ECON207 Session 7

Prediction with Regression Models

Anthony Tay

This Version: 31 Aug 2024

Session 7

- Main Themes:
  - Statement of the Prediction Problem
  - Loss Functions and Optimal Predictors
  - Estimating Out-of-Sample Root Mean Square Prediction Error
  - Specification and Predictor Selection
  - Bias-Variance Tradeoff
- Focus on Regression-based Prediction Methods

Code in today's session uses

```
library(tidyverse)
library(patchwork)
library(glmnet)
library(leaps)
```

The Prediction Problem

Statement of the Prediction Problem

Objective: To make an educated guess about the value of characteristic Y for a random draw from a population, where the individual drawn has characteristics

$$X_1 = X_1^o \, , \, X_2 = X_2^o \, , \, ... \, , \, X_k = X_k^o$$

E.g. In a certain population of individuals

- We want to predict hourly earnings of someone selected at random from the population, but with $educ = 12$, $age = 30$, $wexp = 5$

E.g., In a certain population of homes

- We want to predict price of a house randomly selected from the population, that is a certain distance from the city center, certain age, number of rooms

The Prediction Problem

In order to make an educated guess, we first have to learn something about the population

Assume we have random sample $\{Y_i, X_{i1}, \ldots, X_{ik}\}_{i=1}^{n}$ from population of interest

We use this to construct a prediction model: a model that we can use to

- predict value of Y_{n+1} for a new observation to be randomly drawn from the population, where the observation drawn satisfies

$$X_{n+1,1} = X_1^o \, , \; X_{n+1,2} = X_2^o \, , \; \ldots \, , \; X_{n+1,k} = X_k^o$$

- estimate size of potential error

Optimal Prediction Rule

Session 7.1 Optimal Prediction Rules

- Loss Function
- Optimal Prediction Rule given Loss Function
- Main Result: Under Mean Squared Error Loss Function, the Optimal Predictor is the Conditional Expectation

Optimal Prediction Rule

Let **Prediction Rule** be: $\hat{Y}^o = \hat{Y}^o(X_1^o, \ldots, X_k^o)$

- This is some deterministic function of $X_1^o, \ldots, X_k^o$

Suppose $(n+1)$th **outcome** of Y, with desired X characteristics, is Y_{n+1}^o

- This is a random variable!

Suppose **Loss Function** is $L(Y_{n+1}^o, \hat{Y}_{n+1}^o) = (Y_{n+1}^o - \hat{Y}_{n+1}^o)^2$

- Cost of error increases with size of error, and increases at increasing rate

Suppose we choose $\hat{Y}_{n+1}^o$ to minimize expected loss: $E((Y_{n+1}^o - \hat{Y}_{n+1}^o)^2)$

- i.e., we minimize "**Mean Square Prediction Error (MSPE)**"

Optimal Prediction Rule

Then the **Optimal Prediction Rule** is: $\hat{Y}^o_{n+1} = E(Y_{n+1} \mid X_{n+1,1} = X^o_1, \ldots, X_{n+1,k} = X^o_k)$

To simplify notation in the proof

- let $h = E(Y^o_{n+1} \mid X_{n+1,1} = X^o_1, \ldots, X_{n+1,k} = X^o_k)$
- let "$\mid X^o$" denote "conditional on $X_{n+1,1} = X^o_1, \ldots, X_{n+1,k} = X^o_k$
- we'll also drop the $n+1$ subscript on Y^o_{n+1}. Then

$$\begin{aligned}(Cond.)\ MSPE &= E((Y^o - \hat{Y}^o)^2 \mid X^o) \\ &= E((Y^o - h + h - \hat{Y}^o)^2 \mid X^o) \\ &= E((Y^o - h)^2 \mid X^o) + E((h - \hat{Y}^o)^2 \mid X^o) \\ &\quad + 2E((Y^o - h)(h - \hat{Y}^o) \mid X^o)\end{aligned}$$

Optimal Prediction Rule

Third term of third line is zero since

$$\begin{aligned} E((Y^o - h)(h - \hat{Y}^o) \mid X^o) &= (h - \hat{Y}^o)E((Y^o - h) \mid X^o) \\ &= (h - \hat{Y}^o)(E(Y^o \mid X^o) - h) \\ &= (h - \hat{Y}^o)(h - h) = 0 \end{aligned}$$

Therefore: $E((Y^o - \hat{Y}^o)^2 \mid X^o) = E((Y^o - h)^2 \mid X^o) + E((h - \hat{Y}^o)^2 \mid X^o)$

To minimize: choose $\hat{Y}^o = h = E(Y^o \mid X^o)$

Since expected loss minimized for $X_{n+1,1} = X_1^o, \ldots, X_{n+1,k} = X_k^o$, also minimized over all possible outcomes for $X_{n+1,1}, \ldots, X_{n+1,k}$

Optimal Prediction Rule

Remarks:

- If loss is not squared error loss, optimal prediction might not be conditional expectation
  - e.g., for absolute error loss, optimal prediction is conditional median
  - optimal predictor under asymmetric loss may be a biased predictor

We will assume squared error loss throughout

- If loss is asymmetric, adjust accordingly
- If loss is absolute loss, use conditional median

Optional Prediction Rule

- In general, conditional expectation must be estimated from available sample, a.k.a., "training" the prediction model
- In this session we assume conditional expectation is (at least approximately) linear-in-parameters

$$E(Y \mid X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

  - $E(Y \mid X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2$
  - $E(Y \mid X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + \beta_5 X_1 X_2$
  - $E(Y \mid X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \beta_2 \ln X_1$

and so on

Optional Prediction Rule

If we assume squared error loss

- Appropriate estimate of potential error size is estimate of

$$E((Y^o_{n+1} - \hat{Y}^o_{n+1})^2)$$

  or its square root (Root MSPE, or RMSPE)

We estimate RMSPE to

- quantify potential error size when using the model
- to choose between alternative specifications

Example 1

We begin with a prediction example *without* predictors

Example 1 Prediction *without* predictors

- Suppose you have an iid sample $\{Y_i\}_{i=1}^n$ from a population with
  - population mean μ
  - population variance σ^2
- You want to predict the $(n+1)$th draw Y_{n+1}
  - What is the optimal predictor?
  - What is the RMSPE?

Example 1

- With no predictors, optimal prediction rule is just unconditional mean $E(Y) = \mu$
- Since sample mean is unbiased for population mean, a reasonable option is to choose

$$\widehat{Y}_{n+1} = \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

- What is the root mean square prediction error if sample mean used as predictor?

$$\begin{aligned} MSPE &= E((Y_{n+1} - \overline{Y})^2) \\ &= E(Y_{n+1}^2 + \overline{Y}^2 - 2Y_{n+1}\overline{Y}) \\ &= E(Y_{n+1}^2) + E(\overline{Y}^2) - 2E(Y_{n+1}\overline{Y}) \end{aligned}$$

Example 1

- $E(Y_{n+1}^2) = \sigma^2 + \mu^2$ and $E(\overline{Y}^2) = \sigma^2/n + \mu^2$
- $E(Y_i Y_j) = \mu^2$ for all $i \neq j$ since $cov(Y_i, Y_j) = E(Y_i Y_j) - E(Y_i)E(Y_j) = 0$
- $E(Y_{n+1}\overline{Y}) = E((1/n)Y_{n+1}\sum_{i=1}^n Y_i) = 1/n\sum_{i=1}^n E(Y_{n+1}Y_i) = (1/n)n\mu^2 = \mu^2$

$$
\begin{aligned}
MSPE &= E((Y_{n+1} - \overline{Y})^2) \\
&= E(Y_{n+1}^2) + E(\overline{Y}^2) - 2E(Y_{n+1}\overline{Y}) \\
&= (\sigma^2 + \mu^2) + (\sigma^2/n + \mu^2) - 2\mu^2 = \left(1 + \frac{1}{n}\right)\sigma^2
\end{aligned}
$$

Replace σ^2 with $\widehat{\sigma^2}$ to get an estimate of the MSPE, take square root to get the RMSPE

Example 1

Remarks:

- MSPE is measurement of **Out-Of-Sample (OOS)** Mean Square Prediction Error
- In this example, MSPE arises because
  - we are predicting a new observation
  - we had to estimate our prediction rule

$$MSPE = \left(1 + \frac{1}{n}\right)\sigma^2 = \underbrace{\sigma^2}_{\text{unpred. of new obs}} + \underbrace{\frac{\sigma^2}{n}}_{\text{sampling or est. error}}$$

In more realistic examples, there may also be specification errors

Example 1

Remarks:

- In this example, there is simple formula for OOS MSPE (seldom the case)
- Relationship to in-sample "training" MSE, defined as

$$\text{Training } MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y})^2 = \widetilde{\sigma^2}$$

If we use $\widehat{\sigma^2}$ to estimate σ^2, we can estimate

$$\widehat{MSPE} = \left(1 + \frac{1}{n}\right)\widehat{\sigma^2} = \widehat{\sigma^2} + \frac{\widehat{\sigma^2}}{n} = \widetilde{\sigma^2} + \frac{2\widehat{\sigma^2}}{n} = \text{Training } MSE + \frac{2\widehat{\sigma^2}}{n}$$

