**Homework For Data Visualization**

**Objective**: The goal of this homework is to deepen your understanding of how data visualizations are designed by creating and analyzing data visualizations in terms of data type, visualization channel, purpose, and insight of the visualization.

**Instructions**:

Please find **two datasets** from either **UCI Machine Learning Repository** or **Kaggle**. For each dataset, follow the steps below to create insightful visualizations and analyze them. Ensure that the visualizations apply the **rule of thumb** for effective data visualization.

**Instructions**:

1. **Data Selection**:
   * Choose two datasets that you find interesting or relevant.
2. **Create Visualizations**:
   * Design visualizations that effectively represent the key information in each dataset. Apply the best practices for data visualization (e.g., clarity, color usage, labeling).
   * You may use any programming tool (Python with Matplotlib, Seaborn, Plotly).
3. **Analyze and Explain**:
   * **Analyze the data type** used in the visualization.
   * **Analyze the channel/visual variables used**: Describe which visual elements (color, shape, size, position) were utilized to present the data.
   * **Analyze the reasons for choosing the type of graph**: Explain why you chose a specific type of graph (e.g., bar chart, scatter plot) and how it suits the data.
   * **State the source of the data**: Clearly mention where you obtained the data, including dataset names and links.
   * **Screenshot the results of the visualization**: Include a clear image of each visualization in your report.
   * **Analyze the results of the insights obtained from your data visualization**: Write a short summary of the key insights you gained from each visualization. Discuss any patterns, trends, or anomalies that were revealed.
   * **Copy your source code**: Provide the full source code you used to generate each visualization, ensuring that it is well-commented in the PDF format.
4. **Additional Requirements**:
   * Each visualization should be accompanied by proper titles, axis labels, and legends where needed. Make sure your visualizations are clear, informative, and easy to interpret.
   * Submit your report in a **PDF** format, including explanations, visualizations, and source code.

**First Dataset: Gym Members Exercise Dataset**

**Data Visualization – 1.A**

|  |
| --- |
| **Bar Chart for Workout Frequency**     * **Graph Type:** Bar chart * **Reason for Choosing This Graph:**   Suitable for categorical data and provides a clear view of the frequency distribution. It allows easy comparison of the number of members based on how many days per week they work out.   * **Data Type:**   + X-axis: Categorical (Workout\_Frequency (days/week))   + Y-axis: Numerical (count of gym members) * **Visual Variables:**   + Position: Used to show the count of members on the Y-axis   + Color: The bars are colored using a palette to make them distinguishable * **Insights from the Visualization:**   The bar chart highlights how frequently members exercise each week. There is a workout frequency of 3 days among members.  This can be used by gym management to tailor their programs or marketing efforts to suit the most common workout habits. |

**Source Code – 1.A**

|  |
| --- |
| import matplotlib.pyplot as plt  import seaborn as sns  # Plot style  sns.set(style="whitegrid")  # Bar chart for Workout Frequency  plt.figure(figsize=(8, 4))  sns.countplot(x='Workout\_Frequency (days/week)', data=gym\_data, palette='viridis')  # Titles and labels  plt.title('Workout Frequency Among Gym Members', fontsize=16)  plt.xlabel('Workout Frequency (days/week)', fontsize=14)  plt.ylabel('Number of Members', fontsize=14)  # Plot Displaying  plt.show() |

**Data Visualization – 1.B**

|  |
| --- |
| * **Graph Type:** Scatter plot * **Reason for Choosing This Graph:**   + Allows us to visually assess the relationship between session duration and calories burned and identify any patterns specific to different workout types. * **Data Type:**   + X-axis: Numerical (Session\_Duration (hours))   + Y-axis: Numerical (Calories\_Burned)   + Color: Categorical (Workout\_Type), helping to distinguish different types of workouts * **Visual Variables:**   + Position: Indicates session duration and calories burned.   + Color: Differentiates various workout types. * **Insights:**   We can see how different workout types correlate with the session duration and calories burned. It reveals which workout types are associated with longer sessions and higher calorie expenditure.  All of the workout types lead to more intense or prolonged sessions. |

