STAT 471: Homework 3

Name

Due: October 24, 2021 at 11:59pm

Contents

Instructions

Setup

Pull the latest version of this assignment from Github and set your working directory to `stat-471-fall-2021/homework/homework-3`. Consult the getting started guide if you need to brush up on `R` or `Git`.

Collaboration

The collaboration policy is as stated on the Syllabus:

> "Students are permitted to work together on homework assignments, but solutions must be written up and submitted individually. Students must disclose any sources of assistance they received; furthermore, they are prohibited from verbatim copying from any source and from consulting solutions to problems that may be available online and/or from past iterations of the course."

In accordance with this policy,

Please list anyone you discussed this homework with:

Alex Chen

Please list what external references you consulted (e.g. articles, books, or websites):

Stackoverflow, API

Writeup

Use this document as a starting point for your writeup, adding your solutions after "**Solution**". Add your R code using code chunks and add your text answers using **bold text**. Consult the preparing reports guide for guidance on compilation, creation of figures and tables, and presentation quality.

Programming

The `tidyverse` paradigm for data wrangling, manipulation, and visualization is strongly encouraged, but points will not be deducted for using base `R`.

Grading

The point value for each problem sub-part is indicated. Additionally, the presentation quality of the solution for each problem (as exemplified by the guidelines in Section 3 of the preparing reports guide will be evaluated on a per-problem basis (e.g. in this homework, there are three problems). There are 100 points possible on this homework, 85 of which are for correctness and 15 of which are for presentation.

Submission

Compile your writeup to PDF and submit to Gradescope.

We'll need to use the following R packages:

```
library(kableExtra) # for printing tables
```

```
## Warning: package 'kableExtra' was built under R version 4.1.1
```

```
library(cowplot)    # for side by side plots
```

```
## Warning: package 'cowplot' was built under R version 4.1.1
```

```
library(glmnet)     # to run ridge and lasso
```

```
## Warning: package 'glmnet' was built under R version 4.1.1
```

```
library(ISLR2)      # necessary for College data
library(pROC)       # for ROC curves
```

```
## Warning: package 'pROC' was built under R version 4.1.1
```

```
library(tidyverse)
```

```
## Warning: package 'tibble' was built under R version 4.1.1
## Warning: package 'tidyr' was built under R version 4.1.1
## Warning: package 'readr' was built under R version 4.1.1
```

```
library(glmnetUtils)
```

```
## Warning: package 'glmnetUtils' was built under R version 4.1.1
```

We'll also need the `plot_glmnet` function from Unit 3 Lecture 3:

```
# install.packages("scales")              # dependency of plot_glmnet
source("../../functions/plot_glmnet.R")[]
```

```
## $value
## function (elnet_fit)
## {
##     alpha = elnet_fit$alpha
##     error = sapply(elnet_fit$modlist, function(mod) {
##         min(mod$cvm)
##     })
##     out = elnet_fit$modlist[[which.min(error)]]
##     out$alpha = alpha[which.min(error)]
##     out$use.model.frame = elnet_fit$use.model.frame
##     out$call$formula = elnet_fit$call$formula
##     out
## }
##
## $visible
## [1] FALSE
```

1 Framingham Heart Study

Heart disease is the leading cause of the death in United States, accounting for one out of four deaths. It is important to identify risk factors for this disease. Many studies have indicated that high blood pressure, high cholesterol, age, gender, race are among the major risk factors.

Starting from the late 1940s, National Heart, Lung and Blood Institute (NHLBI) launched its famous Framingham Heart Study. By now subjects of three generations together with other people have been monitored and followed in the study. Over thousands research papers have been published using these longitudinal data sets.

Using a piece of the data gathered at the beginning of the study, we illustrate how to identify risk factors of heart disease and how to predict =this disease.

The data contain the following eight variables for each individual:

Variable	Description
`HD`	Indicator of having heart disease or not
`AGE`	Age
`SEX`	Gender
`SBP`	Systolic blood pressure
`DBP`	Diastolic blood pressure
`CHOL`	Cholesterol level
`FRW`	age and gender adjusted weight
`CIG`	Self-reported number of cigarettes smoked each week

1.1 Data import and exploration

i. Import the data from `stat-471-fall-2021/data/Framingham.dat` into a tibble called `hd_data`, specifying all columns to be integers except `SEX`, which should be a factor. Rename `Heart Disease?` to `HD`, and remove any rows containing `NA` values using `na.omit()`.

```r
# Importing the table
hd_data <- read_csv(file = "../../data/Framingham.dat") %>%
  na.omit() %>% # Removing NA
  rename("HD" = 'Heart Disease?') %>%
  mutate_at(vars(SEX), # Specifying factors
            list(factor)) %>%
  mutate_at(vars(-SEX), # Specifying integer
            list(as.numeric))
#sum(is.na(hd_data))
```

ii. What is the number of people in this data? What percentage of them have heart disease?

```r
# Counting the people and people with heart disease
count(hd_data)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  1393
```

```r
count(hd_data[hd_data$HD==1,])
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   307
```

There are 1393 people in the data. 307/1393 or 22.03% of them have heart disease.

iii. Split `hd_data` into training (80%) and test (20%) sets, using the rows in `train_samples` below for training. Store these in tibbles called `hd_train` and `hd_test`, respectively.

```r
# Splitting the sets
set.seed(5) # seed set for reproducibility (DO NOT CHANGE)
```

```r
n = nrow(hd_data)
train_samples = sample(1:n, round(0.8*n))
hd_train <- hd_data[train_samples,]
hd_test <- hd_data[-train_samples,]
```

iv. Display the age distribution in `hd_train` with a plot. What is the median age?

```r
# Plotting age distribution
ggplot(hd_train, aes(x = AGE)) +
  geom_histogram(bins = 15) +
  theme_bw() +
  labs(x = "Age", y = "Number of Patients")
```

