Practicing Data Wrangling with Tree Equity Data

Instructions

In this worksheet, you'll practice writing **tidyverse-style R code** for data wrangling tasks. Each question guides you through filtering, selecting, mutating, and summarizing the `ca_tes` dataset.

Don't stress yourself out over perfect syntax – the goal is to practice what you've learned about data wrangling so far

Practice Questions

1. Filter for neighborhoods with low tree equity scores

Show only neighborhoods where the `tes` (Tree Equity Score) is **less than 50**.

Hint: Use `filter()`

2. Focus on specific columns of interest

From the filtered data, keep only the following columns: `treecanopy`, `tes`, and `unemplnorm`.

Hint: Use `select()`

3. Calculate percent of tree coverage goal met

Compute a new column that represents **the percent of the tree coverage goal that has been met**.

Hint: Use `mutate()`
- You'll need to make an assumption about how to calculate this — discuss with a partner if unsure!

4. Compute average tree equity score by place

Group the data by `place` (city or neighborhood), and compute the **average `tes` score** in each one.

Hint: Use `group_by()` and `summarise()`

5. How does tree equity relate to both poverty and health disparity?

In this question, you'll explore the relationship between **Tree Equity Score (`tes`)**, **poverty**, and **health disparity**.

- Create a new variable called `combined_index` that combines `pctpovnorm` and `health_nor`.
- Cut the dataset down to just the relevant variables: `place`, `combined_index`, and `tes`.
- Organize the data so you can easily compare which places have the **highest combined index**.
    - **Hint:** Use `arrange()` to sort by the index in descending order

3. Tree Equity vs. Linguistic Isolation

Explore whether there's a relationship between **linguistic isolation** and **tree equity** in different areas.

- Use `ggplot()` to visualize the relationship between `lingnorm` (linguistic isolation) and `te`
    - **Hint:** Choose the best type of plot for these two numerical variables.
- Add meaningful labels to your plot:
    - Use `labs()` to set an informative **title** and **axis labels**.