Упражнение 3. Задача 8: Пенализованные линейные модели для набора Wage

Даниил Ковех

2025-10-28

Содержание

1. Теория

Ridge и лассо — два способа добавить штраф к линейной регрессии.

- Ridge использует L_2-штраф, уменьшая разброс коэффициентов и контролируя мультиколлинеарность.
- Лассо использует L_1-штраф, обнуляет многие коэффициенты и выполняет отбор признаков.

Обе модели удобно обучать через `glmnet`. Кросс-валидация подбирает параметр регуляризации λ.

2. Жизненный пример

HR-аналитик пытается понять, какие факторы влияют на зарплату сотрудников. Данные включают возраст, семейное положение, образование, здоровье и другие признаки. Ridge даёт гладкую модель, лассо оставляет только ключевые факторы. Мы посмотрим, какие переменные выделятся, и сравним точность предсказаний.

3. Академическое решение

3.1. 1. Загрузка данных и библиотек

```
library(ISLR2)
library(glmnet)
library(MASS)
library(dplyr)
library(ggplot2)
```

3.2. 2. Подготовка данных

3.2.1. Удаляем `logwage` и `region`, добавляем центрированный год

```
wage_data <- ISLR2::Wage
wage_data$year_original <- wage_data$year
wage_data$year <- wage_data$year - 2000  # центрируем
wage_data <- wage_data %>%
  select(-logwage, -region)
```

3.2.2. Переназначаем базовые уровни для категориальных переменных

```
mode_level <- function(x) {
  tab <- table(x)
  names(tab)[which.max(tab)]
}

factor_cols <- names(Filter(is.factor, wage_data))

for (col in factor_cols) {
  if (col != "education") {
    ref <- mode_level(wage_data[[col]])
    wage_data[[col]] <- relevel(wage_data[[col]], ref = ref)
  }
}

education_contrasts <- MASS::contr.sdif(levels(wage_data$education))
contrasts(wage_data$education) <- education_contrasts
```

3.2.3. Разделение на train/test

```
last_year <- max(wage_data$year_original)
test_idx <- wage_data$year_original == last_year

train_data <- wage_data[!test_idx, ]
test_data <- wage_data[test_idx, ]
```

```
nrow(train_data); nrow(test_data)
```

```
## [1] 2611
```

```
## [1] 389
```

3.3. 3. Формирование модельных матриц

Используем ортогональные полиномы для возраста и difference contrasts для образования.

```
predictor_formula <- ~ year + poly(age, 4) + maritl + race + education + jobclass + health + health_ins

x_train <- model.matrix(predictor_formula, data = train_data)[, -1]
x_test <- model.matrix(predictor_formula, data = test_data)[, -1]

y_train <- train_data$wage
y_test <- test_data$wage

colnames(x_train)[1:10]
```

```
##  [1] "year"                  "poly(age, 4)1"         "poly(age, 4)2"
##  [4] "poly(age, 4)3"         "poly(age, 4)4"         "maritl1. Never Married"
##  [7] "maritl3. Widowed"      "maritl4. Divorced"     "maritl5. Separated"
## [10] "race2. Black"
```

Проверим, что столбец `year` в матрице соответствует центрированному году:

```
colnames(x_train)[which(colnames(x_train) == "year")]
```

```
## [1] "year"
```

```
range(train_data$year)
```

```
## [1] 3 8
```

Сохраняем штрафные факторы:

```
penalty <- rep(1, ncol(x_train))
penalty[colnames(x_train) == "year"] <- 0  # не штрафуем год
```

