STA 235H - Model Selection II: Shrinkage

Fall 2021

McCombs School of Business, UT Austin

Announcements

Homework 4 is due on Thursday

- Please check out the notes for submissions on HW4.
- Check out answers on Canvas discussion board!
- I'll complement what we see in class with R code videos when needed.
    - Make sure all packages are installed and work.

Prediction project

It's a competition!

- **Teams of three or four**, depending on your section (you should enter your team on Canvas if you haven't done so).
- You will have a **classification** and a **prediction** task.
    - Choose your best models (and compare them to <u>one</u> other method)
- Grade will be based on: **data + models (including analysis) + accuracy ranking**
- Get an <u>early start</u> (you can start downloading and getting the data ready now)
    - **No extension for the final project**.
    - There are many **deadlines** at the end of the semester.

Last week

- Introduction to **prediction**:
  - Bias vs. Variance, validation set approach, cross-validation.
- One method for model selection: **stepwise subsetting**:
  - We start with a null (full) model and add (subtract) one variable at a time. We choose the best one through CV.

Today: Continuing our journey

- Regularization and model selection: **Shrinkage**
  - Advantages of regularization over OLS
  - Ridge regression and Lasso regression
  - When is ridge regression better? When do we prefer lasso?

Honey, I shrunk the coefficients!

What is shrinkage?

- Last class, we saw **stepwise procedure**: Subsetting model selection approach.
  - Select k out of p total predictors
- **Shrinkage** ***(a.k.a Regularization)***: Fitting a model with all p predictors, but introducing bias (i.e. shrinking coefficients towards 0) for improvement in variance.
  - Ridge regression
  - Lasso regression

Let's build a ridge.

Ridge Regression: An example

- Window-shoppers vs. High rollers

Ordinary Least Squares

- In an OLS: Minimize sum of squared-errors, i.e. $\min_\beta \sum_{i=1}^n (\text{spend}_i - \beta \text{freq}_i)^2$

What about fit?

- Does the OLS fit the testing data well?

Ridge Regression

- Let's shrink the coefficients!: Ridge Regression

Why does Ridge Regression reduce its slope compared to OLS?

Ridge Regression: What does it do?

- Ridge regression introduces bias to reduce variance in the testing data set.
- In a simple regression (i.e. one regressor/covariate):

$$\min_{\beta} \sum_{i=1}^{n} \underbrace{(y_i - \beta_0 - x_i\beta_1)^2}_{OLS}$$

Ridge Regression: What does it do?

- Ridge regression **introduces bias to reduce variance** in the testing data set.
- In a simple regression (i.e. one regressor/covariate):

$$\min_{\beta} \sum_{i=1}^{n} \underbrace{(y_i - \beta_0 - x_i\beta)^2}_{OLS} + \underbrace{\lambda \cdot \beta_1^2}_{Ridge Penalty}$$

