATION TRUST

Bar Plot in Matplotlib

- Comparing categories.

```
categories = ['A', 'B', 'C', 'D']
values = [4, 7, 1, 8]
plt.bar(categories, values)
plt.title('Bar Plot')
plt.xlabel('Category')
plt.ylabel('Value')
plt.show()
```

NITTE EDUCATION TRUST

Box Plot in Matplotlib

- Visualizing distributions and outliers.

```
data = [list(range(1, 8)), list(range(2, 9)), list(range(3, 10))]
plt.boxplot(data)
plt.title('Box Plot')
plt.show()
```

NITTE EDUCATION TRUST

Scatter Plot in Matplotlib

- Exploring relationships between variables.

```
x = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11]
y = [99, 86, 87, 88, 100, 86, 103, 87, 94, 78]
plt.scatter(x, y)
plt.title('Scatter Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

NITTE EDUCATION TRUST

Heatmap in Matplotlib

- Displaying matrix data such as correlations.

```
import numpy as np
data = np.random.rand(10, 10)
plt.imshow(data, cmap='coolwarm', interpolation='nearest')
plt.colorbar()
plt.title('Heatmap')
plt.show()
```

NITTE EDUCATION TRUST

Subplots in Matplotlib

- Subplots divide figure into plotting grid
- Use subplot function with three arguments
- nrows specifies number of subplot rows
- ncols defines number of subplot columns
- Index determines subplot position in grid
- Enables organizing multiple plots within figure

NITTE EDUCATION TRUST

Subplots in Matplotlib

```
import matplotlib.pyplot as plt
# Create a 2x2 grid of subplots
plt.subplot(2, 2, 1)  # First subplot
plt.plot([1, 2, 3], [1, 4, 9])
plt.title('Plot 1')

plt.subplot(2, 2, 2)  # Second subplot
plt.plot([1, 2, 3], [1, 2, 3])
plt.title('Plot 2')

plt.subplot(2, 2, 3)  # Third subplot
plt.plot([1, 2, 3], [3, 1, 4])
plt.title('Plot 3')

plt.subplot(2, 2, 4)  # Fourth subplot
plt.plot([1, 2, 3], [9, 4, 1])

plt.title('Plot 4')
plt.tight_layout()  # Adjust spacing between subplots
plt.show()
```

NITTE EDUCATION TRUST

Subplots in Matplotlib

2. Using plt.subplots()

The plt.subplots() function is more flexible and is often preferred for more complex subplot configurations.

fig, axes = plt.subplots(nrows, ncols)

Where:

nrows is the number of rows,

ncols is the number of columns,

axes is an array of subplots (when more than 1 subplot is created).

NITTE EDUCATION TRUST

Subplots in Matplotlib

```
import matplotlib.pyplot as plt
# Create a 2x2 grid of subplots
fig, axes = plt.subplots(2, 2)

# First subplot
axes[0, 0].plot([1, 2, 3], [1, 4, 9])
axes[0, 0].set_title('Plot 1’)

# Second subplot
axes[0, 1].plot([1, 2, 3], [1, 2, 3])
axes[0, 1].set_title('Plot 2')

# Third subplot
axes[1, 0].plot([1, 2, 3], [3, 1, 4])
axes[1, 0].set_title('Plot 3')

# Fourth subplot
axes[1, 1].plot([1, 2, 3], [9, 4, 1])
axes[1, 1].set_title('Plot 4')

plt.tight_layout()  # Adjust spacing between subplots
plt.show()
```

NITTE EDUCATION TRUST

Customizing Subplots

1. **Adjust Spacing:** You can use plt.tight_layout() to automatically adjust the spacing between subplots to prevent overlapping labels.
2. **Share Axis:** You can also share the x- or y-axis between subplots by using the sharex or sharey arguments in plt.subplots().

fig, axes = plt.subplots(2, 2, sharex=True, sharey=True)

Setting Figure Size: You can adjust the overall size of the figure with the figsize argument in plt.subplots():

fig, axes = plt.subplots(2, 2, figsize=(10, 8))

NITTE EDUCATION TRUST

Three-Dimensional Plotting using Matplotlib

- Visualizing data in 3D space

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x = [1, 2, 3, 4, 5]
y = [5, 6, 7, 8, 9]
z = [10, 11, 12, 13, 14]
ax.scatter(x, y, z)
plt.title('3D Scatter Plot')
plt.show()
```