**Source Code – 1.B**

|  |
| --- |
| # Scatter plot for Session Duration vs. Calories Burned  plt.figure(figsize=(10, 6))  scatter\_plot = sns.scatterplot(      x='Session\_Duration (hours)',      y='Calories\_Burned',      hue='Workout\_Type',      data=gym\_data,      palette='Set2',      alpha=0.7  )  # Add titles and labels  plt.title('Session Duration vs. Calories Burned by Workout Type', fontsize=16)  plt.xlabel('Session Duration (hours)', fontsize=14)  plt.ylabel('Calories Burned', fontsize=14)  plt.legend(title='Workout Type', bbox\_to\_anchor=(1.05, 1), loc='upper left')  # Display the plot  plt.show() |

**Data Visualization – 1.C**

|  |
| --- |
| **Box Plot - Resting BPM by Experience Level**     * **Graph Type**: Box plot * **Reason for Choosing This Graph:**   + Because is a summary of data distribution, including medians, quartiles, and potential outliers. They are ideal for comparing distributions across different categories. * **Data Type:**   + X-axis: Categorical (Experience\_Level).   + Y-axis: Numerical (Resting\_BPM). * **Visual Variables:**   + Position: Displays the range and spread of Resting\_BPM for each experience level.   + Color: Differentiates the experience levels for clearer comparison. * **Insights:**   We can see that more experienced gym members have lower resting BPMs, which might indicate better cardiovascular health.  There are no outliers. |

**Source Code – 1.C**

|  |
| --- |
| # Box plot for Resting BPM by Experience Level  plt.figure(figsize=(10, 6))  box\_plot = sns.boxplot(      x='Experience\_Level',      y='Resting\_BPM',      data=gym\_data,      palette='coolwarm'  )  # Add titles and labels  plt.title('Resting BPM by Experience Level', fontsize=12)  plt.xlabel('Experience Level (1 = Beginner, 3 = Expert)', fontsize=10)  plt.ylabel('Resting BPM', fontsize=10)  # Display the plot  plt.show() |

**Data Visualization – 1.D**

|  |
| --- |
| **Histogram - Age Distribution**     * **Graph Type**: Histogram * **Reason for Choosing This Graph:**   + Histograms are ideal for showing the distribution of a single numerical variable, making it easy to identify which age ranges are most common**.** * **Data Type:**   + X-axis: Numerical (Age).   + Y-axis: Numerical (Count of members). * **Visual Variables:**   + Position: Shows the count of gym members in each age bin.   + Color: Used a single color for simplicity and clarity. * **Insights:**   This histogram reveals which age groups are most prevalent at the gym. Early 20s people, have higher member counts, while Middle 30s show less represented age groups. The KDE line provides a smooth estimate of the distribution curve. |

**Source Code – 1.D**

|  |
| --- |
| # Histogram for Age Distribution  plt.figure(figsize=(10, 6))  sns.histplot(      gym\_data['Age'],      bins=20,      kde=True,      color='lightpink'  )  # Add titles and labels  plt.title('Age Distribution of Gym Members', fontsize=16)  plt.xlabel('Age', fontsize=12)  plt.ylabel('Number of Members', fontsize=12)  # Display the plot  plt.show() |

**Data Visualization – 1.E**

|  |
| --- |
| **Heatmap - Correlation Matrix**     * **Graph Type:** Heatmap * **Reason for Choosing This Graph:**   + Clear visual representation of the correlation between numerical variables, making it easy to spot strong and weak relationships. * **Data Type:**   + Numerical: Includes variables such as age, weight, height, calories burned, session duration. * **Visual Variables:**   + Color: Indicates the strength of correlations, with positive correlations in warmer colors and negative correlations in cooler colors.   + Position: Displays the relationships between all numerical variables. * **Insights:**   This heatmap shows how different variables relate to each other. There is a strong positive correlation between weight and BMI or between session duration and calories burned.  Identifying significant correlations can help in understanding which variables might have the most impact on workout outcomes or member behaviour. |