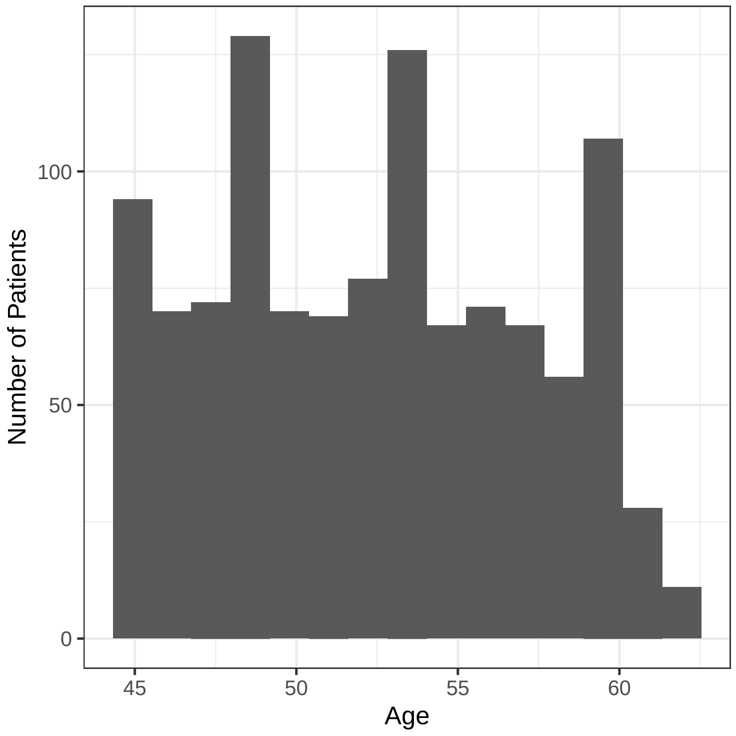

Figure 1: Age Distribution of Patients

```r
# Getting median
median(hd_train$AGE)
```

```
## [1] 52
```

The median age is 52.

v. Use a plot to explore the relationship between heart disease and systolic blood pressure in `hd_train`. What does this plot suggest?

```r
# Plotting HD vs. Blood Pressure
ggplot(hd_train, aes(x = SBP, y =HD)) +
  geom_point() +
  theme_bw()  +
  geom_jitter(height=0.05) +
  labs(x = "Systolic Blood Pressure", y = "Prob(Heart Disease=1)")
```

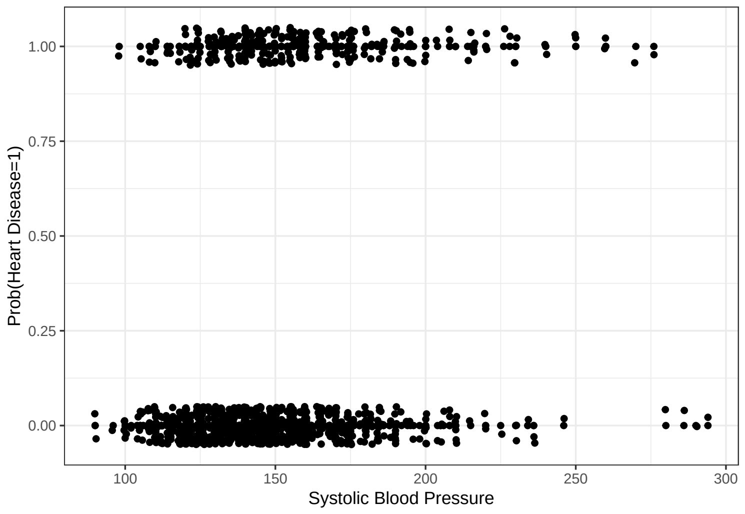

Figure 2: Heart Disease by Systolic Blood Pressure

Most of the points for those with heart disease and those without have roughly similar systolic blood pressure levels. There is a lot of overlap between the two. However, it does appear that those with heart disease do have slightly higher levels. Their distribution appears to be shifted rightwards ever so slightly. Those without heart disease do have some outlier systolic blood pressure levels that are greater than those with heart disease.

1.2 Univariate logistic regression

In this part, we will study the relationship of heart disease with systolic blood pressure using univariate logistic regression.

1.2.1 Logistic regression building blocks

Let's take a look under the hood of logistic regression using a very small subset of the data.

i. Define and print a new data frame called `hd_train_subset` containing `HD` and `SBP` for the individuals in `hd_train` who smoke (exactly) 40 cigarettes per week and have a cholesterol of at least 260.

```
# Getting the subset
hd_train_subset <- hd_train %>%
  filter(CIG == 40) %>%
  filter(CHOL >= 260) %>%
  select(HD, SBP)
```

ii. Write down the logistic regression likelihood function using the observations in `hd_train_subset`.

The logistic regression likelihood function is:

$$\frac{(e^{\beta_0+\beta_1*190})*(e^{\beta_0+\beta_1*150})*(e^{\beta_0+\beta_1*130})}{(1+e^{\beta_0+\beta_1*190})(1+e^{\beta_0+\beta_1*142})(1+e^{\beta_0+\beta_1*150})(1+e^{\beta_0+\beta_1*130})(1+e^{\beta_0+\beta_1*130})}$$

.

iii. Find the MLE based on this subset using `glm()`. Given a value of `SBP`, what is the estimated probability $\mathbb{P}[\text{HD}=1|\text{SBP}]$?

```
# Finding the logistic regression and Coefficients
hd_subset_fit <- glm(HD ~ SBP, data = hd_train_subset, family = "binomial")
coef(hd_subset_fit)
```

```
## (Intercept)         SBP
##    -10.1427      0.0737
```

```
# Calculating the MLE
hd_train_subset <- hd_train_subset %>%
  mutate(logodds = SBP*0.0737 - 10.1427)
```

The MLE for this subset is 0.0679. The estimated probability of heart disease given a particular SBP is:

$$\frac{(e^{-0.9014+0.0101*SBP})}{(1+e^{-0.9014+0.0101*SBP})}$$

iv. Briefly explain how the fitted coefficients in part iii were obtained from the formula in part ii.

