Experiment No. 5

Statistical data analysis plots analysis using Python (Seaborn)

Batch: A3 Roll No: 16010423099 Experiment No.:5

Aim: To perform statistical data analysis plots analysis using Python (Seaborn)

Resources needed: Python IDE

Theory:

- Seaborn creates detailed visualizations that reveal patterns, trends, and relationships within data, aiding in data exploration and interpretation.
- Built on Matplotlib, Seaborn offers a high-level interface for drawing attractive and informative statistical graphics, simplifying complex data visualization.
- Seaborn provides various plot types, including scatter plots, line plots, histograms, and box plots, each designed to highlight different aspects of data.
- Seamlessly integrating with Pandas Data Frames, Seaborn allows efficient visualization directly from tabular data, with built-in themes and color palettes ensuring aesthetically pleasing results.

Basically we do following things in SDA.

Data Collection and Cleaning: Ensure the data is accurate and ready for analysis by gathering relevant data and handling issues like missing values and inconsistencies.

Exploratory Data Analysis (EDA): Use visualizations and statistical techniques to uncover patterns, correlations, and trends, gaining initial insights into the data.

Hypothesis Testing: Validate assumptions and compare groups by formulating and testing hypotheses using appropriate statistical tests.

Predictive Modeling: Develop models to make predictions or identify key factors influencing outcomes, driving data-driven decision-making.

Seaborn is a Python data visualization library built on top of Matplotlib that provides a high-level interface for creating informative and attractive statistical graphics. It simplifies the process of creating complex plots and is particularly well-suited for visualizing datasets that contain multiple variables.

Key Features of Seaborn

- **Integration with Pandas:** Seaborn works seamlessly with Pandas DataFrames, allowing you to easily plot data directly from these structures.
- **Built-in Themes and Color Palettes:** Seaborn provides aesthetically pleasing default themes and color palettes that can be customized.
- **Statistical Plotting:** Seaborn includes functions for visualizing relationships among variables, distribution of data, and comparing categorical data.

1. Common Seaborn Commands and Examples

1. Scatter Plot

- **Purpose:** Visualizes the relationship between two continuous variables.
- **Command:** sns.scatterplot()

2. Line Plot

- Purpose: Displays trends over time or a sequence.
- Command: sns.lineplot()

3. Histogram and KDE Plot

- Purpose: Visualizes the distribution of a single continuous variable.
- Commands: sns.histplot(), sns.kdeplot()

4. Box Plot

- **Purpose:** Displays the distribution of data based on quartiles and highlights outliers.
- **Command:** sns.boxplot()

5. Bar Plot

- **Purpose:** Compares the mean (or other aggregate values) of a continuous variable across categories.
- **Command:** sns.barplot()

6. Pair Plot

- **Purpose:** Visualizes pairwise relationships in a dataset.
- **Command:** sns.pairplot()

7. Heatmap

- **Purpose:** Visualizes matrix-like data (e.g., correlation matrices).
- **Command:** sns.heatmap()

8. Violin Plot

- **Purpose:** Combines box plot and KDE to show the distribution of data across different categories.
- **Command:** sns.violinplot()

Customization

Seaborn allows extensive customization of plots:

- **Themes:** Change the overall style using sns.set_style() (e.g., "whitegrid", "darkgrid").
- **Color Palettes:** Customize colors using sns.set_palette() or specify colors directly within plotting functions.
- **Axes and Labels:** Use Matplotlib functions like plt.xlabel(), plt.ylabel(), and plt.title() to modify labels and titles.