**Source Code – 1.E**

|  |
| --- |
| # Correlation heatmap for numerical variables  plt.figure(figsize=(14, 10))  # Select only numerical columns for correlation calculation  numerical\_data = gym\_data.select\_dtypes(include=['number'])  correlation\_matrix = numerical\_data.corr()  # Calculate correlation on numerical data  heatmap = sns.heatmap(      correlation\_matrix,      annot=True,      cmap='coolwarm',      vmin=-1,      vmax=1  )  # Add titles  plt.title('Gym Members Dataset', fontsize=16)  # Display the heatmap  plt.show() |

**Data Visualization – 1.F**

|  |
| --- |
| **Pair Plot - Selected Variables**    **Purpose: To visualize the relationships between multiple variables at once.**   * **Graph Type:** Pair plot * **Reason for Choosing This Graph:** This helps visualize scatter plots between pairs of numerical variables and the distribution of each variable. * **Data Type:** Numerical and categorical (for color coding). * **Visual Variables:**   + Position: Shows relationships between variable pairs.   + Color: Encodes a categorical variable for deeper insights. * **Insight**:   This highlights how age, calories burned, BMI, and session duration vary by experience level. Experience Level 3 clusters around younger to middle-aged users with a strong positive relationship between session duration and calories burned. Level 2 shows a broader age range and more variability in BMI. While BMI is consistent across levels, age and session duration show noticeable differences, with distinct patterns for beginners. This visualization provides insight into how experience level correlates with workout characteristics. |

**Source Code – 1.F**

|  |
| --- |
| # Pair plot for selected numerical variables  selected\_columns = ['Age', 'Calories\_Burned', 'BMI', 'Session\_Duration (hours)', 'Experience\_Level']  pair\_plot = sns.pairplot(      gym\_data[selected\_columns],      hue='Experience\_Level',      palette='husl',      diag\_kind='kde'  )  # Add a title  pair\_plot.fig.suptitle('Selected Variables by Experience Level', y=1.02, fontsize=16)  # Display the pair plot  plt.show() |

**Data Visualization – 1.G**

|  |
| --- |
| **3D Scatter Plot of Session Duration vs. Calories Burned vs. BMI**     * **Graph Type**: 3D scatter plot * **Reason for Choosing This Graph:**   + A 3D scatter plot is suitable for visualizing complex relationships between multiple numerical variables. It allows us to add depth and better understand how different variables interact simultaneously. * **Data Type:**   + X-axis: Numerical (Session\_Duration (hours)).   + Y-axis: Numerical (Calories\_Burned).   + Z-axis: Numerical (BMI).   + Color: Categorical (Experience\_Level).   + Size of Points: Numerical (Max\_BPM) to represent workout intensity. * **Visual Variables:**   + Position: Maps Session Duration, Calories Burned, and BMI in a 3D space.   + Color: Indicates Experience Level.   + Size: Reflects Max\_BPM to highlight workout intensity. * **Insights:**   The 3D scatter plot shows that higher experience levels (yellow points) are associated with longer session durations and more calories burned, while lower experience levels (darker points) cluster around shorter sessions and lower calorie expenditure. BMI appears to vary across all experience levels without a strong pattern, suggesting that session duration and calories burned are more indicative of experience level than BMI. |