The coefficients in part 3 are the ones that maximize the function found in part 2. This produces the value of the MLE that I found in part 3. I believe R does this through the BFGS algorithm.

v. To illustrate this, fix the intercept at its fitted value and define the likelihood as a function of β_1. Then, plot this likelihood in the range [0, 0.1], adding a vertical line at the fitted value of β_1. What do we see in this plot? [Hints: Define the likelihood as a function in R via `likelihood = function(beta_1)(???)`. Use `stat_function()` to plot it.]

```
# Defining the likelihood function
likelihood <- function(beta_1)(exp(-10.1427*3 + 470*beta_1)
                               /(1+exp(-10.1427 + 190*beta_1))
                               /(1+exp(-10.1427 + 142*beta_1))
                               /(1+exp(-10.1427 + 150*beta_1))
                               /(1+exp(-10.1427 + 130*beta_1))
                               /(1+exp(-10.1427 + 130*beta_1)))
#Plotting the likelihood of Betas
ggplot() +
  xlim(0, 0.1) +
  stat_function(fun =likelihood) +
  geom_vline(aes(xintercept=0.0737)) +
  theme_bw()+
  labs(x = "Beta 1", y = "Likelihood")
```

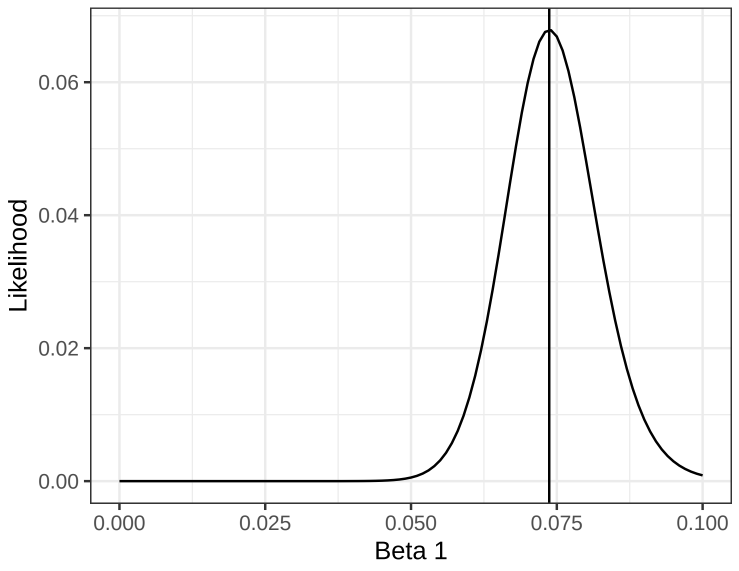

Figure 3: Likelihood Across Different Values of Beta 1

```r
#likelihood(0.0737)
```

1.2.2 Univariate logistic regression on the full data

i. Run a univariate logistic regression of `HD` on `SBP` using the full training data `hd_train`. According to the estimated coefficient, how do the odds of heart disease change when `SBP` increases by 1?

```r
#Running the Univariate Logistic Regression
hd_logistic <- glm(HD ~ SBP, hd_train, family = "binomial")
coef(hd_logistic)
```

```
## (Intercept)         SBP
##     -3.7558      0.0166
```

The odds of heart disease are multiplied by $e^{0.016}$.

ii. Plot the logistic regression fit along with a scatter plot of the data. Use `geom_jitter()` instead of `geom_point()` to better visualize the data. Based on the plot, roughly what is the estimated probability of heart disease for someone with `SBP` $= 100$?

```r
#Plotting the Logistic Regression on the Heart Disease Plot
hd_train %>%
  ggplot(aes(x = SBP, y = HD))+
  geom_jitter(height = .05) +
  geom_smooth(method = "glm",
  formula = "y~x",
  method.args = list(family = "binomial"),
  se = FALSE) +
  xlab("Systolic Blood Pressure") +
  ylab("Prob(Heart Disease=1)") +
  theme_bw()
```

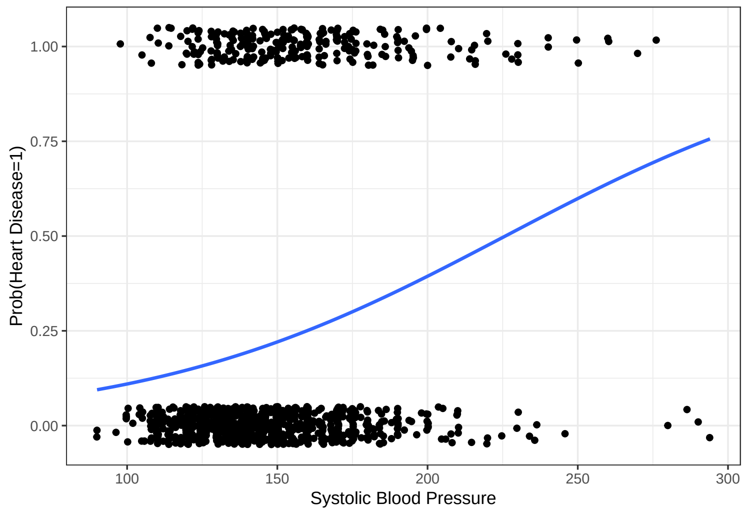

Figure 4: Title

Based on Figure 4, the probability of someone having heart disease with SBP = 100 is about 20%.

1.3 Multiple logistic regression

i. Run a multiple logistic regression of HD on all of the other variables in the data. Other things being equal, do the estimated coefficient suggest that males are more or less prone to heart disease? Other things being equal, what impact does an increase in AGE by 10 years have on the odds of heart disease (according to the estimated coefficients)?

```r
## Running a multivariate logistic regression
hd_logistic_all <- glm(HD ~ ., hd_train, family = "binomial")
coef(hd_logistic_all)
```

```
## (Intercept)         AGE     SEXMALE         SBP         DBP        CHOL
##    -9.43399     0.06151     0.96813     0.01568     0.00302     0.00439
##         FRW         CIG
##     0.00618     0.01080
```

Being male does seem to imply that people are more prone to heart disease, as the coefficient for SEXMALE is positive. Additionally, being older by ten years increases the odds of heart disease by $e^{0.615}$.

ii. Mary is a patient with the following readings: AGE=50, SEX=FEMALE, SBP=110, DBP=80, CHOL=180, FRW=105, CIG=0. According to the fitted model, what is the estimated probability Mary has heart disease?

```r
# Creating a tribble representing Mary
mary <- tribble(
  ~AGE, ~SEX, ~SBP, ~DBP, ~CHOL, ~FRW, ~CIG,
  50, "FEMALE", 110, 80, 180, 105, 0
)
# Getting the probability
fitted_prob_mary <-  predict(hd_logistic_all,
  newdata = mary,
  type = "response")
head(fitted_prob_mary)
```

```
##      1
## 0.0496
```

Mary's estimated probability of heart disease is 0.0496.