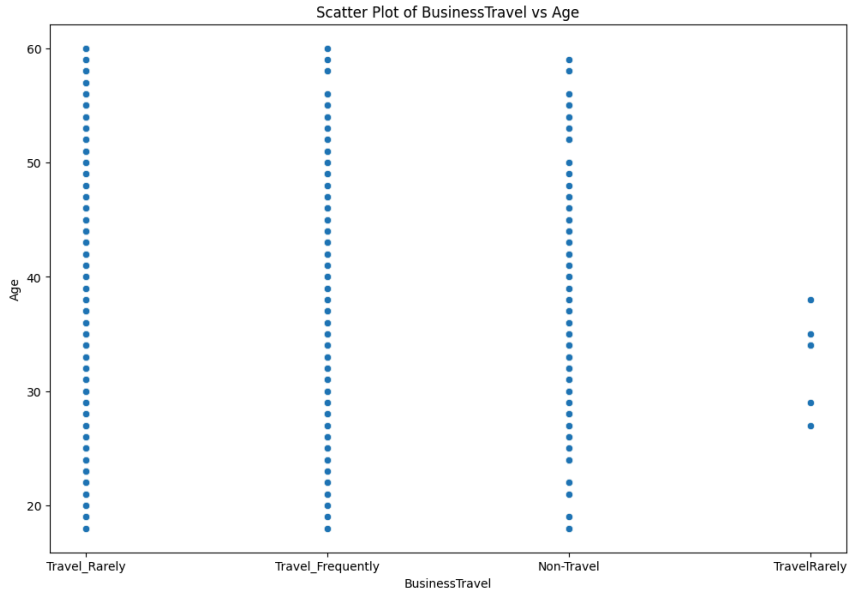
Activities:

1. Download data set with at least 1500 rows and 10-20 columns (numeric and non-numeric) from valid data sources
2. Perform in detail statistical data analysis of this dataset
    - Visualize Distribution
    - Analyse Relationships
    - Explore Categorical Data
    - Examine Pairwise Relationships
    - Compare Groups
    - Trend Analysis
    - Visualize Data Distributions by Groups
    - Customize Plots
    - Create Regression Plots

Result: (script and output)

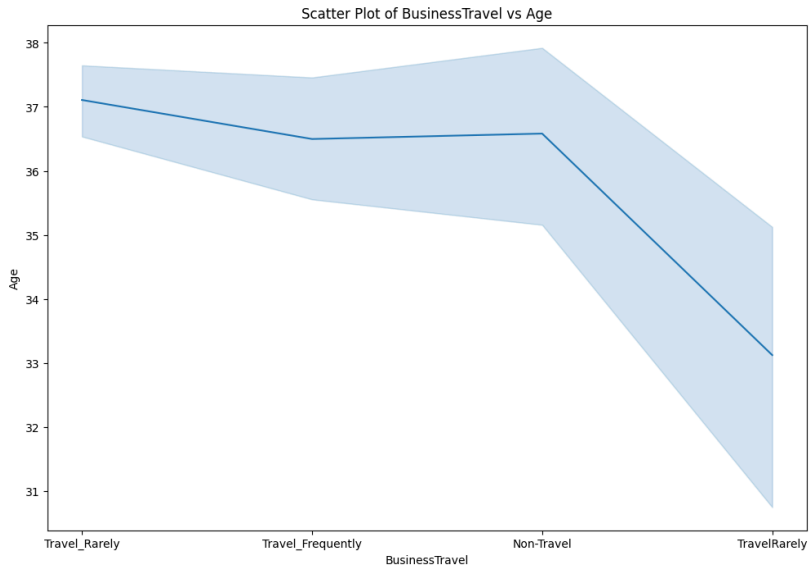
- Scatter :

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd


df = pd.read_csv('/content/HR_Analytics.csv')
plt.figure(figsize=(12, 8))

sns.scatterplot(data=df, x='BusinessTravel', y='Age')


plt.title('Scatter Plot of BusinessTravel vs Age')
plt.xlabel('BusinessTravel')
plt.ylabel('Age')