**Source Code – 1.G**

|  |
| --- |
| from mpl\_toolkits.mplot3d import Axes3D  # Prepare the 3D scatter plot  fig = plt.figure(figsize=(10, 8))  ax = fig.add\_subplot(111, projection='3d')  # Plotting the data  scatter = ax.scatter(      gym\_data['Session\_Duration (hours)'],      gym\_data['Calories\_Burned'],      gym\_data['BMI'],      c=gym\_data['Experience\_Level'],      s=gym\_data['Max\_BPM'] / 2,      cmap='viridis',      alpha=0.8  )  # Setting the axes labels  ax.set\_xlabel('Session Duration (hours)', fontsize=12)  ax.set\_ylabel('Calories Burned', fontsize=12)  ax.set\_zlabel('BMI', fontsize=12)  # Adding color bar for Experience Level  cbar = fig.colorbar(scatter, ax=ax, pad=0.1)  cbar.set\_label('Experience Level')  # Add a title  plt.title('Session Duration vs. Calories Burned vs. BMI', fontsize=16)  # Display the plot  plt.show() |

**Second Dataset: Mobile Device Usage and User Behaviour Dataset**

**Data Visualization – 2.A**

|  |
| --- |
| **Distribution Of User Behavior Classes**     * **Graph Type:** Bar chart * **Reason for Choosing This Graph:**   + A bar chart is effective for showing the distribution of categorical data, such as user behavior classes, and allows for easy comparison between categories. * **Data Type:**   + X-axis: Categorical (User Behavior Class).   + Y-axis: Numerical (count of users). * **Visual Variables:**   + Position: Represents the count of users in each behavior class.   + Color: Used for visual appeal and distinction between classes. * **Insights:**   The bar chart shows that the distribution of users across behavior classes is relatively balanced, with a slight concentration in classes 2 and 3. This indicates that most users fall into moderate behavior categories, with fewer users in the lowest and highest behavior classes. |

**Source Code – 2.A**

|  |
| --- |
| # Bar chart for the distribution of User Behavior Classes  plt.figure(figsize=(10, 6))  sns.countplot(x='User Behavior Class', data=user\_behavior\_data, palette='plasma')  # Add titles and labels  plt.title('Distribution of User Behavior Classes', fontsize=16)  plt.xlabel('User Behavior Class (1 to 5)', fontsize=14)  plt.ylabel('Number of Users', fontsize=14)  # Display the plot  plt.show() |

**Data Visualization – 2.B**

|  |
| --- |
| **Box Plot for App Usage Time by User Behavior Class**     * **Graph Type:** Box plot * **Reason for Choosing This Graph:**   + This is ideal for comparing distributions and identifying medians, quartiles, and potential outliers for different user behavior classes. * **Data Type:**   + X-axis: Categorical (User Behavior Class).   + Y-axis: Numerical (App Usage Time in minutes per day). * **Visual Variables:**   + Position: Shows the distribution of app usage time for each behavior class.   + Color: Distinguishes between the classes for better readability. * **Insights:**   The box plot illustrates that app usage time increases progressively across user behavior classes, with class 5 showing the highest median and range. This suggests that higher user behavior classes are associated with significantly greater daily app usage, highlighting distinct usage patterns as behavior class increases. |

**Source Code – 2.B**

|  |
| --- |
| # Box plot for App Usage Time by User Behavior Class  plt.figure(figsize=(12, 8))  sns.boxplot(      x='User Behavior Class',      y='App Usage Time (min/day)',      data=user\_behavior\_data,      palette='rainbow'  )  # Add titles and labels  plt.title('App Usage Time by User Behavior Class', fontsize=16)  plt.xlabel('User Behavior Class (1 to 5)', fontsize=14)  plt.ylabel('App Usage Time (min/day)', fontsize=14)  # Display the plot  plt.show() |

**Data Visualization – 2.C**

|  |
| --- |
| **Scatter Plot for Screen On Time vs. Battery Drain**     * **Graph Type:** Scatter plot * **Reason for Choosing This Graph:**   + Because it’s ideal for visualizing the relationship between two numerical variables, with the addition of color coding for categorization. * **Data Type:**   + X-axis: Numerical (Screen On Time in hours per day).   + Y-axis: Numerical (Battery Drain in mAh per day).   + Color: Categorical (Operating System) for added analysis. * **Visual Variables:**   + Position: Represents the relationship between screen-on time and battery drain.   + Color: Differentiates between iOS and Android to observe any trends specific to operating systems. * **Insights:**   The scatter plot shows a clear positive relationship between screen-on time and battery drain, with both Android and iOS devices following a similar pattern. Devices with longer screen-on times tend to exhibit higher battery consumption, but there is no significant distinction between the operating systems in terms of battery drain behavior. |

