Summary Sheet

AUTHOR
Julia Gallucci

Class 6

Types of graphs

Bar Graph, Histogram, and Scatter plot are all commonly used types of graphs to visually represent data. While they share some similarities, they have distinct characteristics and purposes. Here's a brief overview of each:

Bar graph	Histogram	Scatter plot
Uses rectangular bars to represent different categories or groups of data. The height of each bar corresponds to the values or frequency of the category it represents.	Graphical representation of a continuous data set divided into intervals or bins. The X axis represents the range of values, while the Y axis represents the frequency or count of data points within each interval.	Represents data as a collection of points, where each point represents an individual data point with its respective values on the x-axis and y-axis.
Useful for displaying categorical data and comparing values between different groups.	Useful for visualizing the distribution of numerical data and identifying patterns, such as the shape (e.g., normal distribution, skewed distribution) and central tendency.	Useful for identifying relationships or correlations between two variables.

ggplot()

two arguments

1. data: dataset you wish you plot
2. aesthetics: (abbreviated as "aes") refers to how variables are mapped to visual properties of a plot. Aesthetics define how the data attributes are visually represented, such as position, size, shape, color, etc.

```
aes(x = variable1,

y = variable2,

shape = variable3,

color = variable4,

size = variable5,
fill = variable6)
```

- x and y: map variables to the x-axis and y-axis, respectively, determining the position of data points on the plot.
- shape: defines the marker shape of data points, typically used to differentiate groups or levels of a categorical variable.
- color: assigns different colors to data points based on a categorical or continuous variable, allowing for visual differentiation of groups or levels.
- size: controls the size of data points, which can be mapped to a continuous variable to represent an additional dimension of information.
- fill: used to set the color or pattern of the area inside plotted objects, such as bars.

Adding layers

Bar plots `geom_bar()`

- Defaults:

  -
    ```
    geom_bar(

    stat = "count",

    position = "stack",

    width = NULL

    )
    ```
- default **stat = “count”** counts the number of occurrences of each category and represents it as the height of the bars. However, you can change the statistic by specifying a different value for the **`stat`** parameter, such as **`"identity"`** (for providing explicit heights) or **`"bin"`** (for binning continuous variables).

- default **position= “stack”**, stacks the bars vertically on top of each other. Other common options include **`"fill"`** (stacked bars with proportional heights), **`"dodge"`** (side-by-side bars for grouped categories), and **`"identity"`** (no stacking or grouping).

- default **width** is determined automatically based on the data and plot dimensions. However, you can manually adjust the width by specifying a value for the **`width`** parameter.

Histograms `geom_histogram()`

- Defaults:

  -
    ```
    geom_histogram(

    stat = "bin",

    position = "stack",
    ```

```
binwidth = NULL,

bins = NULL

)
```

- Default **stat =“bin”**. This statistical transformation bins the continuous data into intervals or bins and counts the number of data points falling into each bin. The resulting counts are then used to create the histogram.

- Default **position=“identity”**, which creates overlapping bars. You can change the positioning behavior by setting **`position`** to **`"stack"`** (stacked bars) or **`"dodge"`** (side-by-side bars).

- Default **binwidth (**width of the bins) is based on the data range and the number of bins. However, you can manually set the bin width by specifying a value. Alternatively, you can specify the number of bins using the **bins** parameter.

Scatterplots `geom_point()`

- Defaults:

  -
    ```
    geom_point(

    position = "identity",

    size = NULL,

    alpha = 1

    )
    ```

- Default **position = identity**. This means that the points are plotted as they are without any adjustment for overlapping. If multiple points have the same coordinates, they will overlap visually. can “jitter” points which adds a small random displacement to the points to visually separate them.

- Default **size** of the markers is determined automatically based on the data and plot dimensions. However, you can manually adjust the size by mapping a variable to the **size** aesthetic or specifying a fixed size value.

- Default **alpha = 1**, indicating complete opacity. Refers to the transparency or opacity of the markers. You can specify a value between 0 and 1 to adjust the transparency level.

Customizing

1. add labels `labs(x= <x axis label>,y= <y axis label>, title= <title of plot>)`

2. manipulate axis i.e., `scale_x_continuous` , `scale_x_discrete`

3. `fill/color` used to set the color inside plotted objects, such as bars based on looks (outside aesthetic) or variable (inside aesthetic)

4. `size` used to change the size of a datapoint based on looks (outside aesthetic) or variable (inside aesthetic)

5. `facets` used to give side-by-side graphs for different categories

6. `themes` used for different overall looks

   Note, you can layer multiple geoms!