OOS MSPE $>$ Training MSE in general, since in-sample fit is optimized to the sample

Example 2

Example 2 Simulation Experiment to Illustrate Example 1

Suppose population is well-represented by $Y \sim \chi^2(1)$ so that

- $E(Y) = 1$
- $Var(Y) = 2$

Suppose 1000 people each draw sample of 10 observations from population

They each predict the value of an 11th draw using the sample mean of their respective samples

We compare the average of their in-sample MSE with the average of their OOS MSPE

Example 2

The file `S7_Chisq1_samples.csv` contains 1000 columns of 11 iid Chi-sq(1) draws

- First 10 rows of column "rep_i" represents the sample for "replication i"
- Sample means of each of these 1000 samples used to predict respective 11th draws
- 11th row represents the actual 11th draws for the 1000 reps

Our theory says that

- their predictions will be centered around 1
- the Estimated MSPE of their predictions will be approx $(2 \times 2)/10$ greater than the Training MSE

Example 2

We have

```
# Read Data
Y <- read_csv("data\\S7_Chisq1_samples.csv", show_col_types=FALSE)
# Calculate predictions
pred <- colMeans(Y[1:10,])
# The actual outcomes are
Y11 <- as.matrix(Y[11,])    # Outcome
# Behaviour of Predictions
cat("Ave. prediction:", round(mean(pred),3))
cat("\nAve. Training MSE:", round(mean(colMeans((Y[1:10,] - pred)^2)),3))
cat("\nAve. OOS Squared Pred. Error:", round(mean((Y11-pred)^2),3))
```

```
Ave. prediction: 0.965
Ave. Training MSE: 2.081
Ave. OOS Squared Pred. Error: 2.383
```

Session 7.2

Session 7.2 Prediction with Linear Regression Models (One Predictor)

- Prediction *with* Predictors
- Optimal Prediction Rule is $E(Y_{n+1} \mid X_{n+1,1} = X_1^o, \ldots, X_{n+1,k} = X_k^o)$
- Assume $E(Y \mid X_1, \ldots, X_k)$ is at least approximately linear-in-parameters
- Main Themes
  - How to choose specification
  - How to estimate parameters
  - How to estimate MSPE
  - Variance-Bias Trade-off
- Also: Prediction intervals vs confidence intervals

Session 7.2

Variance-bias trade-off is based on the fact that MSPE is

$$E(e_{n+1}^2) = Var(e_{n+1}) + E(e_{n+1})^2$$

$$= \text{Pred. Err. Variance} + (\text{Pred. Err. Bias})^2$$

where $e_{n+1} = Y_{n+1}^o - \hat{Y}_{n+1}^o$

- Even if unbiased predictor is available, maybe a biased predictor can produce lower OOS MSPE
- If linear-in-specification is approximate, maybe "chasing unbiasedness" (by using more and more complicated specifications) can lead to greater OOS MSPE

We assume single predictor for now, will deal with multiple predictors in next section

Recollection (Linear Regression)

We first recall some results from OLS theory

$$\text{MLR: } Y = X\beta + \epsilon \text{ or } Y_i = \beta_0 + \beta_1 X_{i1} + \dots \beta_k X_{ik} + \epsilon_i, \ i = 1, \dots, n$$

If $E(Y \mid X) = X\beta$, i.e, $E(\epsilon \mid X) = 0_{n\times 1}$, then

- $\hat{\beta}^{ols} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$ is unbiased estimator for β (also linear)

If $Var(\epsilon \mid X_1, \dots, X_k) = \sigma^2 I_n$

- $Var(\hat{\beta}^{ols} \mid X) = \sigma^2(X^{\mathrm{T}}X)^{-1}$
- OLS estimators are *best* linear unbiased estimators

Recollection (Linear Regression)

The one regressor case with intercept: $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$

$$\begin{bmatrix} \hat{\beta}_0^{ols} \\ \hat{\beta}_1^{ols} \end{bmatrix} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y \;\Rightarrow\; \widehat{\beta}_0^{ols} = \overline{Y} - \widehat{\beta}_1^{ols}\overline{X} \text{ and } \widehat{\beta}_1^{ols} = \frac{\sum_{i=1}^n (X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^n (X_i - \overline{X})^2}$$

The variance-covariance matrix of $\widehat{\beta}^{ols}$ is

$$Var(\hat{\beta}^{ols} \mid X) = \begin{bmatrix} Var(\hat{\beta}_0^{ols} \mid X) & Cov(\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols} \mid X) \\ Cov(\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols} \mid X) & Var(\hat{\beta}_1^{ols} \mid X) \end{bmatrix} = \sigma^2 (X^{\mathrm{T}}X)^{-1}$$

with

$$Var(\hat{\beta}_0^{ols} \mid X) = \frac{\sigma^2 \sum_{i=1}^n X_i^2}{n\sum_{i=1}^n (X_i - \overline{X})^2} \quad , \quad Var(\hat{\beta}_1^{ols} \mid X) = \frac{\sigma^2}{\sum_{i=1}^n (X_i - \overline{X})^2}$$

$$\text{and} \quad Cov(\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols} \mid X) = \frac{-\sigma^2 \overline{X}}{\sum_{i=1}^n (X_i - \overline{X})^2}$$

Prediction with Linear Regression

We start with the simple linear regression case

- Suppose that $E(Y \mid X_1) = \beta_0 + \beta_1 X_1$
- Predict using $\hat{Y}^o_{n+1} = \hat{\beta}^{ols}_0 + \hat{\beta}^{ols}_1 X^o_1$
- Estimate β_0 and β_1 by OLS using sample $\{Y_i, X_{i1}\}_{i=1}^n$

We get

- Unbiased predictor
- Under homoskedasticity: *best* linear unbiased predictor

How might we estimate the mean square prediction error?

Prediction with Linear Regression

Potential new observation: $Y^o = \beta_0 + \beta_1 X_1^o + \epsilon^o$ (we'll skip the $n+1$ subscripts again)

Prediction: $\hat{Y}^o = \hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} X_1^o$

Prediction Error:

$$\begin{aligned} Y^o - \hat{Y}^o &= \beta_0 + \beta_1 X_1^o + \epsilon^o - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_1^o \\ &= \beta_0 - \hat{\beta}_0^{ols} + (\beta_1 - \hat{\beta}_1^{ols}) X_1^o + \epsilon^o \end{aligned}$$

Can show that

$$MSPE(X_1^o) = E((Y^o - \hat{Y}^o)^2 \mid X, X_1^o) = \sigma^2 \left(1 + \frac{1}{n} + \frac{(X_1^o - \overline{X}_1)^2}{\sum_{i=1}^n (X_{i1} - \overline{X}_1)^2} \right)$$

where "X" denotes the sample obs. of X_1, i.e.,$\{X_{11}, X_{21}, \ldots, X_{n1}\}$

Prediction with Linear Regression

Sketch of proof:

$$
\begin{aligned}
E((Y^o - \hat{Y}^o)^2 \mid X, X_1^o) &= E((\beta_0 - \hat{\beta}_0^{ols} + (\beta_1 - \hat{\beta}_1^{ols})X_1^o + \epsilon^o)^2 \mid X, X_1^o) \\
&= Var(\hat{\beta}_0^{ols} \mid X, X_1^o) + (X_1^o)^2\, Var(\hat{\beta}_1^{ols} \mid X, X_1^o) + Var(\epsilon_o \mid X, X_1^o) \\
&\qquad + 2X^o\, Cov(\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols} \mid X, X_1^o) \\
&= \frac{\sigma^2}{n\sum_{i=1}^n (X_{i1} - \overline{X}_1)^2} \left(\sum_{i=1}^n X_{i1}^2 + n(X_1^o)^2 - 2n\overline{X}_1 X_1^o \right) + \sigma^2 \\
&= \frac{\sigma^2}{n\sum_{i=1}^n (X_{i1} - \overline{X}_1)^2} \left(\sum_{i=1}^n (X_{i1} - \overline{X}_1)^2 + n(X_1^o - \overline{X}_1)^2 \right) + \sigma^2
\end{aligned}
$$

and simplify

Prediction with Linear Regression

Remarks about the $MSPE(X_1^o)$:

- Replace σ^2 in $MSPE(X_1^o) = \sigma^2 \left(1 + \frac{1}{n} + \frac{(X_1^o - \overline{X}_1)^2}{\sum_{i=1}^n (X_{i1} - \overline{X}_1)^2}\right)$ with $\widehat{\sigma^2}$
- Formula gives MSPE at various values of X_1^o
- Validity of formula depends on correct specification, homoskedasticity
- $n \uparrow$ or higher variation in X_{i1} implies MSPE $\downarrow$
- Harder to predict if X_1^o far from sample mean
- Often the "1" dominates, especially if n is large (unpredictability $\gg$ estimation err.)