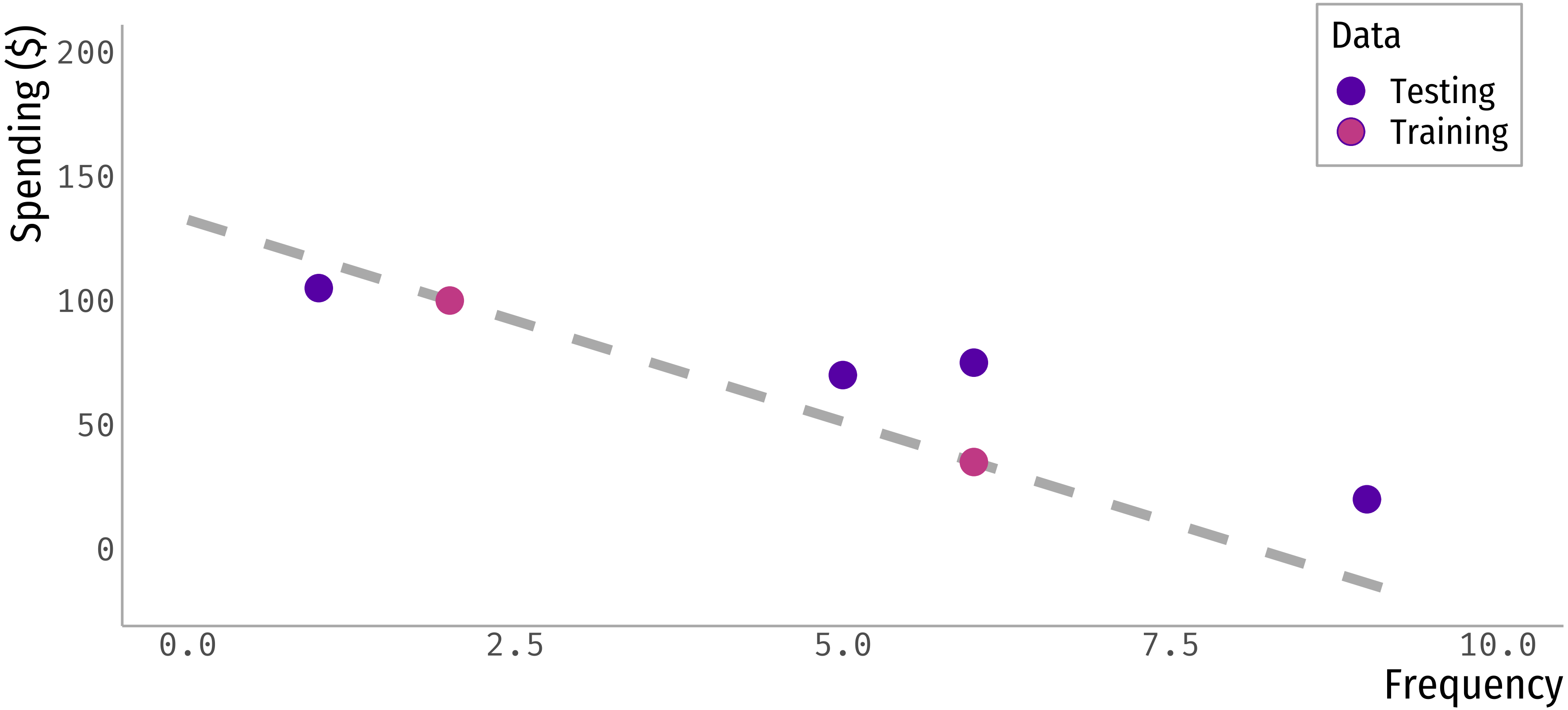
- λ is the **penalty factor** $\rightarrow$ indicates how much we want to shrink the coefficients.

Back to the plots...

- Let's solve the minimization problem for ridge regression. **What line do we choose?**