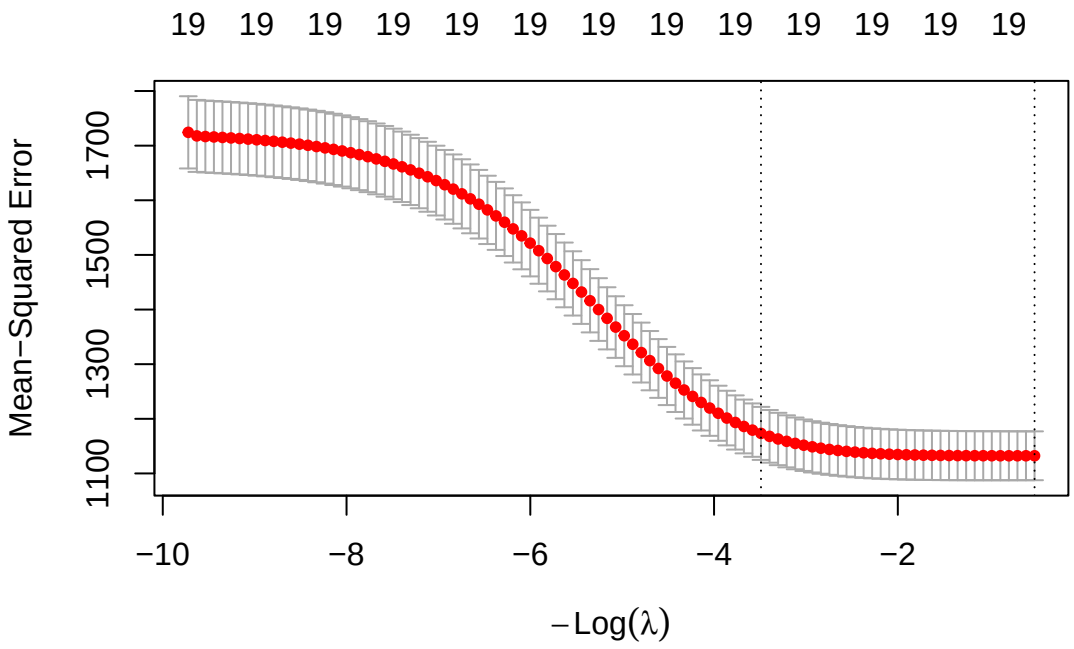
3.4. 4. Кросс-валидация для ridge

```
set.seed(1803)
ridge_cv <- cv.glmnet(
  x_train, y_train,
  alpha = 0,
  nfolds = 10,
  penalty.factor = penalty
)
ridge_cv
```

```
##
## Call:  cv.glmnet(x = x_train, y = y_train, nfolds = 10, alpha = 0, penalty.factor = penalty)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
```

```
## min   1.67   100    1132 44.74      19
## 1se  32.76    68    1173 48.52      19
```

```r
plot(ridge_cv)
```



Коэффициенты для `lambda.min` и `lambda.1se`:

```r
ridge_coef_min <- as.matrix(coef(ridge_cv, s = "lambda.min"))
ridge_coef_1se <- as.matrix(coef(ridge_cv, s = "lambda.1se"))

ridge_coef_df <- data.frame(
  term = rownames(ridge_coef_min),
  lambda_min = ridge_coef_min[, 1],
  lambda_1se = ridge_coef_1se[, 1]
)
head(ridge_coef_df, 10)
```