plt.show()
```



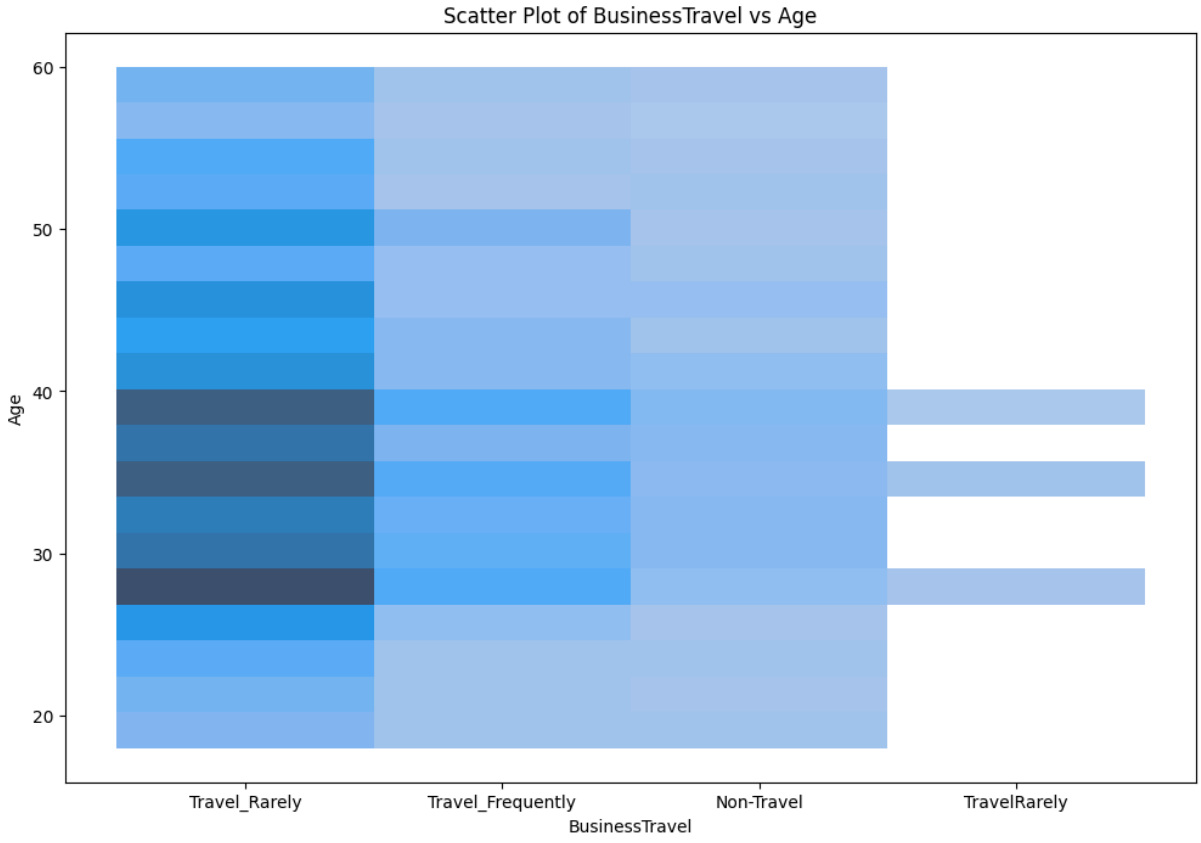
- Lineplot:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd



df = pd.read_csv('/content/HR_Analytics.csv')
plt.figure(figsize=(12, 8))

sns.lineplot(data=df, x='BusinessTravel', y='Age')



plt.title('Scatter Plot of BusinessTravel vs Age')
plt.xlabel('BusinessTravel')
plt.ylabel('Age')

plt.show()
```