Remarks about the $MSPE(X_1^o)$

- Prediction usually reported as "Prediction Interval", e.g., "95% P.I."

$$\hat{Y}^o(X_1^o) \pm 1.96 RMSPE \quad \text{where} \quad \hat{Y}^o(X_1^o) = \hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} X_1^o$$

- Sometimes 2 used instead of 1.96

"Confidence Interval" is $\hat{Y}^o(X_1^o) \pm 1.96\widehat{\sigma^2}\left(\frac{1}{n} + \frac{(X_1^o - \overline{X}_1)^2}{\sum_{i=1}^n (X_{i1} - \overline{X}_1)^2}\right)$

- measure of how well the conditional expectation was estimated
- does not account for ϵ^o

Remarks about the $MSPE$

- Sometimes we want "overall" out-of-sample MSPE (not for specific X_1^o)
- We can estimate this by averaging $MSPE(X_1^o)$ over all $X_1^o = X_{i1}$, $i = 1, \ldots, n$

$$
\begin{aligned}
MSPE &= \frac{1}{n}\sum_{i=1}^{n} MSPE(X_1^o = X_{i1}) \\
&= \frac{\sigma^2}{n}\sum_{i=1}^{n}\left(1 + \frac{1}{n} + \frac{(X_{i1} - \overline{X}_1)^2}{\sum_{i=1}^{n}(X_{i1} - \overline{X}_1)^2}\right) \\
&= \frac{\sigma^2}{n}\left(n + 1 + \frac{\sum_{i=1}^{n}(X_{i1} - \overline{X}_1)^2}{\sum_{i=1}^{n}(X_{i1} - \overline{X}_1)^2}\right) = \sigma^2\left(1 + \frac{2}{n}\right)
\end{aligned}
$$

Remarks about the $MSPE$

If we estimate σ^2 with $\widehat{\sigma^2} = \frac{1}{n-2}\sum_{i=1}^{n}(Y_i - \hat{Y}_i^{ols})^2$ and

define Training $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i^{ols})^2$

then we have

$$\widehat{MSPE} = \widehat{\sigma^2}\left(1 + \frac{2}{n}\right) = \widetilde{\sigma^2} + \frac{2}{n}\widehat{\sigma^2} + \frac{2}{n}\widehat{\sigma^2} = \text{Training } MSE + \frac{2p}{n}\widehat{\sigma^2}$$

where $p = 2$ is the number of coefficients in the model

What we have called $\widehat{MSPE}$ is often referred to as Mallow's C_p statistic

Cross-Validation

Alternative method for estimating MSPE: "Leave-One-Out Cross-validation"

- For each i
  - fit model to dataset without observation i
  - use model to predict Y_i
  - get prediction error $Y_i - \hat{Y}_{i,-i}$
- After doing this for all i, estimate MSPE as

$$MSPE_{LOOCV} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_{i,-i})^2$$

Prediction with Linear Regression

For the general MLR case with k regressors

$y = X\beta + \epsilon$, y is $n \times 1$, X is $n \times (k+1)$, $k < n$, with

- $E(y \mid X) = X\beta$ or $E(\epsilon \mid X) = 0_{n\times 1}$
- $Var(\epsilon \mid X) = \sigma^2(X^{\mathrm{T}}X)^{-1}$

Suppose we want to predict h new observations, at the specific values

$$X^o = \begin{bmatrix} 1 & X^o_{n+1,1} & X^o_{n+1,2} & \cdots & X^o_{n+1,k} \\ 1 & X^o_{n+2,1} & X^o_{n+2,2} & \cdots & X^o_{n+2,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X^o_{n+h,1} & X^o_{n+h,2} & \cdots & X^o_{n+h,k} \end{bmatrix}$$

Prediction with Linear Regression

Actual: $y^o = X^o\beta + \epsilon^o$

Prediction: $\hat{y}^o = X^o\hat{\beta}^{ols}$

The $h \times h$ MSPE *Matrix* is

$$
\begin{aligned}
&E((y^o - \hat{y}^o)(y^o - \hat{y}^o)^{\mathrm{T}}) = E((X^o\beta - X^o\hat{\beta}^{ols} + \epsilon^o)(X^o\beta - X^o\hat{\beta}^{ols} + \epsilon^o)^{\mathrm{T}}) \\
&= E((X^o(\beta - \hat{\beta}^{ols}) + \epsilon^o)(X^o(\beta - \hat{\beta}^{ols}) + \epsilon^o)^{\mathrm{T}}) \\
&= X^o E((\beta - \hat{\beta}^{ols})(\beta - \hat{\beta}^{ols})^{\mathrm{T}})X^{o\mathrm{T}} + E(\epsilon^o\epsilon^{o\mathrm{T}}) \\
&= X^o E((\hat{\beta}^{ols} - E(\hat{\beta}^{ols}))(\hat{\beta}^{ols} - E(\hat{\beta}^{ols}))^{\mathrm{T}})X^{o\mathrm{T}} + E(\epsilon^o\epsilon^{o\mathrm{T}}) \\
&= \sigma^2(X^o(X^{\mathrm{T}}X)^{-1}X^{o\mathrm{T}} + I_h)
\end{aligned}
$$

nb. Expectations are conditional on X and X^o, individual conditional MSPEs are along diagonal

Prediction with Linear Regression

(Unconditional) MSPE can be calculated using

- LOOCV
- Taking average of conditional MSPEs evaluated at each in-sample X observations, i.e., evaluate Conditional MSPE at $X^o = X$, add up the diagonals, divide by n

$$
\begin{aligned}
MSPE &= \frac{\sigma^2}{n}\text{trace}\left(X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}} + I_n\right) \\
&= \frac{\sigma^2}{n}\left(\text{trace}(X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}) + \text{trace}(I_n)\right) \\
&= \frac{\sigma^2}{n}\left(\text{trace}((X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}X) + n\right) = \frac{\sigma^2}{n}\left(\text{trace}(I_p) + n\right) = \sigma^2\left(1 + \frac{p}{n}\right)
\end{aligned}
$$

where p=k+1 is the number of coef. incl. intercept

Example 2

Example 2: Predicting $\ln earn$ using $educ$

- Data set is `earnings2019.csv` with 4946 observations
- Use estimation sample to construct prediction model of
  - $\ln(earn)$ at various levels of $educ$
  - size of prediction error in terms of RMSPE
- Model 1: $E(\ln earn \mid educ) = \beta_0 + \beta_1 educ$

Example 2

```
# Read data
dat <- read_csv("data\\earnings2019.csv", show_col_types=FALSE) %>%
  mutate(ln_earn=log(earn))

# Model 1
mdl1 <- lm(ln_earn~educ, data=dat)

# Prediction
# -- Predict at these values
newdata <- data.frame(educ = seq(6, 18, 1))
# -- Generate predictions and "prediction interval"
prd1a <- predict(mdl1, newdata, se.fit=TRUE, interval="prediction", level=0.95)
# -- Generate predictions and "confidence interval"
prd1b <- predict(mdl1, newdata, se.fit=TRUE, interval="confidence", level=0.95)
```

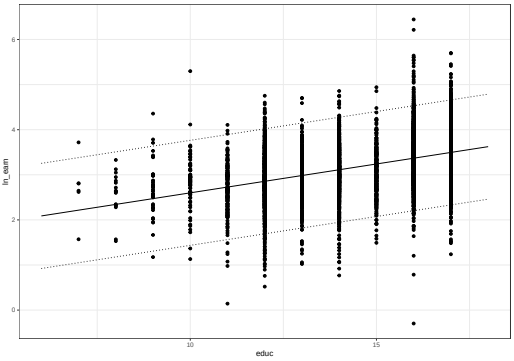
Example 2 (Prediction Interval)

```
print(cbind(newdata, prd1a$fit, se=prd1a$se.fit),
      row.names=F)
cat("df:", prd1a$df, "sghat:", prd1a$residual.scale)
```

educ	fit	lwr	upr	se
6	2.087968	0.9222934	3.253643	0.034107272
7	2.215965	1.0506974	3.381232	0.030268303
8	2.343961	1.1790493	3.508873	0.026470648
9	2.471958	1.3073490	3.636567	0.022735022
10	2.599954	1.4355966	3.764312	0.019097858
11	2.727951	1.5637920	3.892110	0.015628054
12	2.855947	1.6919351	4.019959	0.012466151
13	2.983944	1.8200260	4.147862	0.009911342
14	3.111940	1.9480646	4.275816	0.008527921
15	3.239937	2.0760509	4.403823	0.008881278
16	3.367933	2.2039849	4.531881	0.010802301
17	3.495930	2.3318667	4.659993	0.013644166
18	3.623926	2.4596963	4.788156	0.016949861

df: 4944 sghat: 0.5936184

```
prd1adat <- cbind(newdata, prd1a$fit)
ggplot() + geom_point(data=dat, aes(y=ln_earn, x=educ)) +
  geom_line(data=prd1adat, aes(y=fit, x=educ)) +
  geom_line(data=prd1adat, aes(y=lwr, x=educ), linetype="dotted") +
  geom_line(data=prd1adat, aes(y=upr, x=educ), linetype="dotted") +
  theme_bw()
```

