Statistics II

Week 4: **Causal Graphs**

Content for Today

1. Directed Acyclic Graphs (DAGs)
2. Thinking about bias
3. Plotting with `ggplot`
4. R tutorial

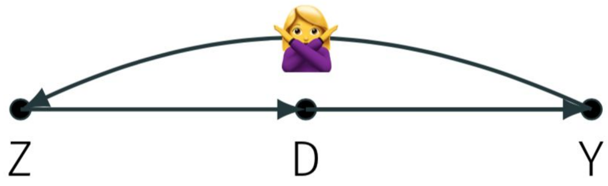
Why use causal graphs?

- Layout our assumptions about causal relations between variables in an intuitive way.
- Practical to assess whether a causal effect can be identified, or not.
- Useful to define what variables to control for and to communicate why the model was specified that way.
- They are non-parametric, i.e. do not imply assumptions about *distributions* of variables or functional form of relationships: they just state the assumed *directionality* of effects.

Directed Acyclic Graphs – DAGs

DAGs represent our qualitative causal assumptions about the data-generating process in the population (i.e. how we think stuff works).

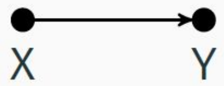
1. They are **directed**: all edges have a direction (→)
2. They are **acyclic**: no feedback loops.

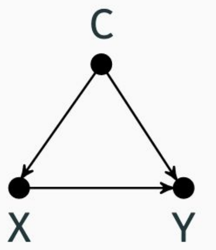


Characteristics of DAGs

Variables are the nodes (or *vertices*) of the graph.

Links between nodes are called edges (or *arcs*).

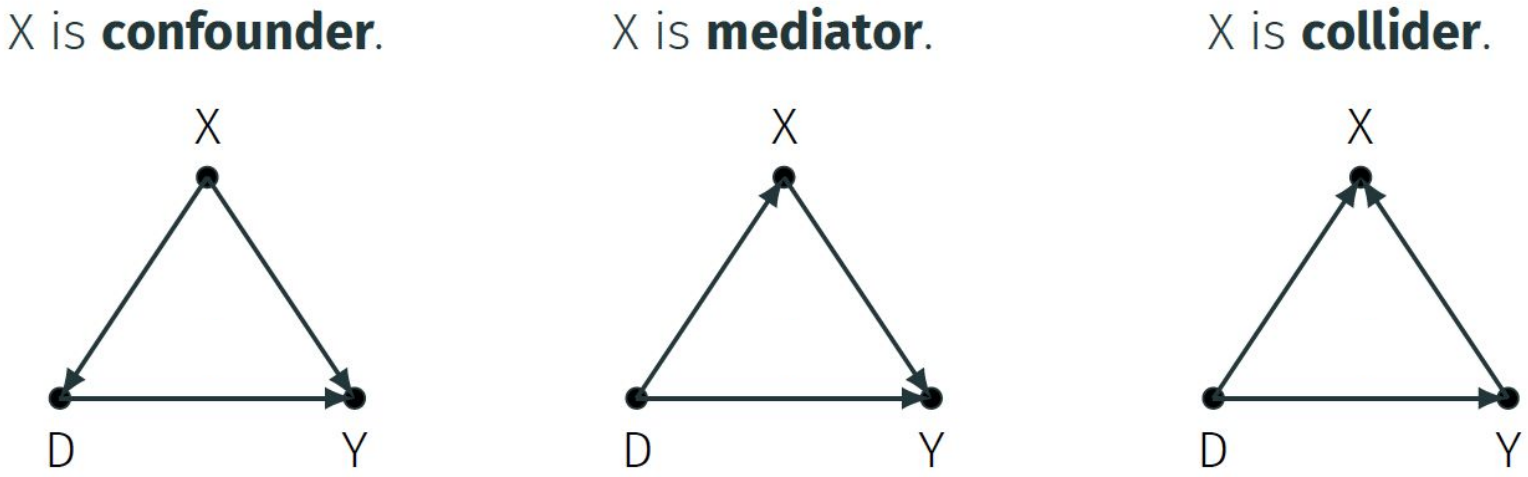




C is exogenous and a parent of X and Y

X is endogenous and a child of C

Common node types



Paths

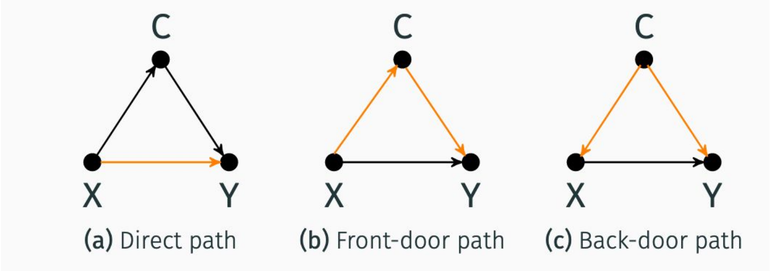
Paths can be causal or non-causal

$P \rightarrow Q \rightarrow R$ is a causal path

$P \rightarrow Q \rightarrow R \leftarrow S$ is a non-causal path (but it is a path)