iii. What are the misclassification rate, false positive rate, and false negative rate of the logistic regression classifier (based on the probability threshold of 0.5) on `hd_test`? Print these in a nice table. Plot the ROC curve, and add a red point to the plot corresponding to the threshold of 0.5 (recalling that the true positive rate is one minus the false negative rate). What is the AUC? How does it compare to that of a classifier that guesses randomly?

```
fitted_probabilities = predict(hd_logistic_all,
  newdata = hd_test,
  type = "response")
predictions = as.numeric(fitted_probabilities > 0.5)
hd_test <- hd_test %>%
  mutate(predicted_hd = predictions)
hd_test %>%
  select(HD, predicted_hd) %>%
  table()
```

```
##    predicted_hd
## HD    0   1
##   0 217   5
##   1  51   6
```

```
tribble(
  ~"Misclssification Rate", ~"False Positive Rate", ~"False Negative Rate",
  0.201, 0.0225, 0.895
) %>%
  kable(format = "latex", row.names = NA,
      booktabs = TRUE, digits = 3,
      caption = "Error Rates of the Logistic Regression") %>%
  kable_styling(position = "center", latex_options = "HOLD_position")
```

Table 2: Error Rates of the Logistic Regression

Misclssification Rate	False Positive Rate	False Negative Rate
0.201	0.022	0.895

The misclassification rate is 56/279. The false positive rate is 5/222. The false negative rate is 51/57.

```
# Getting the ROC data
roc_data <-  roc(hd_test %>% pull(HD),
  fitted_probabilities)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Plotting the AUC Curve
tibble(FPR = 1-roc_data$specificities,
  TPR = roc_data$sensitivities) %>%
  ggplot(aes(x = FPR, y = TPR)) +
  geom_line() +
  geom_abline(slope = 1, linetype = "dashed") +
  geom_point(x = 5/222, y = 1-51/57, colour = "red") +
  theme_bw()
```

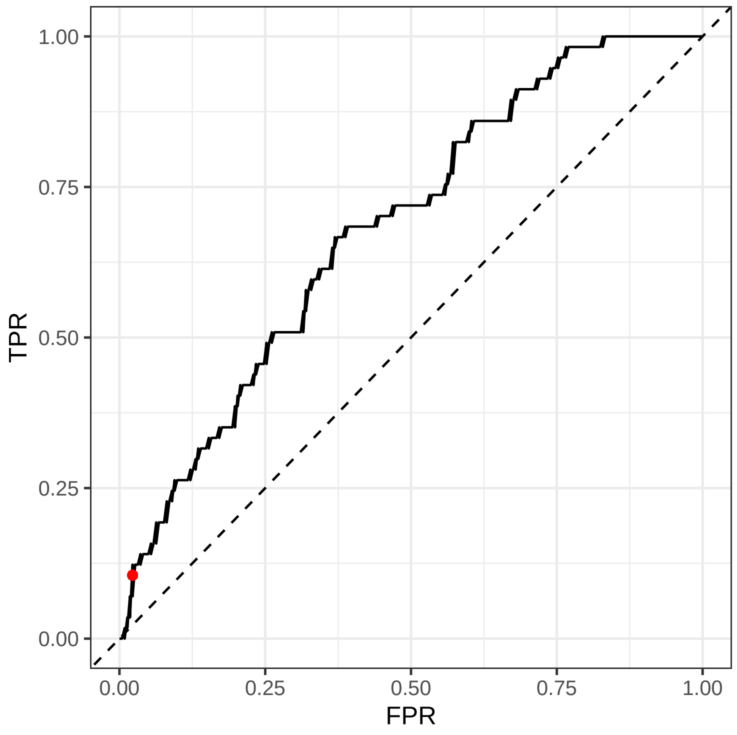

Figure 5: AUC Curve

The AUC is 0.681. 0.681 is better than a classifier that guesses randomly (that's AUC would be around 0.5), but it is not significantly better. AUC's of atleast 0.9 are common in industry.

2 College Applications

Next, we will examine the `College` dataset from the `ISLR` package. According to the documentation, these data contain "statistics for a large number of US Colleges from the 1995 issue of US News and World Report." The goal will be to predict the acceptance rate.

Next, let us make a few small adjustments to the data:

```
college_data = ISLR2::College %>%
  bind_cols(Name = rownames(ISLR2::College)) %>% # add college names
  relocate(Name) %>%                              # put name column first
  mutate(Accept = Accept/Apps) %>%                # redefine `Accept`
  select(-Private,-Apps) %>%                      # remove `Private` and `Apps`
  as_tibble()                                     # cast to tibble
```

Now, let's take a look at the data and its documentation:

```
college_data                                    # take a look at the data
```

```
## # A tibble: 777 x 17
##    Name       Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate
##    <chr>       <dbl>  <dbl>     <dbl>     <dbl>       <dbl>       <dbl>    <dbl>
##  1 Abilene C~  0.742    721        23        52        2885         537     7440
##  2 Adelphi U~  0.880    512        16        29        2683        1227    12280
##  3 Adrian Co~  0.768    336        22        50        1036          99    11250
##  4 Agnes Sco~  0.837    137        60        89         510          63    12960
##  5 Alaska Pa~  0.756     55        16        44         249         869     7560
##  6 Albertson~  0.816    158        38        62         678          41    13500
##  7 Albertus ~  0.963    103        17        45         416         230    13290
##  8 Albion Co~  0.906    489        37        68        1594          32    13868
##  9 Albright ~  0.808    227        30        63         973         306    15595
## 10 Alderson-~  0.856    172        21        44         799          78    10468
## # ... with 767 more rows, and 9 more variables: Room.Board <dbl>, Books <dbl>,
## #   Personal <dbl>, PhD <dbl>, Terminal <dbl>, S.F.Ratio <dbl>,
## #   perc.alumni <dbl>, Expend <dbl>, Grad.Rate <dbl>
```

```
?College                                   # read the documentation
```

```
## starting httpd help server ... done
```

Note that `Accept` is now the acceptance *rate*, and will serve as our response variable. We will use the 15 variables aside from `Name` and `Accept` as our features.

Let's define the 80%/20% train/test partition:

```
# Setting the training and testing sets
set.seed(471) # seed set for reproducibility (DO NOT CHANGE)
n = nrow(college_data)
train_samples = sample(1:n, round(0.8*n))
college_train = college_data %>% filter(row_number() %in% train_samples)
college_test = college_data %>% filter(!(row_number() %in% train_samples))
```

In what follows, we will do some exploratory data analysis and build some predictive models on the training data `college_train`.