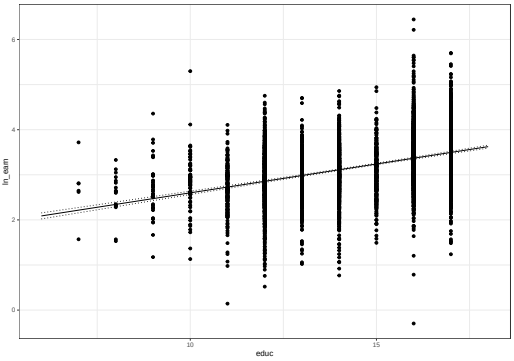


Example 2 (Confidence Interval)

```
print(cbind(newdata, prd1b$fit, se=prd1b$se.fit),
      row.names=F)
cat("df:", prd1b$df, "sghat:", prd1b$residual.scale)
```

```
 educ      fit      lwr      upr         se
    6 2.087968 2.021103 2.154834 0.034107272
    7 2.215965 2.156626 2.275304 0.030268303
    8 2.343961 2.292067 2.395856 0.026470648
    9 2.471958 2.427387 2.516529 0.022735022
   10 2.599954 2.562514 2.637395 0.019097858
   11 2.727951 2.697313 2.758589 0.015628054
   12 2.855947 2.831508 2.880386 0.012466151
   13 2.983944 2.964513 3.003374 0.009911342
   14 3.111940 3.095222 3.128659 0.008527921
   15 3.239937 3.222525 3.257348 0.008881278
   16 3.367933 3.346756 3.389111 0.010802301
   17 3.495930 3.469181 3.522678 0.013644166
   18 3.623926 3.590697 3.657155 0.016949861
df: 4944 sghat: 0.5936184
```

```
prd1bdat <- cbind(newdata, prd1b$fit)
ggplot() + geom_point(data=dat, aes(y=ln_earn, x=educ)) +
  geom_line(data=prd1bdat, aes(y=fit, x=educ)) +
  geom_line(data=prd1bdat, aes(y=lwr, x=educ), linetype="dotted") +
  geom_line(data=prd1bdat, aes(y=upr, x=educ), linetype="dotted") +
  theme_bw()
```



Example 2 (MSPE)

```
# LOOCV
n <- nrow(dat)
SqErr_LOOCV <- 0
for (i in 1:n){
  Ypred_i <- predict(lm(ln_earn~educ, data=dat[-i,]), newdata=dat[i,])
  SqErr_LOOCV <- SqErr_LOOCV + (pull(log(dat[i,"earn"])) - Ypred_i)^2
}
cat("Training MSE:", mean(mdl1$residuals^2),
    "  MSPE_LOOCV:", SqErr_LOOCV/n,
    "  C_p:", sum(mdl1$residuals^2)/(n-2)*(1+2/n))
```

```
Training MSE: 0.3522403   MSPE_LOOCV: 0.3525308   C_p: 0.3525253
```

- MSPE_LOOCV and C_p are both greater than Training Error

Example 2

Remarks

- Although we predict for $educ = 18$, in this example it does not make sense to do so
  - In this dataset, $educ = 17$ means masters and above
  - We included prediction at $educ = 18$ to make this point
- PI much wider than CI, unpredictable elements greater than estimation error
- However, specification seems obviously inappropriate
- Very likely to be getting biased predictions
- Model 2: $E(\ln earn \mid educ) = \beta_0 + \beta_1 educ + \beta_2 educ^2$

Example 2

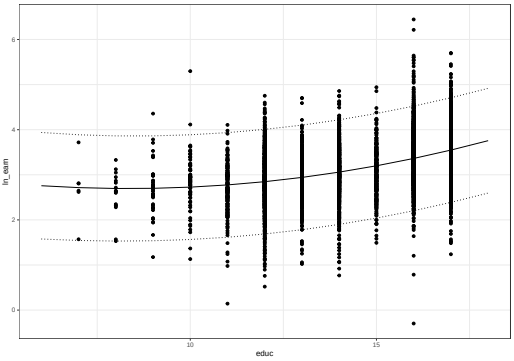
```
# Model 2
mdl2 <- lm(ln_earn ~ educ + I(educ^2), data=dat)

# Prediction
# -- Predict at these values
newdata <- data.frame(educ = seq(6, 18, 1))
# -- Generate predictions and "prediction interval"
prd2a <- predict(mdl2, newdata, se.fit=TRUE, interval="prediction", level=0.95)
# -- Generate predictions and "confidence interval"
prd2b <- predict(mdl2, newdata, se.fit=TRUE, interval="confidence", level=0.95)
```

Prediction with Linear Regression (Example, Prediction Interval)

```
print(cbind(newdata, prd2a$fit, se=prd2a$se.fit),
      row.names=F)
cat("df:", prd2a$df, "sghat:", prd2a$residual.scale)
```

```
 educ      fit      lwr      upr         se
    6 2.757544 1.576700 3.938388 0.11413042
    7 2.715622 1.543773 3.887472 0.08671738
    8 2.696460 1.530420 3.862499 0.06313147
    9 2.700056 1.537474 3.862638 0.04348219
   10 2.726412 1.565658 3.887165 0.02802613
   11 2.775527 1.615573 3.935480 0.01738867
   12 2.847401 1.687689 4.007112 0.01249768
   13 2.942033 1.782342 4.101725 0.01200074
   14 3.059425 1.899732 4.219119 0.01205032
   15 3.199577 2.039923 4.359230 0.01101936
   16 3.362487 2.202841 4.522132 0.01079879
   17 3.548156 2.388278 4.708034 0.01603157
   18 3.756584 2.595888 4.917281 0.02740707
df: 4943 sghat: 0.5914234
```

```
prd2adat <- cbind(newdata, prd2a$fit)
ggplot() + geom_point(data=dat, aes(y=ln_earn, x=educ)) +
  geom_line(data=prd2adat, aes(y=fit, x=educ)) +
  geom_line(data=prd2adat, aes(y=lwr, x=educ), linetype="dotted") +
  geom_line(data=prd2adat, aes(y=upr, x=educ), linetype="dotted") +
  theme_bw()
```

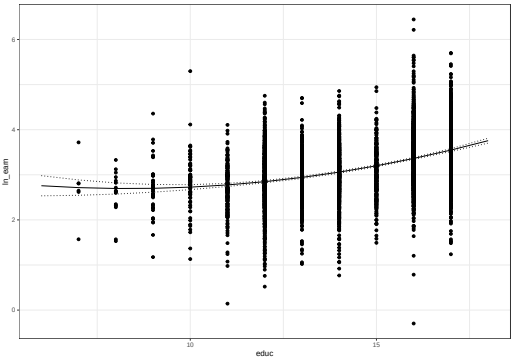


Prediction with Linear Regression (Example, Confidence Interval)

```
print(cbind(newdata, prd2b$fit, se=prd2b$se.fit),
      row.names=F)
cat("df:", prd2b$df, "sghat:", prd2b$residual.scale)
```

```
 educ      fit      lwr      upr         se
    6 2.757544 2.533797 2.981290 0.11413042
    7 2.715622 2.545618 2.885627 0.08671738
    8 2.696460 2.572694 2.820225 0.06313147
    9 2.700056 2.614812 2.785301 0.04348219
   10 2.726412 2.671468 2.781356 0.02802613
   11 2.775527 2.741437 2.809616 0.01738867
   12 2.847401 2.822900 2.871902 0.01249768
   13 2.942033 2.918507 2.965560 0.01200074
   14 3.059425 3.035802 3.083049 0.01205032
   15 3.199577 3.177974 3.221179 0.01101936
   16 3.362487 3.341316 3.383657 0.01079879
   17 3.548156 3.516727 3.579585 0.01603157
   18 3.756584 3.702854 3.810314 0.02740707
df: 4943 sghat: 0.5914234
```

```
prd2bdat <- cbind(newdata, prd2b$fit)
ggplot() + geom_point(data=dat, aes(y=ln_earn, x=educ)) +
  geom_line(data=prd2bdat, aes(y=fit, x=educ)) +
  geom_line(data=prd2bdat, aes(y=lwr, x=educ), linetype="dotted") +
  geom_line(data=prd2bdat, aes(y=upr, x=educ), linetype="dotted") +
  theme_bw()
```



Example 2 (MSPE)

```
# LOOCV
n <- nrow(dat)
SqErr_LOOCV <- 0
for (i in 1:n){
  Ypred_i <- predict(lm(ln_earn~educ + I(educ^2), data=dat[-i,]), newdata=dat[i,])
  SqErr_LOOCV <- SqErr_LOOCV + (pull(log(dat[i,"earn"])) - Ypred_i)^2
}
cat("Training MSE:", mean(mdl2$residuals^2),
    "  MSPE_LOOCV:", SqErr_LOOCV/n,
    "  C_p:", sum(mdl2$residuals^2)/(n-3)*(1+3/n))
```

```
Training MSE: 0.3495694   MSPE_LOOCV: 0.3499906   C_p: 0.3499938
```

- MSPE_LOOCV (Quadratic) $>$ Training Error (Quadratic)
- MSPE_LOOCV (Quadratic) $<$ MSPE_LOOCV (Linear)

Prediction with Linear Regression (Example, comments)

Remarks:

- Specification is probably not *exactly* correct, but much closer
- C.I. wider in quadratic case, but OOS MSPE (Quadratic) $<$ OOS MSPE (Linear)
  - more "complicated" model results in larger estimation variance
  - But as bias is likely to be much smaller, there is favorable bias-variance trade-off
- Example shows danger of extrapolation
- Can we try to eliminate bias further/completely?