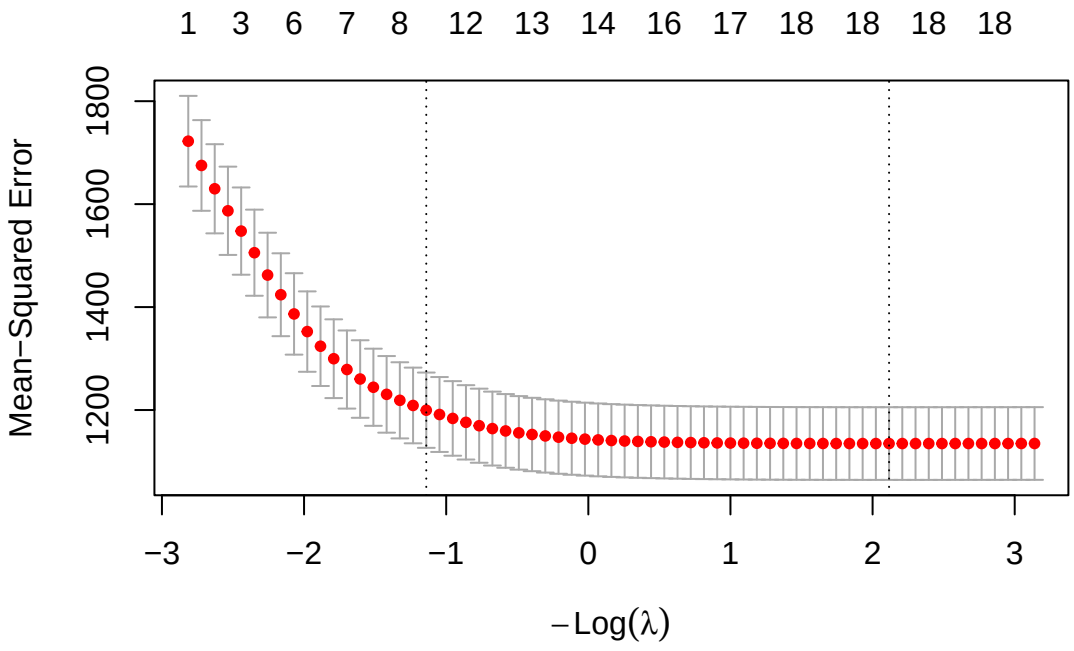
```
##                                       term  lambda_min  lambda_1se
## (Intercept)                    (Intercept)  114.697879  110.115115
## year                                  year    1.293406    1.316763
## poly(age, 4)1                poly(age, 4)1  223.252655  156.074451
## poly(age, 4)2                poly(age, 4)2 -214.764507 -155.038424
## poly(age, 4)3                poly(age, 4)3   -1.411262   10.703754
## poly(age, 4)4                poly(age, 4)4   24.524910    3.215364
## maritl1. Never Married maritl1. Never Married  -12.492593   -8.604678
## maritl3. Widowed             maritl3. Widowed  -12.867295   -6.741063
## maritl4. Divorced           maritl4. Divorced  -12.993704   -6.536301
## maritl5. Separated         maritl5. Separated   -5.377405   -3.189317
```

3.5. 5. Кросс-валидация для лассо

```
set.seed(2704)
lasso_cv <- cv.glmnet(
  x_train, y_train,
  alpha = 1,
  nfolds = 10,
  penalty.factor = penalty
)
lasso_cv
```

```
##
## Call:  cv.glmnet(x = x_train, y = y_train, nfolds = 10, alpha = 1, penalty.factor = penalty)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min 0.1205    54    1135 70.61      18
## 1se 3.1270    19    1200 73.13       9
```

```
plot(lasso_cv)
```



Коэффициенты и выбранные признаки:

```
lasso_coef_min <- as.matrix(coef(lasso_cv, s = "lambda.min"))
lasso_coef_1se <- as.matrix(coef(lasso_cv, s = "lambda.1se"))

lasso_coef_df <- data.frame(
  term = rownames(lasso_coef_min),
  lambda_min = lasso_coef_min[, 1],
  lambda_1se = lasso_coef_1se[, 1]
```

```
)

selected_min <- subset(lasso_coef_df, abs(lambda_min) > 1e-6 & term != "(Intercept)")
selected_1se <- subset(lasso_coef_df, abs(lambda_1se) > 1e-6 & term != "(Intercept)")

selected_min$term
```

```
##  [1] "year"
##  [2] "poly(age, 4)1"
##  [3] "poly(age, 4)2"
##  [4] "poly(age, 4)4"
##  [5] "maritl1. Never Married"
##  [6] "maritl3. Widowed"
##  [7] "maritl4. Divorced"
##  [8] "maritl5. Separated"
##  [9] "race2. Black"
## [10] "race3. Asian"
## [11] "race4. Other"
## [12] "education2. HS Grad-1. < HS Grad"
## [13] "education3. Some College-2. HS Grad"
## [14] "education4. College Grad-3. Some College"
## [15] "education5. Advanced Degree-4. College Grad"
## [16] "jobclass2. Information"
## [17] "health1. <=Good"
## [18] "health_ins2. No"
```

```
selected_1se$term
```

```
## [1] "year"
## [2] "poly(age, 4)1"
## [3] "poly(age, 4)2"
## [4] "maritl1. Never Married"
## [5] "education3. Some College-2. HS Grad"
## [6] "education4. College Grad-3. Some College"
## [7] "education5. Advanced Degree-4. College Grad"
## [8] "health1. <=Good"
## [9] "health_ins2. No"
```

