Medical Statistics – Lab 8
R version

Welcome to lab 8. In this lab, we will build prediction models using backward elimination and automated procedures, and we will practice reasoning with causal diagrams (DAGs).

We will use the Worcester Heart Attack Study dataset (`whas500.sav`, available from the Datasets menu) and the following R packages: `haven`, `dplyr`, `ggplot2`, `car`, and `MASS`.

```r
library(haven)
library(dplyr)
library(ggplot2)
library(car)
library(MASS)


whas500 <- read_sav("whas500.sav")
whas500 <- whas500 |> mutate(across(where(is.labelled), as_factor))
```

Part 1: Building prediction models using backward elimination

In this part of the lab, we will build a prediction model for hospital length of stay (`los`) in patients with acute myocardial infarction. The dataset comes from the Worcester Heart Attack Study (WHAS) and includes data from 500 patients admitted in Worcester, Massachusetts in 1997, 1999, and 2001.

Key variables in the dataset include:

- `los`: Length of hospital stay (days, continuous outcome)
- `age`: Age at hospital admission (years)
- `gender`: Gender (0 = Male, 1 = Female)
- `hr`: Initial heart rate (beats per minute)
- `sysbp` and `diasbp`: Initial systolic and diastolic blood pressure (mmHg)
- `bmi`: Body mass index (kg/m^2)
- `cvd`: Presence of cardiovascular disease (0 = No, 1 = Yes)
- `sho`: Presence of cardiogenic shock (0 = No, 1 = Yes)

Step 1: Fit the initial linear regression model

Fit an initial linear regression model for hospital length of stay (`los`) using `lm()` with predictors `age`, `gender`, `hr`, `sysbp`, `diasbp`, `bmi`, `cvd`, and `sho`. Summarize the model output to inspect coefficients and p-values.

Step 2: Eliminate the least significant predictor

To identify the least significant predictor, we use the Type III ANOVA table:

- Significance threshold: $p > 0.10$
- Remove the predictor with the largest p-value above this threshold.

Use the `Anova()` function from the `car` package to obtain the Type III ANOVA table (see lab week 6).

Step 3: Repeat the steps

Iteratively remove the least significant predictor until all predictors have $p < 0.10$. At each step:

- Rerun the regression model
- Generate the Type III ANOVA table
- Remove the least significant predictor

Step 4: Final model

Present the final linear regression model:

- Report the final model, including regression coefficients and 95% CIs.
- Create residual plots to assess the model assumptions (normality, homoscedasticity, linearity).

> 💡 95% CIs for regression coefficients
>
> To obtain 95% CIs for the regression coefficients after fitting your final model (e.g., `fit <- lm(...)`), use:
>
> ```
> confint(fit)
> ```

Part 2: Automated procedures for building prediction models (logistic regression)

In this part, we explore automated procedures for predictor selection in **logistic regression** prediction models. We use the same WHAS dataset but now focus on predicting in-hospital death (`dstat`: alive/dead) from candidate predictors.

Automated model selection (AIC)

`stepAIC()` from the `MASS` package can be used for automated selection based on AIC. Because AIC compares overall model fit (not individual p-values), the selected model may differ from manual backward elimination.

```
# Create a 0/1 outcome (1 = dead)
whas500 <- whas500 |>
  mutate(dstat01 = ifelse(dstat == "dead", 1, 0))

fit_full <- glm(
  dstat01 ~ age + gender + hr + sysbp + diasbp + bmi + cvd + sho,
  family = binomial,
  data = whas500
)

fit_step <- stepAIC(fit_full, direction = "backward", trace = FALSE)
summary(fit_step)
```

> Question
>
> Inspect your final selected logistic regression model. Which predictors are retained in the prediction model?