2.1 Exploratory data analysis

Please use the training data `college_train` to answer the following EDA questions.

i. Create a histogram of `Accept`, with a vertical line at the median value. What is this median value? Which college has the smallest acceptance rate in the training data, and what is this rate? How does this acceptance rate (recall the data are from 1995) compare to the acceptance rate for the same university in 2020? Look up the latter figure on Google.

```
# Plotting the histogram
ggplot(college_train, aes(x=Accept)) +
  geom_histogram() +
  theme_bw() +
  geom_vline(aes(xintercept = median(college_train$Accept))) +
  labs(x = "Acceptance Rate", y = "Number of Colleges")
```

```
## Warning: Use of `college_train$Accept` is discouraged. Use `Accept` instead.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

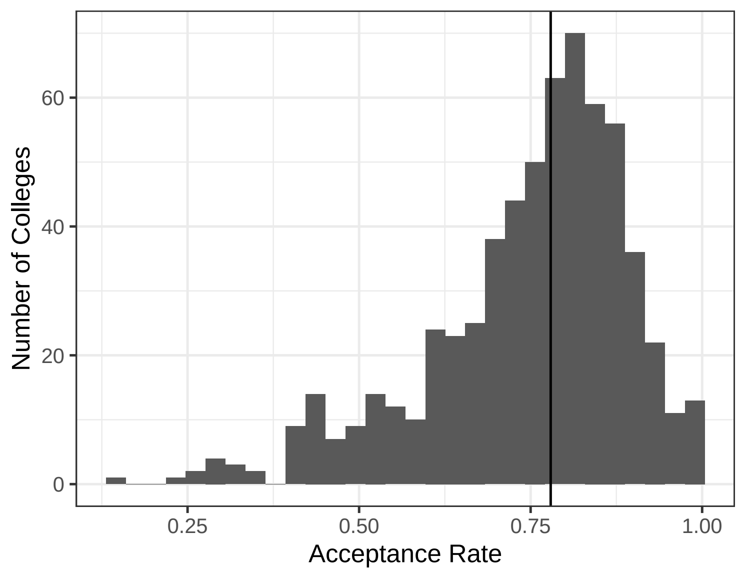

Figure 6: Acceptance Rate Distribution

```r
# Finding the lowest acceptance rate
college_train %>%
  filter(Accept == min(college_train$Accept))
```

```
## # A tibble: 1 x 17
##   Name         Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate
##   <chr>         <dbl>  <dbl>     <dbl>     <dbl>       <dbl>       <dbl>    <dbl>
## 1 Harvard Un~  0.156   1606        90       100        6862         320    18485
## # ... with 9 more variables: Room.Board <dbl>, Books <dbl>, Personal <dbl>,
## #   PhD <dbl>, Terminal <dbl>, S.F.Ratio <dbl>, perc.alumni <dbl>,
## #   Expend <dbl>, Grad.Rate <dbl>
```

The college with the lowest acceptance rate was Harvard University at 15.6%. The acceptance rate in 2020 is 4.6%, which is a third of the 1995 rate. This is a marked decline.

ii. Produce separate plots to explore the relationships between `Accept` and the following three features: `Grad.Rate`, `Top10perc`, and `Room.Board`.

```r
## Creating the scatter plots for the three variables
accept_grad <- ggplot(college_train, aes(x=Grad.Rate, y = Accept)) +
  geom_point() +
  theme_bw() +
  labs(x = "Graduation Rate", y = "Acceptance Rate")
accept_Top <- ggplot(college_train, aes(x=Top10perc, y = Accept)) +
  geom_point() +
  theme_bw() +
  labs(x = "Percent of Students Who Were Top 10 Precent", y = "Acceptance Rate")
accept_Room <- ggplot(college_train, aes(x=Room.Board, y = Accept)) +
  geom_point() +
  theme_bw() +
  labs(x = "Room and Board Costs (USD)", y = "Acceptance Rate")
plot_grid(accept_grad, accept_Top, accept_Room)
```

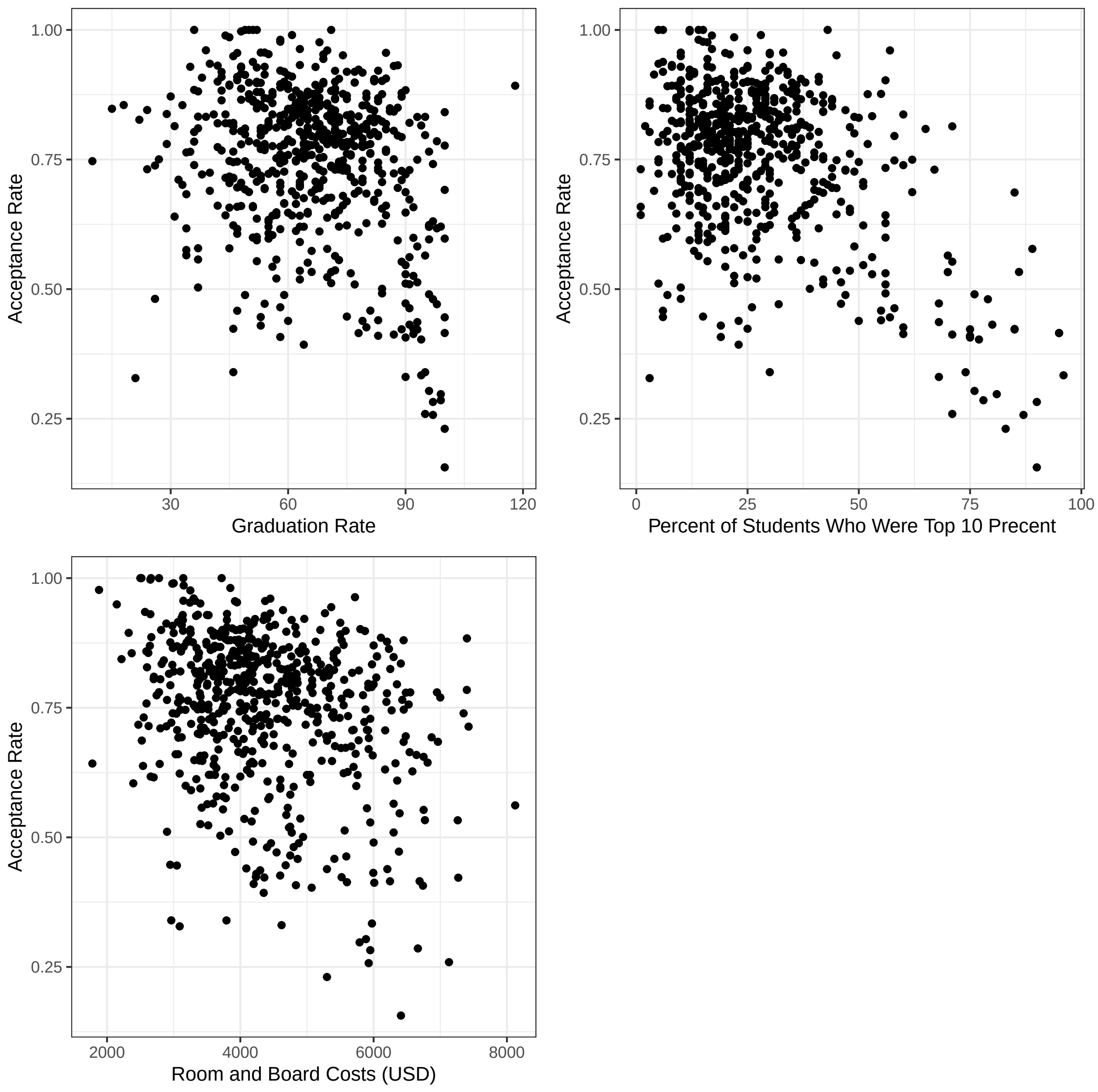

Figure 7: Acceptance Rate by Various Variables

iii. For the most selective college in the training data, what fraction of new students were in the top 10% of their high school class? For the colleges with the largest fraction of new students in the top 10% of their high school class (there may be a tie), what were their acceptance rates?

