STAC67: Regression Analysis

Lecture 22

Thibault Randrianarisoa

November 27, 2026

University of Toronto Scarborough

Announcements

- Project: report, R Markdown file, dataset, contribution statement and slides (PDF), on Quercus by Tuesday 1 December, 23:59. No late submission.
- Presentations, in this room: Wednesday 2 December from 16:10, and Friday 4 December from 13:10. Both sessions are class time: everyone attends both.
- Final exam: Thursday 17 December, 9:00 to 12:00, IA 1150. It covers the whole term, today included.

Review of Wednesday

- Unequal variances, $\epsilon_i \sim N(0, \sigma_i^2)$: least squares stays unbiased, but its standard errors, tests and intervals are wrong.
- Weighted least squares: $w_i = 1/\sigma_i^2$, minimize $Q_w = \sum_i w_i \left(Y_i - \beta_0 - \beta_1 X_{i1} - \dots\right)^2$, so $\hat{\boldsymbol{\beta}}_w = (\boldsymbol{X'WX})^{-1}\boldsymbol{X'WY}$: least squares on the data multiplied by $\sqrt{w_i}$.
- Known up to a constant: $\sigma_i^2 = \sigma^2/w_i$. For means of n_i measures, $w_i = n_i$, and $MSE_w = \sum_i w_i e_i^2/(n-p)$ estimates σ^2.

Today: end of weighted linear regression, then shrinkage: ridge regression and the LASSO.

Weighted least squares (continued)

Variances estimated from the residuals

Usually the σ_i^2 are unknown. So we estimate how the spread changes, from a first fit:

1. Fit by ordinary least squares, and plot the residuals.
2. A funnel: regress $|e_i|$ on the predictor (or on $\hat{Y}_i$). The fitted values $\hat{s}_i$ estimate σ_i: the standard deviation function. Or regress e_i^2, for the variance function $\hat{v}_i$.
3. Take $w_i = 1/\hat{s}_i^2$ (or $1/\hat{v}_i$).
4. Refit with these weights.
5. If the coefficients changed a lot, repeat steps 2 to 4 with the new residuals.

Variances estimated from the residuals

Usually the σ_i^2 are unknown. So we estimate how the spread changes, from a first fit:

1. Fit by ordinary least squares, and plot the residuals.
2. A funnel: regress $|e_i|$ on the predictor (or on $\hat{Y}_i$). The fitted values $\hat{s}_i$ estimate σ_i: the standard deviation function. Or regress e_i^2, for the variance function $\hat{v}_i$.
3. Take $w_i = 1/\hat{s}_i^2$ (or $1/\hat{v}_i$).
4. Refit with these weights.
5. If the coefficients changed a lot, repeat steps 2 to 4 with the new residuals.

The weights are estimated, so the tests and intervals are approximate: fine unless n is small. Check that every $\hat{s}_i$ is positive.

Blood pressure: in R

The 54 women of Wednesday, in the data frame bp: Y is the diastolic blood pressure (mm Hg), X the age (years).

```
fit  <- lm(Y ~ X, data = bp)
# the standard deviation function, then the weights
sfit <- lm(abs(resid(fit)) ~ X, data = bp)
fitw <- lm(Y ~ X, data = bp, weights = 1 / fitted(sfit)^2)
```

Blood pressure: in R

The 54 women of Wednesday, in the data frame `bp`: Y is the diastolic blood pressure (mm Hg), X the age (years).

```
fit  <- lm(Y ~ X, data = bp)
# the standard deviation function, then the weights
sfit <- lm(abs(resid(fit)) ~ X, data = bp)
fitw <- lm(Y ~ X, data = bp, weights = 1 / fitted(sfit)^2)
```

$\hat{s} = -1.55 + 0.198\,X$: from 2.4 at age 20 to 10.1 at age 59. An older person gets about $(10.1/2.4)^2 \approx 18$ times less weight.

Blood pressure: in R

The 54 women of Wednesday, in the data frame `bp`: Y is the diastolic blood pressure (mm Hg), X the age (years).

```
fit  <- lm(Y ~ X, data = bp)
# the standard deviation function, then the weights
sfit <- lm(abs(resid(fit)) ~ X, data = bp)
fitw <- lm(Y ~ X, data = bp, weights = 1 / fitted(sfit)^2)
```

$\hat{s} = -1.55 + 0.198\,X$: from 2.4 at age 20 to 10.1 at age 59. An older person gets about $(10.1/2.4)^2 \approx 18$ times less weight.

	$\hat{\beta}_1$	$s\{\hat{\beta}_1\}$	95% interval for β_1
Ordinary	0.580	0.0970	(0.385, 0.775)
Weighted	0.596	0.0792	(0.437, 0.755)

Blood pressure: in R

The 54 women of Wednesday, in the data frame `bp`: Y is the diastolic blood pressure (mm Hg), X the age (years).

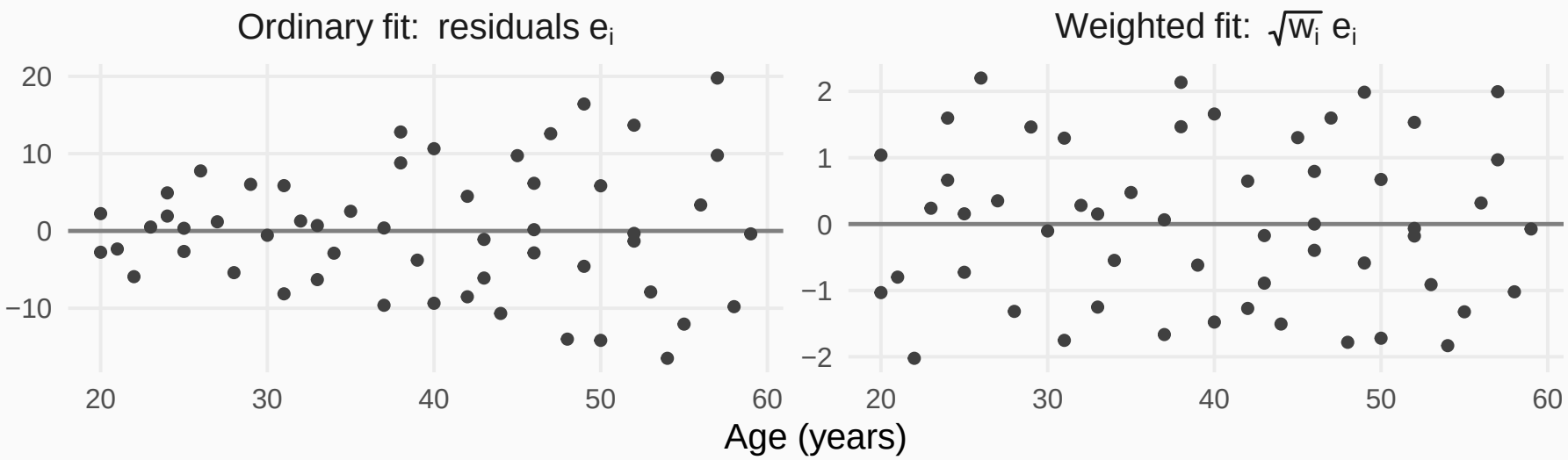
```
fit  <- lm(Y ~ X, data = bp)
# the standard deviation function, then the weights
sfit <- lm(abs(resid(fit)) ~ X, data = bp)
fitw <- lm(Y ~ X, data = bp, weights = 1 / fitted(sfit)^2)
```

$\hat{s} = -1.55 + 0.198\,X$: from 2.4 at age 20 to 10.1 at age 59. An older person gets about $(10.1/2.4)^2 \approx 18$ times less weight.