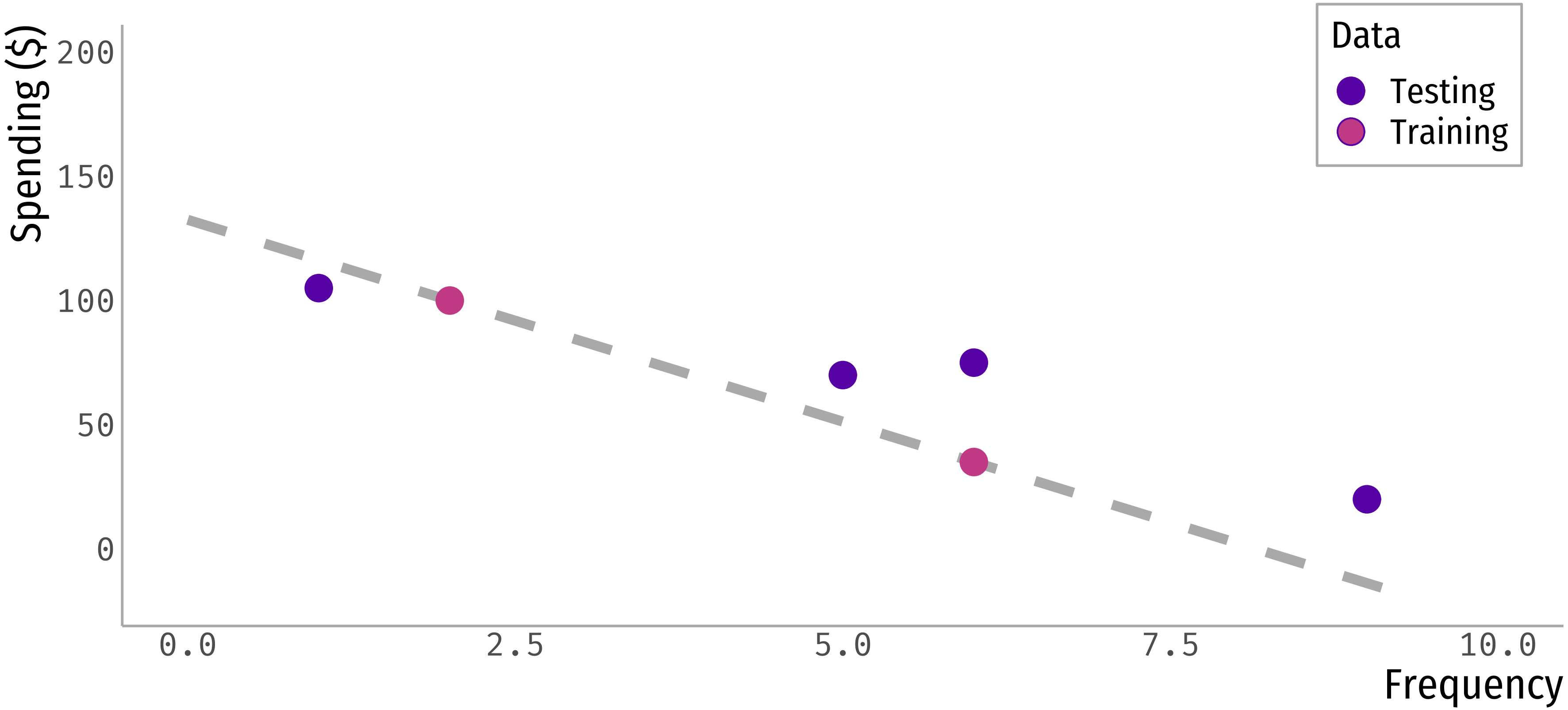
For the OLS line

$$0 + \lambda \cdot (-16.25)^2 = 264.1\lambda$$



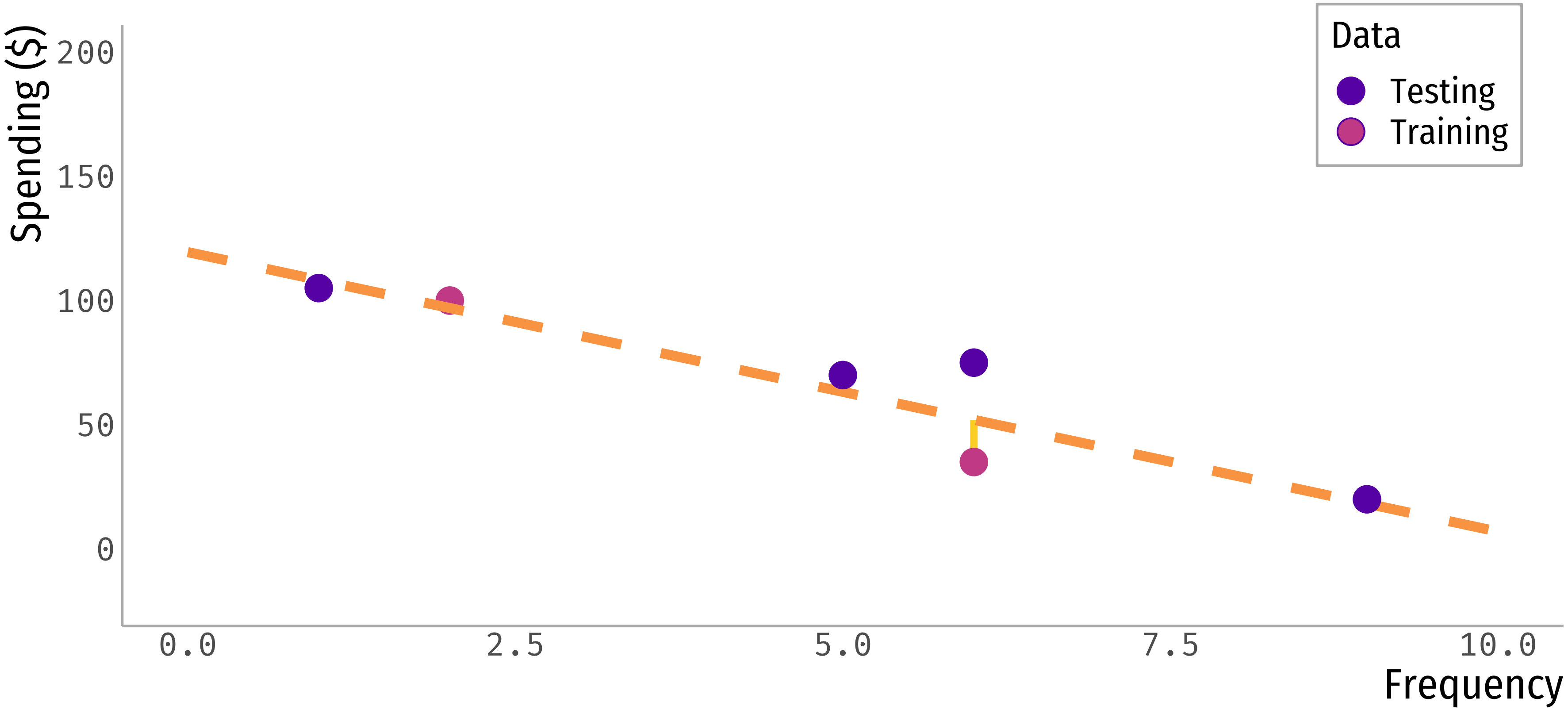
For the OLS line

$$0 + \lambda \cdot (-16.25)^2 = 264.1 \times 3 = 792.3$$



Now, for the ridge regression line

$$(3^2 + (-17)^2) + \lambda \cdot (-11.25)^2 = 298 + 126.6 \times 3 = 677.8$$



But remember... we care about accuracy in the testing dataset!

RMSE on the testing dataset: OLS

$$RMSE = \sqrt{\frac{1}{4}\sum_{i=1}^{4}(\text{spend}_i - (132.5 - 16.25 \cdot \text{freq}_i))^2} = 28.36$$

RMSE on the testing dataset: Ridge Regression

$$RMSE = \sqrt{\frac{1}{4}\sum_{i=1}^{4}(\text{spend}_i - (119.5 - 11.25 \cdot \text{freq}_i))^2} = 12.13$$