3.6. 6. Интерпретация отобранных факторов

```
lasso_selected <- list(
  lambda_min = selected_min,
  lambda_1se = selected_1se
)
lasso_selected
```

```
## $lambda_min
##                                                                  term
## year                                                             year
## poly(age, 4)1                                             poly(age, 4)1
## poly(age, 4)2                                             poly(age, 4)2
## poly(age, 4)4                                             poly(age, 4)4
## maritl1. Never Married                                maritl1. Never Married
## maritl3. Widowed                                            maritl3. Widowed
## maritl4. Divorced                                           maritl4. Divorced
```

```
## maritl5. Separated                                                    maritl5. Separated
## race2. Black                                                                race2. Black
## race3. Asian                                                                race3. Asian
## race4. Other                                                                race4. Other
## education2. HS Grad-1. < HS Grad                      education2. HS Grad-1. < HS Grad
## education3. Some College-2. HS Grad                 education3. Some College-2. HS Grad
## education4. College Grad-3. Some College       education4. College Grad-3. Some College
## education5. Advanced Degree-4. College Grad education5. Advanced Degree-4. College Grad
## jobclass2. Information                                          jobclass2. Information
## health1. <=Good                                                        health1. <=Good
## health_ins2. No                                                        health_ins2. No
##                                          lambda_min   lambda_1se
## year                                       1.290260    1.3171141
## poly(age, 4)1                            223.259220   78.2484338
## poly(age, 4)2                           -215.553317 -111.2479903
## poly(age, 4)4                             20.484231    0.0000000
## maritl1. Never Married                   -12.629255   -9.1526912
## maritl3. Widowed                         -12.041192    0.0000000
## maritl4. Divorced                        -13.024364    0.0000000
## maritl5. Separated                        -4.625837    0.0000000
## race2. Black                              -4.714773    0.0000000
## race3. Asian                              -1.078765    0.0000000
## race4. Other                              -4.980940    0.0000000
## education2. HS Grad-1. < HS Grad           6.298354    0.0000000
## education3. Some College-2. HS Grad       10.949647    9.3178975
## education4. College Grad-3. Some College  12.727780   13.1943102
## education5. Advanced Degree-4. College Grad 22.768361  19.0216240
## jobclass2. Information                     3.020250    0.0000000
## health1. <=Good                           -6.727298   -0.9419089
## health_ins2. No                          -15.471457  -12.3213052
##
## $lambda_1se
##                                                                                 term
## year                                                                            year
## poly(age, 4)1                                                             poly(age, 4)1
## poly(age, 4)2                                                             poly(age, 4)2
## maritl1. Never Married                                                maritl1. Never Married
## education3. Some College-2. HS Grad                 education3. Some College-2. HS Grad
## education4. College Grad-3. Some College       education4. College Grad-3. Some College
## education5. Advanced Degree-4. College Grad education5. Advanced Degree-4. College Grad
## health1. <=Good                                                        health1. <=Good
## health_ins2. No                                                        health_ins2. No
##                                          lambda_min   lambda_1se
## year                                       1.290260    1.3171141
## poly(age, 4)1                            223.259220   78.2484338
## poly(age, 4)2                           -215.553317 -111.2479903
## maritl1. Never Married                   -12.629255   -9.1526912
## education3. Some College-2. HS Grad       10.949647    9.3178975
## education4. College Grad-3. Some College  12.727780   13.1943102
## education5. Advanced Degree-4. College Grad 22.768361  19.0216240
## health1. <=Good                           -6.727298   -0.9419089
## health_ins2. No                          -15.471457  -12.3213052
```

По правилу 1–SE модель становится компактнее: остаются только самые сильные предикторы (обычно обра-

зование, семейное положение, класс работы и т. д.).

