STAT 471: Homework 3

Nico Melton

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:

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

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
library(cowplot)    # for side by side plots
library(glmnetUtils)     # to run ridge and lasso
library(ISLR2)      # necessary for College data
library(pROC)       # for ROC curves
library(tidyverse)
```

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")
```

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().

Solution.

```
# import data
hd_data = read_csv("../../data/Framingham.dat", col_types = "iifiiiii")
# clean data
hd_data = hd_data %>%
  rename(HD = `Heart Disease?`) %>%  # rename `Heart Disease?`
  na.omit()  # remove any rows containing NA
```

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

Solution.

```
# number of people in data
hd_data %>% nrow()
# percent of people that have heart disease
hd_data %>% pull(HD) %>% mean() * 100
```

There are a 1393 people in this data. 22.039% 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.

Solution.

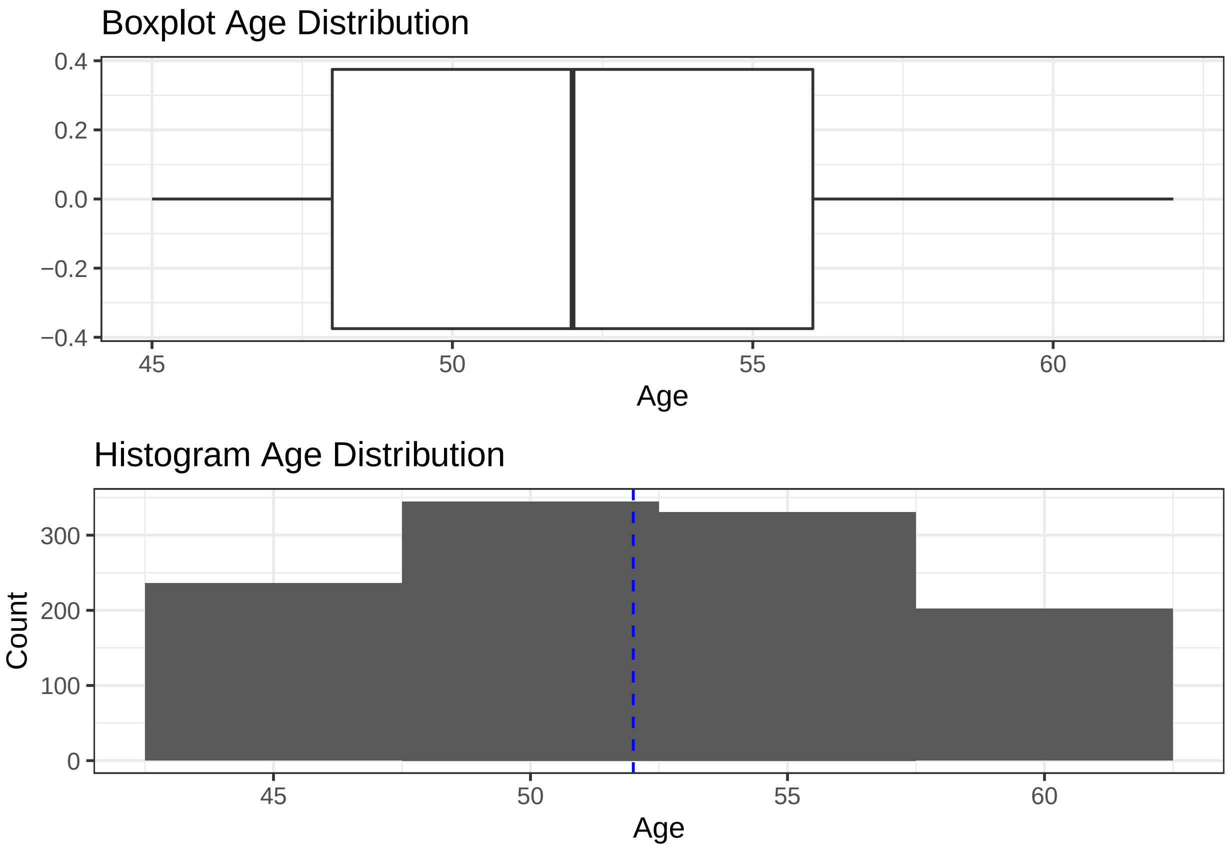
```
set.seed(5) # seed set for reproducibility (DO NOT CHANGE)
n = nrow(hd_data)
train_samples = sample(1:n, round(0.8*n))
# split head into training and testing
hd_train = hd_data %>% filter(row_number() %in% train_samples)
hd_test = hd_data %>% filter(!row_number() %in% train_samples)
```

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

Solution.

```
# boxplot of age distribution
train_dist_1 = hd_train %>%
  ggplot() +
  geom_boxplot(aes(AGE)) +
  labs(x = "Age", y = " ", title = "Boxplot Age Distribution") +
  theme_bw()
# histograme of age distribution
train_dist_2 = hd_train %>%
  ggplot() +
  geom_histogram(aes(AGE), binwidth = 5) +
  geom_vline(xintercept = hd_train %>% pull(AGE) %>% median(),
             linetype = "dashed", color = "blue") +
  labs(x = "Age", y = "Count", title = "Histogram Age Distribution") +
  theme_bw()
# plot in the same figure
plot_grid(train_dist_1, train_dist_2, ncol = 1)
```

See Figure 1. The median age in `hd_train` is 52.