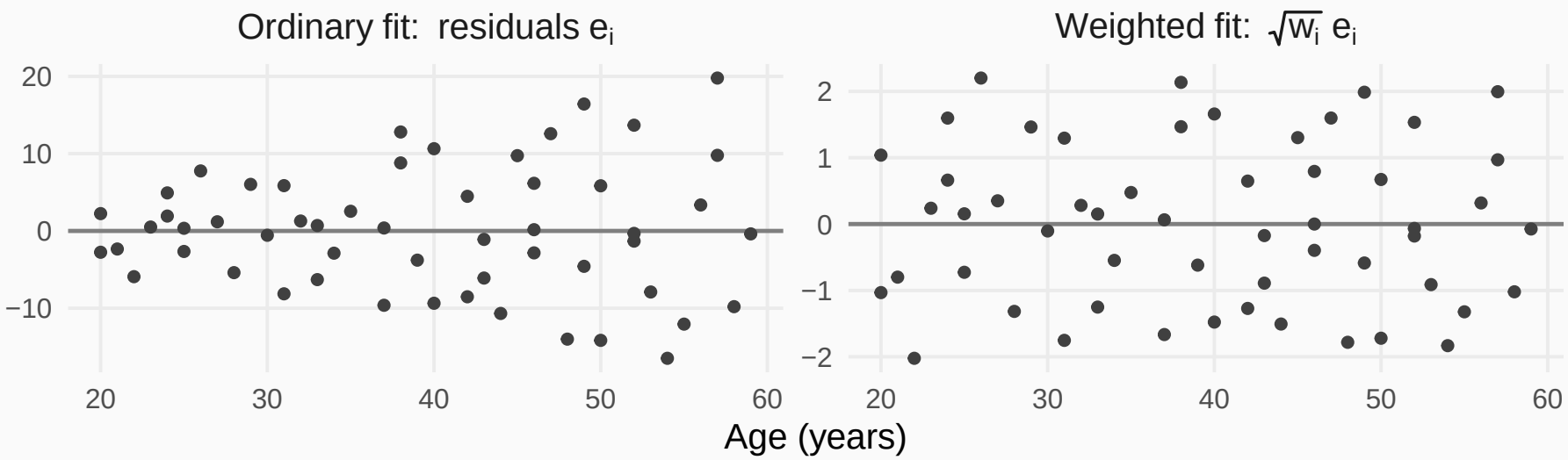
	$\hat{\beta}_1$	$s\{\hat{\beta}_1\}$	95% interval for β_1
Ordinary	0.580	0.0970	(0.385, 0.775)
Weighted	0.596	0.0792	(0.437, 0.755)

The slope barely moves; the interval is narrower, and now valid: the mean blood pressure rises by 0.44 to 0.76 mm Hg per year of age. A second pass gives $\hat{\beta}_1 = 0.5964$: no need to iterate.

Check again


Ordinary fit: residuals e_i
Weighted fit: $\sqrt{w_i}\, e_i$
20
10
0
−10
2
1
0
−1
−2
20
30
40
50
60
Age (years)

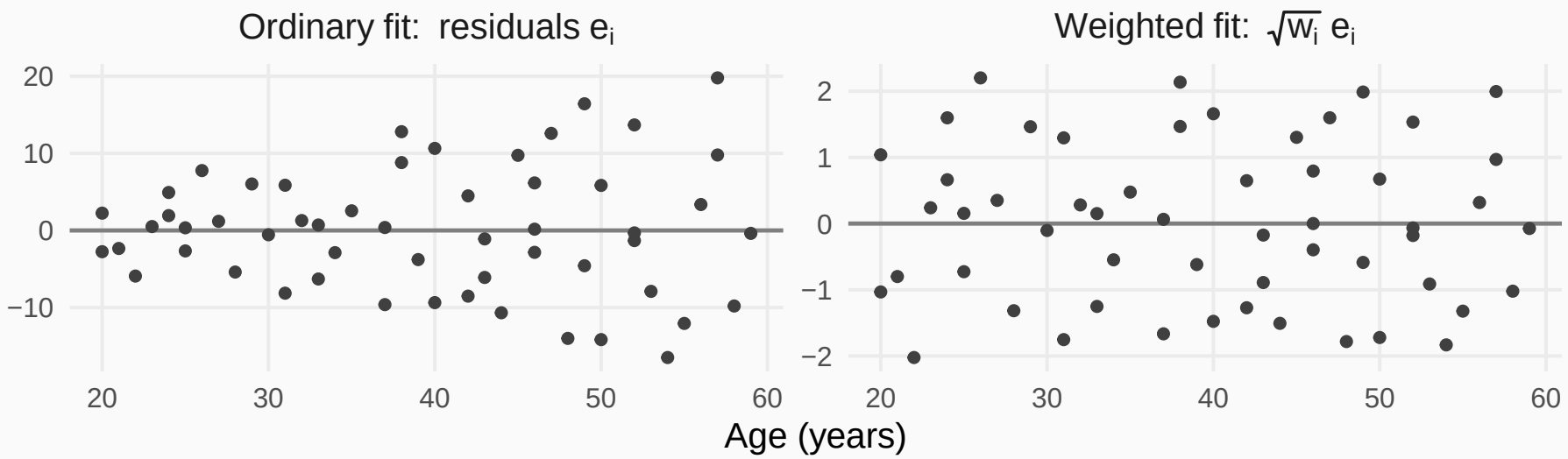
Check again



The weighted residuals $\sqrt{w_i}\, e_i$ have a constant spread. Brown-Forsythe on them: $t_{BF} = -0.18$, no evidence left.

For the project: the rubric's Validation criterion asks for remedial measures "taken where needed and checked again".

Check again



The weighted residuals $\sqrt{w_i}\, e_i$ have a constant spread. Brown-Forsythe on them: $t_{BF} = -0.18$, no evidence left.

For the project: the rubric's Validation criterion asks for remedial measures "taken where needed and checked again".

⚠ Careful

`resid(fitw)` returns $e_i = Y_i - \hat{Y}_i$, which still fan out. Plot `weighted.residuals(fitw)` or `rstandard(fitw)`, which use $\sqrt{w_i}\, e_i$; `plot(fitw)` uses them too.

What weighting changes, and what it does not

- Unchanged: the response and its units, the regression function, how a coefficient is read.
- Changed: the estimates, a little; the standard errors, tests and intervals, which become valid.
- The R^2 of a weighted fit has no simple meaning (here 0.52 against 0.41): do not compare it with the ordinary fit's.

What weighting changes, and what it does not

- Unchanged: the response and its units, the regression function, how a coefficient is read.
- Changed: the estimates, a little; the standard errors, tests and intervals, which become valid.
- The R^2 of a weighted fit has no simple meaning (here 0.52 against 0.41): do not compare it with the ordinary fit's.

Weights or a transformation?

- the mean looks right in the units of Y, and only the spread changes: weights;
- the mean is curved too, or the errors are skewed: transform Y (Lecture 18), then check again.

What weighting changes, and what it does not

- Unchanged: the response and its units, the regression function, how a coefficient is read.
- Changed: the estimates, a little; the standard errors, tests and intervals, which become valid.
- The R^2 of a weighted fit has no simple meaning (here 0.52 against 0.41): do not compare it with the ordinary fit's.

Weights or a transformation?

- the mean looks right in the units of Y, and only the spread changes: weights;
- the mean is curved too, or the errors are skewed: transform Y (Lecture 18), then check again.

For the project: weighted least squares is now a tool from the course. If you use it, show the standard deviation function and the weighted residuals.

From question to final model

The report, step by step

Step	What the reader should find	Tools
Question	What you explain or predict, and why it matters	
Description	Only what the model uses: the response, its relations with the predictors, unusual cases, data issues	Scatterplots, boxplots, correlations
Model	Predictors considered (following from the question): factors, transformations, interactions, all justified	Lectures 8, 15 to 18
Selection	How you reached the final model	R^2_a, C_p, AIC, BIC, PRESS, `step()`
Validation	Residuals (L, I, N, E); outliers and influential points; multicollinearity; remedies checked again	Residual plots; t_i, h_{ii}, DFFITS, Cook's D, DFBETAS; VIF
Interpretation	Coefficients and intervals in context; the answer, and its limits	Lectures 15 to 18

Ridge regression

A biased estimator can be better

To judge an estimator $\hat{\theta}$ of θ, measure how far it falls from θ, on average:

$$E\left[(\hat{\theta} - \theta)^2\right] = Var(\hat{\theta}) + \left[E(\hat{\theta}) - \theta\right]^2 = \text{variance} + \text{bias}^2.$$

This is the mean squared error of an estimator, not the MSE of a fit.

A biased estimator can be better

To judge an estimator $\hat{\theta}$ of θ, measure how far it falls from θ, on average:

$$E\left[(\hat{\theta}-\theta)^2\right] = Var(\hat{\theta}) + \left[E(\hat{\theta})-\theta\right]^2 = \text{variance} + \text{bias}^2.$$

This is the mean squared error of an estimator, not the MSE of a fit.

Least squares is unbiased, with the smallest variance among the linear unbiased estimators (Gauss-Markov, Lecture 3). But with multicollinearity, that smallest variance is large (Lecture 20):

$$Var(\hat{\beta}_k) = VIF_k \times \frac{\sigma^2}{\sum_i (X_{ik}-\bar{X}_k)^2}.$$

A biased estimator can be better

To judge an estimator $\hat{\theta}$ of θ, measure how far it falls from θ, on average:

$$E\left[(\hat{\theta}-\theta)^2\right] = Var(\hat{\theta}) + \left[E(\hat{\theta})-\theta\right]^2 = \text{variance} + \text{bias}^2.$$

This is the mean squared error of an estimator, not the MSE of a fit.

Least squares is unbiased, with the smallest variance among the linear unbiased estimators (Gauss-Markov, Lecture 3). But with multicollinearity, that smallest variance is large (Lecture 20):

$$Var(\hat{\beta}_k) = VIF_k \times \frac{\sigma^2}{\sum_i (X_{ik}-\bar{X}_k)^2}.$$

A little bias can buy a large drop in variance: shrink the estimates toward 0.