Seems like these data points are cherry-picked...

- Yes! This is a stylized example to show what's happening in the background when we are running OLS and Ridge regression.
- How can we know whether OLS or Ridge Regression is better without running the risk of cherry-picking training and testing data?
- If the data is linear, OLS might be the right model:
  - Penalty term λ will most likely be 0.

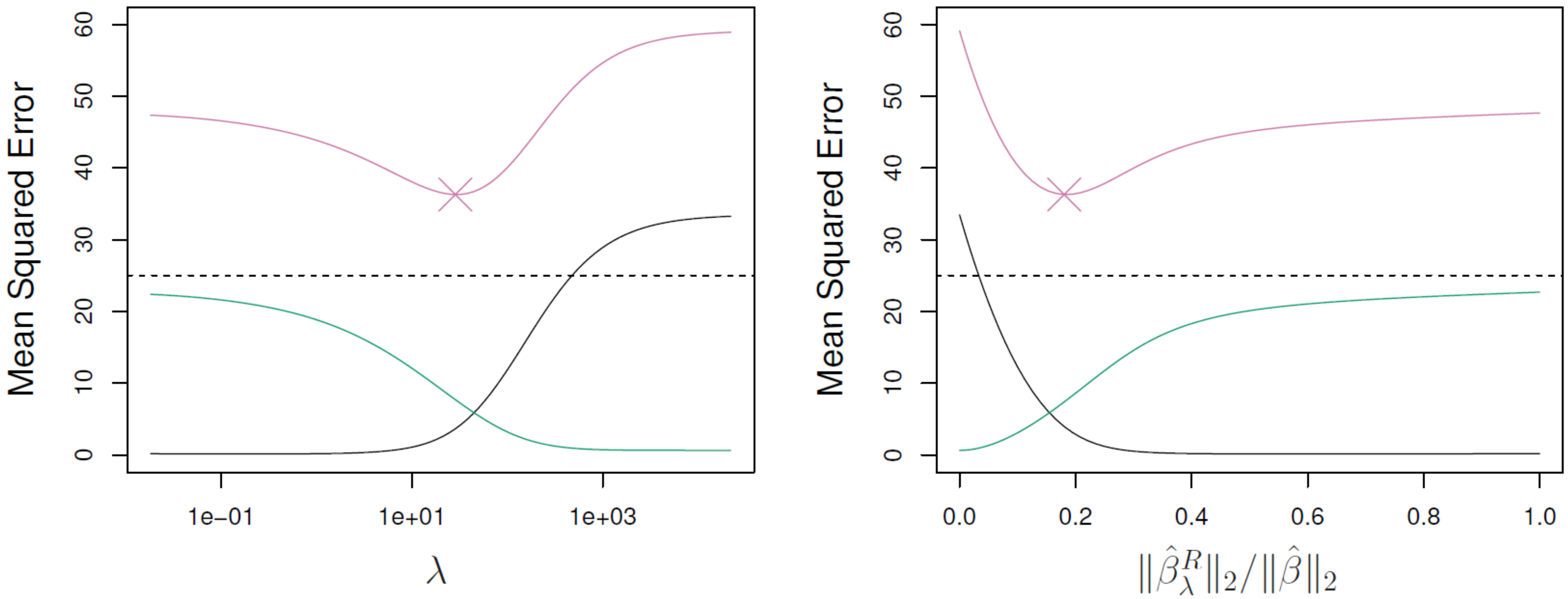


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of λ and $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$. The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