Depends on direction of edges

Paths can be open or closed



We can alter this depending on:

1. Whether or not we control for variables.
2. And which type of variables we control for.

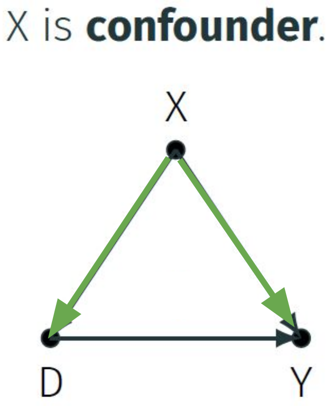
Remember from session 3 on regression...

If **D** and the **error term** are independent, our β_1 *could* be the **ATE** since there would be no selection bias.

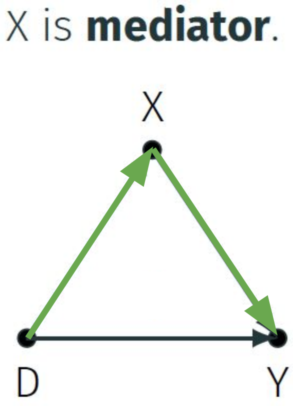
In order to achieve this, we need to have the **true model**.

This is where putting our qualitative assumptions in **causal graphs** can help us lay out our models in a very intuitive way and help us answer the key question:

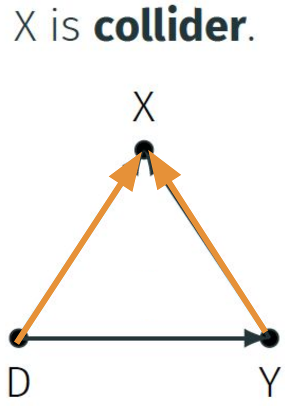
What should we control for?



Paths are open at **confounders**.



Paths are open at **mediators**.



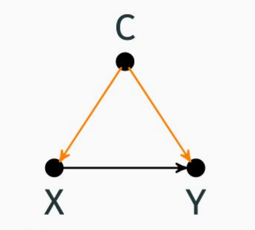
Paths are closed at **colliders**.

Key:

- An open path induces statistical association between two variables.
- Absence of an open path implies statistical independence.


What do we want?
D → Y
Treatment assignment as good as random
How do we get it?
Closing back-door paths
imgflip.com

Back-door criterion



(c) Back-door path

Back-door paths are non-causal paths between X and Y that start with an arrow into **X.**

How to close them?

If path contains confounder, condition on confounder

If path contains collider, it is already closed. Do not condition on collider.

Strategy for d-separation

1. Lay out your assumptions in a DAG based on empirical a theoretical knowledge.
2. Identify the causal and non-causal paths from **D** to **Y.**
3. Identify the adjustments that would close the non-causal paths.

   Be careful with colliders. If path includes colliders, it is already closed.
4. Include the identified variables in the model specifications.

   Be careful not to include any variables in a causal path from D to Y.
5. Do not give a causal interpretation to any coefficient other than that for D.

We can incorporate these steps in our own research

Thinking about bias

Remember:

1. A path is **open** or **unblocked** at non-colliders (confounders or mediators)
2. A path is **(naturally) blocked at colliders**
3. An **open path induces statistical association between two variables**
4. Absence of an open path implies statistical independence
5. Two variables are **d-connected** if there is an open path between them
6. Two variables are **d-separated** if the path between them is blocked

Thinking about bias

Conditioning on a **collider** (or a descendant) leads to **collider bias** or **endogenous bias**