```
college_train %>%
  filter(Top10perc == max(college_train$Top10perc))
```

```
## # A tibble: 1 x 17
##   Name        Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate
##   <chr>        <dbl>  <dbl>     <dbl>     <dbl>       <dbl>       <dbl>    <dbl>
## 1 Massachuse~  0.334   1078        96        99        4481          28    20100
## # ... with 9 more variables: Room.Board <dbl>, Books <dbl>, Personal <dbl>,
## #   PhD <dbl>, Terminal <dbl>, S.F.Ratio <dbl>, perc.alumni <dbl>,
## #   Expend <dbl>, Grad.Rate <dbl>
```

The most selective school was Harvard, and 90% of students admitted to Harvard were at the top 10% of their high school class. MIT has the highest fraction of new students in the top 10% of their high school class at 96%, and MIT had a 33.4% acceptance rate.

2.2 Predictive modeling

Now we will build some predictive models for `Accept`. For convenience, let's remove the `Name` variable from the training and test sets since it is not a feature we will be using for prediction:

```
# Removing Name
college_train = college_train %>% select(-Name)
college_test = college_test %>% select(-Name)
```

2.2.1 Ordinary least squares

i. Using the training set `college_train`, run a linear regression of `Accept` on the other features and display the regression summary. What fraction of the variation in the response do the features explain?

```
lm_fit <- lm(Accept~., data = college_train)
summary(lm_fit)
```

```
##
## Call:
## lm(formula = Accept ~ ., data = college_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5515 -0.0701  0.0115  0.0855  0.2988
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.10e+00   5.17e-02   21.29  < 2e-16 ***
## Enroll       5.59e-05   2.15e-05    2.60  0.00951 **
## Top10perc   -3.26e-03   7.20e-04   -4.52  7.3e-06 ***
## Top25perc    3.20e-05   5.83e-04    0.05  0.95622
## F.Undergrad -8.85e-06   4.42e-06   -2.00  0.04552 *
## P.Undergrad -9.74e-06   4.07e-06   -2.39  0.01710 *
## Outstate     6.74e-06   2.26e-06    2.98  0.00304 **
## Room.Board  -1.92e-05   6.19e-06   -3.10  0.00205 **
## Books       -7.93e-05   3.03e-05   -2.62  0.00901 **
## Personal     7.43e-06   8.13e-06    0.91  0.36093
## PhD         -1.66e-04   5.93e-04   -0.28  0.78012
## Terminal    -1.04e-04   6.55e-04   -0.16  0.87403
## S.F.Ratio   -5.76e-03   1.69e-03   -3.42  0.00068 ***
## perc.alumni  1.18e-03   5.51e-04    2.14  0.03290 *
## Expend      -7.28e-06   1.54e-06   -4.73  2.8e-06 ***
## Grad.Rate   -1.13e-03   3.84e-04   -2.94  0.00338 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.123 on 606 degrees of freedom
## Multiple R-squared:  0.32,   Adjusted R-squared:  0.303
## F-statistic:   19 on 15 and 606 DF,  p-value: <2e-16
```

The value for R^2 is 0.32, so 32% of the variation is explained by the regression.

ii. Do the signs of the fitted coefficients for `Grad.Rate`, `Top10perc`, and `Room.Board` align with the directions of the univariate relationships observed in part iii of the EDA section?

In Figure 7, for Grad.Rate, it does as the data appears to trend downward in the scatterplot and the coefficient is negative. For Top10perc, it does as the data trends downward in the scatterplot and the coefficient is negative. For Room.Board, it does as the data trends downward in the scatterplot and the coefficient is negative.

2.2.2 Ridge regression

i. Fit a 10-fold cross-validated ridge regression to the training data and display the CV plot. What is the value of lambda selecting according to the one-standard-error rule?

```r
set.seed(3) # set seed before cross-validation for reproducibility
ridge_fit = cv.glmnet(Accept ~ ., # formula notation, as usual
  alpha = 0, # alpha = 0 for ridge
  nfolds = 10, # number of folds
  data = college_train) # data to run ridge on
```

