

Predictor Selection

Choosing which variables to include in a regression model

Contents

Configuring R	2
Acknowledgements	3
1 Introduction	3
1.1 Why predictor selection matters	3
1.2 Strategies for predictor selection	3
1.3 The role of subject-matter knowledge	4
2 Inferential Goals	4
2.1 Three inferential goals	4
2.1.1 Goal 1: Prediction	4
2.1.2 Goal 2: Evaluating a predictor of primary interest	4
2.1.3 Goal 3: Identifying the important predictors of an outcome	5
3 Causal Thinking and DAGs	5
4 Prediction	5
4.1 Bias–variance trade-off and overfitting	5
4.2 Numerical example of overfitting	5
4.3 Measures of prediction error	5
4.4 Optimism and cross-validation	7
4.5 Cross-validation for LASSO in the HERS data	8
4.6 Development and validation sets	8
5 Evaluating a Predictor of Primary Interest	9
5.1 Evaluating a predictor of primary interest	9
5.2 Including predictors for face validity	9
5.3 Selecting confounders on statistical grounds	9
5.3.1 Numerical example: 10% rule in WCGS	10
5.4 Example: backward selection for CHD confounders	13
5.5 Interactions with the primary predictor	14
5.6 Randomized experiments	14
6 Identifying Multiple Important Predictors	15
6.1 Identifying multiple important predictors	15
6.2 Allen–Cady modified backward selection	15
6.3 Cautious interpretation	15
6.4 Example: risk factors for CHD in HERS	16
7 Testing-based selection	16
7.1 Stepwise regression	16
7.2 Example: stepwise selection for LDL	16
8 Criterion-based selection	17

8.1	AIC and BIC	17
8.2	Comparing models for LDL	17
9	Penalized regression	17
9.1	The LASSO	17
9.2	Example: LASSO for LDL	19
10	Some Details	21
10.1	Collinearity	21
10.1.1	Variance inflation factor	21
10.2	Number of predictors	22
10.3	Alternatives to backward selection	22
10.4	Model selection complicates inference	22
11	Summary	23
	Learning Objectives	24
	References	24

Configuring R

Functions from these packages will be used throughout this document:

```
library(conflicted) # check for conflicting function definitions
# library(printr) # inserts help-file output into markdown output
library(rmarkdown) # Convert R Markdown documents into a variety of formats.
library(pander) # format tables for markdown
library(ggplot2) # graphics
library(ggfortify) # help with graphics
library(dplyr) # manipulate data
library(tibble) # `tibble`s extend `data.frame`s
library(magrittr) # `%>%` and other additional piping tools
library(haven) # import Stata files
library(knitr) # format R output for markdown
library(tidyr) # Tools to help to create tidy data
library(plotly) # interactive graphics
library(dobson) # datasets from Dobson and Barnett 2018
library(parameters) # format model output tables for markdown
library(haven) # import Stata files
library(latex2exp) # use LaTeX in R code (for figures and tables)
library(fs) # filesystem path manipulations
library(survival) # survival analysis
library(survminer) # survival analysis graphics
library(KMsurv) # datasets from Klein and Moeschberger
library(parameters) # format model output tables for
library(webshot2) # convert interactive content to static for pdf
library(forcats) # functions for categorical variables ("factors")
library(stringr) # functions for dealing with strings
library(lubridate) # functions for dealing with dates and times
library(broom) # Summarizes key information about statistical objects in tidy tibbles
library(broom.helpers) # Provides suite of functions to work with regression model 'broom::tidy()' t
```

Here are some R settings I use in this document:

```
rm(list = ls()) # delete any data that's already loaded into R

conflicts_prefer(dplyr::filter)
```

```

ggplot2::theme_set(
  ggplot2::theme_bw() +
    # ggplot2::labs(col = "") +
    ggplot2::theme(
      legend.position = "bottom",
      text = ggplot2::element_text(size = 12, family = "serif")))

knitr::opts_chunk$set(message = FALSE)
options('digits' = 6)

panderOptions("big.mark", ",")
pander::panderOptions("table.emphasize.rownames", FALSE)
pander::panderOptions("table.split.table", Inf)
conflicts_prefer(dplyr::filter) # use the `filter()` function from dplyr() by default
legend_text_size = 9
run_graphs = TRUE

```

Acknowledgements

This content is adapted from:

- (Vittinghoff et al. 2012, chap. 10)
- (Heinze et al. 2018)

1 Introduction

1.1 Why predictor selection matters

In regression modeling, we often have many potential predictors available, but including too many predictors can cause problems:

- **Overfitting:** the model fits noise in the training data and predicts poorly for new data.
- **Multicollinearity:** highly correlated predictors make coefficients unstable and hard to interpret.
- **Loss of precision:** including unnecessary predictors increases standard errors.

On the other hand, excluding important predictors can lead to:

- **Confounding bias:** omitting a common cause of the exposure and outcome distorts the estimated association.
- **Reduced power:** failing to adjust for strong predictors of the outcome can reduce statistical power.

1.2 Strategies for predictor selection

Vittinghoff et al. (2012, chap. 10) describes several general approaches to predictor selection:

1. **Science-driven (purposeful) selection:** Use subject-matter knowledge and a causal diagram (DAG) to identify which variables to include. Science-driven selection is the preferred approach when the goal is causal inference.
2. **Testing-based selection:** Use statistical tests to decide which predictors to include. Examples include forward selection, backward elimination, and stepwise selection. These methods have well-known problems and should be used with caution (see Section 7).
3. **Criterion-based selection:** Compare models using information criteria such as AIC or BIC. These criteria balance model fit against complexity (see Section 8).

4. **Penalized regression:** Use regularization methods (such as the LASSO) that shrink coefficients toward zero, effectively performing selection and estimation simultaneously (see Section 9).

1.3 The role of subject-matter knowledge

Regardless of which selection strategy is used, subject-matter knowledge should guide model building.

A regression model can only tell us about *associations* — whether certain predictors are correlated with the outcome in our data. It cannot, on its own, tell us whether those associations reflect causal effects.

To draw causal conclusions, we need:

1. A causal model (such as a DAG) specifying assumed causal relationships.
2. Appropriate adjustment for confounders.
3. Careful avoidance of adjusting for mediators or colliders.

These concepts are discussed in the causal inference chapter¹.

2 Inferential Goals

2.1 Three inferential goals

The appropriate strategy for predictor selection depends on the inferential goal of the analysis. Vittinghoff et al. (2012, chap. 10) distinguishes three distinct goals:

2.1.1 Goal 1: Prediction

The primary aim is to minimize prediction error for new observations. The predictors' individual causal interpretations are secondary.

Exm

Example 2.1. *Example:* Walter et al. (2001) developed a model to identify older adults at high risk of death following hospitalization, using backward selection on a development set and validating predictions in an independent hospital.

For this goal, we use measures of **prediction error** (PE) to choose among candidate models. Overfitting — the tendency of flexible models to fit noise — is the central concern. Strategies such as cross-validation and penalized regression (see Section 4 and Section 9) protect against overfitting.