Ridge Regression in general

- For regressions that include **more than one regressor**:

$$\min_{\beta} \underbrace{\sum_{i=1}^{n} (y_i - \sum_{k=0}^{p} x_i \beta_k)^2}_{OLS} + \underbrace{\lambda \cdot \sum_{k=1}^{p} \beta_k^2}_{RidgePenalty}$$

- In our previous example, if we had two regressors, $female$ and $freq$:

$$\min_{\beta} \sum_{i=1}^{n} (\text{spend}_i - \beta_0 - \beta_1 \text{female}_i - \beta_2 \text{freq}_i)^2 + \lambda \cdot (\beta_1^2 + \beta_2^2)$$

- Because the ridge penalty includes the β's coefficients, **scale matters**:
    - Standardize coefficients to $SD = 1 \rightarrow x'_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}(x_{ij} - \bar{x}_j)^2}}$

Some jargon

- Ridge regression is also referred to as l_2 regularization:
  - l_2 norm $\rightarrow ||\beta||_2 = \sqrt{\sum_{k=1}^{p} \beta^2}$
- Some important notes:
  - $||\hat{\beta}_\lambda^R||_2$ will always decrease in λ.
  - $||\hat{\beta}_\lambda^R||_2 / ||\hat{\beta}||_2$ will always decrease in λ.

If λ=0, what is the value of l_2 norm for the ridge regression over the l_2 norm of OLS?

How do we choose λ?

Cross-validation!

1) Choose a grid of λ values

- The grid you choose will be context dependent (play around with it!)

2) Compute cross-validation error (e.g. RMSE) for each

3) Choose the smallest one.

λ vs RMSE?

λ vs RMSE? A zoom

How do we do this in R?

```
library(caret)

set.seed(100)

data <- read.csv("https://raw.githubuserconten

lambda_seq <- c(0,10^seq(-3, 3, length = 100))

ridge <- train(spend ~., data = train.data,
            method = "glmnet",
            preProcess = "scale",
            trControl = trainControl("cv", num
            tuneGrid = expand.grid(alpha = 0,
                          lambda = lambda_seq)
  )

cv_lambda <- data.frame(lambda = ridge$results
                        rmse = ridge$results$R
```

- We will be using the `caret` package

How do we do this in R?

```
library(caret)

set.seed(100)

data <- read.csv("https://raw.githubuserconten

lambda_seq <- c(0,10^seq(-3, 3, length = 100))

ridge <- train(spend ~., data = train.data,
            method = "glmnet",
            preProcess = "scale",
            trControl = trainControl("cv", numl
            tuneGrid = expand.grid(alpha = 0,
                        lambda = lambda_seq)
  )

cv_lambda <- data.frame(lambda = ridge$results$
                        rmse = ridge$results$RI
```

- We will be using the `caret` package
- We are doing **cross-validation**, so remember to set a seed!

How do we do this in R?

```r
library(caret)

set.seed(100)

data <- read.csv("https://raw.githubuserconten

lambda_seq <- c(0,10^seq(-3, 3, length = 100))

ridge <- train(spend ~., data = train.data,
            method = "glmnet",
            preProcess = "scale",
            trControl = trainControl("cv", num
            tuneGrid = expand.grid(alpha = 0,
                        lambda = lambda_seq)
  )

cv_lambda <- data.frame(lambda = ridge$results
                        rmse = ridge$results$R
```

- We will be using the `caret` package
- We are doing **cross-validation**, so remember to set a seed!
- You need to create a grid for the λ's **that will be tested**

How do we do this in R?

```r
library(caret)

set.seed(100)

data <- read.csv("https://raw.githubusercontent

lambda_seq <- c(0,10^seq(-3, 3, length = 100))

ridge <- train(spend ~., data = train.data,
               method = "glmnet",
               preProcess = "scale",
               trControl = trainControl("cv", numl
               tuneGrid = expand.grid(alpha = 0,
                                      lambda = lambda_seq)
  )

cv_lambda <- data.frame(lambda = ridge$results$
                        rmse = ridge$results$RI
```

