



Here's a comprehensive set of revision notes for the class on data visualization techniques, specifically focusing on bivariate analysis using various types of plots. The notes are structured to facilitate a thorough review and understanding of the topic.

Data Visualization Revision Notes

Introduction

This class covered advanced data visualization techniques with a focus on bivariate data. We explored how to visualize relationships between two variables, whether they are numerical or categorical, through a variety of plots.

Key Concepts

Bivariate Data Visualization

Bivariate data involves two variables; we can categorize these kinds of relationships into:

1. **Continuous-Continuous:** Both variables are numerical.
2. **Categorical-Categorical:** Both variables are categorical.
3. **Categorical-Continuous:** One variable is categorical, the other is numerical [\[4:2+source\]](#) .

Types of Plots

Continuous-Continuous Relationships

1. Line Plot:

- Used to show trends over time, such as sales variation over years.
- It's important to choose line plots when you have a clear order in your data like time progression [\[4:3+source\]](#) [\[4:10+source\]](#) .

Example:



2. Scatter Plot:

- More suitable than line plots when data points are not inherently ordered or the data is "messy".
- Ideal for visualizing correlations between two numerical variables.

Example:

```
sns.scatterplot(x='Variable1', y='Variable2', data=my_data)
```

Observations:

- Scatter plots can reveal correlations; e.g., a negative correlation means as one variable increases, the other decreases **【4:10+source】** **【4:13+source】** .

Categorical-Categorical Relationships

1. Dodged Countplot:

- Displays multiple categorical distributions side by side, ideal for comparing counts across categories.
- For example, comparing the frequency of different genres across multiple platforms **【4:14+source】** .

Example:

```
sns.countplot(x='Category1', hue='Category2', data=my_data)
```

2. Stacked Countplot:

- Uses stacked bars to show the total composition within categories, helpful for showing proportions.
- Misinterpretations can occur with stacking, as it might not be clear where sections start **【4:14+source】** .

Categorical-Continuous Relationships

1. Boxplot:

- Useful for visualizing the distribution of a numerical variable across categories.
- Provides insights into medians, quartiles, and potential outliers **【4:12+source】** **【4:15+source】** .



```
sns.boxplot(x='Category', y='Sales', data=my_data)
```

2. Bar Plot:

- Useful for comparing average values across different categories.
- For example, average sales per genre [\[4:15+source\]](#) .

Example:

```
sns.barplot(x='Genre', y='Average_Sales', estimator=np.mean, da
```

Subplots

- **Creating Subplots:** Necessary when plotting multiple charts in a coherent manner on a single canvas. It allows different visualizations to be displayed together for comparative insights [\[4:4+source\]](#) .

Example:

```
fig, axes = plt.subplots(nrows=2, ncols=2)
sns.lineplot(ax=axes[0,0], x='Year', y='NA_Sales', data=my_data)
sns.scatterplot(ax=axes[0,1], x='Rank', y='Global_Sales', data=
```

- **Customizing Subplots:** Adjust sizes and layout for better readability using `figsize` and subplot matrix indexing [\[4:17+source\]](#) .

Concluding Observations

- **Data Cleanliness:** Ensure the dataset is clean for more accurate visualizations; messy data can lead to misinterpretations.
- **Choosing the Right Plot:** Selection of a plot type is crucial—it should align with the data characteristics and the message you intend to convey.
- **Interactivity and Exploration:** Visualization is not just for presentation but also for exploring data insights and discovering patterns [\[4:14+source\]](#) [\[4:16+source\]](#) .

These notes capture the essence of the class on bivariate data visualization, providing learners with a framework to understand