2.1.2 Goal 2: Evaluating a predictor of primary interest

The aim is to obtain a valid, minimally confounded estimate of the association between a specific exposure and the outcome.

Exm

Example 2.2. *Example:* Grodstein et al. (2001) estimated the effect of hormone therapy on CHD risk in the Nurses' Health Study (NHS), controlling for a wide range of known CHD risk factors to minimize confounding.

In observational studies, confounding is the primary concern. In randomized trials (like HERS), treatment is already unconfounded by design, but adjusting for strong prognostic covariates can improve precision. DAGs (see the causal inference chapter²) are especially useful for this goal.

¹[causal-inference.qmd#sec-pred-sel-dag](#)

²[causal-inference.qmd#sec-pred-sel-dag](#)

2.1.3 Goal 3: Identifying the important predictors of an outcome

The aim is to discover which predictors are independently associated with the outcome.

Exm

Example 2.3. *Example:* Orwoll et al. (1996) examined independent predictors of bone mass in the Study of Osteoporotic Fractures (SOF), including all predictors statistically significant at $p < 0.05$ in age-adjusted models.

Identifying important predictors is the most challenging goal because inferences about multiple predictors are of direct interest, making false-positive findings and unstable selections particular concerns. Inclusive models with cautious interpretation are preferred (see Section 6.1).

In practice, these goals often overlap. For example, an analysis can estimate the causal effect of treatment (Goal 2) while also selecting additional prognostic covariates to improve individualized risk prediction performance (Goal 1) in the same model-building workflow.

3 Causal Thinking and DAGs

DAG material for this course is covered in the causal inference chapter³.

For the worked examples, see:

- The HERS DAG example⁴
- The WCGS DAG example⁵

4 Prediction

4.1 Bias–variance trade-off and overfitting

When building a prediction model, adding more predictors generally **reduces bias**: the model better captures the true relationships in the population. But it also **increases variance**: more parameters to estimate means more sampling variability, and the model can start to fit noise in the training data.

Definition 4.1 (Overfitting). **Overfitting** occurs when a model fits the training data well but predicts poorly for new observations. It results from including too many predictors relative to the effective sample size.

A good prediction model minimizes the sum of bias and variance — the **bias–variance trade-off**. Adding a predictor helps only if the reduction in bias outweighs the increase in variance.

4.2 Numerical example of overfitting

Table 1 gives a simple numerical example. A parsimonious model and an intentionally overfit model are both trained on the same development set. The overfit model adds many noise predictors that have no true relationship with LDL.

In this example, the overfit model usually has lower training RMSE but higher validation RMSE, which is the defining pattern of overfitting.

4.3 Measures of prediction error

To select a model that minimizes prediction error, we need a measure of PE that does not simply increase with the number of predictors.

³[causal-inference.qmd#sec-pred-sel-dag](#)

⁴[causal-inference.qmd#sec-pred-sel-hers-dag](#)

⁵[causal-inference.qmd#sec-pred-sel-wcgs-dag](#)

Table 1: Numerical example showing overfitting: lower training error but worse validation error.

```

set.seed(204)
n_noise_predictors <- 20

hers_base <-
  hers_ldl |>
  dplyr::select(LDL, HT, age, statins) |>
  tidyr::drop_na()

n_overfit <- nrow(hers_base)
idx_overfit <- sample(seq_len(n_overfit), size = floor(0.7 * n_overfit))

train_overfit <- hers_base[idx_overfit, ]
valid_overfit <- hers_base[-idx_overfit, ]

for (j in seq_len(n_noise_predictors)) {
  zname <- paste0("z", j)
  train_overfit[[zname]] <- rnorm(nrow(train_overfit))
  valid_overfit[[zname]] <- rnorm(nrow(valid_overfit))
}

model_parsimonious <- lm(LDL ~ HT + age + statins, data = train_overfit)
noise_terms <- paste0("z", seq_len(n_noise_predictors))
overfit_formula <- reformulate(c("HT", "age", "statins", noise_terms), "LDL")
model_overfit <- lm(
  overfit_formula,
  data = train_overfit
)

rmse <- function(y, yhat) {
  sqrt(mean((y - yhat)^2))
}

tibble::tibble(
  Model = c(
    "Parsimonious model",
    paste0("Overfit model (+", n_noise_predictors, " noise predictors)")
  ),
  Training_RMSE = c(
    rmse(
      train_overfit$LDL,
      predict(model_parsimonious, newdata = train_overfit)
    ),
    rmse(train_overfit$LDL, predict(model_overfit, newdata = train_overfit))
  ),
  Validation_RMSE = c(
    rmse(
      valid_overfit$LDL,
      predict(model_parsimonious, newdata = valid_overfit)
    ),
    rmse(valid_overfit$LDL, predict(model_overfit, newdata = valid_overfit))
  )
) |>
dplyr::mutate(
  dplyr::across(c(Training_RMSE, Validation_RMSE), ~ round(.x, 2))
)

#> # A tibble: 2 x 3
#>   Model                                Training_RMSE Validation_RMSE
#>   <chr>                                <dbl>          <dbl>
#> 1 Parsimonious model                    6           37.0
#> 2 Overfit model (+20 noise predictors) 36.9          37.0

```

For **continuous outcomes**:

- **Adjusted R^2** penalizes the number of predictors:

$$\bar{R}^2 = 1 - \frac{n-1}{n-p-1}(1-R^2)$$

Adding a variable increases $\bar{R}^2$ only if it reduces the residual variance more than the penalty.

- **AIC** and **BIC** impose steeper penalties:

$$\text{AIC} = -2\hat{\ell} + 2p, \quad \text{BIC} = -2\hat{\ell} + p \log n$$

(see Section 8).

For **binary outcomes**, common measures include **balanced accuracy** and the **C-statistic**.

Balanced accuracy:

$$\text{BA} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

where

$$\text{Sensitivity} = \Pr(\hat{Y} = 1 \mid Y = 1), \quad \text{Specificity} = \Pr(\hat{Y} = 0 \mid Y = 0).$$

If J is **Youden's J statistic**,

$$J = \text{Sensitivity} + \text{Specificity} - 1,$$

then

$$\text{BA} = \frac{J + 1}{2}.$$

So BA is a linear transformation of J , and BA and J are equivalent for comparing models.

C-statistic (area under the ROC curve, AUC):

$$\begin{aligned} C &\stackrel{\text{def}}{=} \Pr(\hat{\pi}_i > \hat{\pi}_j \mid Y_i = 1, Y_j = 0) \\ &\quad + \frac{1}{2} \Pr(\hat{\pi}_i = \hat{\pi}_j \mid Y_i = 1, Y_j = 0) \end{aligned}$$

where $\hat{\pi}$ is the predicted event probability. An empirical estimator is

$$\hat{C} = \frac{1}{n_1 n_0} \sum_{i: Y_i=1} \sum_{j: Y_j=0} \left[I(\hat{\pi}_i > \hat{\pi}_j) + \frac{1}{2} I(\hat{\pi}_i = \hat{\pi}_j) \right].$$

It measures discrimination: the model's ability to rank events above non-events.