Example 2

Two Approaches

- Allow more flexibility in specification
- Take a "local approach"

In this example

- increasing specification flexibility not likely to work (we demonstrate this in the next example)
- local approach can eliminate bias completely: since predictor educ is discrete, with many observations per level of educ in dataset, we can use

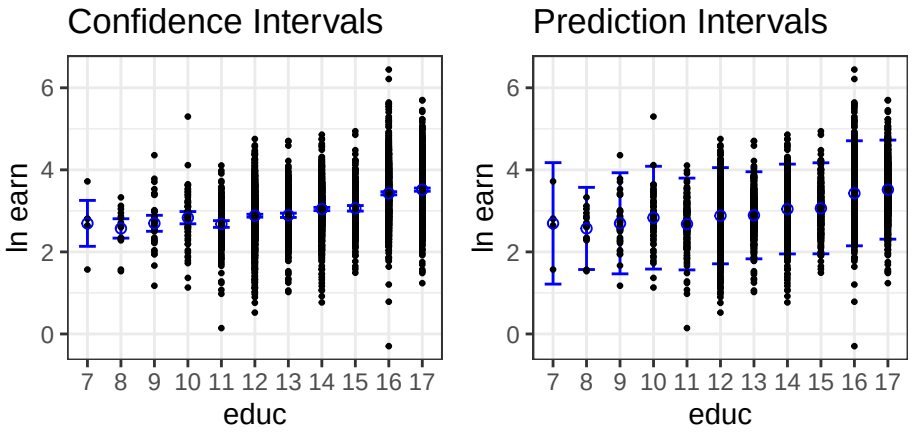
$$\widehat{\ln earn}(educ = j) = \frac{1}{n_j} \sum_{educ_i = j} \ln earn_i \quad \text{where} \quad n_j \text{ is no. obs with } educ = j$$

Example 2

```
mdl0 <- dat %>% group_by(educ) %>%
  summarize(m_ln_earn = mean(ln_earn), n = n(),
            v_ln_earn=var(ln_earn)/n, mspe_est=var(ln_earn)*(1+1/n)) %>%
  mutate(upp1 = m_ln_earn+2*sqrt(v_ln_earn), low1 = m_ln_earn-2*sqrt(v_ln_earn),
         upp2 = m_ln_earn+2*sqrt(mspe_est), low2 = m_ln_earn-2*sqrt(mspe_est))
#Plot predictions with CI and PI
p0a <- ggplot() +
  geom_errorbar(data=mdl0, aes(x = factor(educ), ymin=low1, ymax=upp1),
                width=0.5, color="blue", linewidth=0.5) +
  geom_point(data=dat, aes(x=factor(educ), y= ln_earn), size=0.5) +
  geom_point(data=mdl0, aes(x=factor(educ), y=m_ln_earn), size=1.5, color="blue", shape=1, stroke=0.5) +
  labs(title="Confidence Intervals", x = "educ", y="ln earn") + theme_bw()
p0b <- ggplot() +
  geom_errorbar(data=mdl0, aes(x = factor(educ), ymin=low2, ymax=upp2),
                width=0.5, color="blue", linewidth=0.5) +
  geom_point(data=dat, aes(x=factor(educ), y= ln_earn), size=0.5) +
  geom_point(data=mdl0, aes(x=factor(educ), y=m_ln_earn), size=1.5, color="blue", shape=1, stroke=0.5) +
  labs(title="Prediction Intervals", x = "educ", y="ln earn") + theme_bw()
```

Example 2

```
p0a | p0b
```



Example 2

Remarks:

- Cost of removing bias completely by taking this approach is probably not worth the additional variance
- Cannot extrapolate outside range
- *Can* be adapted for continous predictors (leave for more advanced courses)
- Not straightforward to extend to multi-predictor case

Example 2

(Digression) What if predictions of $earn$ are required, rather than $\ln earn$?

- If we assume

$$\ln earn \mid educ \overset{a}{\sim} \text{Normal}(\beta_0 + \beta_1 educ, \sigma^2)$$

then

$$earn \mid educ \overset{a}{\sim} \text{Log-Normal}(\beta_0 + \beta_1 educ, \sigma^2)$$

- $E(earn \mid educ) = \exp(\beta_0 + \beta_1 educ + \sigma^2/2)$
- $Median(earn \mid educ) = \exp(\beta_0 + \beta_1 educ)$
- Many would say, for an asymmetric distribution (such as Log-Normal), the median is more meaningful

Example 3

Example 3: More on bias-variance tradeoff

- Suppose $E(Y \mid X) = \exp(\beta_1 X + \sin(\beta_2 X))$ (not linear-in-parameters)
- In particular, suppose

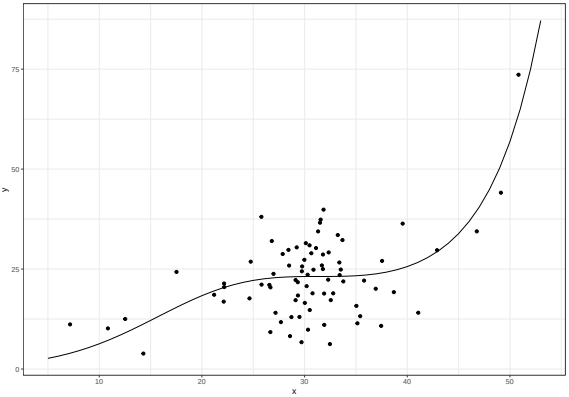
$$Y_i = \exp(0.1X + \sin(0.1X)) + \epsilon_i \,,\ \epsilon_i \overset{iid}{\sim} \text{Normal}(0, 5) \,,\ i = 1, \ldots, 50$$

- Try approximating with polynomial regressions, i.e., we'll assume
  - $E(Y \mid X) = \beta_0 + \beta_1 X$
  - $E(Y \mid X) = \beta_0 + \beta_1 X + \beta_2 X^2$
  - ...
  - $E(Y \mid X) = \beta_0 + \beta_1 X + \cdots + \beta_8 X^8$

Example 3

```
set.seed(1701)
n <- 80
# True Conditional Mean
x0 <- seq(5, 53, 1)
fx0 <- exp(0.1*x0 + sin(0.1*x0))
dat0 <- tibble(x=x0, fx=fx0)
# Simulated Data
x <- 6*rt(n,4) + 30
#x  <- rnorm(n, 30, 8)   # X taken as exogenous
y  <- exp(0.1*x + sin(0.1*x)) + rnorm(n, 0, 8)
dat1 <- tibble(x=x, y=y)
# plot data
p1 <- ggplot() +
  geom_point(data=dat1, aes(x=x, y=y)) +
  geom_line(data=dat0, aes(x=x, y=fx)) +
  theme_bw()
```

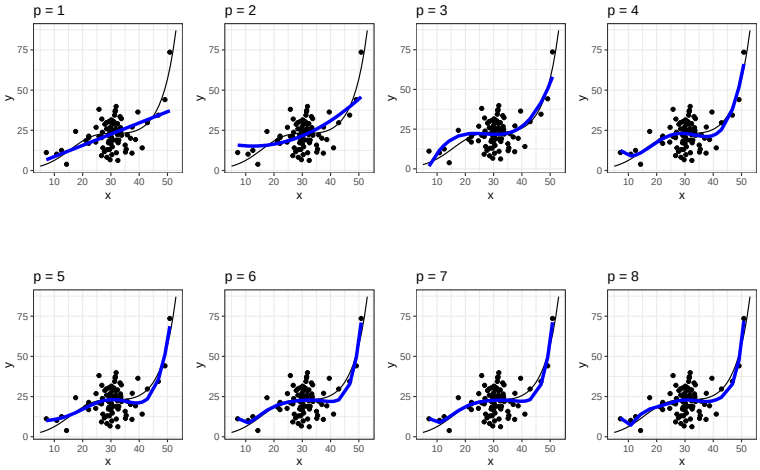
```
p1
```



Example 3

```
P <- 8
models <- list()
plots <- list()
IS_MSE <- rep(NA, P); MSPE_LOOCV <- rep(NA, P); C_p <- rep(NA, P)
n <- nrow(dat1)
for (p in 1:P){
  models[[p]] <- lm(y~poly(x,p), data=dat1)
  dat2 <- cbind(dat1, yhat=models[[p]]$fitted.values)
  plots[[p]] <- p1 + geom_line(data=dat2, aes(x=x, y=yhat), color='blue', linewidth=1.5) +
    ggtitle(paste0("p = ", p)) + theme(aspect.ratio=1)
  IS_MSE[p] <- mean(residuals(models[[p]])^2)
  SqErr_LOOCV <- 0
  for (i in 1:n){
    Ypred_i <- predict(lm(y~poly(x,p), data=dat1[-i,]), newdata=dat1[i,])
    SqErr_LOOCV <- SqErr_LOOCV + (pull(dat1[i,"y"]) - Ypred_i)^2
  }
  MSPE_LOOCV[p] <- SqErr_LOOCV/n
  C_p[p] <- sum(residuals(models[[p]])^2)/(n-p-1)*(1 + (p+1)/n)
}
```