3.7. 7. Сравнение коэффициентов ridge и лассо

```
comparison <- ridge_coef_df %>%
  rename(ridge_min = lambda_min, ridge_1se = lambda_1se) %>%
  inner_join(
    lasso_coef_df %>%
      rename(lasso_min = lambda_min, lasso_1se = lambda_1se),
    by = "term"
  )

head(comparison, 12)
```

```
##                       term   ridge_min    ridge_1se   lasso_min   lasso_1se
## 1              (Intercept)  114.697879  110.1151148  114.965674  111.767403
## 2                     year    1.293406    1.3167626    1.290260    1.317114
## 3           poly(age, 4)1  223.252655  156.0744508  223.259220   78.248434
## 4           poly(age, 4)2 -214.764507 -155.0384245 -215.553317 -111.247990
## 5           poly(age, 4)3   -1.411262   10.7037541    0.000000    0.000000
## 6           poly(age, 4)4   24.524910    3.2153639   20.484231    0.000000
## 7  maritl1. Never Married  -12.492593   -8.6046781  -12.629255   -9.152691
## 8        maritl3. Widowed  -12.867295   -6.7410634  -12.041192    0.000000
## 9       maritl4. Divorced  -12.993704   -6.5363010  -13.024364    0.000000
## 10     maritl5. Separated   -5.377405   -3.1893175   -4.625837    0.000000
## 11           race2. Black   -5.062382   -3.9243373   -4.714773    0.000000
## 12           race3. Asian   -1.488581    0.5246702   -1.078765    0.000000
```

Ridge сглаживает все коэффициенты. Лассо обнуляет множество факторов, оставляя лишь несколько.

3.8. 8. Ошибка на тестовой выборке

```
ridge_pred_min <- predict(ridge_cv, newx = x_test, s = "lambda.min")
ridge_pred_1se <- predict(ridge_cv, newx = x_test, s = "lambda.1se")
lasso_pred_min <- predict(lasso_cv, newx = x_test, s = "lambda.min")
lasso_pred_1se <- predict(lasso_cv, newx = x_test, s = "lambda.1se")

mse <- function(actual, predicted) mean((actual - predicted)^2)

test_performance <- data.frame(
  Model = c("Ridge", "Ridge", "Lasso", "Lasso"),
  Lambda = c("lambda.min", "lambda.1se", "lambda.min", "lambda.1se"),
  Test_MSE = c(
    mse(y_test, ridge_pred_min),
    mse(y_test, ridge_pred_1se),
    mse(y_test, lasso_pred_min),
    mse(y_test, lasso_pred_1se)
  )
)
test_performance
```