**Source Code – 2.C**

|  |
| --- |
| # Scatter plot for Screen On Time vs. Battery Drain  plt.figure(figsize=(12, 8))  scatter\_plot = sns.scatterplot(      x='Screen On Time (hours/day)',      y='Battery Drain (mAh/day)',      hue='Operating System',      data=user\_behavior\_data,      palette='Set2',      alpha=0.7  )  # Add titles and labels  plt.title('Screen On Time vs. Battery Drain by Operating System', fontsize=16)  plt.xlabel('Screen On Time (hours/day)', fontsize=14)  plt.ylabel('Battery Drain (mAh/day)', fontsize=14)  plt.legend(title='Operating System', bbox\_to\_anchor=(1.05, 1), loc='upper left')  # Display the plot  plt.show() |

**Data Visualization – 2.D**

|  |
| --- |
| **Heatmap for Correlation Matrix**     * **Graph Type:** Heatmap * **Reason for Choosing This Graph:**   + Provides a visual summary of correlations between numerical variables. It’s an effective way to quickly identify strong and weak relationships in the dataset. * **Data Type:**   + Numerical Variables: Includes variables like app usage time, screen on time, battery drain, data usage, etc. * **Visual Variables:**   + Color: Indicates the strength and direction of correlations, with positive correlations shown in warmer colors and negative in cooler colors. * **Insights:**   The heatmap shows strong positive correlations between user behavior class and variables such as app usage time, screen-on time, battery drain, and the number of apps installed, indicating that higher behavior classes are associated with increased device usage and consumption. Age has a negligible correlation with these factors, suggesting that behavior class and usage patterns are not significantly age-dependent. |

**Source Code – 2.D**

|  |
| --- |
| # Correlation heatmap for numerical variables in the user behavior dataset  plt.figure(figsize=(10, 6))  # Select only numerical features for correlation analysis  numerical\_data = user\_behavior\_data.select\_dtypes(include=['number'])  correlation\_matrix = numerical\_data.corr()  heatmap = sns.heatmap(      correlation\_matrix,      annot=True,      cmap='coolwarm',      vmin=-1,      vmax=1  )  # Add titles  plt.title('User Behavior Dataset', fontsize=16)  # Display the heatmap  plt.show() |

**Data Visualization – 2.E**

|  |
| --- |
| **3D Scatter Plot for Multi-Dimensional Analysis**     * **Graph Type:** 3D scatter plot * **Reason for Choosing This Graph:**   + This visualization allows us to observe relationships between three numerical variables simultaneously and adds an extra layer of analysis with color coding for user behavior class and point size for the number of apps installed. * **Data Type:**   + X-axis: Numerical (Age).   + Y-axis: Numerical (Data Usage in MB per day).   + Z-axis: Numerical (Battery Drain in mAh per day).   + Color: Categorical (User Behavior Class).   + Size of Points: Numerical (Number of Apps Installed) to represent the number of apps. * **Visual Variables:**   + Position: Maps age, data usage, and battery drain in 3D space.   + Color: Indicates different user behavior classes.   + Size: Reflects the number of apps installed, showing user engagement. * **Insights:**   The 3D scatter plot illustrates that as user behavior class increases (from 1 to 5), there is a notable rise in both data usage and battery drain, with class 5 showing the highest values. Additionally, users in higher behavior classes tend to be younger, highlighting that more intensive device use is associated with higher data consumption and power demands, particularly among younger users. |

