Visualizing Categorical Data

Visualizing Categorical Data

Bar Charts and *Pie Charts* are used to visualize categorical data. Both types of graphs contain variations as displayed in the visual.

Bar Chart Area

The bars of a bar chart have a couple of key features:

- They have lengths that are proportional to the counts they represent
- Each bar's width in the bar chart is the same, meaning each bar's area is also proportional to the counts they represent.

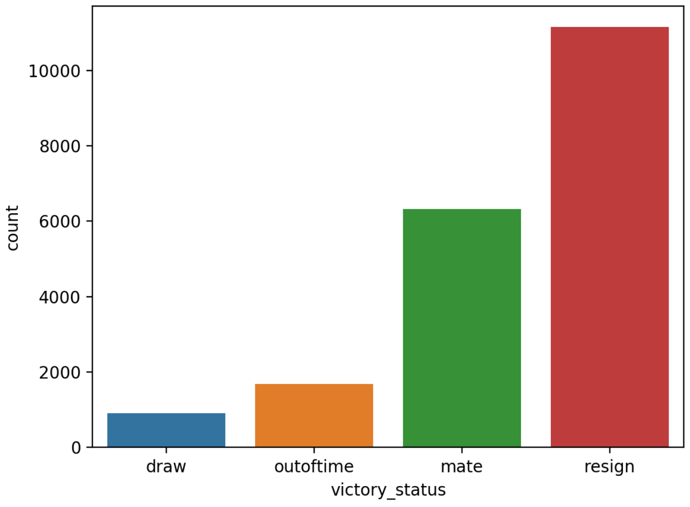
These features make a bar chart super dependable for representing categorical data.
For any chart like a bar chart, the areas we use to represent values must always be equivalent to the relative sizes of the values they represent. Otherwise, readers could be misled and potentially identify patterns that do not actually exist.

Bar Chart Nominal Data Ordering

Nominal data has labels with no specific order. Thus, we have a lot of creative freedom when choosing where each bar goes on our chart. One way would order our data is by ascending or descending order.

You can do this in Python with the `order` parameter in the `.counplot()` seaborn method. Within the `order` parameter, the `.value_counts()` pandas method can order the values in either ascending or descending format. The code snippet below gives an example of how this can be done.

```
sns.countplot(df["victory_status"], c
```

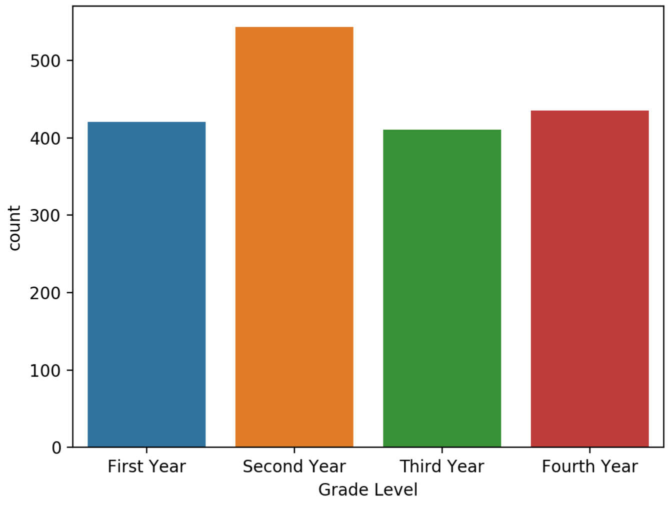


Ordering the bars of nominal data is useful because it makes keys findings, such as the mode of the data, easy to spot. For example, in the graph pictured, we can see that the `resign` value is the mode of the `victory_status` variable, while `draw` is the least common value.

Bar Chart Ordinal Data Ordering

If we are working with ordinal data, we should plot the data according to our categorical variables. For example, let's say we want to plot the number of students per grade level at a college. We have a table below, which is a preview data from a **students.csv** file.

Grade Level
Second Year
Second Year
First Year
Third year
Fourth Year



We can order the categorical values as `First Year`, `Second Year`, `Third Year`, and `Fourth Year` since they are ordinal. Using `.counplot()`, we can input these as a list in the `order` parameter.

```
sns.countplot(df["Grade Level"], orde
```

From the chart shown, we get takeaways that we couldn't just from looking at the column of data. We can see that the college had an influx of students for the second year class, as `Second Year` is the mode (most observed value) of the `Grade Level` column of data. All other years have roughly the same number of students. Putting bars in order according to the ordinal data helps us see these patterns and make our visuals informative for readers.

Pie Chart Rules

It is crucial that any pie chart you see or create follows these two rules:

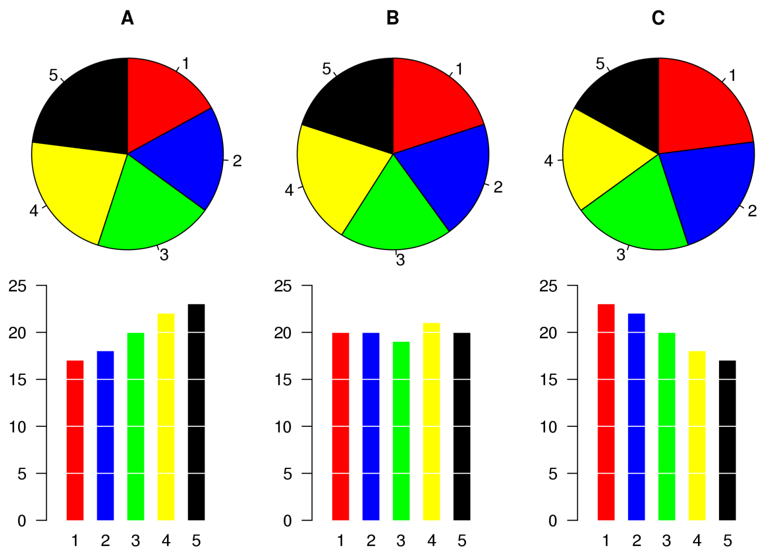
- Both the arc length and area of each sector are proportional to the percentage of the circle it makes up.
- The sum of each sector adds up to 100 percent.

Pie Chart Pitfalls

Pie Charts have two common pitfalls:

- It can be difficult for viewers to compare sector sizes within the chart.
- If a pie chart contains too many sectors, it is difficult for a viewer to decipher any useful information.

If you ever run into this issue, a bar chart may be the best solution. The picture comparing pie charts and bar charts shows why.
With each pie chart, it is almost impossible to compare separate sectors. However, the bar chart makes the comparisons much easier to decipher.



Print Share