Ridge regression

Standardize each predictor (mean 0, standard deviation 1), and minimize

$$Q_R = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1}\right)^2 + \lambda \sum_{k=1}^{p-1} \beta_k^2, \qquad \lambda \geq 0.$$

Ridge regression

Standardize each predictor (mean 0, standard deviation 1), and minimize

$$Q_R = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1}\right)^2 + \lambda \sum_{k=1}^{p-1} \beta_k^2, \qquad \lambda \geq 0.$$

- The penalty $\lambda \sum_k \beta_k^2$ makes large coefficients costly: the estimates are shrunk toward 0.
- $\lambda = 0$ gives least squares. As $\lambda \to \infty$, every $\hat{\beta}_k \to 0$ and the fit tends to $\bar{Y}$.
- The intercept is not penalized. The predictors are standardized so that the penalty does not depend on their units.

Ridge regression

Standardize each predictor (mean 0, standard deviation 1), and minimize

$$Q_R = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1}\right)^2 + \lambda \sum_{k=1}^{p-1} \beta_k^2, \qquad \lambda \geq 0.$$

- The penalty $\lambda \sum_k \beta_k^2$ makes large coefficients costly: the estimates are shrunk toward 0.
- $\lambda = 0$ gives least squares. As $\lambda \to \infty$, every $\hat{\beta}_k \to 0$ and the fit tends to $\bar{Y}$.
- The intercept is not penalized. The predictors are standardized so that the penalty does not depend on their units.

λ is a tuning parameter: chosen from the data, by cross-validation (later today).

The ridge estimator

Centre Y too. Then $\hat{\beta}_0 = \bar{Y}$, and with $\boldsymbol{X}$ the $n \times (p-1)$ matrix of the standardized predictors (no column of 1s):

$$Q_R = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\, \boldsymbol{\beta}'\boldsymbol{\beta}.$$

The ridge estimator

Centre Y too. Then $\hat{\beta}_0 = \bar{Y}$, and with $\boldsymbol{X}$ the $n \times (p-1)$ matrix of the standardized predictors (no column of 1s):

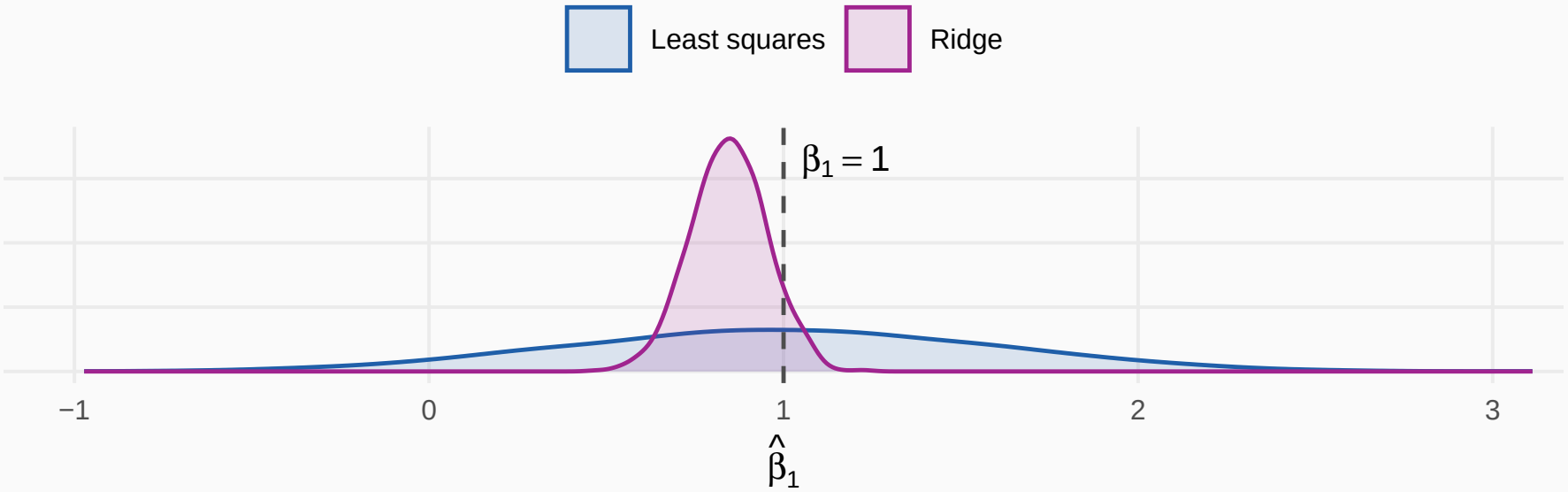
$$Q_R = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\,\boldsymbol{\beta}'\boldsymbol{\beta}.$$

$$\hat{\boldsymbol{\beta}}^R = (\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{Y}.$$

For $\lambda > 0$, $\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I}$ can always be inverted, even when $\boldsymbol{X}'\boldsymbol{X}$ is nearly singular: ridge regression was introduced for multicollinearity.

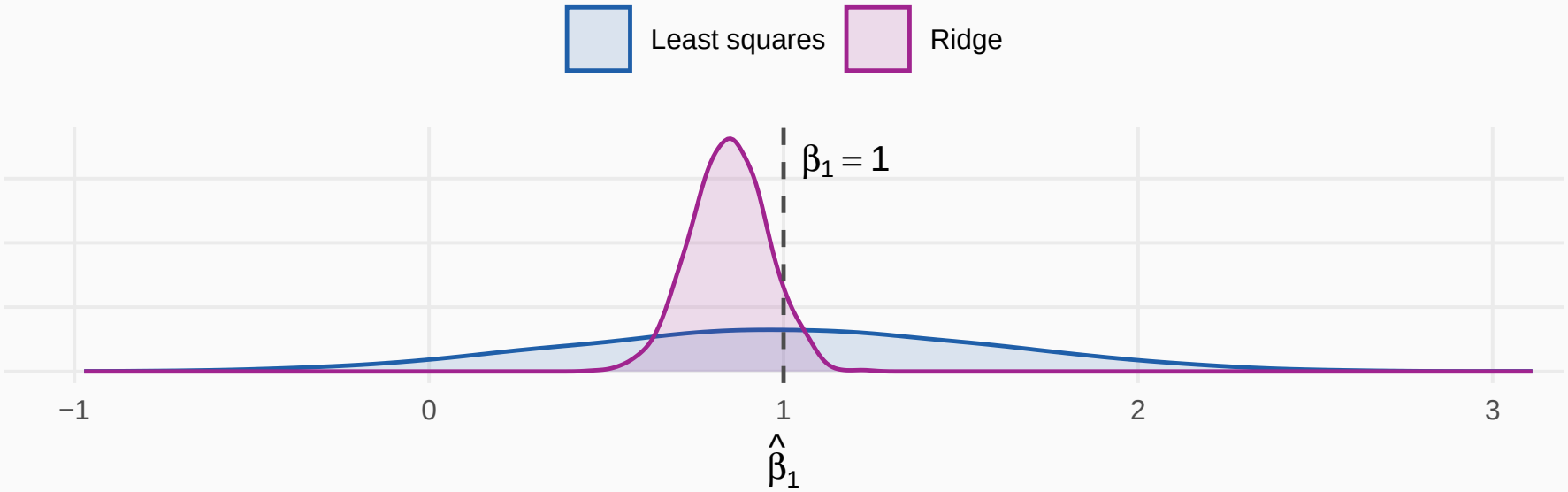
Ridge on two correlated predictors

The same two predictors in every sample ($n = 30$, $r_{12} = 0.95$), $\beta_1 = \beta_2 = 1$, $\sigma = 1$: 2000 samples of Y. In each, $\hat{\beta}_1$ by least squares and by ridge ($\lambda = 10$).



Ridge on two correlated predictors

The same two predictors in every sample ($n = 30$, $r_{12} = 0.95$), $\beta_1 = \beta_2 = 1$, $\sigma = 1$: 2000 samples of Y. In each, $\hat{\beta}_1$ by least squares and by ridge ($\lambda = 10$).