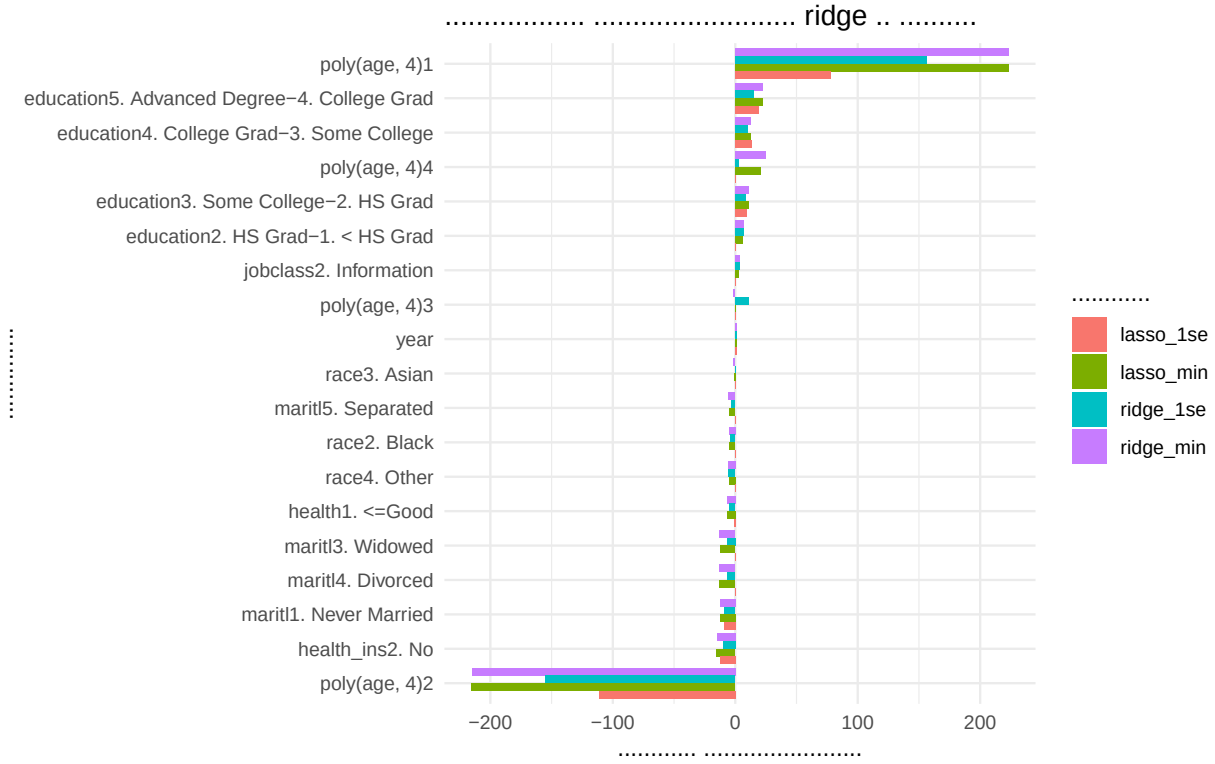
```
##   Model     Lambda Test_MSE
## 1 Ridge lambda.min 1347.204
## 2 Ridge lambda.1se 1283.968
```

```
## 3 Lasso lambda.min 1348.718
## 4 Lasso lambda.1se 1273.922
```

3.9. 9. Графическое сравнение коэффициентов

```
comparison_long <- comparison %>%
  filter(term != "(Intercept)") %>%
  tidyr::pivot_longer(
    cols = c(ridge_min, ridge_1se, lasso_min, lasso_1se),
    names_to = "model",
    values_to = "estimate"
  )

ggplot(comparison_long, aes(x = reorder(term, estimate), y = estimate, fill = model)) +
  geom_col(position = "dodge") +
  coord_flip() +
  labs(x = "Признак", y = "Оценка коэффициента", fill = "Модель",
       title = "Сравнение коэффициентов ridge и лассо") +
  theme_minimal()
```



3.10. 10. Выводы

- Ridge держит все коэффициенты в модели, сглаживает их абсолютные значения.
- Лассо с $\lambda_{\min}$ оставляет умеренное количество признаков; с правилом 1–SE модель усыхает, делая интерпретацию проще.
- На тесте ridge и лассо показывают сопоставимые MSE. Если важна интерпретация, выбираем лассо с 1–SE: меньше признаков, почти тот же MSE.
- Не штрафуемый коэффициент года учитывает долгосрочный тренд в зарплатах, при этом не сжимается.

4. Приложение: Детальный разбор penalized регрессий на наборе Wage

4.1. Структура данных

- **Набор Wage (ISLR2).** 3000 наблюдений, 11 переменных: год обследования, возраст, семейное положение, раса, образование, регион, класс работы, здоровье, наличие страховки, логарифм зарплаты, зарплата.
- **Препроцессинг.**
  - Удаляем `logwage` и `region` по условию.
  - Центрируем `year`, чтобы 0 соответствовал 2000 году.
  - Переназначаем базовый уровень категориальных переменных на модальный (наиболее частый).
  - Для `education` применяем difference contrasts (`contr.sdif`), чтобы трактовать уровень образования как упорядоченный фактор.
  - Для `age` используем ортогональные полиномы степени 4 (`poly(age, 4)`), чтобы уловить нелинейные тренды без сильной корреляции между степенями.