Example 3



Example 3

Training MSE and OOS MSPE for polynomial regression models, $p = 1$ to $p = 8$

```
options(width=300)
names(IS_MSE) <- paste0("p=",1:P)
names(MSPE_LOOCV) <- paste0("p=",1:P)
names(C_p) <- paste0("p=",1:P)
rbind(IS_MSE, MSPE_LOOCV, C_p)
```

	p=1	p=2	p=3	p=4	p=5	p=6	p=7	p=8
IS_MSE	82.77587	78.07600	67.04228	60.37103	59.75112	59.14168	59.13657	58.90400
MSPE_LOOCV	90.38823	93.20659	85.70606	70.13853	73.82322	70.93364	84.23546	582.17952
C_p	87.02079	84.15984	74.09936	68.42050	69.44049	70.48392	72.27803	73.83741

Example 3

As p increases

- Model flexibility increases, Training MSE (IS_MSE) never increases
- OOS MSPE (LOOCV and C_p) falls then increases
  - Prediction bias falls, but prediction variance increases, as p increases
  - At some point, bias-variance tradeoff becomes unfavorable,

Model Selection:

- Choose model with lowest MSPE (can use LOOCV or C_p)
- Both suggest choosing $p = 4$ for this dataset

Example 3

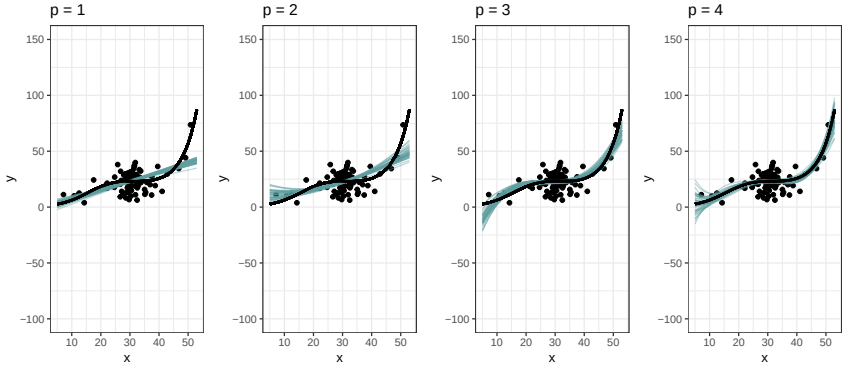
Illustration of the bias-variance trade-off

- plot polynomial models ($p = 1$ to $p = 8$) for 40 similar simulated datasets

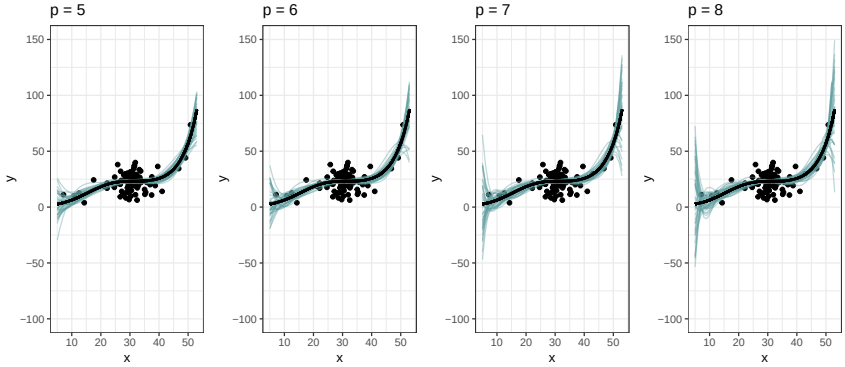
The following code is run for $p = 1$ to $p = 8$

```
p <- 1
p2_1 <- p1
for (i in 1:40){
  y  <- exp(0.1*x + sin(0.1*x)) + rnorm(n, 0, 6)
  dat3 <- tibble(x=x, y=y)
  yhat1 <- predict(lm(y ~ poly(x,p), data=dat3), newdata=tibble(x=x0))
  dat4 <- tibble(x=x0) %>% mutate(yhat=yhat1) %>% arrange(x)
  p2_1 <- p2_1 + geom_line(data=dat4, aes(x=x, y=yhat), color='cadetblue', alpha=0.4)
}
p2_1 <- p2_1 + geom_line(data=dat0, aes(x=x, y=fx), linewidth=1)
```

Example 3



Example 3



Remarks

There are other ways of introducing flexible regression models besides higher-ordered polynomials

- e.g., splines

These are left to other courses (e.g., machine learning)

Same principles apply

- increasing flexibility of regression specification reduces bias, increases variance
- at some point, bias-variance trade-off becomes unfavourable for MSPE
- can use cross-validation to determine optimal level of flexibility

Model Selection Tools

7.3 Model Selection Tools

Up to now, we chose among different specifications by minimizing MSPE estimated by C_p and $MSPE_{LOOCV}$

We can write C_p as $C_p = \widetilde{\sigma^2} + \frac{2p}{n}\widehat{\sigma^2} = \frac{n+p}{n-p}\widetilde{\sigma^2}$

(Recall Training MSE is $\widetilde{\sigma^2}$ and $p = k + 1$, the number of coefficients incl. intercept)

Can show $\frac{d}{dp}\frac{n+p}{n-p} > 0$

- As more parameters included, Training MSE falls, but "penalty factor" increases
- C_p goes down only if fall in Training MSE is greater than rise in penalty factor

Model Selection Tools

Other "model selection" methods:

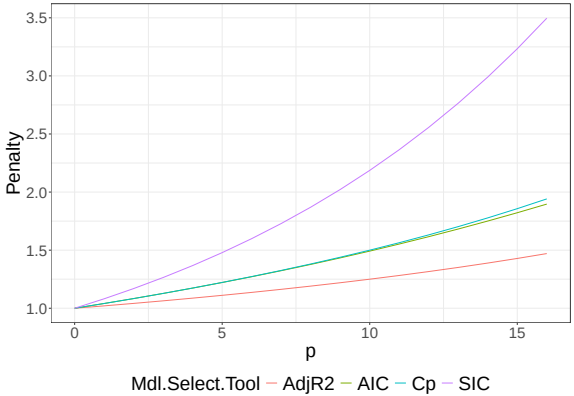
- maximize Adj. $R^2 = 1 - \dfrac{RSS/(n-p)}{TSS/(n-1)}$, equivalent to minimizing $\dfrac{n}{n-p}\widetilde{\sigma^2}$
- minimize "Akaike Information Criterion": $AIC = \exp\left\{\dfrac{2p}{n}\right\}\widetilde{\sigma^2}$
- minimize "Schwarz Information Criterion" or "Bayes Information Criterion": $SIC = n^{(\frac{p}{n})}\widetilde{\sigma^2}$

AIC derived via a "likelihood-based approach", SIC/BIC based on a Bayesian argument

- We will omit these arguments in this course

Model Selection Tools

```
n = 50
p <- seq(from=0, to=16, by=1)
pf <- data.frame(
  p = p,
  Cp = (n + p) / (n - p),
  AdjR2 = n / (n - p),
  AIC = exp(2 * p / n),
  SIC = n^(p/n))
p1 <- pf %>%
  pivot_longer(cols=2:5,
               names_to="Mdl.Select.Tool",
               values_to="Penalty") %>%
  ggplot(aes(x=p, y=Penalty,
             color=Mdl.Select.Tool)) +
  geom_line() + theme_bw() +
  theme(legend.position = "bottom",
        text=element_text(size=24),
        legend.text=element_text(size=24))
```



Model Selection Tools

Notes:

- Adjusted R^2 generally considered "too lenient", allows too many predictors
- AIC and C_p virtually identical (sometimes treated as equivalent)
- SIC is the most strict
- Depending on context, expression for AIC and SIC will be different (the expressions given here are for prediction with linear regression models)
- Different programs calculate AIC and SIC differently (certain transformations applied)
- Never compare AIC or C_p or SIC across different programs, or even across packages

Multiple Predictors

Section 7.4 Multiple Predictors

Returning to prediction of $\ln earn$ example

- more flexible specification unlikely to be helpful with just one predictor
- quadratic regression model works pretty well

Obvious that further improvements will require adding *new* predictors into the model

Incorporating more predictors straightforward with regression approach, e.g.,

$$\ln(earn) = \beta_0 + \beta_1 educ + \beta_2 educ^2 + \beta_3 height + \beta_4 feduc + \beta_5 male + \beta_6 totalwork + \beta_7 raceWhite + \beta_8 \ln(tenure) + \beta_9 \ln(tenure)^2 + \beta_{10} age + \epsilon$$

Example 4

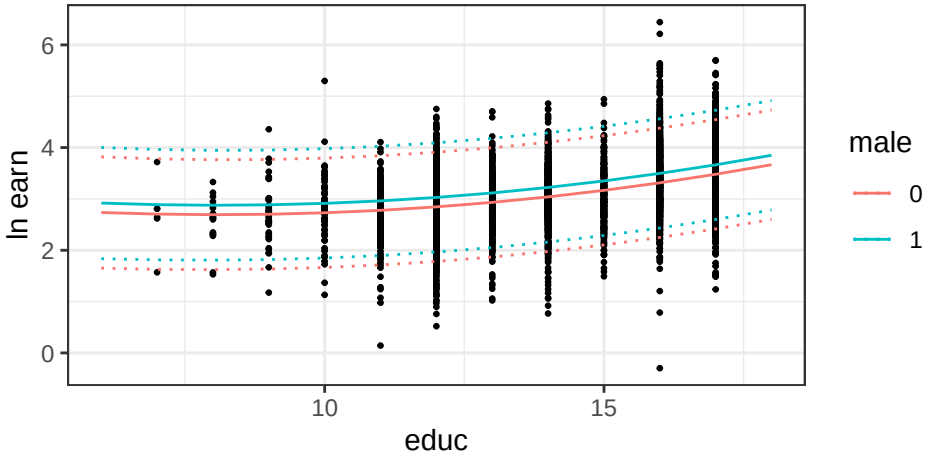
After estimating model, can predict at various values of $educ$, $wexp$, etc.