> 💡 Reporting odds ratios (ORs) and 95% CIs
>
> For logistic regression, `summary()` reports regression coefficients on the **log-odds** scale. For reporting, it is often more useful to report **odds ratios (ORs)** with **95% confidence intervals (CIs)**.
> You can compute these from the `summary()` output (Wald CI):

```
s <- summary(fit_step)$coefficients

OR <- exp(s[, "Estimate"])
CI_lower <- exp(s[, "Estimate"] - 1.96 * s[, "Std. Error"])
CI_upper <- exp(s[, "Estimate"] + 1.96 * s[, "Std. Error"])

cbind(OR = OR, CI_lower = CI_lower, CI_upper = CI_upper)
```

Or compute them more directly by exponentiating the coefficient confidence intervals:

```
exp(cbind(OR = coef(fit_step), confint.default(fit_step)))
```

Part 3: Causal diagrams

For each of the exercises below:

- Try solving the diagrams by hand by using the recipe from the lecture (see lecture slides on Brightspace)
- Check your answer using the DAGitty webtool

Exercise 1

In the graph depicted below, for which variables do you need to adjust to assess the unconfounded effect of E on O (there may be several possibilities)?
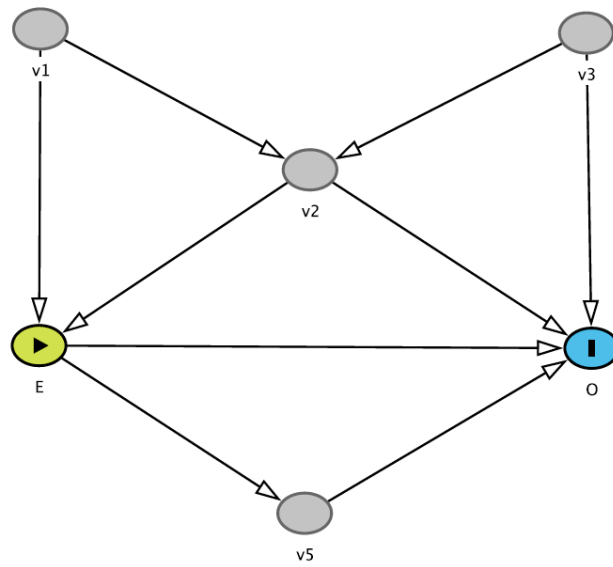
Figure 1: DAG exercise 1

Exercise 2

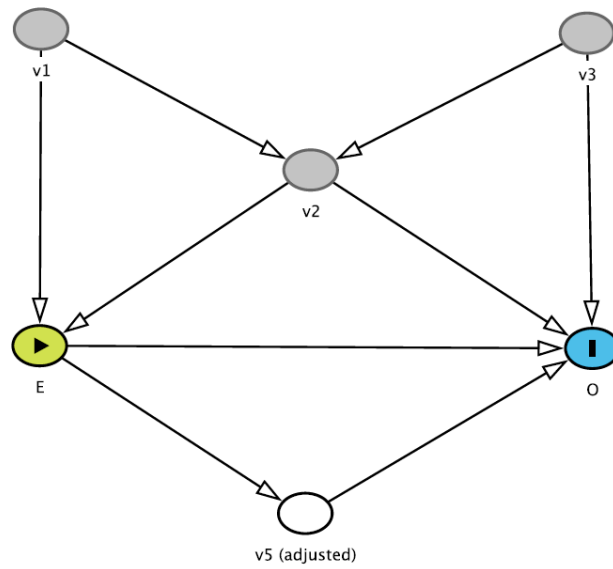In the graph depicted below, what happens when you additionally adjust for **v5**?

Figure 2: DAG exercise 2

Exercise 3

This diagram is slightly different: **v1** now is the exposure. For which variables do you need toadjust to assess the unconfounded effect of **v1** on **O**?
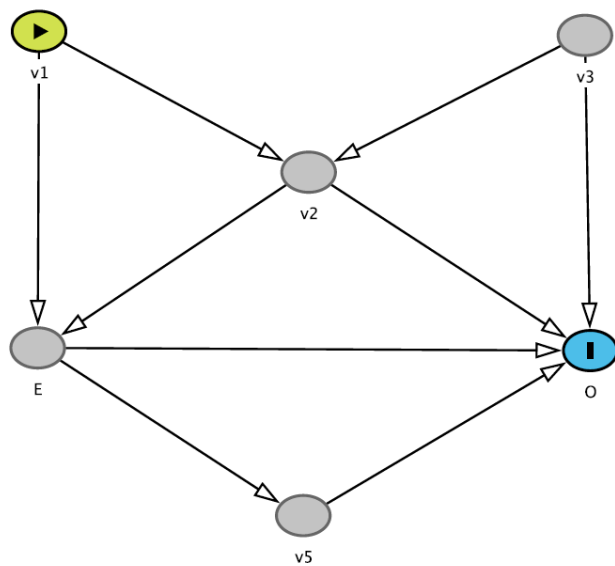
Figure 3: DAG exercise 3