*we will look at this through simulated data in our script

Thinking about bias

Failing to condition on a **confounder** leads to **omitted variable bias**

*we will look at this through simulated data in our script

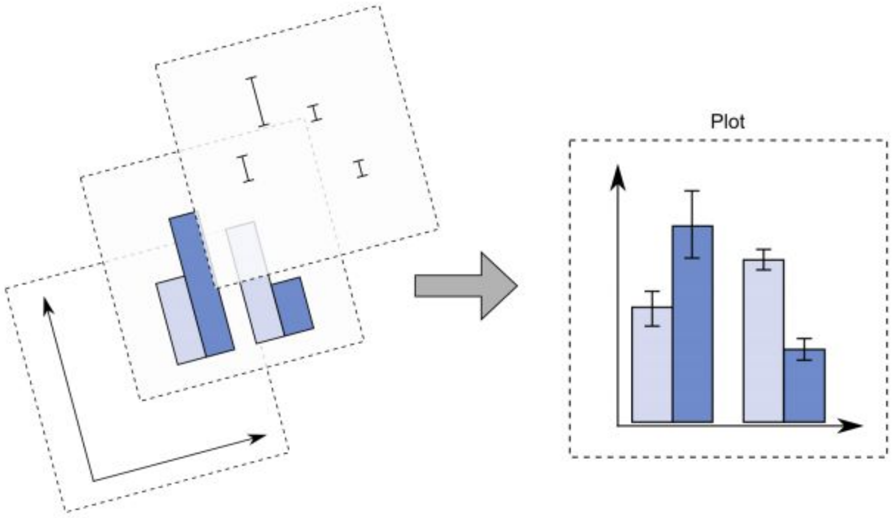
Thinking about bias

Conditioning on a **mediator** leads to **overcontrol** or **post-treatment bias**

Plotting with `ggplot2`

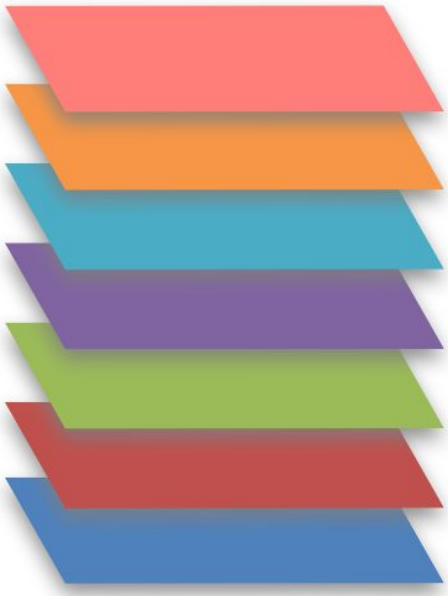
`ggplot2`

In `ggplot2`, a graph is made up of a series of layers



Download the cheat sheet: https://tinyurl.com/h5o9tfq

Describes all the non-data ink	Theme
Plotting space for the data	Coordinates
Statistical models & summaries	Statistics
Rows and columns of sub-plots	Facets
Shapes used to represent the data	Geometries
Scales onto which data is mapped	Aesthetics
The actual variables to be plotted	Data



ggplot2

- The appearance and location of these geoms (such as size and color) are controlled by the **aesthetics properties**: `aes()`

  The variables you want to plot are are referred to here.

```
myGraph <- ggplot(data,
                  aes(x = variable_for_x_axis,
                      y = variable_for_y_axis)) +
                  geom()
```

Geometric objects are the visual elements such as bars and points: **`geom()`**

- There are many kinds of geoms, such as scatterplots (`geom_point`) or barplots (`geom_bar`)

ggplot2

Types of Geoms:

Scatterplot: `geom_point()`

Histogram: `geom_histogram()`

Barplot: `geom_bar()`

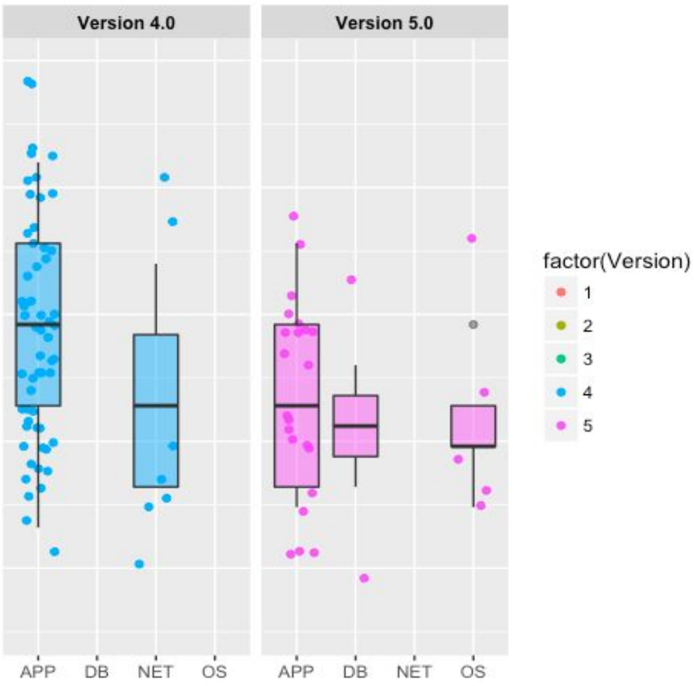
Boxplot: `geom_boxplot()`

Density: `geom_density()`

Adding a “linear regression” line:

`geom_smooth(model = lm)`

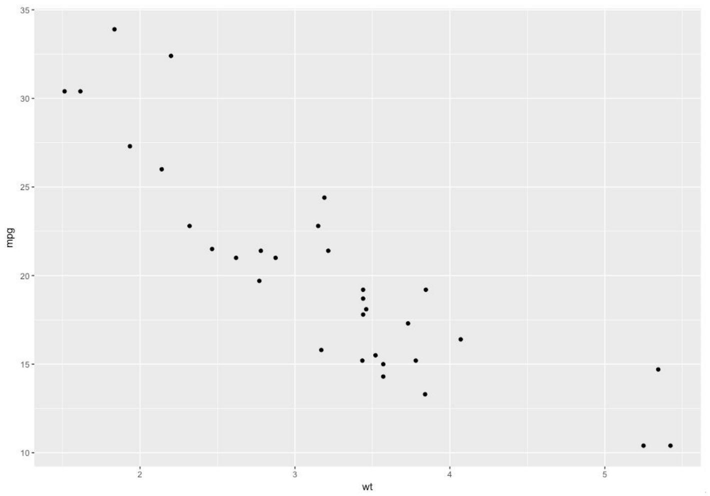
Or a combination of multiple.



ggplot2

Using the build-in `mtcars` data frame, we can plot a basic scatterplot:

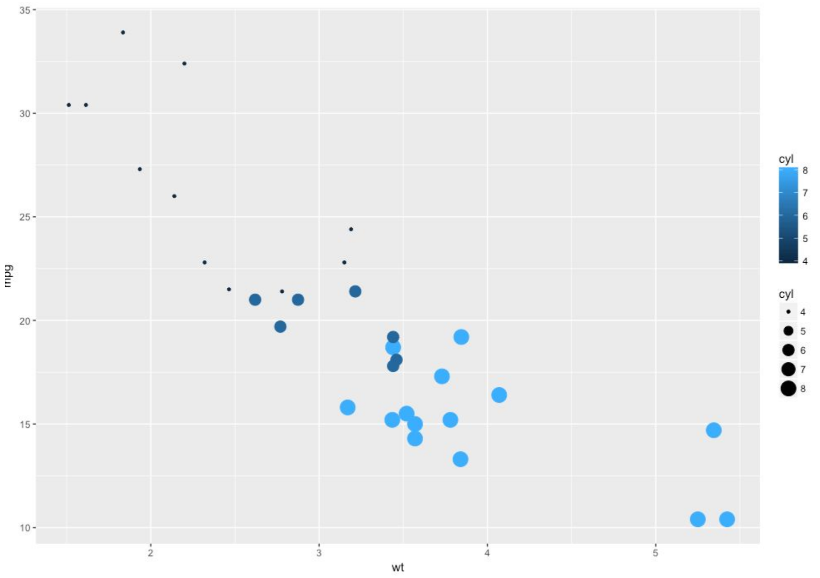
```
scatterplot <- ggplot(
    data = mtcars,
    aes(x = wt, y = mpg)) +
  geom_point()
```



ggplot2

From here we can add extra design elements, like changing the color and size of the points based on cylinder size:

```
scatterplot <- ggplot(
    data = mtcars,
    aes(x = wt,
        y = mpg,
        col = cyl,
        size = cyl)) +
  geom_point()
```



ggplot2

There are lots of other things we can do, too!

Change the transparency using `alpha` and a number from 0-1: ex. `alpha = 0.6`

Add a theme: ex. `geom_bar() + theme_bw()`

Add labels and titles: ex. `+ xlab('X label')+ ylab('Y label') + ggtitle(' Title')`

Change the size and shape of lines, points, etc.

Zoom in on a certain part of a graph

And more :)

Let’s move to R!