For **survival outcomes**, the analogous measure is the **C-index**, defined as

$$\hat{C}_{\text{index}} = \frac{N_{\text{concordant}} + 0.5 N_{\text{tied}}}{N_{\text{comparable}}},$$

where pairs are comparable when one subject is observed to fail before the other is known to fail or be censored, and concordance means the subject with earlier failure has the higher predicted risk score.

4.4 Optimism and cross-validation

Naive estimates of PE evaluated on the same data used to fit the model are **optimistic**: they overstate the model's ability to predict new data. For example, R^2 always increases when a predictor is added, even if the predictor is pure noise.

Cross-validation provides a less optimistic estimate of PE by evaluating predictions on observations not used to estimate the model.

Definition 4.2 (k -fold cross-validation). In **k -fold cross-validation**:

1. The data are randomly divided into k equal-sized subsets (“folds”).
 2. For each fold $i = 1, \dots, k$:
 - Fit the model using all data *except* fold i .
 - Compute predicted values for the observations in fold i .
 3. Compute a summary PE measure across all folds.
- Values of $k = 5$ or $k = 10$ are typical.

Using cross-validation to select predictors avoids overfitting because the predictive performance is evaluated on held-out data that did not influence the model fit.

4.5 Cross-validation for LASSO in the HERS data

The LASSO (see Section 9) uses cross-validation to select the penalty parameter λ that minimizes prediction error. The `cv.glmnet` function implements 10-fold cross-validation, fitting the LASSO for a grid of λ values and evaluating mean squared error on held-out folds. The fitted object provides two key λ values: `lambda.min` (minimizes CV mean squared error) and `lambda.1se` (largest λ within one standard error of the minimum). See Section 9 for the HERS example with coefficient paths and CV results.

The two recommended values of λ represent different points on the bias–variance trade-off:

- `lambda.min` minimizes cross-validated MSE, yielding a model with good predictive accuracy.
- `lambda.1se` uses a larger penalty (farther from zero, more shrinkage), resulting in a sparser, more conservative model that is often preferred for interpretability.

4.6 Development and validation sets

The most transparent approach to measuring prediction error is to split the data into a **development set** (used to fit the model) and a **validation set** (used only to evaluate predictions).

When a single data set must be split, a common allocation is two-thirds to development and one-third to validation. External validation using a completely independent sample (from a different time period or study site) provides the most rigorous assessment of generalizability.

```
set.seed(42)
n_hers <- nrow(hers_ldl)
train_idx <- sample(seq_len(n_hers), size = floor(0.67 * n_hers))

hers_train <- hers_ldl[train_idx, ]
hers_valid <- hers_ldl[-train_idx, ]

# Fit model on training data
model_train <- lm(
  LDL ~ HT + age + BMI + diabetes + statins,
  data = hers_train
)

# Evaluate on validation data
pred_valid <- predict(model_train, newdata = hers_valid)
rmse_valid <- sqrt(mean((hers_valid$LDL - pred_valid)^2, na.rm = TRUE))
rmse_train <- sqrt(mean(residuals(model_train)^2))

cat(
  "Training RMSE:", round(rmse_train, 2), "\n",
  "Validation RMSE:", round(rmse_valid, 2), "\n"
)
#> Training RMSE: 36.78
#> Validation RMSE: 36.85
```

If the validation RMSE is much larger than the training RMSE, this discrepancy is a sign of overfitting. In well-specified models with adequate sample sizes, the two values are typically close.

5 Evaluating a Predictor of Primary Interest

5.1 Evaluating a predictor of primary interest

When the goal is to estimate the causal effect of a specific exposure on an outcome, the central challenge in observational data is **confounding**. Relatively inclusive models — those including more potential confounders — are better at minimizing confounding bias.

The following principles guide predictor selection for this goal.

5.2 Including predictors for face validity

Some predictors are such well-established causal antecedents of the outcome that they should be included regardless of their statistical significance in the current data. Including these variables ensures the **face validity** of the model: readers and reviewers can be confident that obvious confounders have been accounted for.

For example, age and smoking are established CHD risk factors and would be included in any model for CHD outcomes, even if they are not statistically significant in a small sample.

5.3 Selecting confounders on statistical grounds

In addition to variables included for face validity, we may consider other potential confounders identified in previous studies or on substantive grounds.

A practical approach is **backward selection with a liberal criterion**:

1. Start with a full model including all pre-specified candidates.
2. At each step, remove the predictor with the largest p-value, provided it exceeds a liberal threshold (e.g., $p > 0.2$).
3. Stop when all remaining predictors meet the retention criterion.

The liberal threshold ($p > 0.2$ rather than $p > 0.05$) is important for confounding control: in small samples, even important confounders may have large p-values, and removing them could introduce bias.

An alternative retention criterion is the **10% change-in-estimate rule** (Kleinbaum and Klein 2010, chap. 8, pp. 194–212; Vittinghoff et al. 2012, chap. 10, pp. 257–275; Mickey and Greenland 1989, 125; Greenland 1989, 340):

Retain a candidate confounder C in the model if dropping it changes the estimated coefficient for the primary predictor by more than 10–15%.

More precisely, let $\hat{\beta}$ be the estimated coefficient for the exposure from the **full model** (including C), and let $\hat{\beta}_0$ be the estimate from the **reduced model** (dropping C). Then C is judged a meaningful confounder if:

$$\left| \frac{\hat{\beta} - \hat{\beta}_0}{\hat{\beta}} \right| > 0.10$$

The threshold is often taken as 10–20%, depending on the context. A smaller threshold (e.g., 10%) retains more variables; a larger threshold (e.g., 20%) allows more aggressive pruning.

i Note

The 10% change-in-estimate rule focuses on **bias** rather than **statistical significance**. A variable can be a meaningful confounder even if it is not statistically significant (because of

small sample size or measurement error), and conversely, a statistically significant variable may cause negligible confounding. The p-value criterion and the change-in-estimate rule can disagree, and Vittinghoff et al. (2012) recommends using both as complementary checks.

5.3.1 Numerical example: 10% rule in WCGS

The WCGS (Western Collaborative Group Study) is an observational prospective cohort of 3,154 men followed for CHD events. Unlike HERS (a randomized trial), the WCGS exposure — Type A behavior pattern — was not randomly assigned, so confounding is a genuine concern.

In the WCGS study, the exposure of interest is Type A behavior pattern (`dibpat`), and the outcome is CHD incidence (`chd69`). We fit logistic regression models and apply the 10% change-in-estimate rule to assess whether age, cholesterol, SBP, and BMI are meaningful confounders of the Type A → CHD relationship, while also examining smoking as an additional covariate.