	Mean of $\hat{\beta}_1$	Standard deviation	Mean of $(\hat{\beta}_1 - \beta_1)^2$
Least squares	0.99	0.59	0.35
Ridge	0.85	0.11	0.04

One predictor: bias and variance

One centred predictor x_i, with $S = \sum_i x_i^2$ and least squares slope $\hat{\beta} = \sum_i x_i Y_i / S$. Ridge gives

$$\hat{\beta}^R = \frac{\sum_i x_i Y_i}{S + \lambda} = \frac{S}{S + \lambda}\,\hat{\beta}.$$

One predictor: bias and variance

One centred predictor x_i, with $S = \sum_i x_i^2$ and least squares slope $\hat{\beta} = \sum_i x_i Y_i / S$. Ridge gives

$$\hat{\beta}^R = \frac{\sum_i x_i Y_i}{S + \lambda} = \frac{S}{S + \lambda}\,\hat{\beta}.$$

$$E(\hat{\beta}^R) = \frac{S}{S + \lambda}\,\beta, \qquad Var(\hat{\beta}^R) = \frac{\sigma^2 S}{(S + \lambda)^2}, \qquad E\big[(\hat{\beta}^R - \beta)^2\big] = \frac{\sigma^2 S + \lambda^2 \beta^2}{(S + \lambda)^2}.$$

One predictor: bias and variance

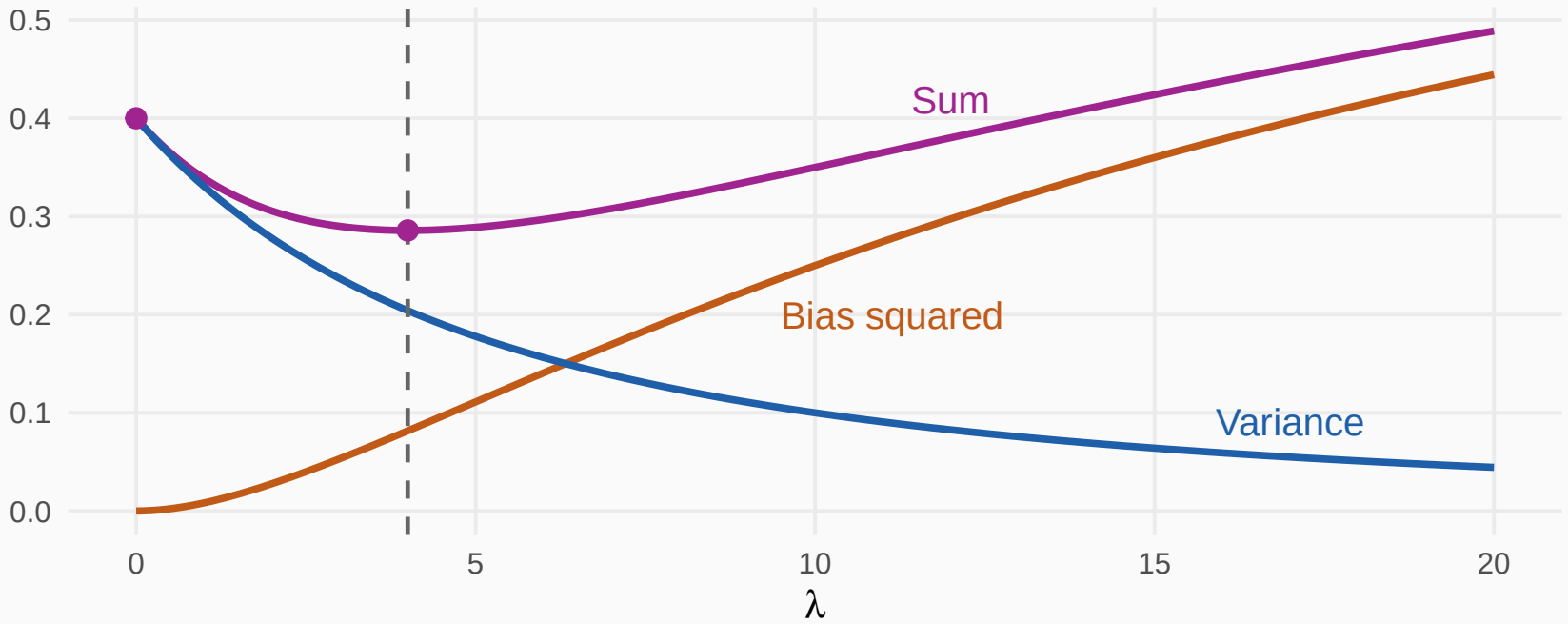
One centred predictor x_i, with $S = \sum_i x_i^2$ and least squares slope $\hat{\beta} = \sum_i x_i Y_i / S$. Ridge gives

$$\hat{\beta}^R = \frac{\sum_i x_i Y_i}{S + \lambda} = \frac{S}{S + \lambda}\,\hat{\beta}.$$

$$E(\hat{\beta}^R) = \frac{S}{S + \lambda}\,\beta, \qquad Var(\hat{\beta}^R) = \frac{\sigma^2 S}{(S + \lambda)^2}, \qquad E\big[(\hat{\beta}^R - \beta)^2\big] = \frac{\sigma^2 S + \lambda^2 \beta^2}{(S + \lambda)^2}.$$

Its derivative in λ is $2S(\lambda\beta^2 - \sigma^2)/(S + \lambda)^3$: negative at $\lambda = 0$, zero at $\lambda = \sigma^2/\beta^2$. Some ridge estimate always beats least squares, but the best λ depends on the unknown σ^2/β^2.

The trade-off



$S = 10$, $\sigma = 2$, $\beta = 1$. Least squares ($\lambda = 0$): 0.40. At the best $\lambda = \sigma^2/\beta^2 = 4$: 0.29.

The school data: multicollinearity

70 schools: ACHV, an index of the students' achievement, and three indices of the school: FAM (faculty credentials), PEER (peer group), SCHOOL (facilities).

	Alone		Together		
Predictor	$\hat{\beta}_k$ ($s\{\hat{\beta}_k\}$)	t^*	$\hat{\beta}_k$ ($s\{\hat{\beta}_k\}$)	t^*	VIF_k
FAM	0.88 (0.23)	3.81	1.10 (1.41)	0.78	37.6
PEER	1.08 (0.27)	4.04	2.32 (1.48)	1.57	30.2
SCHOOL	0.93 (0.24)	3.80	-2.28 (2.22)	-1.03	83.2

The school data: multicollinearity

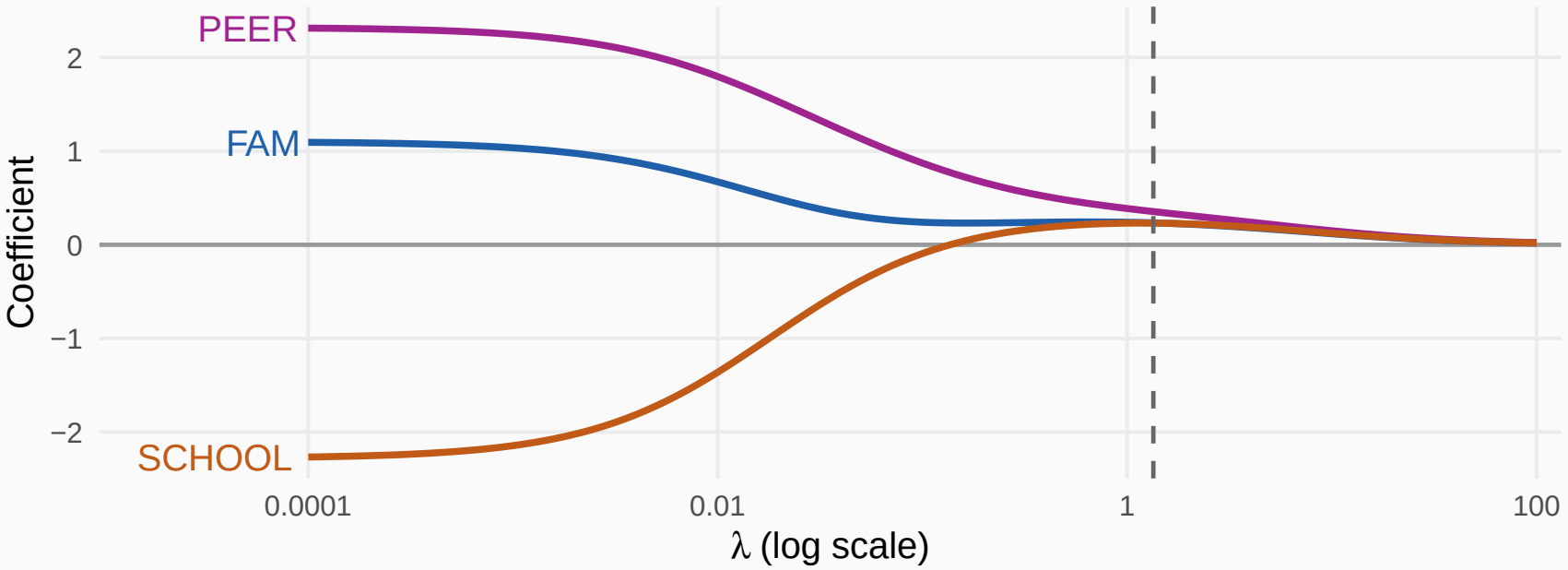
70 schools: ACHV, an index of the students' achievement, and three indices of the school: FAM (faculty credentials), PEER (peer group), SCHOOL (facilities).