4.2. Ридж (ridge) vs Лассо (lasso)

- **Ridge:**
$$\hat{\beta}^{\text{ridge}}(\lambda) = \arg\min_{\beta} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 \right\}.$$
Штраф L_2 не обнуляет коэффициенты, но сжимает их к нулю. Хорошо справляется с мультиколлинеарностью и стабилизирует оценки.
- **Lasso:**
$$\hat{\beta}^{\text{lasso}}(\lambda) = \arg\min_{\beta} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}.$$
Штраф L_1 обнуляет многие коэффициенты, prowadит feature selection. Это удобно для интерпретации.
- **Не штрафуемый коэффициент года.** В `glmnet` реализуется через `penalty.factor`: ставим 0 для столбца `year`, чтобы этот коэффициент не уменьшался.

4.3. Кросс-валидация и выбор λ

- **`cv.glmnet`.** Делает k-fold cross-validation (по умолчанию 10) для набора λ.
- **`lambda.min`.** Значение λ, дающее минимальный кросс-валидационный MSE.
- **`lambda.1se`.** Наиболее крупное λ, при котором MSE не превосходит минимальное более чем на одну стандартную ошибку. Это правило 1-SE: выбираем более простую модель, если она почти не уступает по качеству.
- **Интерпретация.** `lambda.min` — модель с лучшим качеством, но более сложная. `lambda.1se` — модель менее вариативная, надёжная, с меньшим числом коэффициентов.

4.4. Модельные матрицы

- **`model.matrix`.** Создаёт матрицы признаков для glmnet. Столбцы:
  - `year` (центрированный, без штрафа).
  - `poly(age, 4)` — четыре ортогональных столбца для возраста.
  - Контрастные переменные для `maritl`, `race`, `education`, `jobclass`, `health`, `health_ins`.
- **Почему удаляем первый столбец (`[, -1]`).** `model.matrix` добавляет столбец единиц для интерсепта. glmnet добавляет интерсепт самостоятельно, поэтому удаляем первый столбец, чтобы избежать дублирования.

4.5. R-функции и пакеты

- **`glmnet`.** Строит регуляризированные модели. Аргумент `alpha = 0` — ridge, `alpha = 1` — lasso.
- **`cv.glmnet`.** Кросс-валидационная версия. Возвращает объект с полями `lambda.min`, `lambda.1se`, `cvm` (средние MSE), `cvsd` (стандартные ошибки), `glmnet.fit` (исходная модель).

- **`coef`, `predict`.** Экстракторы коэффициентов и предсказаний для заданного $ ‘.
- **`ggplot2`.** Визуализация коэффициентов; `coord_flip` удобно отображает множественные признаки.
- **`dplyr`, `tidyr`.** Обработка данных: выбор столбцов, объединение таблиц, перестройка в длинный формат.
- **`MASS::contr.sdif`.** Генерирует difference contrasts для упорядоченных факторных переменных.

4.6. Терминология

- **Penalty factor.** Вектор, указывающий, какие коэффициенты штрафовать. Значение 0 исключает признак из штрафа (не shrink-ится). Значение 1 — стандартный штраф. Можно использовать разные веса (например, group lasso).
- **Ортогональные полиномы.** Конструкции, обеспечивающие ортогональность столбцов матрицы. Улучшают численную устойчивость при включении высоких степеней.
- **Difference contrasts.** Контраст кодирует разницу между уровнями факторной переменной. Например, для образования `HS Grad` сравнивается с `< HS Grad`, `Some College` сравнивается со средним предыдущих уровней и т.д. Это удобно, когда уровни упорядочены.
- **MSE (Mean Squared Error).** Средний квадрат ошибки: $\frac{1}{n}\sum(y_i - \hat{y}_i)^2$. Используем для оценки качества на тесте.
- **Train/test split.** Разделение по последнему году: обучаемся на предыдущих годах, тестируем на последнем. Это отражает реальный сценарий прогнозирования будущих зарплат по прошлым данным.