Adapted from Kleinbaum and Klein (2010, chap. 8), which describes the change-in-estimate procedure for logistic regression models.

```
wcgs <- rmb::wcgs |> haven::as_factor()
wcgs$dibpat <- stats::relevel(wcgs$dibpat, ref = "Type B")

# Full model (all candidate confounders)
m_full <- glm(
  I(chd69 == "Yes") ~ dibpat + age + chol + sbp + smoke + bmi,
  data = wcgs,
  family = binomial
)

# Unadjusted model (Type A only)
m_unadj <- glm(
  I(chd69 == "Yes") ~ dibpat,
  data = wcgs,
  family = binomial
)

# Estimate of Type A effect from full model
beta_full <- coef(m_full)["dibpatType A"]
beta_unadj <- coef(m_unadj)["dibpatType A"]

# Models dropping one confounder at a time
confounders <- c("age", "chol", "sbp", "smoke", "bmi")

results <- lapply(confounders, function(v) {
  f <- as.formula(
    paste("I(chd69 == \"Yes\") ~ dibpat +",
          paste(setdiff(confounders, v), collapse = " + "))
  )
  m_red <- glm(f, data = wcgs, family = binomial)
  beta_red <- coef(m_red)["dibpatType A"]
  pct_change <- abs((beta_full - beta_red) / beta_full) * 100
  data.frame(
    Confounder = v,
    Beta_full = round(beta_full, 4),
    Beta_reduced = round(beta_red, 4),
    Pct_change = round(pct_change, 1),
    Meaningful = ifelse(pct_change > 10, "Yes (>10%)", "No (<=10%)")
  )
})
```

```
results_df <- do.call(rbind, results)
rownames(results_df) <- NULL
```

```
knitr::kable(
  results_df,
  col.names = c(
    "Dropped variable",
    " (full model)",
    " (reduced model)",
    "% change",
    "Meaningful confounder?"
  )
)
```

Table 2: 10% change-in-estimate rule applied to the WCGS data. The Type A coefficient from the full model is compared with coefficients from models that drop one confounder at a time. A percentage change exceeding 10% indicates meaningful confounding.

Dropped variable	(full model)	(reduced model)	% change	Meaningful confounder?
age	0.6972	0.7420	6.4	No ($\leq 10\%$)
chol	0.6972	0.7183	3.0	No ($\leq 10\%$)
sbp	0.6972	0.7303	4.8	No ($\leq 10\%$)
smoke	0.6972	0.7242	3.9	No ($\leq 10\%$)
bmi	0.6972	0.7031	0.8	No ($\leq 10\%$)

In this example, none of the individual confounders causes a $>10\%$ change when dropped, suggesting they are not individually strong confounders of the Type A $\rightarrow$ CHD relationship in WCGS. Comparing the unadjusted log-OR for Type A (≈ 0.86) with the fully adjusted log-OR (≈ 0.7) shows that the overall set of covariates provides modest adjustment. In practice, covariates with established biological relationships to the outcome (such as age, smoking, and blood pressure for CHD) are typically retained regardless of the change-in-estimate result, for reasons of face validity.

When multiple covariates are candidates for removal, checking all possible subsets is often impractical. A practical alternative is a backward-elimination procedure guided jointly by the 10% rule and precision.

At each step:

1. Start from the current model.
2. Fit models that drop one remaining candidate covariate at a time.
3. Keep only candidates with absolute percent change from the full-model exposure coefficient no larger than 10%, where the full-model coefficient is always from the original gold-standard model.
4. Among those candidates, drop the covariate that gives the largest reduction in the exposure coefficient's standard error.
5. Repeat until no remaining covariate can be dropped without either exceeding the 10% threshold or increasing the exposure standard error.

```
beta_full <- coef(m_full)["dibpatType A"]
se_full <- sqrt(vcov(m_full)["dibpatType A", "dibpatType A"])
exposure_var <- "dibpat"

current_covars <- confounders
current_se <- se_full
step_num <- 1
bwe_steps <- list()

repeat {
```

```

candidate_results <- lapply(current_covars, function(v) {
  keep_covars <- setdiff(current_covars, v)
  reduced_formula <- if (length(keep_covars) == 0) {
    as.formula("I(chd69 == \"Yes\") ~ dibpat")
  } else {
    as.formula(
      paste(
        "I(chd69 == \"Yes\") ~",
        paste(c(exposure_var, keep_covars), collapse = " + ")
      )
    )
  }

  m_red <- glm(reduced_formula, data = wgs, family = binomial)
  beta_red <- coef(m_red)["dibpatType A"]
  se_red <- sqrt(vcov(m_red)["dibpatType A", "dibpatType A"])
  pct_change <- abs((beta_full - beta_red) / beta_full) * 100
  se_reduction <- current_se - se_red

  data.frame(
    Step = step_num,
    Candidate_drop = v,
    Beta_reduced = beta_red,
    Pct_change_from_full = pct_change,
    SE_reduced = se_red,
    SE_reduction = se_reduction,
    Eligible = pct_change <= 10 & se_reduction > 0
  )
})

candidates_df <- do.call(rbind, candidate_results)
eligible_df <- candidates_df[candidates_df$Eligible, , drop = FALSE]

if (nrow(eligible_df) == 0) {
  break
}

best_idx <- which.max(eligible_df$SE_reduction)
best_drop <- eligible_df[best_idx, , drop = FALSE]
dropped_var <- as.character(best_drop$Candidate_drop)

bwe_steps[[step_num]] <- data.frame(
  Step = step_num,
  Dropped = dropped_var,
  Pct_change_from_full = round(best_drop$Pct_change_from_full, 1),
  SE_before = round(current_se, 4),
  SE_after = round(best_drop$SE_reduced, 4),
  SE_reduction = round(best_drop$SE_reduction, 4)
)

current_covars <- setdiff(current_covars, dropped_var)
current_se <- best_drop$SE_reduced
step_num <- step_num + 1
}

bwe_steps_df <- if (length(bwe_steps) == 0) {
  data.frame(

```

```

    Step = NA_integer_,
    Dropped = "None",
    Pct_change_from_full = NA_real_,
    SE_before = round(se_full, 4),
    SE_after = round(se_full, 4),
    SE_reduction = 0
  )
} else {
  do.call(rbind, bwe_steps)
}

retained_covars <- current_covars

knitr::kable(
  bwe_steps_df,
  col.names = c(
    "Step",
    "Dropped covariate",
    "% change from full model",
    "SE before drop",
    "SE after drop",
    "SE reduction"
  )
)

```

Table 3: Backward elimination under the 10% change-in-estimate rule, with standard-error reduction used to choose among eligible drops. The reference for the percent-change criterion is always the Type A coefficient in the original full model.

Step	Dropped covariate	% change from full model	SE before drop	SE after drop	SE reduction
1	chol	3.0	0.1444	0.1430	0.0013
2	sbp	8.4	0.1430	0.1422	0.0008

For further reading on model selection strategies in epidemiologic analyses, see Vittinghoff et al. (2012, chap. 10, pp. 257–275), Kleinbaum and Klein (2010, chaps. 6–8), and Rothman et al. (2021, chap. 21, section “Model Selection”).

i Collinearity between confounders is not a problem

Unlike collinearity between a predictor of primary interest and other covariates (which inflates its standard error), collinearity *among* adjustment variables does not affect the precision of the primary predictor’s estimate. Thus, two collinear confounders can both be retained if both are judged necessary on substantive grounds.

