Creating Multiple Line Graphs: Takeaways

Syntax

- Splitting data into subplots based on values of a variable:

```
ggplot(data = data_frame,
  aes(x = variable_1, y = variable_2)) +
  geom_line() +
  facet_wrap(~variable_3)
```

- Creating a line graph with different values of a variable using different styles of lines:

```
ggplot(data = data_frame,
  aes(x = variable_1, y = variable_2, lty = variable_3)) +
  geom_line()
```

- Creating a line graph with different values of a variable using different colors:

```
ggplot(data = data_frame,
  aes(x = variable_1, y = variable_2, color = variable_3)) +
  geom_line()
```

- Viewing a specific part of a graph:

```
ggplot(data = data_frame,
  aes(x = variable_1, y = variable_2, color = variable_3)) +
  geom_line() +
  xlim(1915, 1920) +
  ylim(35, 60)
```

- Changing the color and line types of a line:

```
ggplot(data = life_expec_sex_race,
  aes(x = Year, y = Avg_Life_Expec, color = Sex, lty = Race)) +
  geom_line() +
  scale_color_manual(values = c("magenta", "orange")) +
  scale_linetype_manual(values = c("longdash", "dotdash"))
```

Concepts

- The `facet_wrap()` function splits data into subplots based on values of a variable in your data set. By default, this creates two panels side by side. You can specify arrangements of the panels using the `ncol` and `nrow` arguments.
- Chart legends contain a list of variables that appear in a graph and an example of their appearance. Legends help the person looking at your graph understand it.
- Changing the scale limits changes the range of your axes so you can display only a portion of your data.
- The way you choose to graph your data ultimately depends on the question you're trying to answer and requires some trial and error.
- The consistent, systematic syntax of `ggplot2` will allow you to create multiple graphs as part of your workflow without taking too much of your time.

Resources

- Color names
- Line type names