	Alone		Together		
Predictor	$\hat{\beta}_k$ ($s\{\hat{\beta}_k\}$)	t^*	$\hat{\beta}_k$ ($s\{\hat{\beta}_k\}$)	t^*	VIF_k
FAM	0.88 (0.23)	3.81	1.10 (1.41)	0.78	37.6
PEER	1.08 (0.27)	4.04	2.32 (1.48)	1.57	30.2
SCHOOL	0.93 (0.24)	3.80	-2.28 (2.22)	-1.03	83.2

All six of Lecture 20's warning signs:

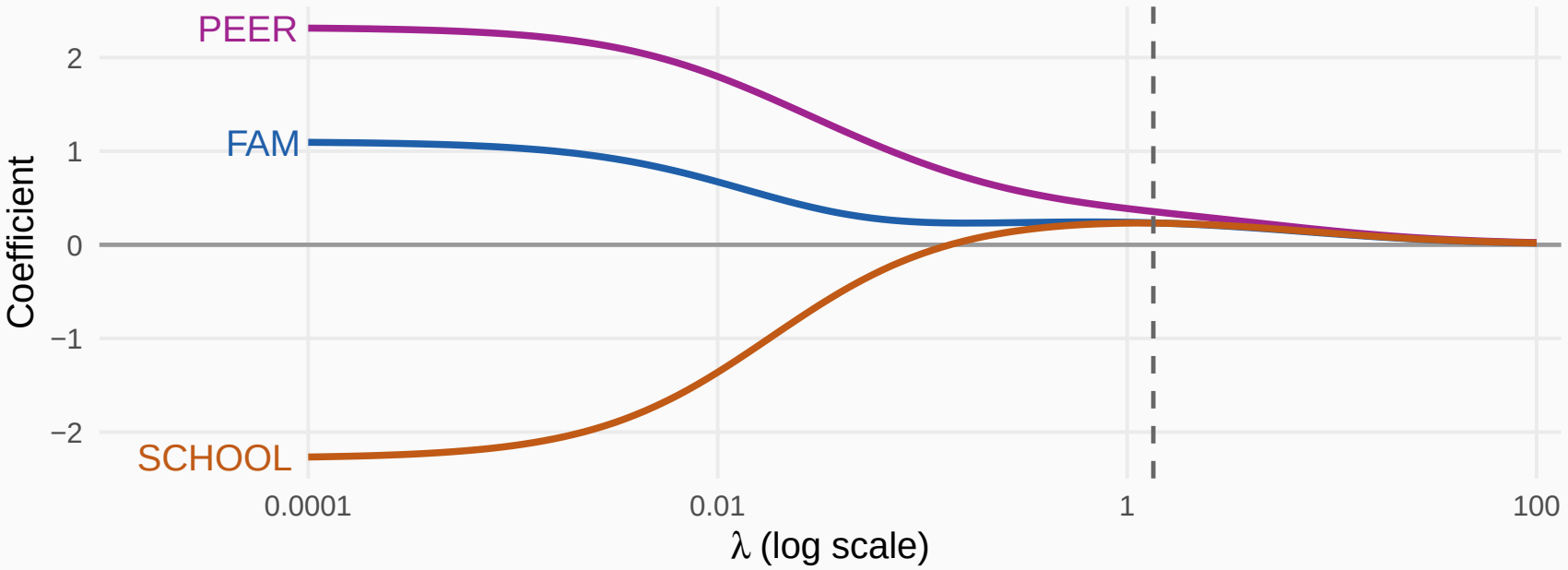
1. The coefficients change: SCHOOL goes from 0.93 alone to -2.28 together.
2. No $|t^*|$ reaches $t(0.975; 66) = 2.00$, yet $F^* = 5.72$, $p = 0.002$.
3. An unexpected sign: better facilities, lower achievement.
4. Large standard errors: 1.41 to 2.22, against 0.23 to 0.27 alone.
5. R^2 barely moves: 0.19 with PEER alone, 0.21 with all three.
6. The predictors are highly correlated: 0.96 to 0.99.

The ridge trace



From least squares (left) to 0 (right). For small λ the coefficients already come together: SCHOOL turns positive. The dashed line marks the λ chosen by cross-validation, the method of the next slide.

The ridge trace



From least squares (left) to 0 (right). For small λ the coefficients already come together: SCHOOL turns positive. The dashed line marks the λ chosen by cross-validation, the method of the next slide.

How large should λ be? "Where the trace settles" is a judgement call. Instead: the λ that predicts best.

Choosing λ: k-fold cross-validation

Lecture 20 split the cases once, into a training set and a validation set.

Two drawbacks: the $MSPE$ depends on that random split, and the validation cases are not used to fit the model.

Instead, every case takes a turn in the validation set:

1. Split the cases at random into k folds of about equal size (often $k = 5$ or 10).
2. For each fold: fit on the other $k - 1$ folds, and predict the cases of the fold.
3. CV_k is the average of the n squared prediction errors.

Choosing λ: k-fold cross-validation

Lecture 20 split the cases once, into a training set and a validation set.

Two drawbacks: the $MSPE$ depends on that random split, and the validation cases are not used to fit the model.

Instead, every case takes a turn in the validation set:

1. Split the cases at random into k folds of about equal size (often $k = 5$ or 10).
2. For each fold: fit on the other $k - 1$ folds, and predict the cases of the fold.
3. CV_k is the average of the n squared prediction errors.

Every case is predicted once, with a fitted model that did not use it. With $k = n$, each fold is one case, and $CV_n = PRESS_p/n$.

Choosing λ: k-fold cross-validation

Lecture 20 split the cases once, into a training set and a validation set.

Two drawbacks: the $MSPE$ depends on that random split, and the validation cases are not used to fit the model.

Instead, every case takes a turn in the validation set:

1. Split the cases at random into k folds of about equal size (often $k = 5$ or 10).
2. For each fold: fit on the other $k - 1$ folds, and predict the cases of the fold.
3. CV_k is the average of the n squared prediction errors.

Every case is predicted once, with a fitted model that did not use it. With $k = n$, each fold is one case, and $CV_n = PRESS_p/n$.

For shrinkage: compute $CV_k(\lambda)$ for a grid of values of λ, always with the same folds, and take the λ with the smallest CV_k. Using the same folds for least squares and for ridge also makes their comparison fair.

The school data in R

```r
library(glmnet)
x <- as.matrix(eeo[, c("FAM", "PEER", "SCHOOL")])
set.seed(1); fold <- sample(rep(1:10, length.out = 70))
# alpha = 0: ridge; the same folds for every lambda
cvr <- cv.glmnet(x, eeo$ACHV, alpha = 0, foldid = fold)
coef(cvr, s = "lambda.min")
```

The school data in R

```r
library(glmnet)
x <- as.matrix(eeo[, c("FAM", "PEER", "SCHOOL")])
set.seed(1); fold <- sample(rep(1:10, length.out = 70))
# alpha = 0: ridge; the same folds for every lambda
cvr <- cv.glmnet(x, eeo$ACHV, alpha = 0, foldid = fold)
coef(cvr, s = "lambda.min")
```

	FAM	PEER	SCHOOL	CV_{10}
Least squares	1.10	2.32	-2.28	4.68
Ridge, λ by cross-validation	0.23	0.35	0.23	4.50

The school data in R

```
library(glmnet)
x <- as.matrix(eeo[, c("FAM", "PEER", "SCHOOL")])
set.seed(1); fold <- sample(rep(1:10, length.out = 70))
# alpha = 0: ridge; the same folds for every lambda
cvr <- cv.glmnet(x, eeo$ACHV, alpha = 0, foldid = fold)
coef(cvr, s = "lambda.min")
```

	FAM	PEER	SCHOOL	CV_{10}
Least squares	1.10	2.32	-2.28	4.68
Ridge, λ by cross-validation	0.23	0.35	0.23	4.50

All three coefficients now have the sign of their correlation with ACHV, and the predictions are a little better. `glmnet` standardizes the predictors itself and reports the coefficients in their original units.

`glmnet` needs a matrix, not a formula. It measures λ on its own scale: the values differ from the slides', the method does not.

The LASSO

The LASSO

The same idea, with absolute values:

$$Q_L = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1}\right)^2 + \lambda \sum_{k=1}^{p-1} |\beta_k|.$$

The LASSO

The same idea, with absolute values:

$$Q_L = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1}\right)^2 + \lambda \sum_{k=1}^{p-1} |\beta_k|.$$

- Like ridge: standardized predictors, estimates shrunk toward 0, biased, with a smaller variance than least squares.
- Unlike ridge: some coefficients are set exactly to 0, more of them as λ grows. The LASSO selects the predictors and shrinks the others, in one step.
- No formula for the estimates in general: they are computed numerically (`glmnet` with `alpha = 1`).

The LASSO

The same idea, with absolute values:

$$Q_L = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1}\right)^2 + \lambda \sum_{k=1}^{p-1} |\beta_k|.$$

- Like ridge: standardized predictors, estimates shrunk toward 0, biased, with a smaller variance than least squares.
- Unlike ridge: some coefficients are set exactly to 0, more of them as λ grows. The LASSO selects the predictors and shrinks the others, in one step.
- No formula for the estimates in general: they are computed numerically (`glmnet` with `alpha = 1`).

LASSO: least absolute shrinkage and selection operator.