NITTE EDUCATION TRUST

Time Series Plot using Pandas

- Plotting time series data.

```
import pandas as pd
data = pd.Series([10, 20, 30, 40],
index=pd.date_range('20230101', periods=4))
data.plot(title='Time Series Plot')
plt.show()
```

NITTE EDUCATION TRUST

Geospatial Data Visualization

- Plotting data on maps.

```
import geopandas as gpd
world =
gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world.plot()
plt.title('Geospatial Data Plot')
plt.show()
```

NITTE EDUCATION TRUST

Data Visualization with Seaborn

- Seaborn builds on Matplotlib for more complex statistical plot

```
import seaborn as sns
iris = sns.load_dataset('iris')
sns.pairplot(iris)
plt.title('Pair Plot with Seaborn')
plt.show()
```

NITTE EDUCATION TRUST

1. Scatter Plot

Used to visualize the relationship between two continuous variables.

```
import seaborn as sns
import matplotlib.pyplot as plt # Load a sample dataset

df = sns.load_dataset("tips") # Create a scatter plot
sns.scatterplot(x="total_bill", y="tip", data=df)
plt.show()
```

NITTE EDUCATION TRUST

2. Line Plot

Used to display the trend of a variable over time or some continuous index

```
sns.lineplot(x="size", y="tip", data=df)
plt.show()
```

NITTE EDUCATION TRUST

3. Bar Plot

Displays the relationship between a categorical variable and a continuous variable.

```
sns.barplot(x="day", y="total_bill", data=df)
plt.show()
```

NITTE EDUCATION TRUST

4. Count Plot

Used to count the number of observations in each categorical bin.

```
sns.countplot(x="day", data=df)
plt.show()
```

NITTE EDUCATION TRUST

5. Box Plot

Displays the distribution of a continuous variable and highlights outliers.

```
sns.boxplot(x="day", y="total_bill", data=df)
plt.show()
```

NITTE EDUCATION TRUST

6. Violin Plot

Combines aspects of a box plot and a KDE plot to provide deeper insights into the distribution.

```
sns.violinplot(x="day", y="total_bill", data=df)
plt.show()
```

7. **Swarm Plot**

A categorical scatter plot where points are adjusted to avoid overlapping.

```
sns.swarmplot(x="day", y="total_bill", data=df)
plt.show()
```

NITTE EDUCATION TRUST

8. Strip Plot

A basic scatter plot for categorical data, often used in combination with a box or violin plot.

```
sns.stripplot(x="day", y="total_bill", data=df)
plt.show()
```

NITTE EDUCATION TRUST

9. Histogram

Displays the distribution of a continuous variable.

```
sns.histplot(df["total_bill"], kde=True)
plt.show()
```

NITTE EDUCATION TRUST

10. **KDE Plot (Kernel Density Estimate)**

Shows the distribution of a continuous variable, smoothed by a Gaussian kernel.

```
sns.kdeplot(df["total_bill"], shade=True)
plt.show()
```

NITTE EDUCATION TRUST

11. **Pair Plot**

Displays pairwise relationships across an entire dataframe. Particularly useful for visualizing the relationships between numerical variables.

```
sns.pairplot(df)
plt.show()
```

NITTE EDUCATION TRUST

12. **Heatmap**

- Used to visualize data as a color-coded matrix. Often used for correlation matrices.

```
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.show()
```

NITTE EDUCATION TRUST

13. **FacetGrid**

- Creates subplots of different subsets of your data using categorical variables.

```
g = sns.FacetGrid(df, col="sex")
g.map(sns.scatterplot, "total_bill", "tip")
plt.show()
```

NITTE EDUCATION TRUST

14. **Joint Plot**

Displays the relationship between two variables along with their individual distributions.

```
sns.jointplot(x="total_bill", y="tip", data=df, kind="scatter")
plt.show()
```

NITTE EDUCATION TRUST

15. **Pair** Grid

Similar to pairplot, but with more flexibility to define how to plot different types of relationships.

```
g = sns.PairGrid(df)
g.map_upper(sns.scatterplot)
g.map_lower(sns.kdeplot)
 g.map_diag(sns.histplot)
plt.show()
```