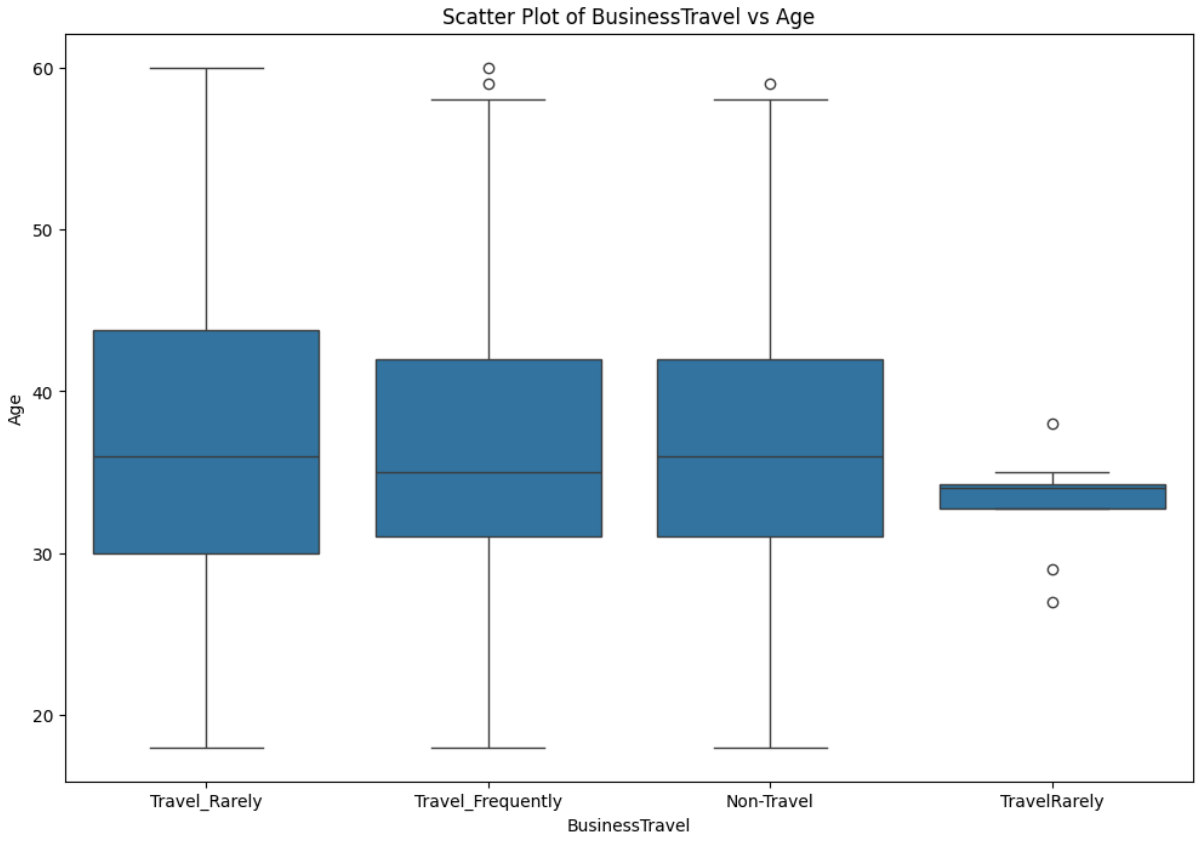
- Histplot:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd


df = pd.read_csv('/content/HR_Analytics.csv')
plt.figure(figsize=(12, 8))

sns.histplot(data=df, x='BusinessTravel', y='Age')


plt.title('Scatter Plot of BusinessTravel vs Age')
plt.xlabel('BusinessTravel')
plt.ylabel('Age')

plt.show()
```



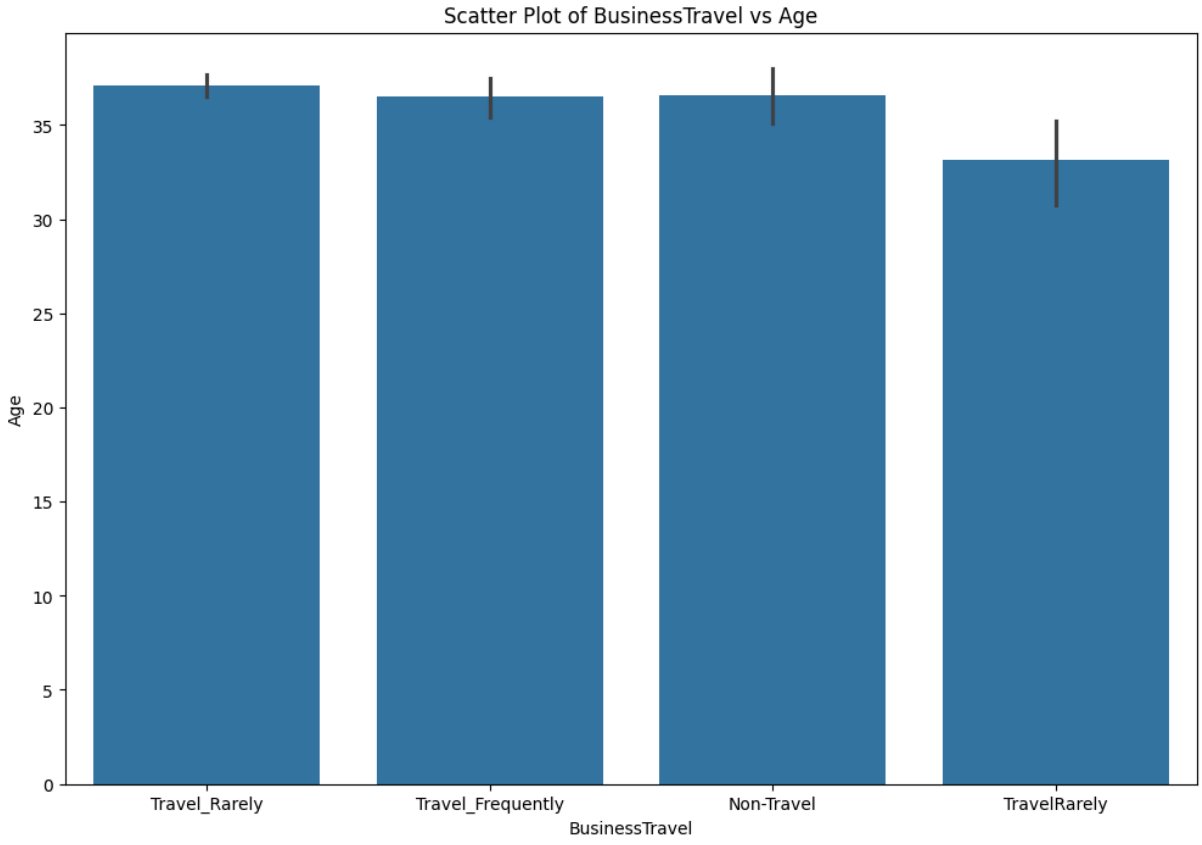
- Boxplot:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd


df = pd.read_csv('/content/HR_Analytics.csv')
plt.figure(figsize=(12, 8))

sns.boxplot(data=df, x='BusinessTravel', y='Age')


plt.title('Scatter Plot of BusinessTravel vs Age')
plt.xlabel('BusinessTravel')
plt.ylabel('Age')

plt.show()
```