One predictor: why exactly 0

With one centred predictor, $\sum_i (Y_i - \bar{Y} - \beta x_i)^2 = S(\beta - \hat{\beta})^2 +$ a constant. So ridge and the LASSO minimize

$$S(\beta - \hat{\beta})^2 + \lambda\beta^2 \qquad \text{and} \qquad S(\beta - \hat{\beta})^2 + \lambda|\beta|.$$

One predictor: why exactly 0

With one centred predictor, $\sum_i (Y_i - \bar{Y} - \beta x_i)^2 = S(\beta - \hat{\beta})^2 +$ a constant. So ridge and the LASSO minimize

$$S(\beta - \hat{\beta})^2 + \lambda\beta^2 \qquad \text{and} \qquad S(\beta - \hat{\beta})^2 + \lambda|\beta|.$$

$$\hat{\beta}^R = \frac{S}{S+\lambda}\,\hat{\beta}, \qquad \hat{\beta}^L = \begin{cases} \hat{\beta} - \lambda/(2S) & \text{if } \hat{\beta} > \lambda/(2S), \\ 0 & \text{if } |\hat{\beta}| \le \lambda/(2S), \\ \hat{\beta} + \lambda/(2S) & \text{if } \hat{\beta} < -\lambda/(2S). \end{cases}$$

One predictor: why exactly 0

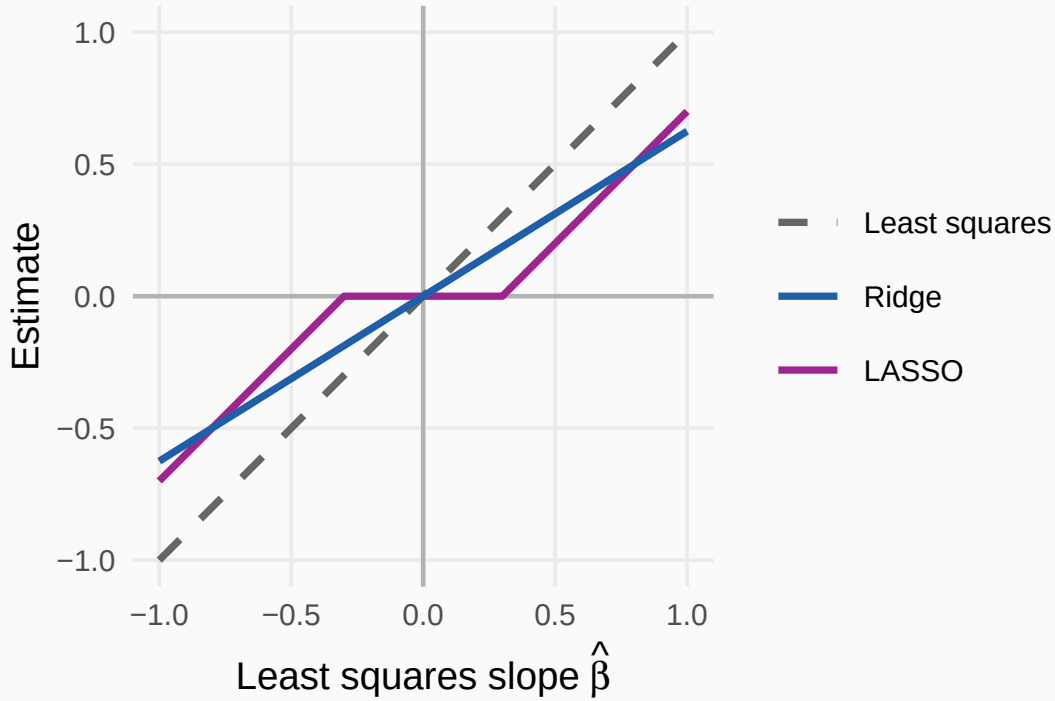
With one centred predictor, $\sum_i (Y_i - \bar{Y} - \beta x_i)^2 = S(\beta - \hat{\beta})^2 +$ a constant. So ridge and the LASSO minimize

$$S(\beta - \hat{\beta})^2 + \lambda\beta^2 \qquad \text{and} \qquad S(\beta - \hat{\beta})^2 + \lambda|\beta|.$$

$$\hat{\beta}^R = \frac{S}{S+\lambda}\,\hat{\beta}, \qquad \hat{\beta}^L = \begin{cases} \hat{\beta} - \lambda/(2S) & \text{if } \hat{\beta} > \lambda/(2S), \\ 0 & \text{if } |\hat{\beta}| \le \lambda/(2S), \\ \hat{\beta} + \lambda/(2S) & \text{if } \hat{\beta} < -\lambda/(2S). \end{cases}$$

Ridge multiplies $\hat{\beta}$ by a factor below 1: never exactly 0. The LASSO moves $\hat{\beta}$ toward 0 by a fixed amount $\lambda/(2S)$, and stops at 0.

Least squares, ridge and the LASSO



Here $S = 10$ and $\lambda = 6$: ridge keeps $10/16 = 0.625$ of $\hat{\beta}$; the LASSO subtracts 0.3, and gives 0 when $|\hat{\beta}| \leq 0.3$.

Exercise

One centred predictor, with $S = \sum_i x_i^2 = 40$, and $\lambda = 10$. Three samples give the least squares slopes

$$\hat{\beta} = 0.9, \qquad \hat{\beta} = -0.5, \qquad \hat{\beta} = 0.1.$$

1. Compute the ridge estimate $\hat{\beta}^R$ and the LASSO estimate $\hat{\beta}^L$ for each.
2. Which of the two shrinks the large slope $\hat{\beta} = 0.9$ more?
3. For which values of $\hat{\beta}$ is $\hat{\beta}^L = 0$?

Exercise: solution

$S/(S+\lambda) = 40/50 = 0.8$ and $\lambda/(2S) = 10/80 = 0.125$.

$\hat{\beta}$	0.9	-0.5	0.1
$\hat{\beta}^R = 0.8\,\hat{\beta}$	0.72	-0.40	0.08
$\hat{\beta}^L$	$0.9 - 0.125 = 0.775$	$-0.5 + 0.125 = -0.375$	0

Exercise: solution

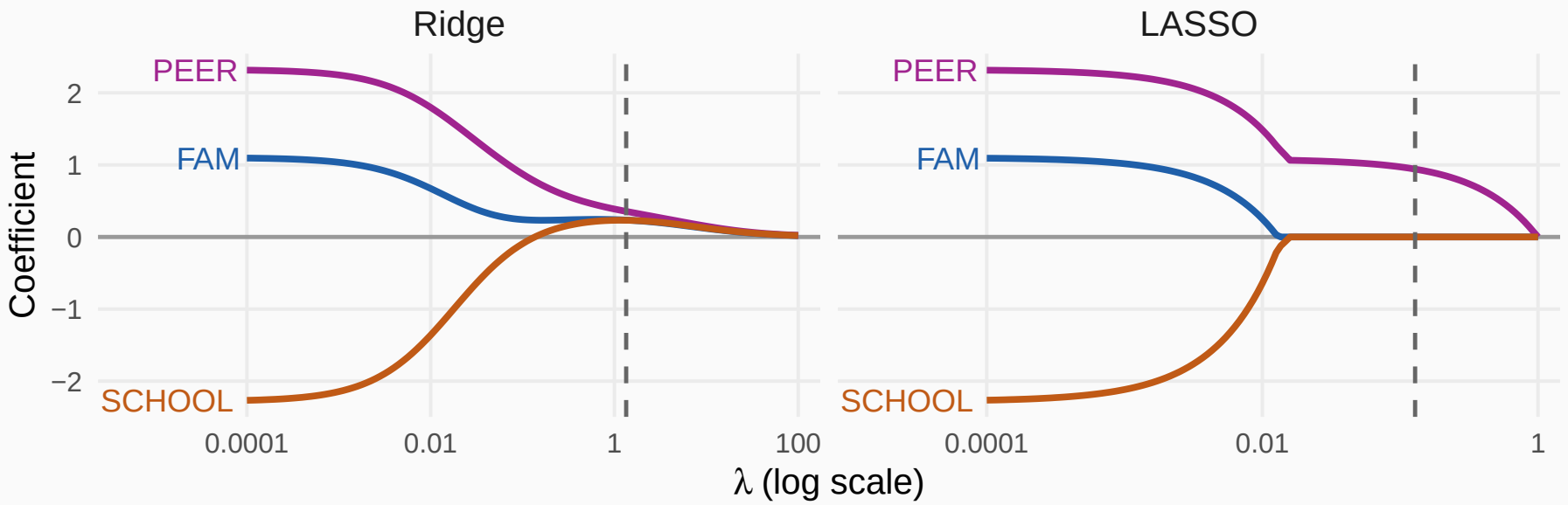
$S/(S+\lambda) = 40/50 = 0.8$ and $\lambda/(2S) = 10/80 = 0.125$.