- We will be using the `caret` package
- We are doing **cross-validation**, so remember to set a seed!
- You need to create a grid for the λ's **that will be tested**
- The function we will use is `train`: Same as before
  - `method="glmnet"` means that it will run an elastic net.
  - `alpha=0` means is a **ridge regression**
  - `lambda = lambda_seq` is not necessary (you can provide your own grid)

How do we do this in R?

```
library(caret)

set.seed(100)

data <- read.csv("https://raw.githubuserconten

lambda_seq <- c(0,10^seq(-3, 3, length = 100))

ridge <- train(spend ~., data = train.data,
            method = "glmnet",
            preProcess = "scale",
            trControl = trainControl("cv", num
            tuneGrid = expand.grid(alpha = 0,
                         lambda = lambda_seq)
  )

cv_lambda <- data.frame(lambda = ridge$results
                        rmse = ridge$results$R
```

- We will be using the `caret` package
- We are doing **cross-validation**, so remember to set a seed!
- You need to create a grid for the λ's **that will be tested**
- The function we will use is `train`: Same as before
- Important objects in `cv`:
  - `results$lambda`: Vector of λ that was tested
  - `results$RMSE`: RMSE for each λ
  - `bestTune$lambda`: λ that minimizes the error term.

How do we do this in R?

OLS regression:

```
lm1 <- lm(spend ~ freq + female,
          data = train.data)

coef(lm1)
```

```
## (Intercept)        freq      female
## 118.2605090  -3.4339875  -0.6391956
```

```
rmse(lm1, test.data)
```

```
## [1] 22.79557
```

Ridge regression:

```
coef(ridge$finalModel, ridge$bestTune$lambda)
```

```
## 3 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) 117.4837717
## freq         -9.4095221
## female       -0.2773666
```

```
rmse(ridge, test.data)
```

```
## [1] 22.7896
```

Let's look at this in R!

Throwing a lasso

Lasso regression

- Very similar to ridge regression, except it **changes the penalty term**:

$$\min_{\beta} \sum_{i=1}^{n} \underbrace{\left(y_i - \sum_{k=0}^{p} x_i \beta_k\right)^2}_{OLS} + \underbrace{\lambda \cdot \sum_{k=1}^{p} |\beta_k|}_{LassoPenalty}$$

- In our previous example:

$$\min_{\beta} \sum_{i=1}^{n} (\text{spend}_i - \beta_0 - \beta_1 \text{female}_i - \beta_2 \text{freq}_i)^2 + \lambda \cdot (|\beta_1| + |\beta_2|)$$

- Lasso regression is also called l_1 regularization:

$$||\beta||_1 = \sum_{k=1}^{p} |\beta|$$

Ridge vs Lasso

Ridge

- Final model will have p coefficients
- Usually better with multicollinearity

Lasso

- Can set coefficients = 0
- Improves interpretability of model
- Can be used for model selection

And how do we do Lasso in R?

```
library(caret)

set.seed(100)

data <- read.csv("https://raw.githubuserconten

lambda_seq <- 10^seq(-3, 3, length = 100)

lasso <- train(spend ~., data = train.data,
            method = "glmnet",
            preProcess = "scale",
            trControl = trainControl("cv", numl
            tuneGrid = expand.grid(alpha = 1,
                         lambda = lambda_seq)
  )

cvl_lambda <- data.frame(lambda = lasso$result
                         rmse = lasso$results$
```

Exactly the same!

- … But change `alpha=1`!!

And how do we do Lasso in R?

Ridge regression:

```
coef(ridge$finalModel, ridge$bestTune$lambda)
```

```
## 3 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept) 117.4837717
## freq         -9.4095221
## female       -0.2773666
```

```
rmse(ridge, test.data)
```

```
## [1] 22.7896
```

Lasso regression:

```
coef(lasso$finalModel, lasso$bestTune$lambda)
```

```
## 3 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept) 117.032965
## freq         -9.429349
## female        .
```

```
rmse(lasso, test.data)
```

```
## [1] 22.79291
```

- Why isn't every coefficient smaller in the Ridge Regression?

Main takeway points

- You can **shrink coefficients** to introduce bias and decrease variance.
- Ridge and Lasso regression are **similar**:
    - Lasso can be used for model selection.
- Importance of understanding **how to estimate the penalty coefficient**.

References

- James, G. et al. (2021). "Introduction to Statistical Learning with Applications in R". *Springer. Chapter 6.*
- STDHA. (2018). "Penalized Regression Essentials: Ridge, Lasso & Elastic Net"