Welcome to

The Session Will Begin Shortly

START

Statistics in R with Tidyverse

Session 3: Data Wrangling and Tidy Data

instats

Data Wrangling

- Overview of the `tidyverse`
- Importance of Data Wrangling in Research
- Key Packages: `tidyr`, `dplyr`

Filter Rows

- Use `filter()`to select rows based on conditions
- Focuses on rows
    - Similar to `slice()` which selects rows by position, not condition
- Combine conditions with `&` (AND) and `|` (OR)
- **Tip**: Use `!=` to filter out specific values

Mutate Columns

- Use `mutate()`to create new columns based on existing ones
- Adds new columns; unlike `transmute()`, which drops all other columns
- Useful for transforming or calculating new values from existing data
- **Tip**: Can also be used to modify an existing column

Summarize Data

- Use `summarize()` to calculate summary statistics
- Reduces data to a single row or value; unlike `mutate()` which keeps original data format
- **Tip**: Can handle missing data with `na.rm = TRUE`

Group By and Summarize

- Use `group_by()` to split data into groups, then apply `summarize()`
- Organizes data into groups; unlike `arrange()`, which only sorts data
- Combine `group_by()` with `summarize()` to create grouped statistics
- **Tip**: `ungroup()` data after grouping if further processing is needed

Arrange Data

- Use `arrange()` to sort rows based on specific columns
- Sorts data; unlike `filter()` which selects rows without changing order
- **Tip**: Sort in ascending order by default; use `desc()` for descending

Select Columns

- Use `select()` to choose specific columns
- Different from `mutate()`, which adds new columns
- Can deselect columns using – (e.g., `select(-year)`)
- **Tip**: Use helpers like `starts_with()` to select columns by pattern

Tidy Data

- "Tidy" data means
    - each variable has its own column
    - each observation has its own row
    - each kind of thing you're observing is its own table
- Different from "wide" data in that it is often longer to be tidy
- **Tip**: Use `pivot_longer()` to convert wide data for easier analysis

Pipe Operator (|>)

- Use the pipe operator to chain multiple operations together
- Chains operations unlike using nested functions, which is harder to read
- Often improves workflows
- **Tip**: Think of |> as “then” to improve readability

Demo & Exercises

Q & A

STOP