NITTE EDUCATION TRUST

16. Cat Plot

Combines different categorical plots such
as bar, box, strip, swarm, etc. into one flexible interface.

```
sns.catplot(x="day", y="total_bill", kind="box", data=df)
plt.show()
```

NITTE EDUCATION TRUST

17. **Point Plot**

Shows the mean value of a point and displays error bars to indicate confidence intervals.

```
sns.pointplot(x="day", y="total_bill", data=df)
plt.show()
```

NITTE EDUCATION TRUST

18. **Regression Plot (LM Plot)**

Fits and plots a regression line along with confidence intervals.

```
sns.lmplot(x="total_bill", y="tip", data=df)
plt.show()
```

19. **Residual Plot**

Shows the residuals of a regression, helping to assess the quality of the fit.

```
sns.residplot(x="total_bill", y="tip", data=df)
plt.show()
```

NITTE EDUCATION TRUST

20. **Matrix Plot (Clustermap)**

Displays a matrix where rows and columns are reordered based on similarity.

```
sns.clustermap(df.corr(), cmap="coolwarm", annot=True)
plt.show()
```

NITTE EDUCATION TRUST

Customizing Seaborn plots

Change Aesthetics: You can use sns.set_style() to modify the overall appearance of your plots.

```
sns.set_style("whitegrid")
#Add Titles and Labels: Always add titles and labels for readability.
plt.title("Plot Title")
plt.xlabel("X-axis label")
plt.ylabel("Y-axis label")


#Color Palettes: You can change the color palette with sns.set_palette().
sns.set_palette("husl")
```

NITTE EDUCATION TRUST

Interactive Visualizations with Plotly

Create interactive plots with Plotly.

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df, x='sepal_width', y='sepal_length',
color='species')
fig.show()
```

NITTE EDUCATION TRUST

Scatter Plot

A scatter plot shows the relationship between two variables.

```
fig = px.scatter(df, x='sepal_width', y='sepal_length',
color='species')
```

Line Plot

Line plots are useful for visualizing trends over time.

```
fig = px.line(df, x='sepal_width', y='sepal_length', color='species')
```

NITTE EDUCATION TRUST

Bar Plot

Bar plots are ideal for comparing categorical data.

```
fig = px.bar(df, x='species', y='sepal_length', color='species')
```

NITTE EDUCATION TRUST

Histogram

Histograms display the distribution of a variable.

```
fig = px.histogram(df, x='sepal_length', nbins=20)
```

NITTE EDUCATION TRUST

Box Plot

Box plots display the distribution of data and highlight outliers.

```
fig = px.box(df, x='species', y='sepal_length', color='species')
```

NITTE EDUCATION TRUST

Violin Plot

Violin plots show the distribution of the data and its probability density.

```
fig = px.violin(df, x='species', y='sepal_length', color='species')
```

NITTE EDUCATION TRUST

Pie Chart

Pie charts display proportions of a whole for different categories.

```
fig = px.pie(df, values='sepal_length', names='species')
```

NITTE EDUCATION TRUST

Sunburst Plot

A sunburst plot is used for hierarchical data visualization.

```
fig = px.sunburst(df, path=['species', 'sepal_width'], values='sepal_length')
```

NITTE EDUCATION TRUST

Treemap

Treemaps are used to represent hierarchical data using nested rectangles.

```
fig = px.treemap(df, path=['species', 'sepal_width'], values='sepal_length')
```

NITTE EDUCATION TRUST

Heatmap/Density Heatmap

A heatmap shows matrix-like data with color representing the values.

```
z = np.random.random((10, 10))
fig = go.Figure(data=go.Heatmap(z=z))
```

A specialized heatmap that represents the density of data points.

```
fig = px.density_heatmap(df, x='sepal_width', y='sepal_length', marginal_x='histogram', marginal_y='violin')
fig.show()
```

NITTE EDUCATION TRUST

3D Scatter Plot/3D Line Plot

A 3D scatter plot is used to plot three variables at once.

```
fig = px.scatter_3d(df, x='sepal_width', y='sepal_length', z='petal_length', color='species')
```

Line plots in 3D space can help visualize trends across three variables.

```
fig = px.line_3d(df, x='sepal_width', y='sepal_length', z='petal_length', color='species')
fig.show()
```

NITTE EDUCATION TRUST

Bubble Plot

A bubble plot is a scatter plot where each point has a third dimension represented by the size of the marker.