```r
plot(ridge_fit)
```

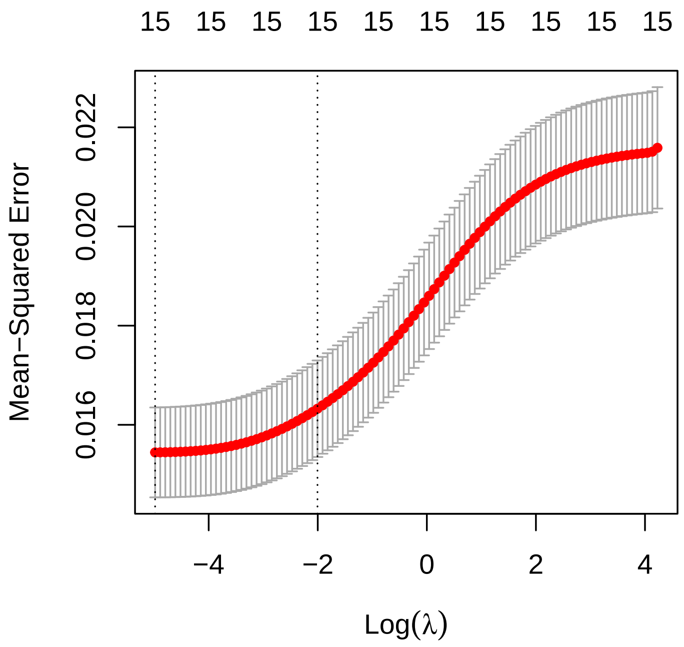

Figure 8: CV Plot for Ridge Regression

The value of lambada according to the one-standard-error rule is 0.135.

ii. UPenn is one of the colleges in the training set. During the above cross-validation process (excluding any subsequent refitting to the whole training data), how many ridge regressions were fit on data that included UPenn?

There were 900 ridge regressions fitted on Upenn, as UPenn was in a fitted fold 9 times and there are a 100 values of lambda being tested.

iii. Use `plot_glmnet` (introduced in Unit 3 Lecture 3) to visualize the ridge regression fitted coefficients, highlighting 6 features using the `features_to_plot` argument. By examining this plot, answer the following questions. Which of the highlighted features' coefficients change sign as lambda increases? Among the highlighted features whose coefficient does not change sign, which feature's coefficient magnitude does not increase monotonically as lambda decreases?

```
plot_glmnet(ridge_fit, college_train, features_to_plot = 6)
```

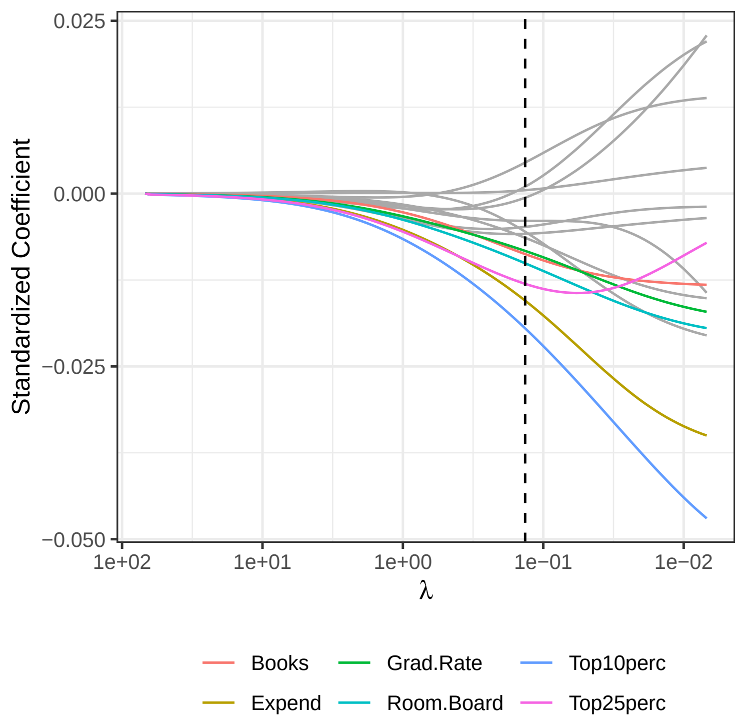

Figure 9: Feature Coefficient Mangitude Plot

In Figure 9, none of the features change sign. However, Top25perc coefficient magnitude doesn't increase monotonically as lambda increases.

iv. Let's collect the least squares and ridge coefficients into a tibble:

```
coeffs = tibble(lm_coef = coef(lm_fit)[-1],
                ridge_coef = coef(ridge_fit, s = "lambda.1se")[-1,1],
                features = names(coef(lm_fit)[-1]))
coeffs
```

```
## # A tibble: 15 x 3
##         lm_coef   ridge_coef features
##           <dbl>        <dbl> <chr>
##   1  0.0000559  -0.000000638 Enroll
##   2 -0.00326    -0.00110     Top10perc
##   3  0.0000320  -0.000658    Top25perc
##   4 -0.00000885 -0.000000814 F.Undergrad
##   5 -0.00000974 -0.00000406  P.Undergrad
##  6  0.00000674  0.000000249 Outstate
##  7 -0.0000192  -0.00000916  Room.Board
##  8 -0.0000793  -0.0000511   Books
##  9  0.00000743  0.000000707 Personal
## 10 -0.000166   -0.000348    PhD
## 11 -0.000104   -0.000321    Terminal
## 12 -0.00576    -0.00140     S.F.Ratio
## 13  0.00118     0.000371    perc.alumni
## 14 -0.00000728 -0.00000294  Expend
## 15 -0.00113    -0.000485    Grad.Rate
```

Answer the following questions by calling `summarise` on `coeffs`. How many features' least squares and ridge regression coefficients have different signs? How many features' least squares coefficient is smaller in magnitude than their ridge regression coefficient?

```r
coeffs %>%
  summarise(
    diff_signs = sum((lm_coef<0) == (ridge_coef<0)),
    smaller_mag = sum(abs(lm_coef)<abs(ridge_coef))
  )
```

```
## # A tibble: 1 x 2
##   diff_signs smaller_mag
##        <int>       <int>
## 1         13           3
```

There were 13 coefficients that had different signs, but only three of the OLS coefficients had smaller magnitudes than their ridge regression counterparts.

v. Suppose instead that we had a set of training features X^{train} such that $n_{\text{train}} = p$ and

$$X_{ij}^{\text{train}} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases}$$

Which of the following phenomena would have been possible in this case?

- Having a feature's ridge regression coefficient change signs based on lambda
- Having a feature's ridge regression coefficient decrease in magnitude as lambda decreases
- Having a feature's coefficients from least squares and ridge regression (the latter based on lambda.1se) have different signs
- Having a feature's coefficient from least squares be smaller in magnitude than its coefficient from ridge regression (based on lambda.1se)

None of these are possible. In the simplified case presented, the ridge regression coefficient is the OLS coefficient divided by (1 + lambda). So, assuming lambda is positive, the ridge regression coefficient would always have the same sign as the OLS coefficient. This is true regardless of whether lambda decreases or increases. The ridge regression coefficient would also have a magnitude always smaller than that of OLS and would increase as lambda decreases.

2.2.3 Lasso regression

i. Fit a 10-fold cross-validated lasso regression to the training data and display the CV plot.