5.4 Example: backward selection for CHD confounders

In the HERS data, we can use backward selection to identify which baseline covariates confound the effect of hormone therapy on LDL. Since HT was randomized in HERS, adjustment is needed for precision, not confounding; but this toy example illustrates the procedure.

```

full_model <- lm(
  LDL ~ HT + age + BMI + diabetes + smoking + statins + SBP,
  data = hers_ldl
)

```

```

# Backward selection using AIC
back_model <- MASS::stepAIC(full_model, direction = "backward", trace = FALSE)
summary(back_model)
#>
#> Call:
#> lm(formula = LDL ~ age + BMI + diabetes + statins + SBP, data = hers_ldl)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -103.69  -24.08   -3.45   19.33   241.89
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  142.6424     9.2977   15.34  <2e-16 ***
#> age          -0.2881     0.1087   -2.65   0.0081 **
#> BMI           0.4349     0.1337    3.25   0.0012 **
#> diabetesYes  -5.3946     1.6742   -3.22   0.0013 **
#> statinsYes   -16.6397     1.4603  -11.40  <2e-16 ***
#> SBP           0.1232     0.0379    3.25   0.0012 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 36.8 on 2741 degrees of freedom
#> (5 observations deleted due to missingness)
#> Multiple R-squared:  0.0565, Adjusted R-squared:  0.0548
#> F-statistic: 32.8 on 5 and 2741 DF,  p-value: <2e-16

```

The backward selection procedure retains the predictors that most improve model fit according to AIC. In this setting, since HT was randomized, we interpret the selected set as prognostic covariates that improve precision of the HT estimate, not as confounders.

5.5 Interactions with the primary predictor

A useful check on the validity of the selected model is to test for **interactions** between the primary predictor and key covariates. Particularly for novel findings, showing that the association is consistent across subgroups strengthens credibility.

If a substantial and biologically plausible interaction is found, the analysis must account for it. However, exploratory interaction tests are susceptible to false-positive findings, and unexpected interactions should be interpreted cautiously.

5.6 Randomized experiments

In randomized trials, participants are allocated to treatment randomly, so the treatment indicator is by design uncorrelated with pre-treatment covariates in expectation. Confounding adjustment is therefore **not required** for causal inference.

However, there are several reasons to include covariates in trial analyses:

1. **Improving precision** (continuous outcomes). Adjusting for strong predictors of the outcome reduces residual variance and tightens the treatment effect estimate. Because treatment is uncorrelated with the covariates by randomization, variance inflation is negligible.
2. **De-attenuation** (binary and survival outcomes). In logistic and Cox models, omitting important predictors attenuates the treatment effect estimate. Adjusting for them moves the estimate closer to the true subject-specific effect.
3. **Accounting for stratified designs**. In stratified or cluster-randomized trials, including stratification variables in the model yields correct standard errors and p-values.

4. **Correcting for chance imbalances.** In small trials, randomization may leave residual imbalance in important prognostic covariates. Adjusted analyses demonstrate that the main inference is not materially affected by any such imbalance.

i Pre-specification is important in trials

Adjusted analyses in clinical trials should ideally be **pre-specified in the study protocol**, to prevent post-hoc selection of covariates that minimize the treatment effect p-value.

6 Identifying Multiple Important Predictors

6.1 Identifying multiple important predictors

When the goal is to identify *all* important predictors of an outcome (Goal 3, Section 2.1), confounding remains a primary concern for each predictor in turn, not just a single primary predictor of interest.

This goal is the most challenging because:

- Overfitting and false-positive results are a greater concern (many predictors, many opportunities for spurious findings).
- Interactions among candidate predictors may be of interest, but systematically testing all possible interactions can easily produce false positives.
- Collinear predictors may be difficult to disentangle.

The same strategies recommended for Goal 2 (face validity, backward selection with liberal criterion) also apply here. However, cautious interpretation of novel and borderline findings becomes even more important.

6.2 Allen–Cady modified backward selection

A modified backward selection procedure due to Allen and Cady (Allen and Cady 1982) limits false-positive findings while still allowing variable selection.

The procedure works as follows:

1. **Specify** a set of variables to be forced into the model: predictors of primary interest and variables needed for face validity.
2. **Rank** the remaining candidate variables in order of hypothesized importance.
3. Start with a model containing all variables in both sets.
4. Delete variables from the ranked set in order of **ascending importance** (least important first), stopping as soon as the first variable that meets the retention criterion is encountered.

This procedure limits false positives because there is a single pre-specified deletion sequence, and selection stops at the first retained variable. In contrast, conventional backward selection considers *all* remaining predictors at each step, producing many more potential opportunities for false findings.

6.3 Cautious interpretation

For Goal 3 analyses, we do **not** recommend using only predictors significant at $p < 0.05$: in small samples, even important predictors may fail to meet this threshold, and a parsimonious model may have substantial residual confounding.

Instead, the recommended approach is:

- Use a relatively **inclusive model** (liberal backward selection criterion: $p < 0.2$).
- Interpret **novel, weak, and borderline significant** associations cautiously.
- Report the **full list of candidate covariates** considered, including those ultimately excluded.
- Acknowledge inflated type-I error as a study limitation.

6.4 Example: risk factors for CHD in HERS

Vittinghoff et al. (2003) identified independent CHD risk factors in the HERS cohort using a multipredictor Cox model. The large number of outcome events ($n = 361$) allowed all previously identified risk factors with $p < 0.2$ in unadjusted models to be included.

Among 11 predictors retained as important on both substantive and statistical grounds were: ethnicity, exercise habits, diabetes, angina, congestive heart failure, prior heart attacks, blood pressure, LDL, HDL, Lp(a), and creatinine clearance.

The model also controlled for age, smoking, alcohol use, and obesity (not statistically significant after adjustment) to rule out confounding and ensure face validity.

7 Testing-based selection

Testing-based selection methods use statistical tests (such as F-tests or likelihood ratio tests) to decide which predictors to include in a model.

7.1 Stepwise regression

Caution about stepwise selection

Stepwise regression has well-known problems:

- It tends to select too many variables (overfitting).
- P-values and confidence intervals are biased after selection.
- It ignores model uncertainty.
- Results can be unstable across different samples.

Consider using subject-matter knowledge (DAGs), cross-validation, or penalized methods (such as LASSO) instead. See Heinze et al. (2018) for a thorough review.

The basic idea of stepwise selection:

- **Forward selection:** Start with no predictors; add the predictor that most improves model fit at each step.
- **Backward elimination:** Start with all predictors; remove the predictor that least worsens model fit at each step.
- **Bidirectional stepwise:** Combine forward and backward steps.

