STAT 333: When Does Correlation Imply Causation? Downstream Variables & Intro to dplyr (Feb. 4 Lecture)

Lecture Notes

February 18, 2025

Overview

This lecture continues our discussion of:

- **Confounding** and **randomization** (RCTs).
- **Multiple linear regression** (MLR) for causal inference when all confounders are measured.
- **Downstream (intermediate) variables**: what they are, and why including them can break causal interpretation.
- A short introduction to **dplyr** (part of the Tidyverse in R) for data manipulation, including:
  - Verbs like `filter`, `select`, `mutate`, `arrange`, and `group_by/summarize`.
  - The **pipe** operator (`%>%`).
  - Handling large raw datasets (e.g. 300k+ rows) by subsetting, transforming, and combining columns.

---

1 When Does Correlation Imply Causation?

1.1 Recap of RCT and Confounding

- **Correlation** does not automatically imply **causation**; often, the correlation between X (treatment/exposure) and Y (outcome) could be due to a *confounder* C that influences both.
- In an **RCT** (Randomized Controlled Trial), X is determined by the researcher's randomization; thus, *no* confounder C can cause X. Therefore, correlation of X and Y in an RCT often does reflect a causal relationship.

1.2 Measuring Confounders in Observational Data

- Often, we cannot randomize (ethical/practical reasons). Instead, we collect observational data.

- **Multiple linear regression (MLR)**: If we *measure all* confounders $C_1, C_2, \ldots, C_{p-1}$, we can put them in a regression model along with X (the treatment of interest). Under the linear model assumption and having truly measured (and included) *all* confounders, the coefficient of X can be interpreted causally.

> *"If you measure all confounders & put them in MLR with the treatment, you can estimate the causal effect (under suitable assumptions)."*

---

2 Intermediate (Downstream) Variables

2.1 Definition and Examples

A **downstream** or **intermediate** variable is caused *by the treatment* X and itself *causes* the outcome Y. For instance:

- Example: **Smoking** (X) $\rightarrow$ **Tar in lungs** (intermediate) $\rightarrow$ **Lung cancer** (Y).
- If you include the intermediate variable (tar in lungs) in the regression model *along with* X, you typically **lose the ability to see the total causal effect** of X on Y. The model might show that only the tar variable affects Y, and thus you incorrectly conclude that X (smoking) has no direct effect.

2.2 Danger of Accidentally Including Downstream Variables

- In practice, researchers sometimes throw *all* available variables into a regression to "adjust for everything." But **if any variable is downstream of** X, including it can *invalidate* the regression's interpretation as a causal effect of X.
- There is no automated "confounder vs. intermediate variable" test in the dataset. *Domain knowledge* about the causal structure is essential to decide which variables are truly confounders (should be included) and which are downstream (should *not* be included if one wants the total effect of X).

---

3 Brief Introduction to `dplyr`

We switched gears to demonstrate how to handle a large dataset (e.g. the `nycflights13::flights` dataset with 336,776 rows and 19 columns) in R. The **dplyr** package (part of the Tidyverse) provides a small set of functions ("verbs") for data wrangling:

3.1 Core Verbs in `dplyr`

- `filter()`: Subset rows by logical conditions.
- `select()`: Subset columns by name (or pattern).
- `mutate()`: Create or transform existing columns (e.g. `distance / air_time` to get speed).

- `arrange()`: Reorder rows.
- `group_by()` + `summarize()`: Aggregate and collapse data by group.
- (Additionally, `join` functions: left-joins, inner-joins, etc. for merging multiple datasets.)

3.2 The Pipe Operator %>%

- The **pipe** (`%>%`) takes an object on the left and feeds it into the function on the right as the first argument.
- Example:

  ```
  flights %>%
    select(origin, dest) %>%
    filter(dest == "IAH")
  ```

  This sequence starts with `flights`, selects only the `origin` and `dest` columns, and then filters rows to keep only `dest == "IAH"`. This yields a smaller data frame.
- Without pipes, the equivalent might be:

  ```
  filter(
    select(flights, origin, dest),
    dest == "IAH"
  )
  ```

  which is less readable.

3.3 Example: Filtering Delayed Flights

```
flights %>%
  filter(dep_delay > 120)
```

keeps only the flights that departed more than 2 hours late. Out of 336,776 rows, one might get (for example) 9,743 rows left.

3.4 Saving Results

- By default, `flights %>% filter(...)` just prints to the screen. To store the result, assign it to a new object, e.g.:

  ```
  delayed_2hr <- flights %>% filter(dep_delay > 120)
  ```
- `delayed_2hr` is now a smaller data frame containing only those delayed flights.

---

4 Looking Ahead

- **Multiple Linear Regression**:
  - Next steps: actually *fitting* the model from data (least squares, parameter estimates, etc.).
  - Interpretation of coefficients, p-values, confidence intervals, and how they relate to the confounder/intermediate-variable discussion.
- **Data Wrangling**:
  - We will further explore `mutate`, `arrange`, `group_by`/`summarize`, and `join` operations, crucial for preparing real datasets.
  - In practice, you often `filter` big raw data to isolate the subset relevant to your study, `mutate` new columns for analyses, then feed it into a regression model.
- **Project/Homework Notes**:
  - You will find and/or create your own dataset, define a clear *treatment* and *outcome*, identify possible confounders, and apply MLR.
  - Be mindful of *downstream variables* and ensure you are not mistakenly controlling for your own outcome or a direct mediator in the model.

End of Feb. 4 Lecture Notes.