```
fig = px.scatter(df, x='sepal_width', y='sepal_length',
size='petal_length', color='species')
```

NITTE EDUCATION TRUST

Polar Chart

A polar chart is used for data plotted on a circular grid.

```
fig = px.line_polar(df, r='sepal_length', theta='species',
color='species')
```

Area Plot

An area plot shows the cumulative totals over time, with areas stacked on top of each other.

```
fig = px.area(df, x='sepal_width', y='sepal_length',
color='species', line_group='species')
fig.show()
```

NITTE EDUCATION TRUST

Funnel Chart

Funnel charts are used to represent stages in a process or pipeline.

```
fig = px.funnel(df, x='species', y='sepal_length')
```

NITTE EDUCATION TRUST

Funnel Area Chart

Similar to a funnel chart, but using an area to display the stages.

```
fig = px.funnel_area(df, names='species',
values='sepal_length')
```

NITTE EDUCATION TRUST

Radar Chart

A radar chart, or spider chart, is used for comparing multiple variables for different categories.

```
fig = px.line_polar(df, r='sepal_length', theta='species', line_close=True)
```

Scatter Matrix (Splom)

A scatter matrix (splom) shows pairwise relationships between variables.

```
fig = px.scatter_matrix(df, dimensions=['sepal_length', 'sepal_width', 'petal_length'], color='species')
```

Contour Plot

A contour plot represents three-dimensional data in two dimensions using contours or color-coded regions.

```
fig = px.density_contour(df, x='sepal_width', y='sepal_length')
```

NITTE EDUCATION TRUST

Facet Plot

Facet plots allow you to split data into multiple plots based on a categorical variable.

```
fig = px.scatter(df, x='sepal_width', y='sepal_length',
color='species', facet_col='species')
```

Marginal Plot

Adds marginal histograms or density plots to scatter plots for better distribution analysis.

```
fig = px.scatter(df, x='sepal_width', y='sepal_length',
color='species', marginal_x='histogram', marginal_y='kde')
```

NITTE EDUCATION TRUST

Animated Scatter Plot

Animate scatter plots over time to show changes in data.

```
df_gap = px.data.gapminder()
fig = px.scatter(df_gap, x='gdpPercap', y='lifeExp',
animation_frame='year')
```

NITTE EDUCATION TRUST

Animated Bar Plot

Create animated bar plots to show changes in categories over time.

```
fig = px.bar(df, x='species', y='sepal_length',
animation_frame='species', color='species')
```

NITTE EDUCATION TRUST

Customizing Plotly Visualizations

Adding Titles and Labels: You can add titles and axis labels using update_layout():

```
fig.update_layout(
    title="Title",
    xaxis_title="X-axis Label",
    yaxis_title="Y-axis Label"
)
```

Updating Markers: You can change the size, color, and style of markers:

```
fig.update_traces(marker=dict(size=12, line=dict(width=2, color='DarkSlateGrey')))
```

NITTE EDUCATION TRUST

Customizing Plotly Visualizations

Hover Text: Customize the hover text using the hover_data argument:

```
fig = px.scatter(df, x='sepal_width', y='sepal_length',
hover_data=['species’])
```

Themes: Plotly offers built-in themes like "plotly_dark" and "ggplot2":

```
fig.update_layout(template="plotly_dark")
```

NITTE EDUCATION TRUST

Interactive Data Visualization with Bokeh

Bokeh creates interactive web-ready plots.

```
from bokeh.plotting import figure, show
p = figure(title="Bokeh Interactive Plot")
p.circle([1, 2, 3, 4], [4, 7, 1, 6], size=15, color="navy",
alpha=0.5)
show(p)
```

NITTE EDUCATION TRUST

Line Plot

- Line plots are useful for visualizing trends over time.

```
from bokeh.plotting import figure, show
p = figure(title='Line Plot', x_axis_label='X', y_axis_label='Y')
p.line([1, 2, 3, 4], [4, 7, 2, 5], line_width=2)
show(p)
```

NITTE EDUCATION TRUST

Scatter Plot

- Scatter plots show relationships between two variables.

```
from bokeh.plotting import figure, show
p = figure(title='Scatter Plot', x_axis_label='X',
y_axis_label='Y')
p.scatter([1, 2, 3, 4], [4, 7, 2, 5], size=10)
show(p)
```

NITTE EDUCATION TRUST

Bar Plot

- Bar plots are useful for comparing categories.