7.2 Example: stepwise selection for LDL

```
# Full model with all candidate predictors
full_model_ldl <- lm(
  LDL ~ HT + age + BMI + diabetes + smoking + statins + SBP,
  data = hers_ldl
)

# Backward elimination using AIC
step_model_ldl <- step(
  full_model_ldl,
  direction = "backward",
  trace = 0
)

summary(step_model_ldl)
#>
#> Call:
#> lm(formula = LDL ~ age + BMI + diabetes + statins + SBP, data = hers_ldl)
#>
```

```

#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -103.69  -24.08   -3.45   19.33  241.89
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 142.6424     9.2977   15.34  <2e-16 ***
#> age         -0.2881     0.1087   -2.65   0.0081 **
#> BMI          0.4349     0.1337    3.25   0.0012 **
#> diabetesYes -5.3946     1.6742   -3.22   0.0013 **
#> statinsYes  -16.6397     1.4603  -11.40  <2e-16 ***
#> SBP          0.1232     0.0379    3.25   0.0012 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 36.8 on 2741 degrees of freedom
#> (5 observations deleted due to missingness)
#> Multiple R-squared:  0.0565, Adjusted R-squared:  0.0548
#> F-statistic: 32.8 on 5 and 2741 DF,  p-value: <2e-16

```

8 Criterion-based selection

8.1 AIC and BIC

When comparing non-nested models, we can use information criteria:

- **AIC** (Akaike Information Criterion):

$$\text{AIC} = -2\hat{\ell} + 2p$$

- **BIC** (Bayesian Information Criterion):

$$\text{BIC} = -2\hat{\ell} + p \log n$$

where $\hat{\ell}$ is the maximized log-likelihood, p is the number of model parameters, and n is the sample size.

Lower values of AIC or BIC indicate a better model. BIC applies a larger penalty for model complexity, especially for large n .

8.2 Comparing models for LDL

9 Penalized regression

When there are many candidate predictors, penalized regression methods simultaneously estimate model coefficients and perform variable selection by shrinking small coefficients toward zero.

9.1 The LASSO

The **LASSO** (Least Absolute Shrinkage and Selection Operator) adds an ℓ_1 penalty to the log-likelihood:

$$\hat{\beta}_\lambda = \arg \max_{\beta} \left\{ \hat{\ell}(\beta) - \lambda \sum_{j=1}^p |\beta_j| \right\}$$

The tuning parameter λ controls the strength of the penalty:

- $\lambda = 0$: no penalty (ordinary least squares).

Table 4: Comparison of LDL models by AIC, BIC, and adjusted R^2 .

```
library(tibble)

models_ldl <- list(
  "HT only" = lm(LDL ~ HT, data = hers_ldl),
  "HT + age" = lm(LDL ~ HT + age, data = hers_ldl),
  "HT + statins" = lm(LDL ~ HT + statins, data = hers_ldl),
  "HT + age + BMI + diabetes + statins" = lm(
    LDL ~ HT + age + BMI + diabetes + statins,
    data = hers_ldl
  ),
  "Full model" = lm(
    LDL ~ HT + age + BMI + diabetes + smoking + statins + SBP,
    data = hers_ldl
  )
)

tibble(
  Model = names(models_ldl),
  AIC = sapply(models_ldl, AIC),
  BIC = sapply(models_ldl, BIC),
  Adj_R2 = sapply(
    models_ldl,
    function(m) summary(m)$adj.r.squared
  )
)

#> # A tibble: 5 x 4
#>   Model                AIC    BIC   Adj_R2
#>   <chr>             <dbl> <dbl>   <dbl>
#> 1 HT only          27807. 27825. -0.000341
#> 2 HT + age         27802. 27826.  0.00202
#> 3 HT + statins      27681. 27704.  0.0451
#> 4 HT + age + BMI + diabetes + statins 27617. 27658.  0.0511
#> 5 Full model        27609. 27662.  0.0545
```

- $\lambda \rightarrow \infty$: all coefficients shrink to zero.

For intermediate values of λ , some coefficients are exactly zero (excluded), while others are shrunk but non-zero.

The optimal λ is usually chosen by cross-validation.

9.2 Example: LASSO for LDL

```
library(glmnet)
library(ggfortify)

hers_ldl_complete <-
  hers_ldl |>
  dplyr::select(LDL, HT, age, BMI, diabetes, smoking, statins, SBP) |>
  tidyr::drop_na()

y_ldl <- hers_ldl_complete$LDL
x_ldl <-
  hers_ldl_complete |>
  dplyr::select(-LDL) |>
  mutate(
    HT = as.integer(HT == "HT"),
    statins = as.integer(statins == "Yes"),
    diabetes = as.integer(diabetes == "Yes"),
    smoking = as.integer(smoking == "Yes")
  ) |>
  as.matrix()

lasso_fit <- glmnet(x_ldl, y_ldl)
autoplot(lasso_fit, xvar = "lambda") +
  theme_bw()
```

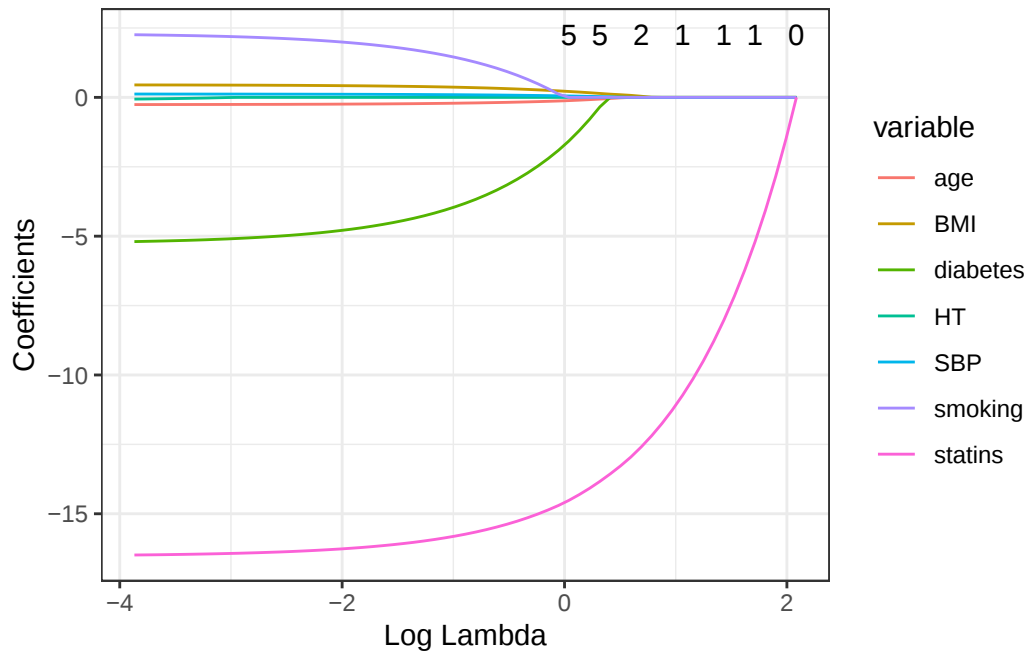


Figure 1: LASSO coefficient paths for LDL predictors in HERS. Each line traces one predictor's coefficient as λ increases. Coefficients shrink toward zero as the penalty grows; some reach exactly zero (variables excluded from the model).