$\hat{\beta}$	0.9	-0.5	0.1
$\hat{\beta}^R = 0.8\,\hat{\beta}$	0.72	-0.40	0.08
$\hat{\beta}^L$	$0.9 - 0.125 = 0.775$	$-0.5 + 0.125 = -0.375$	0

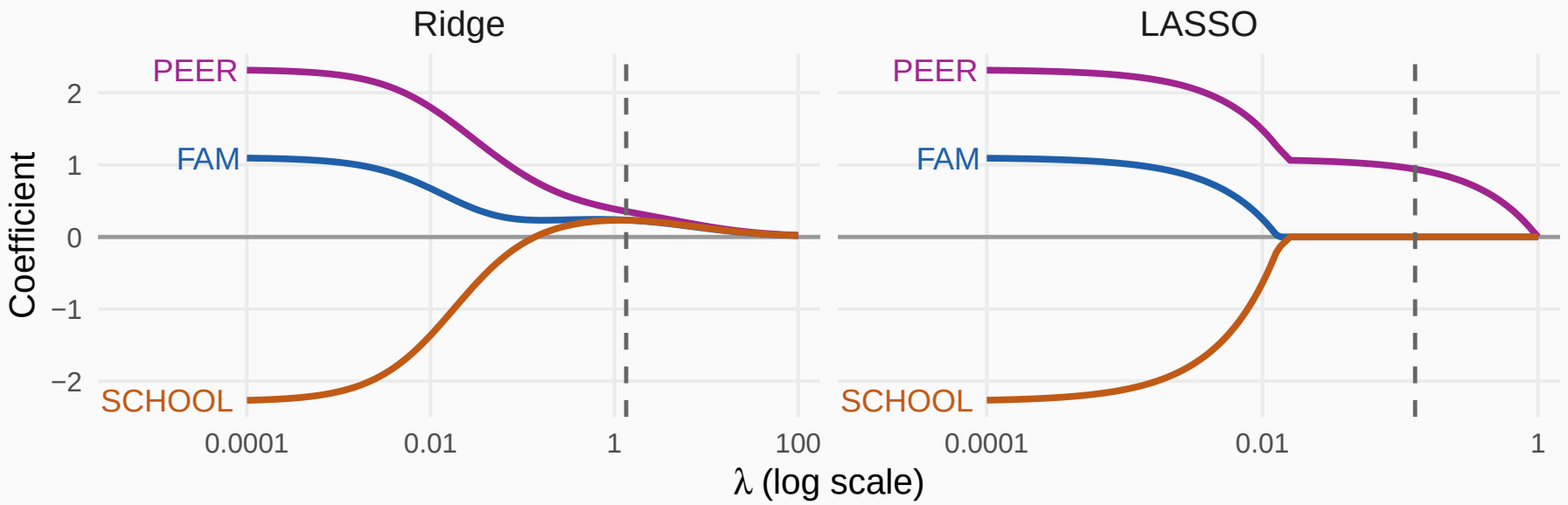
2. Ridge removes 20% of $\hat{\beta}$: 0.18 for $\hat{\beta} = 0.9$, more than the LASSO's 0.125. The LASSO shrinks large slopes less, and sets small ones to 0.

3. $|\hat{\beta}| \leq 0.125$: a slope that small is replaced by 0.

The LASSO on the school data

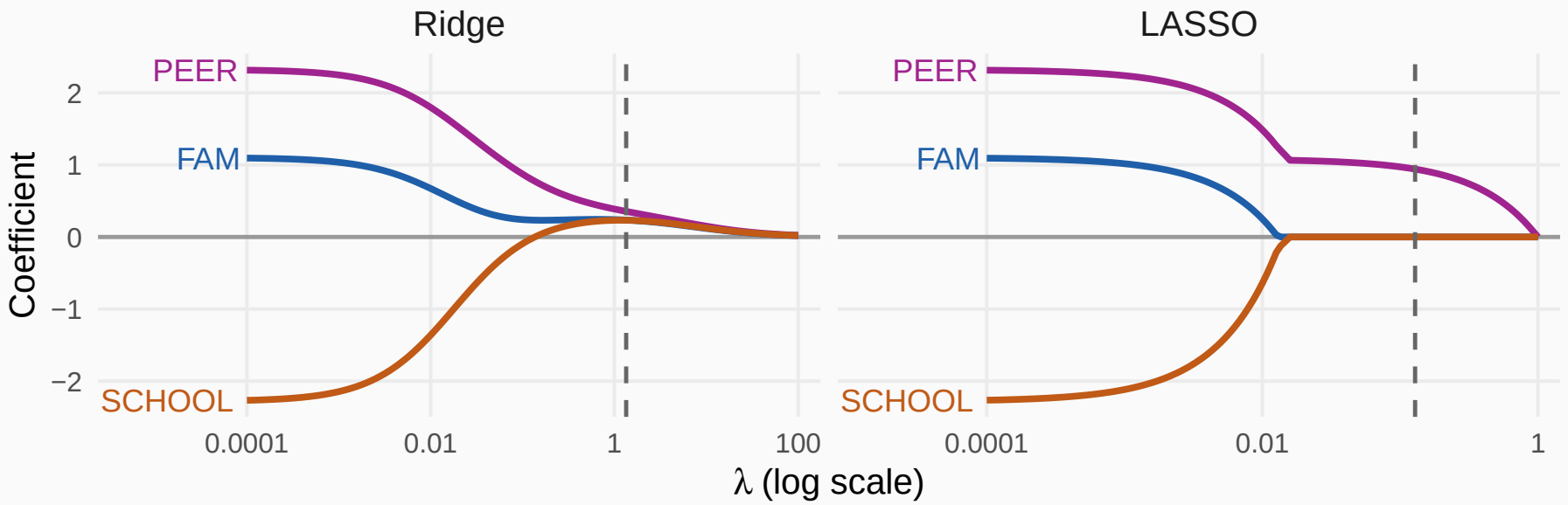


The LASSO on the school data



At the λ chosen by cross-validation (same folds), the LASSO keeps only PEER: 0.94, with $CV_{10} = 4.59$, between ridge (4.50) and least squares (4.68).

The LASSO on the school data

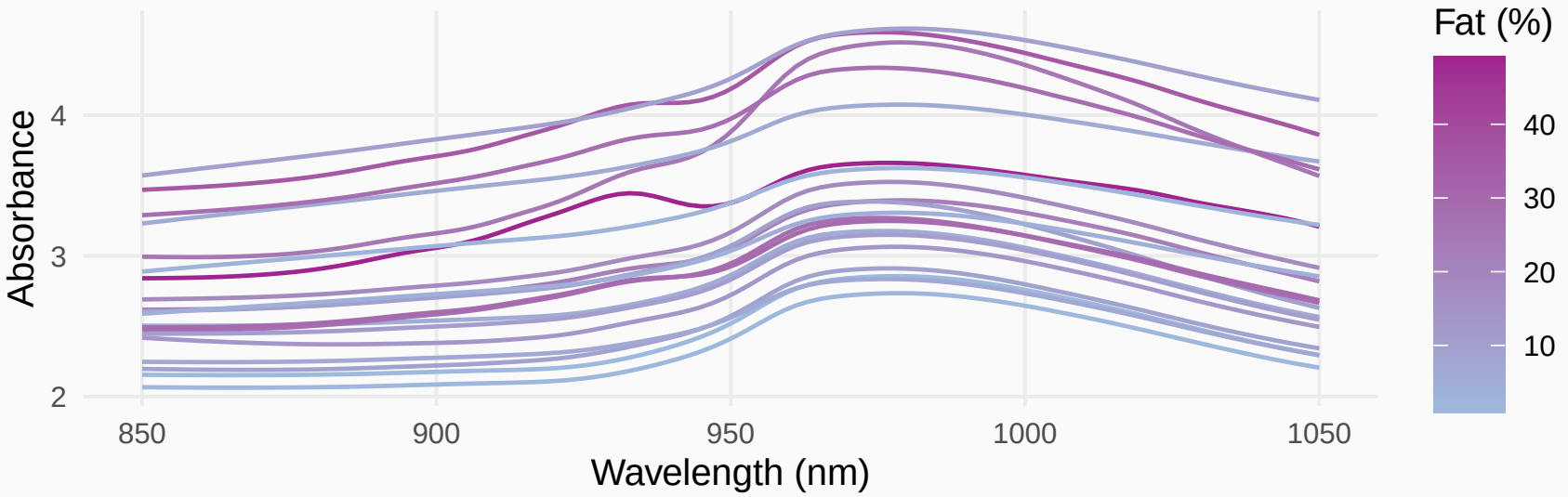


At the λ chosen by cross-validation (same folds), the LASSO keeps only PEER: 0.94, with $CV_{10} = 4.59$, between ridge (4.50) and least squares (4.68).

With three nearly interchangeable predictors, the LASSO keeps one; ridge shares the effect among the three.

