Lecture Notes (Lecture 10 - 2 Hours)

Duration: 2 hours

Factor 1	Factor 2	Measure	Data Visualization	Implementation
Binary	Binary	Binary	Heatmap	Use `seaborn.heatmap()` to visualize a correlation matrix between binary variables. First, create a cross-tabulation (contingency table) and visualize it. `sns.heatmap(pd.crosstab(df['factor1'], df['factor2']), annot=True, cmap='Blues')`
Binary	Binary	Scale	Boxplot or Violin Plot	Use `seaborn.boxplot()` or `sns.violinplot()` to visualize the distribution of a continuous measure across two binary variables. `sns.boxplot(x='factor1', y='measure', hue='factor2', data=df)`
Binary	Nominal	Ordinal	Stacked Bar Chart	Use `matplotlib`'s `bar()` function to create stacked bar charts that show the distribution of ordinal measures for binary variables split by nominal categories. `plt.bar(df['factor1'], df['measure'])`
Ordinal	Ordinal	Scale	Line Plot or Heatmap	Use `plotly.express.line()` or `sns.heatmap()` to visualize trends of a continuous variable between two ordinal variables. `px.line(df, x='factor1', y='measure', color='factor2')`

Nominal	Nominal	Binary	Stacked Bar Chart or Mosaic Plot	Use `seaborn.countplot()` or `plotly.graph_objs.Bar()` to create stacked bars showing counts or proportions of binary outcomes split by two nominal factors. `sns.countplot(x='factor1', hue='factor2', data=df)`
Nominal	Scale	Scale	Scatter Plot with Trendline	Use `seaborn.scatterplot()` or `plotly.express.scatter()` to create a scatter plot with a trendline showing the relationship between two continuous variables categorized by a nominal variable. `sns.scatterplot(x='factor1', y='measure', data=df)`
Ordinal	Nominal	Ordinal	Clustered Bar Plot	Use `seaborn.barplot()` to create clustered bar plots for ordinal measures across nominal categories. `sns.barplot(x='factor1', y='measure', hue='factor2', data=df)`
Scale	Scale	Scale	Pair Plot or 3D Scatter Plot	Use `seaborn.pairplot()` to visualize pairwise relationships between multiple continuous variables. For 3D scatter plots, use `plotly.express.scatter_3d()`. `sns.pairplot(df[['factor1', 'factor2', 'measure']])`
Nominal	Ordinal	Scale	Violin Plot or Swarm Plot	Use `seaborn.violinplot()` or `sns.swarmplot()` to visualize the distribution of a continuous variable grouped by nominal and ordinal factors. `sns.violinplot(x='factor1', y='measure', hue='factor2', data=df)`
Binary	Ordinal	Scale	Boxplot or Bar Plot	Use `seaborn.boxplot()` or `plotly.express.bar()` to show the distribution of a scale variable based on binary and ordinal factors.

```
sns.boxplot(x='factor1',
y='measure', hue='factor2',
data=df)
```

Implementation Summary:

Heatmap: Use Seaborn's `heatmap()` for visualizing relationships or correlations in a matrix-like format.
python
Copy code
```
sns.heatmap(pd.crosstab(df['factor1'], df['factor2']), annot=True,
cmap='Blues')
```

1.

Boxplot/Violin Plot: Use Seaborn's `boxplot()` or `violinplot()` to show the distribution of data for different categories.
python
Copy code
```
sns.boxplot(x='factor1', y='measure', hue='factor2', data=df)
```

2.

Stacked Bar Chart: Use Matplotlib or Plotly for a stacked bar chart. Matplotlib is better for simple cases, while Plotly is interactive.
python
Copy code
```
plt.bar(df['factor1'], df['measure'])
```

3.

Line Plot: Use Plotly Express's `line()` or Seaborn's `lineplot()` to show trends between ordinal variables.
python
Copy code
```
px.line(df, x='factor1', y='measure', color='factor2')
```

4.

Scatter Plot: Use Seaborn or Plotly to visualize relationships between continuous variables.
python
Copy code
```
sns.scatterplot(x='factor1', y='measure', hue='factor2', data=df)
```

5.
6. **Pair Plot**: Seaborn's `pairplot()` is useful for visualizing relationships between several variables at once.
   python
   Copy code
   ```
   sns.pairplot(df[['factor1', 'factor2', 'measure']])
   ```