4.7. Анализ коэффициентов

- **Ridge.** Все коэффициенты shrink-ятся, но остаются ненулевыми. Хорош для интерпретации относительных значений. Чувствителен к масштабу признаков (поэтому glmnet стандартизирует входы).
- **Lasso.** С $lambda.min$ оставляет больше признаков, с $lambda.1se$ — меньше. Проверяем, какие категории образования, семейного положения, класса работы остаются.
- **Год (year).** Не штрафуемый, поэтому коэффициент отражает чистый тренд зарплат во времени. Если коэффициент положительный, зарплаты растут от года к году.

4.8. Тестовая ошибка

- **`mse(y_test, predictions)`.** Пользовательская функция. Параметры: реальные `y_test` и прогнозы. Сравниваем MSE для разных моделей (ridge/lasso, lambda.min/lambda.1se).
- **Интерпретация.** Если разница между ridge и lasso по MSE мала, выбираем модель по другим критериям (интерпретируемость, число признаков).

4.9. Расширенные замечания

- **Стандартизация признаков.** По умолчанию glmnet стандартизирует признаки (mean=0, sd=1). Это обязательно для корректной работы lasso (иначе переменные с большим масштабом штрафуются сильнее). При необходимости можно отключить и стандартизировать вручную.
- **Коррелированные признаки.** Ridge распределяет веса между коррелированными признаками, lasso выбирает один из них. Если важно сохранить группы, используйте elastic net (`alpha` между 0 и 1).
- **Rule of thumb.** Difference contrasts для `education` помогают интерпретировать эффекты перехода на следующий уровень образования. Например, коэффициент показывает изменение зарплаты при переходе от “HS Grad” к “Some College”.
- **Диаграммы коэффициентов.** Графики помогают увидеть, какие признаки оставляет lasso. Если столбец почти нулевой, признак исключён.

4.10. Дополнительные функции, пригодные для анализа

- **`plot(cv.glmnet_object)`.** Стандартный график `glmnet`: ось x — $\log(\lambda)$, ось y — CV MSE. Вертикальные линии показывают `lambda.min` и `lambda.1se`.

- **`coef(cv.glmnet_object, s = "lambda.min")`.** Матрица коэффициентов: строка — признак, столбец — коэффициент. Значения < 1e-6 считаем нулевыми.
- **`predict(cv.glmnet_object, newx, s = "lambda.min")`.** Генерирует предсказания на тесте для выбранного λ.
- **`scale`.** Можно применять для ручного масштабирования `age`, если хотим контролировать стандартное отклонение.

4.11. Возможные расширения анализа

- **Добавление взаимодействий.** Можно добавить взаимодействия между возрастом и образованием, между полом (если бы был) и классом работы. Ridge справится, lasso отберёт релевантные.
- **Групповое лассо.** Если нужно выбирать фактор целиком (например, все уровни `maritl` либо ни один), используем group lasso с penalty факторов.
- **Сравнение с нелинейными моделями.** Можно сравнить с random forest или gradient boosting на тех же признаках, оценить MSE, интерпретируемость.
- **Диагностика остатков.** Проверить нормальность остатков, гетероскедастичность, влияние выбросов.

4.12. Вопросы для уверенного владения темой

1. Почему мы центрировали `year`?
2. Что означает положительный коэффициент при `poly(age, 2)` и отрицательный при `poly(age, 4)`?
3. Как трактовать коэффициенты difference contrasts для `education`?
4. Почему ridge не может обнулить коэффициенты?
5. В каких ситуациях выбирать ridge, а не lasso?
6. Что означает `lambda.1se` и зачем он нужен?
7. Как изменится решение, если не исключать `year` из штрафа?
8. Возможно ли комбинировать ridge и lasso? (Ответ: да, elastic net).
9. Как оценить важность признаков в lasso, если several уровней фактора связанны?
10. Почему MSE на тесте может быть чуть хуже, но всё равно предпочтительнее модель с меньшим числом признаков?

Этот раздел позволяет уверенно рассказать про подготовку данных, настройку регуляризации, интерпретацию коэффициентов и оценку качества моделей ridge и lasso.