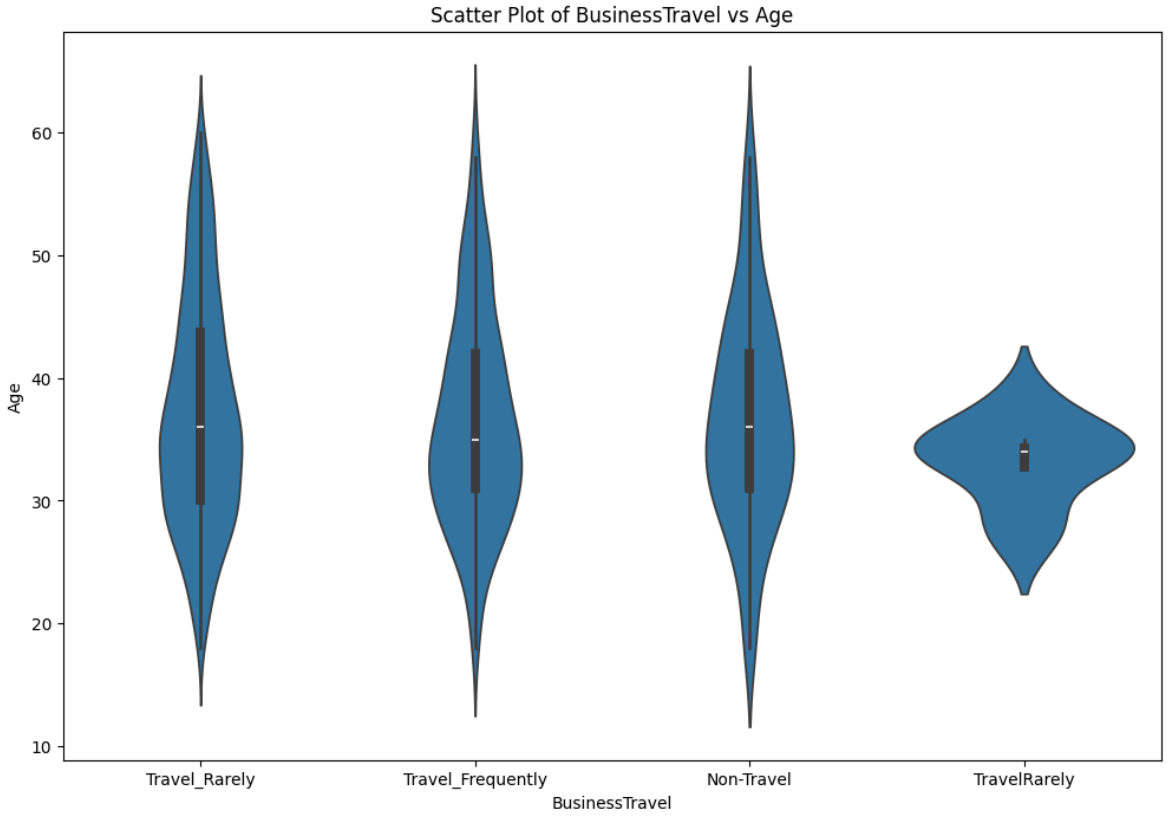
- Barplot:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd


df = pd.read_csv('/content/HR_Analytics.csv')
plt.figure(figsize=(12, 8))

sns.barplot(data=df, x='BusinessTravel', y='Age')


plt.title('Scatter Plot of BusinessTravel vs Age')
plt.xlabel('BusinessTravel')
plt.ylabel('Age')

plt.show()
```



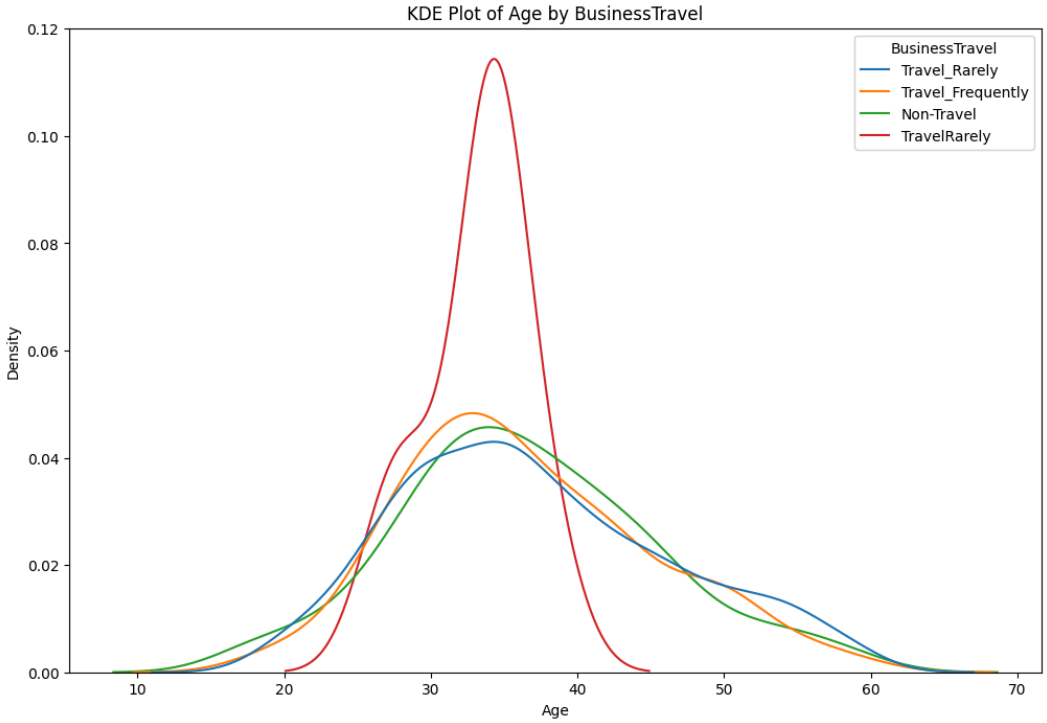
- Violinplot:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd


df = pd.read_csv('/content/HR_Analytics.csv')
plt.figure(figsize=(12, 8))

sns.violinplot(data=df, x='BusinessTravel', y='Age')


plt.title('Scatter Plot of BusinessTravel vs Age')
plt.xlabel('BusinessTravel')
plt.ylabel('Age')

plt.show()
```



- kdeplot:

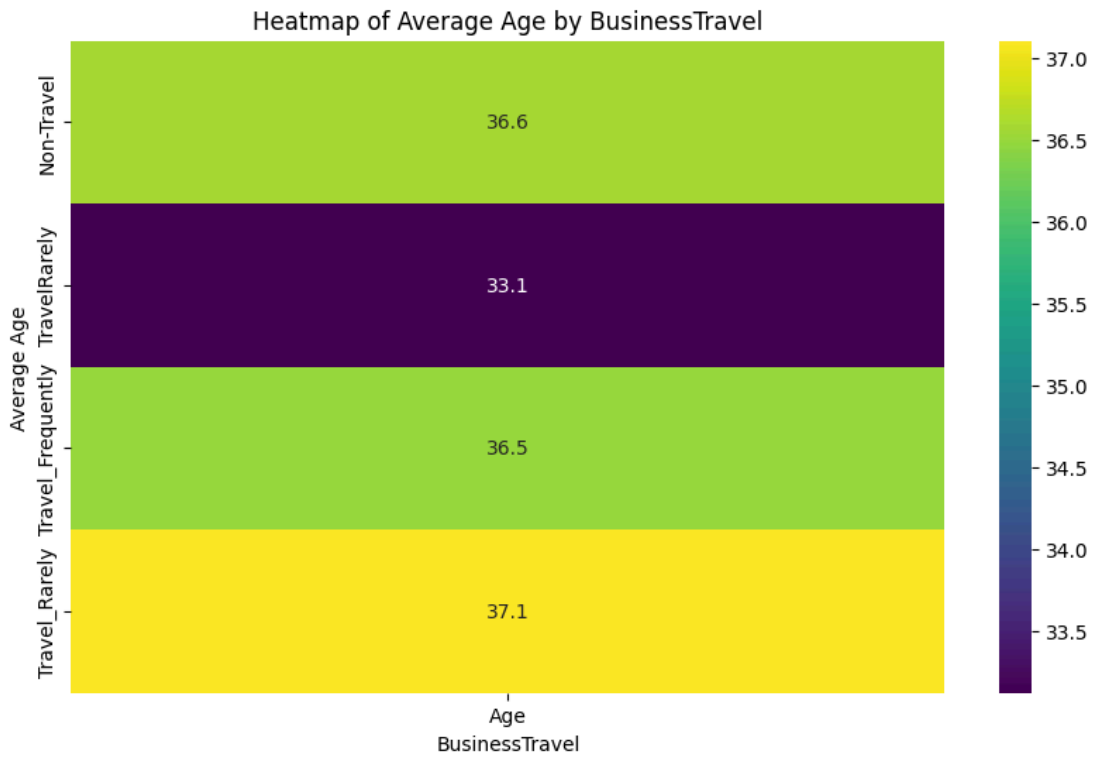
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('/content/HR_Analytics.csv')

plt.figure(figsize=(12, 8))

sns.kdeplot(data=df, x='Age', hue='BusinessTravel', common_norm=False)


plt.title('KDE Plot of Age by BusinessTravel')
plt.xlabel('Age')
plt.ylabel('Density')


plt.show()
```



- Heatmap:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
import pandas as pd


df = pd.read_csv('/content/HR_Analytics.csv')


heatmap_data = df.pivot_table(values='Age', index='BusinessTravel',
aggfunc='mean')


plt.figure(figsize=(10, 6))
sns.heatmap(heatmap_data, annot=True, cmap='viridis', fmt='.1f')


plt.title('Heatmap of Average Age by BusinessTravel')
plt.xlabel('BusinessTravel')
plt.ylabel('Average Age')

plt.show()
```

Outcomes:

CO3: Inculcate the knowledge of python libraries like NumPy, pandas, Matplotlib for scientific- computing and data visualization.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)
Successfully performed statistical data analysis plots analysis using Python (Seaborn)

References:

1. Seaborn Documentation
[Seaborn Documentation](https://seaborn.pydata.org/)

2. Data Camp Seaborn Tutorial
"Seaborn Tutorial," Data Camp, [Online]. Available: https://www.datacamp.com/community/tutorials/seaborn-python-tutorial.

3. J. VanderPlas, Python Data Science Handbook. O'Reilly Media, 2016.
[Online]. Available: https://jakevdp.github.io/PythonDataScienceHandbook.

4. K. Dale, Data Visualization with Python and JavaScript. O'Reilly Media, 2016.
[Online]. Available: https://www.amazon.com/Data-Visualization-Python-JavaScript-Mastering/dp/1491927220.

5. "Journal of Statistical Software,"[Online]. Available: https://www.jstatsoft.org/.

6. "The R Journal,"[Online]. Available: https://journal.r-project.org/.