Table 5: LASSO coefficients at two values of the regularization parameter. `lambda.min` minimizes cross-validation error; `lambda.1se` uses a more conservative (larger) λ , within one standard error of the minimum, producing a sparser model.

```
data.frame(
  lambda.min = as.vector(coef(cv_lasso, s = "lambda.min")),
  lambda.1se = as.vector(coef(cv_lasso, s = "lambda.1se")),
  row.names = rownames(coef(cv_lasso))
)
#> # A tibble: 8 x 2
#>   lambda.min lambda.1se
#>   <dbl>      <dbl>
#> 1    141.      146.
#> 2     0         0
#> 3   -0.239      0
#> 4    0.416      0
#> 5   -4.70      0
#> 6    1.93      0
#> 7   -16.2     -2.84
#> 8    0.110      0
```

```
cv_lasso <- cv.glmnet(x_ldl, y_ldl)
plot(cv_lasso)
```

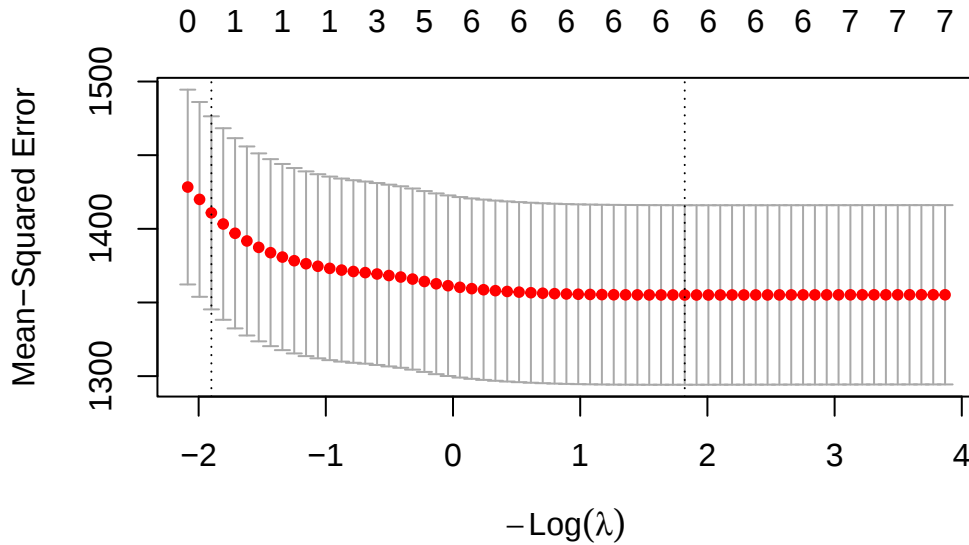


Figure 2: Cross-validated LASSO for LDL: mean squared error vs. $\log(\lambda)$. The left dashed line marks `lambda.min` (minimum CV error); the right dashed line marks `lambda.1se` (largest λ within 1 SE of the minimum).

The LASSO identifies `statins` and `HT` as the strongest predictors of LDL cholesterol in the HERS data, consistent with the biological knowledge encoded in the DAG in the HERS DAG⁶.

⁶[causal-inference.qmd#fig-hers-dag](#)

10 Some Details

10.1 Collinearity

Definition 10.1 (Collinearity). **Collinearity** refers to high correlation between predictors, sufficient to substantially degrade the precision of one or more regression coefficient estimates.

When two predictors are highly correlated, each appears unimportant in the model adjusting for the other, even if both are statistically significant in simpler models. The joint F-test for both predictors may be highly significant, while the individual t-tests are not.

How we handle collinearity depends on the inferential goal:

- **Prediction:** If including both collinear predictors reduces PE, include both. Cross-validation determines whether each contributes.
- **Evaluating a primary predictor:** If the primary predictor remains statistically significant after adjusting for a collinear confounder, the evidence for an independent effect is convincing. If collinearity with a *required* confounder inflates the primary predictor's standard error, we may need to accept reduced precision or present sensitivity analyses.
- **Collinearity among confounders:** Does not affect the precision of the primary predictor's estimate. Both variables can be retained if needed for confounding control.
- **Identifying multiple predictors:** Choose among collinear variables on substantive grounds (causal plausibility, measurement quality, fewer missing values).

10.1.1 Variance inflation factor

The **variance inflation factor (VIF)** for predictor j measures how much its variance is inflated by correlation with other predictors:

$$\text{VIF}_j \stackrel{\text{def}}{=} \frac{1}{1 - R_j^2}$$

where R_j^2 is the R^2 from regressing x_j on all other predictors.

A $\text{VIF} > 10$ (or equivalently, $R_j^2 > 0.9$) indicates problematic collinearity.

Exm

Example 10.1 (Variance inflation and standard error inflation). Suppose x_1 and x_2 are candidate predictors in a linear model. If x_1 and x_2 have correlation $r = 0.95$, then regressing x_1 on x_2 gives $R_1^2 = 0.95^2 = 0.9025$.

The variance inflation factor for x_1 is:

$$\text{VIF}_1 = \frac{1}{1 - 0.9025} = \frac{1}{0.0975} \approx 10.26$$

Because variance is inflated by a factor of 10.26, the standard error of $\hat{\beta}_1$ is increased by a factor of $\sqrt{10.26} \approx 3.20$ compared to an orthogonal design with $R_1^2 = 0$.

```
library(car)

model_full <- lm(
  LDL ~ HT + age + BMI + diabetes + smoking + statins + SBP,
  data = hers_ldl
)
vif(model_full)
#>      HT      age      BMI diabetes  smoking  statins      SBP
#> 1.00157 1.11561 1.12839 1.11188 1.07499 1.00800 1.05820
```

VIFs close to 1 indicate minimal collinearity. In the HERS LDL model, we expect relatively modest collinearity among the predictors, since they represent distinct pathways (see the DAG⁷).

10.2 Number of predictors

A widely used rule of thumb specifies a minimum number of observations (or events) per predictor:

- **Linear models:** at least 10 observations per predictor.
- **Logistic and Cox models:** at least 10 events per predictor (EPV).

This EPV guideline exists because with too many parameters relative to information in the data, coefficient estimates become imprecise and logistic/Cox models can behave poorly (e.g., converge to extreme estimates).

i The EPV rule is a flag, not a hard limit

The precision of estimates depends on more than just the EPV: residual variance, effect size, and correlations among predictors all matter. Adding covariates that are weakly correlated with the primary predictor but explain substantial outcome variance can *improve* precision. Conversely, adding a single collinear covariate can degrade it severely.

The rule is best used as a **warning flag**: if the EPV falls below 10, check whether the standard errors are unusually large, and whether Wald and likelihood ratio test p-values give consistent results. If problems are detected, tighten the backward selection criterion (e.g., to $p < 0.15$ or $p < 0.10$) and verify that a more parsimonious model gives consistent results.