```r
set.seed(5) # set seed before cross-validation for reproducibility
lasso_fit = cv.glmnet(Accept ~ ., # formula notation, as usual
  alpha = 1, # alpha = 1 for lasso
  nfolds = 10, # number of folds
  data = college_train) # data to run lasso on
```

```
plot(lasso_fit)
```

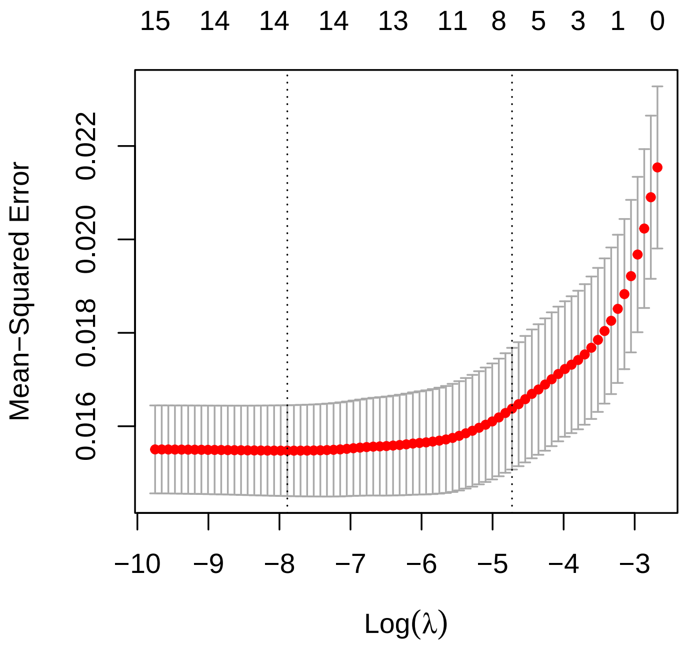

Figure 10: CV Plot of Lasso Regression

ii. How many features (excluding the intercept) are selected if lambda is chosen according to the one-standard-error rule?

Around eight features are chosen according to the one-standard-error rule.

iii. Use `plot_glmnet` to visualize the lasso fitted coefficients, which by default will highlight the features selected by the lasso. By examining this plot, answer the following questions. Which feature is the first to enter the model as lambda decreases? Which feature has the largest absolute coefficient for the most flexibly fitted lasso model?

```
plot_glmnet(lasso_fit, college_train)
```

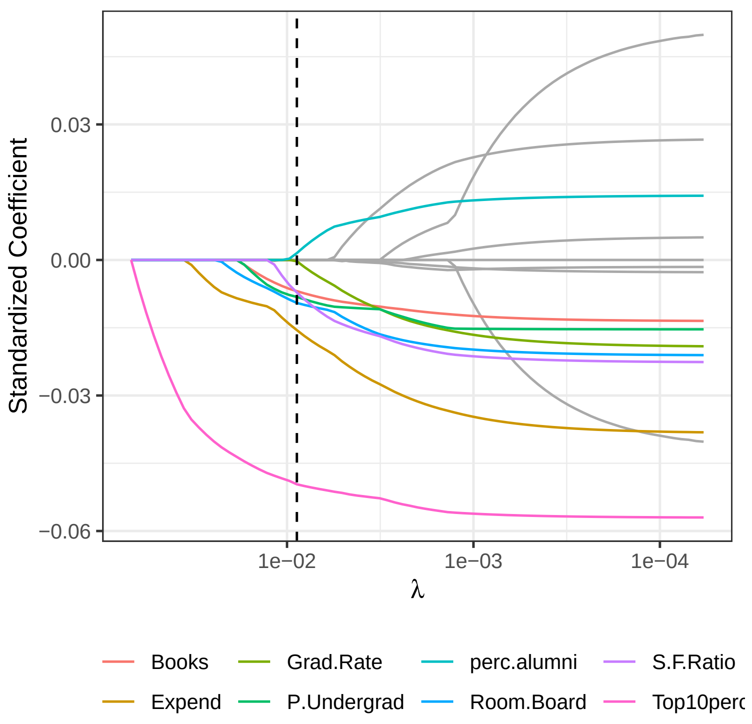

Figure 11: Feature Coefficient Magnitude Plot

In Figure 11, the first feature to enter the model as lambda decreases is Top10perc. It is also the feature with the largest absolute coefficent at the most flexibly fitted lasso model (i.e. lambda = 0).

2.2.4 Test set evaluation

i. Calculate the root mean squared test errors of the linear model, ridge regression, and lasso regression (the latter two using lambda.1se) on `college_test`, and print these in a table. Which of the three models has the least test error?

```
# ridge prediction error
ridge_predictions = predict(ridge_fit,
                            newdata = college_test,
                            s = "lambda.1se") %>%
  as.numeric()
ridge_RMSE = sqrt(mean((ridge_predictions-college_test$Accept)^2))

# lasso prediction error
lasso_predictions = predict(lasso_fit,
                            newdata = college_test,
                            s = "lambda.1se") %>%
  as.numeric()
lasso_RMSE = sqrt(mean((lasso_predictions-college_test$Accept)^2))

# intercept-only prediction error
ols_predictions = predict(lasso_fit,
                            newdata = college_test,
                            s = 0) %>%
    as.numeric()
ols_RMSE = sqrt(mean((ols_predictions-college_test$Accept)^2))

# print nice table
tibble(Ridge = ridge_RMSE, Lasso = lasso_RMSE, "Ordinary Least Squares" = ols_RMSE) %>%
   kable(format = "latex", row.names = NA,
       booktabs = TRUE, digits = 2,
       caption = "Root-mean-squared prediction errors for lasso, ridge,
       and OLS models.") %>%
  kable_styling(position = "center", latex_options = "HOLD_position")
```

Table 3: Root-mean-squared prediction errors for lasso, ridge, and OLS models.

Ridge	Lasso	Ordinary Least Squares
0.12	0.12	0.12

ii. Given which model has the lowest test error from part i, as well as the shapes of the CV curves for ridge and lasso, do we suspect that bias or variance is the dominant force in driving the test error in this data? Why do we have this suspicion? Does this suspicion make sense, given the number of features relative to the sample size?

The model with the least test error is the Ordinary Least Squares regression. This indicates that bias is probably the most dominant force driving test error in this data. Ridge and lasso add bias to the regression to reduce variance. That OLS beats them, then, indicates that this is likely a bad trade-off and that the test error is driven more by bias. That the number of features is dwarfed by the actual number of observations indicates that this suspicion likely makes sense. Usually, a high amount of variance would be due to p being roughly similar or even greater than n, but here p is much smaller than n.