```
from bokeh.plotting import figure, show
p = figure(x_range=['Category A', 'Category B', 'Category C'], title="Bar Plot", toolbar_location=None)
p.vbar(x=['Category A', 'Category B', 'Category C'], top=[4, 7, 2], width=0.5)
show(p)
```

NITTE EDUCATION TRUST

Histogram

- Histograms show the distribution of data.

```
import numpy as np
from bokeh.plotting import figure, show
data = np.random.randn(1000)
p = figure(title='Histogram')
p.quad(top=[10, 20, 30], bottom=0, left=[0, 1, 2], right=[1, 2, 3])
show(p)
```

NITTE EDUCATION TRUST

Box Plot

- Box plots display the distribution and outliers in a dataset.

```
from bokeh.io import output_file, show
from bokeh.plotting import figure
from bokeh.transform import jitter

output_file("boxplot.html")
p = figure(x_range=['Category A', 'Category B'], title="Box Plot with Jitter")
p.circle(jitter('Category', 0.5), [1, 3, 2, 4, 5], size=10)
show(p)
```

NITTE EDUCATION TRUST

Heatmap

- Heatmaps are useful for showing data density and correlations.

```
import numpy as np
from bokeh.plotting import figure, show
from bokeh.models import LinearColorMapper, ColorBar
data = np.random.rand(10, 10)
p = figure(title="Heatmap", x_range=(0, 10), y_range=(0, 10))
p.image(image=[data], x=0, y=0, dw=10, dh=10)
show(p)
```

NITTE EDUCATION TRUST

Pie Chart

- Pie charts show proportions of a whole in different categories.

```
from math import pi
from bokeh.palettes import Category20c
from bokeh.transform import cumsum
from bokeh.plotting import figure, show
data = {'category': ['A', 'B', 'C'], 'value': [10, 20, 30]}
p = figure(title="Pie Chart", toolbar_location=None)
p.wedge(x=0, y=0, radius=1, start_angle=pi/6, end_angle=pi,
color="navy")
show(p)
```

NITTE EDUCATION TRUST

Network Graph

- Network graphs show relationships between entities.

```
from bokeh.io import show
from bokeh.models import GraphRenderer, StaticLayoutProvider, Oval
from bokeh.plotting import figure

plot = figure(title="Network Graph")
graph = GraphRenderer()
plot.renderers.append(graph)
show(plot)
```

NITTE EDUCATION TRUST

Area Plot

- Area plots show the cumulative change in a variable over time.

```
from bokeh.plotting import figure, show
p = figure(title="Area Plot")
p.varea(x=[1, 2, 3, 4], y1=[1, 4, 2, 3], y2=[2, 5, 3, 4])
show(p)
```

NITTE EDUCATION TRUST

Interactive Plot

- Interactive plots allow users to zoom, pan, and hover over data points.

```
from bokeh.plotting import figure, show
p = figure(tools="pan,box_zoom,reset,hover",
title="Interactive Plot")
p.circle([1, 2, 3, 4], [4, 7, 2, 5], size=10)
show(p)
```

NITTE EDUCATION TRUST

Summary – Data Visualization (Unit 3)

- Visualization reveals patterns trends simplifies understanding
- Matplotlib core static charts publication ready
- Seaborn statistical plots built on Matplotlib
- Plotly interactive powerful dashboards web ready
- Bokeh enables interactive browser visualizations easily
- Supports scatter line bar box heatmaps

NITTE EDUCATION TRUST

Summary – Chart Types & Uses

- Scatter line bar plots show relationships
- Box violin histogram KDE distribution insights
- Heatmaps correlation matrices density visualizations effective
- Subplots manage multiple graphs within figure
- 3D plots geospatial visualizations advanced exploration
- Specialized charts include radar funnel treemap
-

NITTE EDUCATION TRUST

Comparison – Visualization Libraries in Python

- **Matplotlib**: Core static plots, highly customizable
- **Seaborn**: Built on Matplotlib, statistical visualizations
- **Plotly**: Interactive, web dashboards, advanced visuals
- **Bokeh**: Interactive browser plots, web integration
- Matplotlib simple Seaborn simplifies aesthetics choices
- Plotly Bokeh enable dynamic engaging interactivity

NITTE EDUCATION TRUST