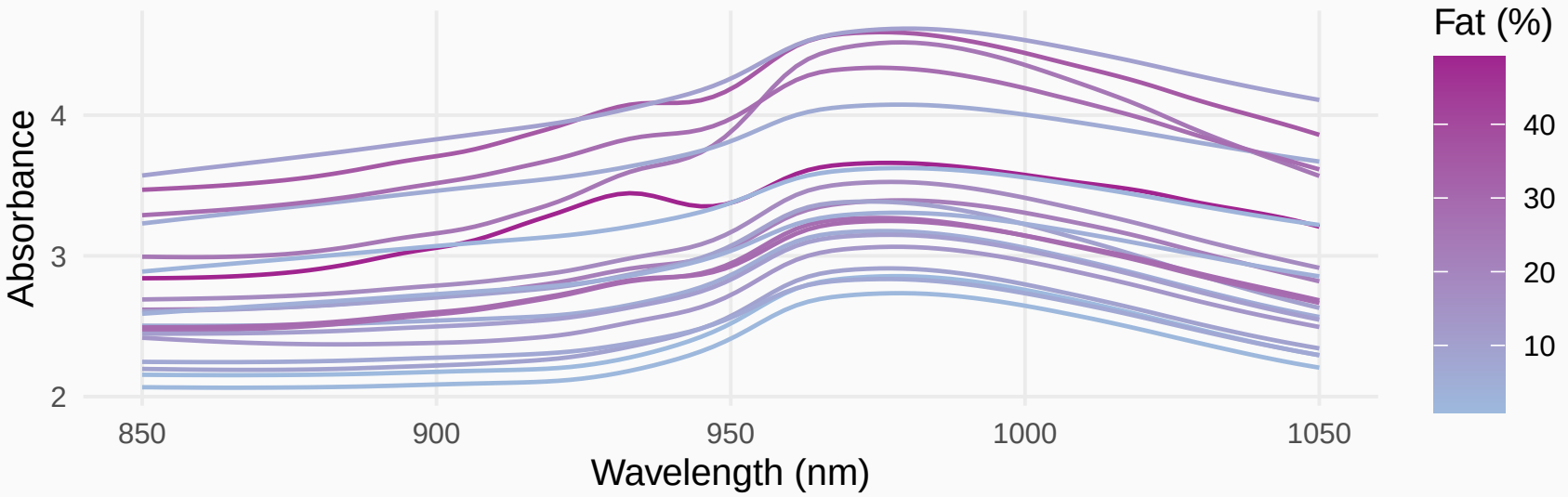
Many predictors: the meat data

215 samples of chopped meat. A spectrometer measures the absorbance of each at 100 wavelengths (850 to 1050 nm) in seconds; the fat content (%) needs a chemical analysis. Can the spectrum predict the fat?



Many predictors: the meat data

215 samples of chopped meat. A spectrometer measures the absorbance of each at 100 wavelengths (850 to 1050 nm) in seconds; the fat content (%) needs a chemical analysis. Can the spectrum predict the fat?



Neighbouring absorbances have correlations above 0.9998. Least squares with all 100 on 172 samples: $R^2 = 0.997$, and the largest VIF is about 2.5×10^{10}.

The meat data in R

`x`: the 100 absorbances, as a matrix; `y`: the fat; `train`: the first 172 samples. The last 43 are kept aside.

```
library(lars)
fitl <- lars(x[train, ], y[train])          # the whole LASSO path
grid <- seq(0, 0.1, by = 0.0005)            # values of t to try
set.seed(1)
cvl <- cv.lars(x[train, ], y[train], K = 10, index = grid)
tbest <- grid[which.min(cvl$cv)]            # chosen by 10-fold CV
pred <- predict(fitl, x[-train, ], s = tbest, mode = "fraction")$fit
mean((y[-train] - pred)^2)                  # MSPE
```

The meat data in R

`x`: the 100 absorbances, as a matrix; `y`: the fat; `train`: the first 172 samples. The last 43 are kept aside.

```
library(lars)
fitl <- lars(x[train, ], y[train])          # the whole LASSO path
grid <- seq(0, 0.1, by = 0.0005)            # values of t to try
set.seed(1)
cvl <- cv.lars(x[train, ], y[train], K = 10, index = grid)
tbest <- grid[which.min(cvl$cv)]            # chosen by 10-fold CV
pred <- predict(fitl, x[-train, ], s = tbest, mode = "fraction")$fit
mean((y[-train] - pred)^2)                  # MSPE
```

`lars` indexes the path by $t = \sum_k |\hat{\beta}_k^L| \,/\, \sum_k |\hat{\beta}_k|$ instead of λ: from $t = 0$ (every coefficient 0, λ very large) to $t = 1$ (least squares, $\lambda = 0$). Here $t = 0.017$.

The meat data in R

`x`: the 100 absorbances, as a matrix; `y`: the fat; `train`: the first 172 samples. The last 43 are kept aside.

```
library(lars)
fitl <- lars(x[train, ], y[train])          # the whole LASSO path
grid <- seq(0, 0.1, by = 0.0005)            # values of t to try
set.seed(1)
cvl <- cv.lars(x[train, ], y[train], K = 10, index = grid)
tbest <- grid[which.min(cvl$cv)]            # chosen by 10-fold CV
pred <- predict(fitl, x[-train, ], s = tbest, mode = "fraction")$fit
mean((y[-train] - pred)^2)                  # MSPE
```

`lars` indexes the path by $t = \sum_k |\hat{\beta}_k^L| \,/\, \sum_k |\hat{\beta}_k|$ instead of λ: from $t = 0$ (every coefficient 0, λ very large) to $t = 1$ (least squares, $\lambda = 0$). Here $t = 0.017$.

Why not `glmnet`? With 100 nearly identical predictors, its numerical method stops the path too early. `lars` computes the LASSO path exactly.

The meat data: three methods compared

CV_{10} on the 172 training samples, with the same 10 folds for the three methods; then one check, the $MSPE$ on the 43 samples kept aside.

	Predictors	CV_{10}	$MSPE$
Least squares	100	14.3	14.5
Ridge, λ by CV_{10}	100	5.6	3.6
LASSO, t by CV_{10}	21	4.5	3.6

The meat data: three methods compared

CV_{10} on the 172 training samples, with the same 10 folds for the three methods; then one check, the $MSPE$ on the 43 samples kept aside.

	Predictors	CV_{10}	$MSPE$
Least squares	100	14.3	14.5
Ridge, λ by CV_{10}	100	5.6	3.6
LASSO, t by CV_{10}	21	4.5	3.6

Both shrinkage methods predict about 4 times better than least squares, on the folds and on the new samples. The LASSO does it with 21 of the 100 wavelengths.

The meat data: three methods compared

CV_{10} on the 172 training samples, with the same 10 folds for the three methods; then one check, the $MSPE$ on the 43 samples kept aside.

	Predictors	CV_{10}	$MSPE$
Least squares	100	14.3	14.5
Ridge, λ by CV_{10}	100	5.6	3.6
LASSO, t by CV_{10}	21	4.5	3.6

Both shrinkage methods predict about 4 times better than least squares, on the folds and on the new samples. The LASSO does it with 21 of the 100 wavelengths.

On the school data, ridge did better than the LASSO: which method wins depends on the data. Compare them by cross-validation, then check once on the cases kept aside.

Ridge or LASSO, and what neither gives

- Ridge keeps every predictor: suited to many correlated predictors that each matter a little.
- The LASSO sets coefficients to 0: a smaller model, easier to read. Among correlated predictors, it tends to keep one.
- Both are biased: the t tests and intervals of `summary()` do not apply to them, and `glmnet` gives none. They are tools for prediction.
- To explain (which predictors matter, by how much, with what uncertainty), as your project does: use ordinary least squares. Choose the model with R^2_a, C_p, AIC, BIC or PRESS, validate it, and interpret its coefficients with their confidence intervals.

Ridge or LASSO, and what neither gives

- Ridge keeps every predictor: suited to many correlated predictors that each matter a little.
- The LASSO sets coefficients to 0: a smaller model, easier to read. Among correlated predictors, it tends to keep one.
- Both are biased: the t tests and intervals of `summary()` do not apply to them, and `glmnet` gives none. They are tools for prediction.
- To explain (which predictors matter, by how much, with what uncertainty), as your project does: use ordinary least squares. Choose the model with R^2_a, C_p, AIC, BIC or PRESS, validate it, and interpret its coefficients with their confidence intervals.

Not needed for the project report. Examinable on the final exam, like the rest of the term.

Summary

- Weighted least squares: check again on the weighted residuals $\sqrt{w_i}\, e_i$, not on `resid(fitw)`.
- $E[(\hat{\theta} - \theta)^2] =$ variance $+$ bias^2: a biased estimator can be closer on average.
- Ridge: penalty $\lambda \sum_k \beta_k^2$, $\hat{\boldsymbol{\beta}}^R = (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{Y}$; shrinks, never exactly to 0; made for multicollinearity.
- LASSO: penalty $\lambda \sum_k |\beta_k|$; shrinks, and sets some coefficients exactly to 0.
- Standardize the predictors, choose λ by k-fold cross-validation, and check on cases kept aside.

Report, R Markdown file, dataset, contribution statement and slides: Tuesday 1 December, 23:59. No late submission.