10.3 Alternatives to backward selection

The main alternatives to backward selection are:

- **Best subsets:** screens all possible subsets of candidate predictors and selects the model minimizing a summary measure (adjusted R^2 , AIC, or a cross-validated PE measure). Computationally intensive for large sets of predictors.
- **Forward selection:** starts with an empty model and adds the predictor providing the greatest improvement at each step. More prone to missing *negatively confounded* sets of variables (sets where no member appears important unless all are included).
- **Stepwise selection:** augments forward selection by also allowing previously added variables to be removed. Better than pure forward but still misses some models that backward selection would find.
- **Bivariate screening:** evaluates each candidate predictor individually before multivariable modeling. Convenient but shares the limitation of forward selection.

i Why backward selection is preferred

Backward selection begins with the full set of candidates, so it is **less likely to miss negatively confounded sets** of variables — groups of variables that appear unimportant individually but are jointly important. Forward and stepwise selection will only include such sets if at least one member passes the inclusion criterion on its own.

10.4 Model selection complicates inference

When predictors are selected from the data, the p-values and confidence intervals computed as if the model were pre-specified are no longer valid.

Specific problems include:

⁷[causal-inference.qmd#fig-hers-dag](#)

- **Inflated type-I error:** searching over many models greatly increases the chance of at least one spurious finding.
- **Testimation bias:** selected predictors tend to have overstated estimated effects, because we select those that appear large by chance.
- **Biased CIs:** confidence intervals are too narrow because they do not account for model uncertainty.

How much these issues matter depends on the inferential goal:

- **Prediction** (Goal 1): Cross-validation of a PE measure protects against overfitting without relying on p-values, so inference is least affected.
- **Evaluating a primary predictor** (Goal 2): The primary predictor is included by default in all candidate models, so its estimate is relatively unaffected by selection. Post-selection inference is most problematic when the primary predictor is of borderline significance.
- **Identifying multiple predictors** (Goal 3): Selection most severely complicates inference here, since p-values for all retained predictors are of direct interest. Careful pre-specification, use of inclusive models, and the Allen–Cady procedure (see Section 6.2) reduce, but do not eliminate, this problem.

Best practices for reporting

When selection procedures are used:

1. Report the **full candidate predictor list**.
2. Report the **selection criterion and procedure** used.
3. Interpret **novel, weak, and borderline significant** findings with explicit caution.
4. Examine **alternative models** with different predictor subsets and report consistency of key results.

11 Summary

Predictor selection is the process of choosing appropriate variables for inclusion in a multivariable regression model. The right approach depends on the **inferential goal**:

For prediction (Goal 1):

- Pre-specify well-motivated candidate predictors.
- Use cross-validation of a target PE measure to select among models without overfitting.
- Shrinkage methods (LASSO, ridge) are effective when there are many candidates.

For evaluating a predictor of primary interest (Goal 2):

- Use a DAG to identify confounders, mediators, and colliders.
- Include all well-established confounders for face validity.
- Use backward selection with a liberal criterion ($p < 0.2$) to remove apparent non-confounders.
- In randomized trials, covariates are not required for confounding control but improve precision (and de-attenuate estimates in logistic/Cox models).
- Pre-specify adjusted analyses in trial protocols.

For identifying multiple important predictors (Goal 3):

- Apply the same inclusive strategy as Goal 2.
- Consider the Allen–Cady modified backward selection procedure to limit false-positive findings.
- Interpret novel, weak, and borderline significant associations cautiously; report all candidate predictors considered.

Across all goals:

- Collinearity between a primary predictor and a confounder can inflate standard errors; between adjustment variables, it does not.
- The EPV rule (10 events per predictor) flags potential problems but is not a hard limit.

- Model selection complicates inference: nominal p-values and confidence intervals are anticonservative when predictors have been selected from the data.

Learning Objectives

After completing this chapter, students should be able to:

1. **Identify the inferential goal** of a predictor selection problem (prediction, evaluating a primary predictor, or identifying multiple important predictors) and choose an appropriate strategy.
2. **Use a DAG** to identify confounders, mediators, and colliders, and determine which covariates to include or exclude.
3. **Apply backward selection with a liberal criterion** to select confounders in observational studies.
4. **Use AIC and BIC** to compare alternative models.
5. **Apply and interpret the LASSO**, including cross-validated selection of λ .
6. **Explain why stepwise selection is problematic** and describe its known pitfalls.
7. **Describe the bias–variance trade-off** and explain why cross-validation provides a less optimistic estimate of PE than naive within-sample measures.
8. **Identify and handle collinearity**, including using the VIF and understanding its implications for different inferential goals.
9. **Articulate how model selection complicates inference** and describe best practices for reporting in each inferential context.

References

- Allen, David M., and Frederick B. Cady. 1982. *Analyzing Experimental Data by Regression*. Lifetime Learning Publications.
- Greenland, Sander. 1989. “Modeling and Variable Selection in Epidemiologic Analysis.” *American Journal of Public Health* 79 (3): 340–49. <https://doi.org/10.2105/AJPH.79.3.340>.
- Grodstein, Francine, JoAnn E. Manson, Meir J. Stampfer, et al. 2001. “Postmenopausal Hormone Use and Secondary Prevention of Coronary Events in the Nurses’ Health Study: A Prospective, Observational Study.” *Annals of Internal Medicine* 135 (1): 1–8. <https://doi.org/10.7326/0003-4819-135-1-200107030-00007>.
- Heinze, Georg, Christine Wallisch, and Daniela Dunkler. 2018. “Variable Selection – A Review and Recommendations for the Practicing Statistician.” *Biometrical Journal* 60 (3): 431–49. <https://doi.org/10.1002/bimj.201700067>.
- Kleinbaum, David G, and Mitchel Klein. 2010. *Logistic Regression: A Self-Learning Text*. 3rd ed. Springer. <https://link.springer.com/book/10.1007/978-1-4419-1742-3>.
- Mickey, Ruth M., and Sander Greenland. 1989. “The Impact of Confounder Selection Criteria on Effect Estimation.” *American Journal of Epidemiology* 129 (1): 125–37. <https://doi.org/10.1093/oxfordjournals.aje.a115101>.
- Orwoll, Eric S., Douglas C. Bauer, Thomas M. Vogt, Kathleen M. Fox, and Study of Osteoporotic Fractures Research Group. 1996. “Axial Bone Mass in Older Women.” *Annals of Internal Medicine* 124 (3): 187–96.

- Rothman, Kenneth J., Timothy L. Lash, Tyler J. VanderWeele, and Sebastien Haneuse. 2021. *Modern Epidemiology*. Fourth edition. Wolters Kluwer.
- Vittinghoff, Eric, David V Glidden, Stephen C Shiboski, and Charles E McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. Springer. <https://doi.org/10.1007/978-1-4614-1353-0>.
- Vittinghoff, Eric, Michael G. Shlipak, Peter D. Varosy, et al. 2003. “Risk Factors and Secondary Prevention in Women with Heart Disease: The Heart and Estrogen/progestin Replacement Study.” *Annals of Internal Medicine* 139 (4): 274–81. <https://doi.org/10.7326/0003-4819-139-4-200308190-00006>.
- Walter, Louise C., Richard J. Brand, Steven R. Counsell, et al. 2001. “Development and Validation of a Prognostic Index for 1-Year Mortality in Older Adults After Hospitalization.” *JAMA* 285 (23): 2987–94. <https://doi.org/10.1001/jama.285.23.2987>.