E.g. we can predict $\ln earn$ at various values of $educ$, for $whites$, at mean values of $height$, $tenure$ and age, and for $males$ and $females$ separately

```
dat <- read_csv("data\\earnings2019.csv", show_col_types=FALSE) %>%
  mutate(ln_earn=log(earn), ln_wexp=log(wexp), ln_tenure=log(tenure),
         raceWhite=recode(race, "White"=1, "Black"=0, "Other"=0))
mdl4 <- lm(ln_earn ~ educ + I(educ^2) + feduc + height + male + totalwork + age +
             ln_tenure + I(ln_tenure^2) + raceWhite, data=dat)
# Prediction
newdata <- data.frame(educ = rep(seq(6, 18, 1), each=2), height = mean(dat$height),
                      male=rep(c(0,1), 13), raceWhite=1, ln_tenure = mean(dat$ln_tenure),
                      feduc=mean(dat$feduc), totalwork=mean(dat$totalwork), age = mean(dat$age))
prd4 <- predict(mdl4, newdata, se.fit=TRUE, interval="prediction", level=0.95)
prd4dat <- cbind(newdata, prd4$fit)
p1 <- ggplot() + geom_point(data=dat, aes(y=ln_earn, x=educ), size=0.5) +
  geom_line(data=prd4dat, aes(y=fit, x=educ, group=male, color=as.factor(male))) +
  geom_line(data=prd4dat, aes(y=lwr, x=educ, group=male, color=as.factor(male)), linetype="dotted") +
  geom_line(data=prd4dat, aes(y=upr, x=educ, group=male, color=as.factor(male)), linetype="dotted") +
```

Example 4

```
p1
```



Example 4

In-Sample Fit, LOOCV MSPE

```
# LOOCV
n <- nrow(dat)
SqErr_LOOCV <- 0
for (i in 1:n){
  Ypred_i <- predict(lm(ln_earn ~ educ + I(educ^2) + feduc + height + male + totalwork + age +
                        ln_tenure + I(ln_tenure^2) + raceWhite, data=dat[-i,]),
                     newdata=dat[i,])
  SqErr_LOOCV <- SqErr_LOOCV + (pull(dat[i,"ln_earn"]) - Ypred_i)^2
}
cat("In-Sample MSE: ", mean(mdl4$residuals^2),
    "   MSPE_LOOCV: ", SqErr_LOOCV/n,
    "   C_p: ", sum(mdl4$residuals^2)/(n-11)*(1+11/n))
```

```
In-Sample MSE:  0.2935153    MSPE_LOOCV:  0.2949743    C_p:  0.2948237
```

Model Selection in Predictive Regressions

With multiple predictors, the main questions are:

- Which predictors should we include?
  - to include age as predictor? $male$? $tenure$?
- How to choose functional form specification?
  - to include $\ln tenure$? $(\ln tenure)^2$? $(\ln tenure) \times male$?
- In the following statements, we consider X and X^2 as separate predictors

Model Selection in Predictive Regressions

Possible approaches

- Use hypothesis testing (not popular)
- Try all possible models out of p preds., choose model that minimizes C_p (or AIC), SIC
  - 2^p possible models! $p = 10 \Rightarrow 1024$ possible models
- Forward Stepwise Selection
  - Choose 1-predictor model with smallest training error out of p predictors (M_1)
  - Choose 2-predictor model (M_2) with smallest training error by adding one predictor out of remaining $p - 1$ predictors, and so on
  - Choose model with smallest AIC, SIC out of $M_0, \ldots, M_p$
  - Total of $1 + \frac{p(p+1)}{2}$ models considered, not guaranteed to give best of 2^p models

Model Selection in Predictive Regressions

- Backward Stepwise Selection
  - Start with model containing all predictors (M_p)
  - Choose $(p-1)$-predictor model with smallest training error by removing 1 predictor out of p predictors (M_{p-1})
  - Choose $p-2$-predictor model (M_{p-1}) with smallest training error by removing one predictor out of remaining $p-2$ predictors, and so on
  - Choose model with smallest AIC, SIC out of $M_0, \ldots, M_p$
  - Total of $1 + \frac{p(p+1)}{2}$ models considered, not guaranteed to give best of 2^p models

Example 5

Application to full earnings2019.csv dataset

```
## THE DATASET
dat <- read_csv("data\\earnings2019.csv", show_col_types=FALSE) %>%
  mutate(ln_earn=log(earn), ln_wexp=log(wexp), ln_tenure=log(tenure), educ2 = educ^2,
         ln_wexp2=log(wexp)^2, ln_tenure2 = log(tenure)^2, age2=age^2) %>%
  select(-c(earn))
cat(names(dat)[1:8], "\n")
cat(names(dat)[9:17])
```

```
age height educ feduc meduc tenure wexp race
male totalwork ln_earn ln_wexp ln_tenure educ2 ln_wexp2 ln_tenure2 age2
```

Example 5

Using `regsubsets()` from `leaps` package

`race` will be converted to 3 dummy variables `raceWhite`, `raceMale`, `raceOther`, and two included

`regsubsets` from `leaps` package will search through all possible models

```
# from "leaps" package, full search
regfit.full <- regsubsets(ln_earn~., dat, nvmax=16)
reg.summary <- summary(regfit.full)
```

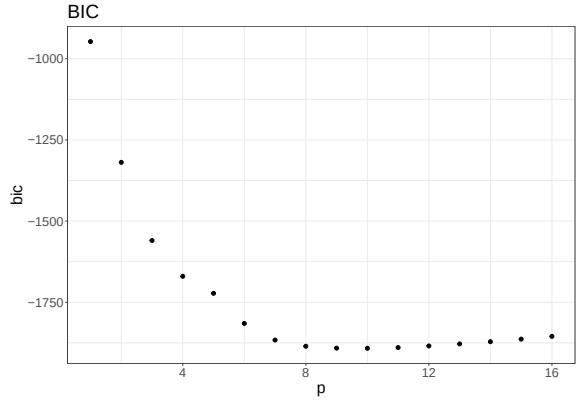
Example 5

```
options(width=300) # To allow me to display all columns on my slides
reg.summary$which
```

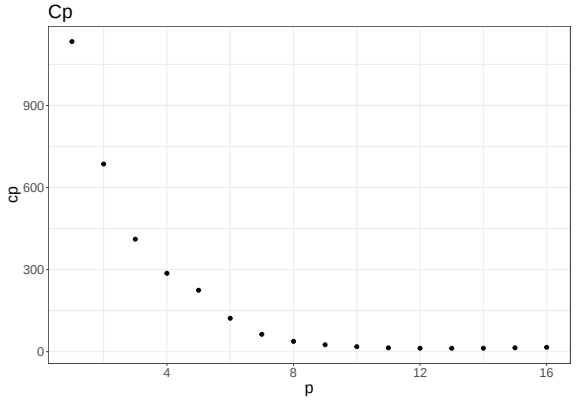
	(Intercept)	age	height	educ	feduc	meduc	tenure	wexp	raceOther	raceWhite	male	totalwork	ln_wexp	ln_tenure	educ2	ln_wexp2	ln_tenure2
1	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
2	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
3	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
4	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
5	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
6	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
7	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
8	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
9	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
10	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
11	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
12	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
13	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
14	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
15	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
16	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

Example 5

```
ggplot(data=data.frame(p=1:16, bic=reg.summary$bic)) +
  geom_point(aes(x=p, y=bic),size=2) + theme_bw() +
  theme(text=element_text(size=20)) + ggtitle("BIC")
```




```
ggplot(data=data.frame(p=1:16, cp=reg.summary$cp)) +
  geom_point(aes(x=p, y=cp),size=2) + theme_bw() +
  theme(text=element_text(size=20)) + ggtitle("Cp")
```



Example 5

```
options(width=300)
print(paste0("BIC: ", which.min(reg.summary$bic)))
round(coef(regfit.full,10)[1:5], 4); round(coef(regfit.full,10)[6:11], 4); cat("\n")
print(paste0("Cp: ", which.min(reg.summary$cp)))
round(coef(regfit.full,13)[1:7], 4); round(coef(regfit.full,13)[8:14], 4)
```

```
[1] "BIC: 10"
(Intercept)         age      height        educ       feduc
     1.0123      0.0581      0.0106     -0.1410      0.0200
 raceWhite        male  totalwork       educ2 ln_tenure2        age2
    0.1700      0.1985     -0.0001      0.0092      0.0370     -0.0006

[1] "Cp: 13"
(Intercept)         age      height        educ       feduc   raceOther   raceWhite
     0.8055      0.0586      0.0116     -0.1263      0.0178      0.0696      0.1870
      male  totalwork    ln_wexp  ln_tenure       educ2 ln_tenure2        age2
    0.1906     -0.0001     -0.0136      0.0499      0.0086      0.0233     -0.0006
```

Use options `method="backward"` and `method="forward"` in `regsubsets()` for backward stepwise selection and forward stepwise selection resp.