Exercise 4

Now, **v2** is the exposure. For which variables do you need to adjust to assess the total unconfounded effect of **v2** on **O**?
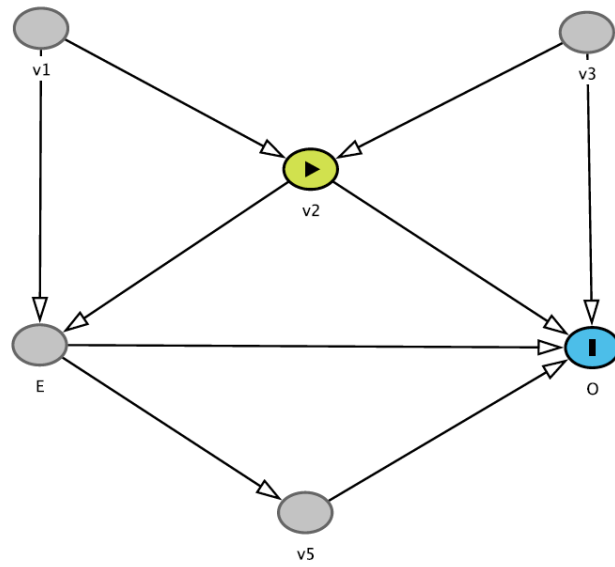
Figure 4: DAG exercise 4

Exercise 5

Back to the first DAG. However, **v2** is now unmeasured. Can we still obtain an unconfounded estimate of the effect of **E** on **O**?
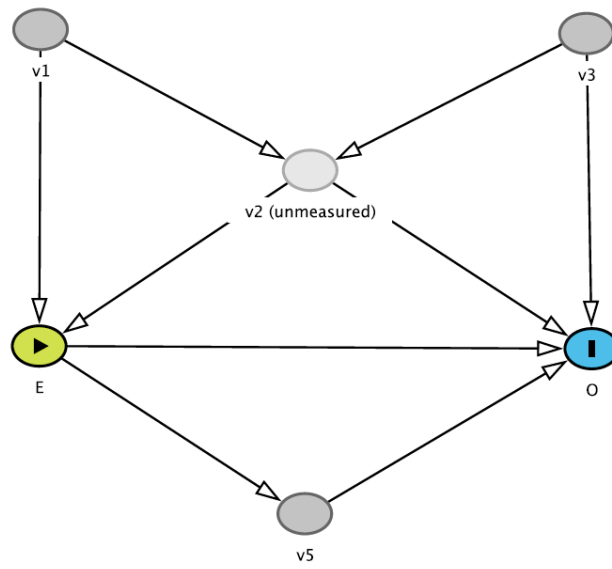
Figure 5: DAG exercise 5

Exercise 6

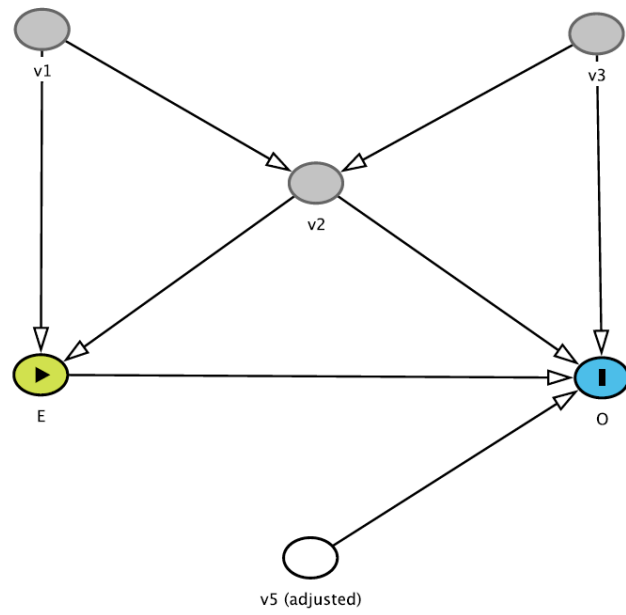See the DAG below: you adjusted for **v5**. What would be the consequence of this action?

Figure 6: DAG exercise 6