**Source Code – 2.E**

|  |
| --- |
| # 3D Scatter plot for Age, Data Usage, and Battery Drain with User Behavior Class  fig = plt.figure(figsize=(14, 10))  ax = fig.add\_subplot(111, projection='3d')  # Plotting the data  scatter = ax.scatter(      user\_behavior\_data['Age'],      user\_behavior\_data['Data Usage (MB/day)'],      user\_behavior\_data['Battery Drain (mAh/day)'],      c=user\_behavior\_data['User Behavior Class'],      s=user\_behavior\_data['Number of Apps Installed'] / 2,      cmap='viridis',      alpha=0.7  )  # Setting the axes labels  ax.set\_xlabel('Age', fontsize=12)  ax.set\_ylabel('Data Usage (MB/day)', fontsize=12)  ax.set\_zlabel('Battery Drain (mAh/day)', fontsize=12)  # Adding color bar for User Behavior Class  cbar = fig.colorbar(scatter, ax=ax, pad=0.1)  cbar.set\_label('User Behavior Class')  # Add a title  plt.title('Age, Data Usage, and Battery Drain by User Behavior Class', fontsize=16)  # Display the plot  plt.show() |

**Data Visualization – 2.F**

|  |
| --- |
| **Facet Grid for App Usage Time by Gender and User Behavior Class**     * **Graph Type:** Facet Grid (histogram per behavior class and gender) * **Reason for Choosing This Graph:**   + Facet grids are powerful for visualizing data across multiple subsets (in this case, gender and user behavior class). This helps identify trends and patterns within subgroups. * **Data Type:**   + X-axis: Numerical (App Usage Time in minutes per day).   + Subplots: Categorical (User Behavior Class) with color coding for gender. * **Visual Variables:**   + Position: Represents app usage time for each gender within each behavior class.   + Color: Differentiates gender within the behavior classes. * **Insights:**   The facet grid shows that app usage time distributions vary by user behavior class, with noticeable gender differences, especially in classes 4 and 5 where usage patterns peak differently for males and females. |

**Source Code – 2.F**

|  |
| --- |
| # Facet Grid for App Usage Time by Gender and User Behavior Class  g = sns.FacetGrid(user\_behavior\_data, col='User Behavior Class', hue='Gender', palette='Set1', col\_wrap=2, height=4, aspect=1.5)  g.map(sns.histplot, 'App Usage Time (min/day)', kde=True, alpha=0.6)  # Add titles and adjust layout  g.add\_legend(title='Gender')  g.set\_axis\_labels('App Usage Time (min/day)', 'Count')  g.set\_titles('User Behavior Class {col\_name}')  plt.subplots\_adjust(top=0.9)  g.fig.suptitle('App Usage Time by Gender and User Behavior Class', fontsize=16)  # Display the facet grid  plt.show() |

**Data Visualization – 2.G**

|  |
| --- |
| **Violin Plot for Data Usage by User Behavior Class**     * **Graph Type:** Violin plot * **Reason for Choosing This Graph:**   + Violin plots display the distribution and density of data, similar to box plots but with additional detail on the data distribution shape. This helps to understand how data usage varies within and across user behavior classes**.** * **Data Type:**   + X-axis: Categorical (User Behavior Class).   + Y-axis: Numerical (Data Usage in MB per day). * **Visual Variables:**   + Position: Represents data usage for each user behavior class.   + Shape and Width: Indicate the density and distribution of data usage, with wider sections showing more frequent data points. * **Insights:**   The violin plot shows that data usage increases significantly from user behavior class 1 to 5, with class 5 exhibiting the widest distribution and highest median, indicating heavier data usage among extreme users. |

**Source Code – 2.G**

|  |
| --- |
| # Violin plot for Data Usage by User Behavior Class  plt.figure(figsize=(10, 6))  sns.violinplot(      x='User Behavior Class',      y='Data Usage (MB/day)',      data=user\_behavior\_data,      palette='muted',      inner='quartile'  )  # Add titles and labels  plt.title('Data Usage by User Behavior Class', fontsize=16)  plt.xlabel('User Behavior Class (1 to 5)', fontsize=14)  plt.ylabel('Data Usage (MB/day)', fontsize=14)  # Display the plot  plt.show() |