Alt Approach: Penalized Least Square

- Ridge Regression

$$\min \left\{ \sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \quad \text{for some} \quad \lambda > 0$$

- LASSO (Least Absolute Shrinkage and Selection Operator)

$$\min \left\{ \sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \quad \text{for some} \quad \lambda > 0$$

- Elastic Net

$$\min \left\{ \sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2 \right] \right\} \quad \text{for some} \quad \lambda > 0$$

Model Selection in Predictive Regressions

- Since $\lambda > 0$, there is "incentive to shrink" estimators toward zero, relative to OLS
- Since OLS estimators are unbiased, the Penalized Least Squares (Ridge/LASSO/Elastic Net) estimators are biased
- But because we are biasing toward zero, this also reduces estimator variance
- May have favorable bias-variance trade-off in terms of minimizing MSPE

E.g. In Example 1, we used sample mean to predict new observation from population

- Unbiased prediction (in fact, best linear unbiased prediction)
- But in some cases, $\frac{1}{n+\lambda}\sum_{i=1}^{n} Y_i = \frac{n}{n+\lambda}\overline{Y}$ will give a smaller MSPE

Model Selection in Predictive Regressions

Ridge Regression:

- Can show that

$$\hat{\beta}_{ridge} = (X^{\mathrm{T}}X + \lambda I_k)^{-1}X^{\mathrm{T}}y$$

and if $Var(\epsilon) = \sigma^2 I_n$, then

$$Var(\hat{\beta}_{ridge}) = \sigma^2(X^{\mathrm{T}}X + \lambda I_k)^{-1}X^{\mathrm{T}}X(X^{\mathrm{T}}X + \lambda I_k)^{-1}$$

- Recall for OLS estimation that $X^{\mathrm{T}}X$ has no inverse if column rank$(X) < k+1$
- However, $X^{\mathrm{T}}X + \lambda I_k$ always has an inverse so this works even in situations with perfect multicollinearity

No formula available for LASSO — $\hat{\beta}_{LASSO}$ must be found "numerically"

Model Selection in Predictive Regressions

- We apply this to our data, using `age`, `height`, `educ`, `feduc`, `meduc`, `tenure`, `wexp`, `raceOther`, `raceWhite`, `male`, `totalwork`, `ln_wexp`, `ln_tenure`, `age^2`, `educ^2`, `ln_tenure^2`, `ln_wexp^2` as our starting set of predictors

```
dat <- read_csv("data\\earnings2019.csv", show_col_types=FALSE) %>%
  mutate(ln_earn=log(earn), ln_wexp=log(wexp), ln_tenure=log(tenure)) %>%
  select(-c(earn))
```

- use `glmnet()` function from the `glmnet` package to do Penalized Least Squares
- We consider a range of values of λ
- All variables will be "standardized" by the `glmnet()` function

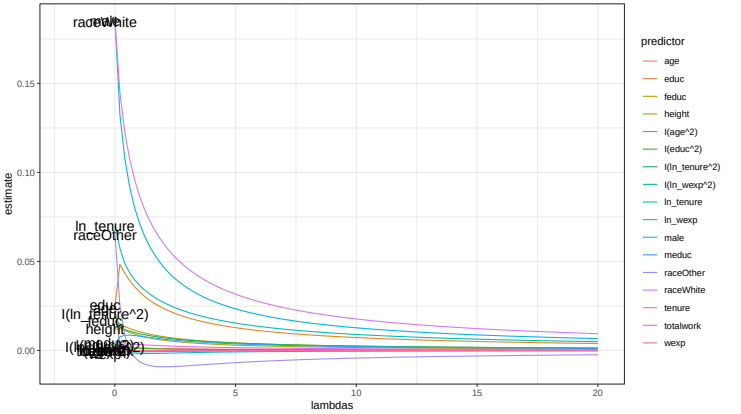
Example 6 Ridge Regression

```
# Set X matrix
x <- model.matrix(ln_earn~. + I(age^2) + I(educ^2) +
                    I(ln_tenure^2) + I(ln_wexp^2), dat)[,-1]
y <- dat$ln_earn
lambdas <- seq(from=20, to=0.01, length.out=100)
# alpha=0 for Ridge Regression
ridge.mod <- glmnet(x, y, alpha=0, lambda=lambdas)
temp <- coef(ridge.mod) %>% as.matrix() %>% t() %>%
  cbind("lambdas"=lambdas) %>% as_tibble() %>% select(-c("(Intercept)"))
# Set up plot of estimates vs lambdas
p1 <- temp %>%
  pivot_longer(cols=-c(lambdas), names_to="predictor", values_to="estimate") %>%
  ggplot(aes(x=lambdas, y=estimate, group=predictor, color=predictor)) +
  geom_line() + theme_bw()
```

Example 6 Ridge Regression

```
J = dim(temp)[2]
p1 <- p1 +
  annotate(
    geom="text",
    x=rep(-0.4, J-1),
    y=as.numeric(
      temp[100, 1:(J-1)]
    ),
    label=colnames(temp)[1:(J-1)],
    size=5
  ) +
  xlim(-2, 20)
```

```
p1
```



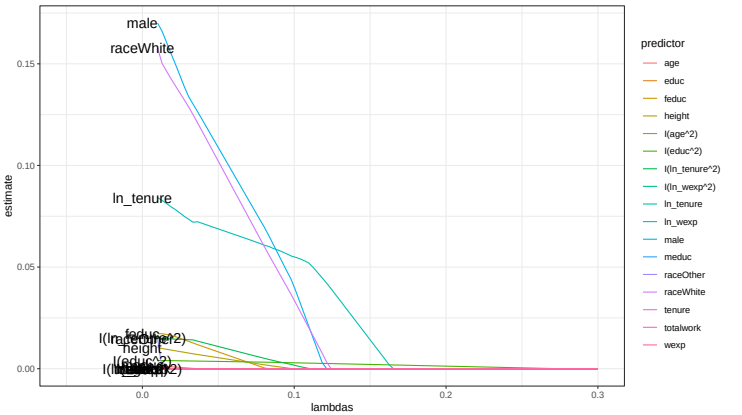
Example 6 LASSO

```
# Set up X matrix
x <- model.matrix(ln_earn~. + I(age^2) + I(educ^2) + I(ln_tenure^2) +
                    I(ln_wexp^2), dat)[,-1]
y <- dat$ln_earn
lambdas <- seq(from=0.3, to=0.01, length.out=100)
ridge.mod <- glmnet(x, y, alpha=1, lambda = lambdas)  # alpha=1 for LASSO
temp <- coef(ridge.mod) %>% as.matrix() %>% t() %>%
  cbind("lambdas"=lambdas) %>% as_tibble() %>% select(-c("(Intercept)"))
p1 <- temp %>%
  pivot_longer(cols=-c(lambdas),
               names_to="predictor",
               values_to="estimate") %>%
  ggplot(aes(x=lambdas, y=estimate, group=predictor, color=predictor)) +
  geom_line() + theme_bw()
```

Example 6 LASSO

```
J = dim(temp)[2]
p1 <- p1 +
  annotate(
    geom="text",
    x=rep(0, J-1),
    y=as.numeric(
      temp[100, 1:(J-1)]),
      label=colnames(temp)[1:(J-1)],
    size=5
    ) +
  xlim(-0.05, 0.3)
```

p1



Penalized Least Squares

Ridge Regression reduces parameter values smoothly but generally does not eliminate predictors ("Shrinkage" or "Regularization")

Lasso performs regularization *and* predictor selection

Elastic Net is combination of the two

Which λ to use? How to do prediction with estimated model?

- Cross Validation
- Explore further in Assignment

Roadmap

- (Previous) Session 1: Statistics Review
- (Previous) Session 2: Simple Linear Regression
- (Previous) Session 3: Estimator Standard Errors; Multiple Linear Regression
- (Previous) Session 4: Matrix Algebra
- (Previous) Session 5: OLS using Matrix Algebra
- (Previous) Session 6: Hypothesis Testing
- **This Session 7: Prediction**
- *Next Session 8: Instrumental Variable Regression*
- Session 9: Logistic and Other Regressions
- Session 10: Panel Data Regressions
- Session 11: Introduction to Time Series
- Session 12: Time Series Regressions