Boxplot Age Distribution

Histogram Age Distribution

Figure 1: Age distribution by boxplot and histogram of training data.

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

Solution.

```
# scatter plot of heart disease against blood pressure
hd_train %>%
  ggplot(aes(x = factor(HD), y = SBP)) +
  geom_boxplot() +
  labs(y = "Systolic blood pressure", x = "Heart disease") +
  theme_bw()
```

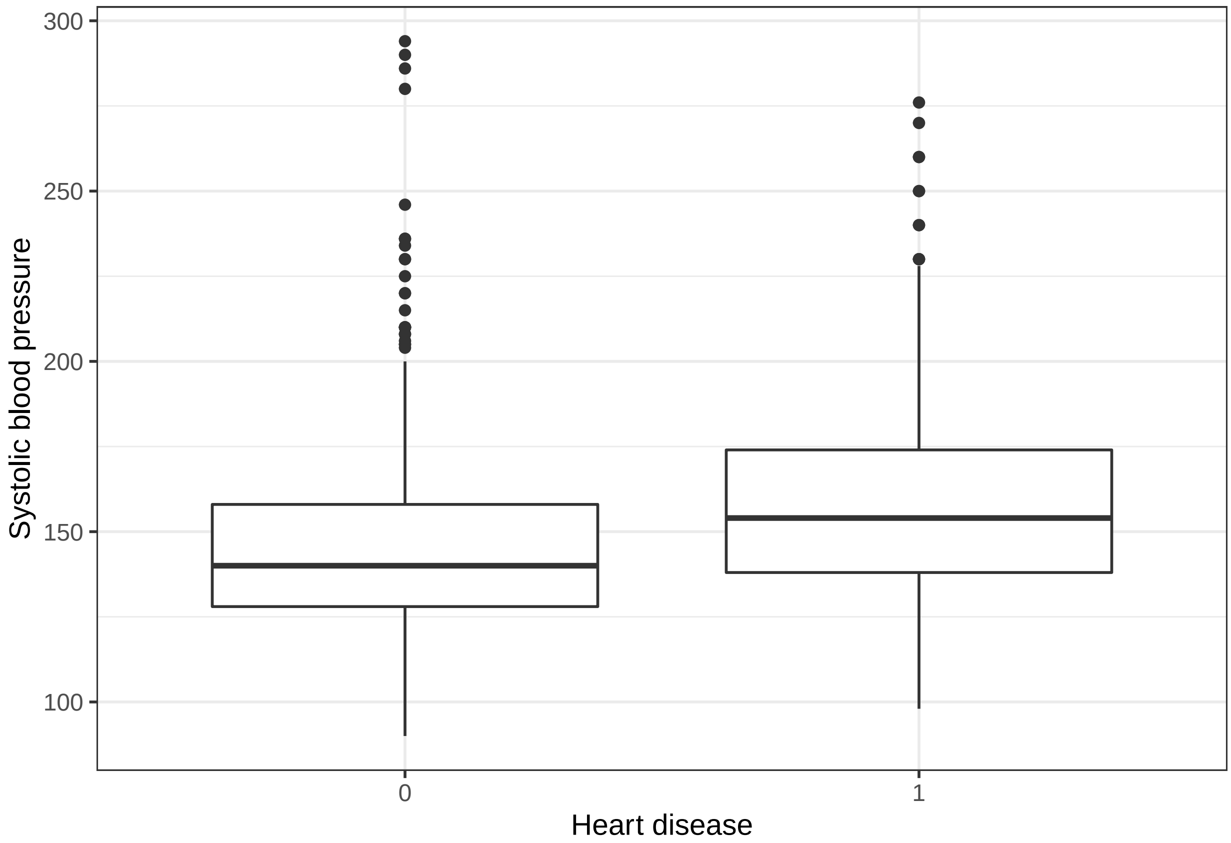


Figure 2: This plot compares the relationship between heart disease and systolic blood pressure. For heart disease, 0 is no heart disease and 1 is has heart disease.

See Figure 2. This plot shows that people with heart disease tend to have higher systolic blood pressure than people without heart disease. This may suggest some correlation between heart disease and systolic blood pressure.

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.

Solution.

```
# create subset
hd_train_subset = hd_train %>%
  filter(CIG == 40, CHOL >= 260) %>%
  select(HD, SBP)
hd_train_subset  # print
```

```
## # A tibble: 5 x 2
##      HD   SBP
##   <int> <int>
## 1     1   150
## 2     0   142
## 3     1   130
## 4     0   130
## 5     1   190
```

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

Solution.

$$\mathbb{P}[\text{HD} = 1|\text{SBP}] = \textit{logistic}(\beta_0 + \beta_1 \times \text{SBP})$$

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

Solution.

```
# maximum likelihood estimate
subset_glm_fit = glm(HD ~ SBP,
                     family = "binomial",
                     data = hd_train_subset)
# print coefficients
coef(subset_glm_fit)
```

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

The MLE is the following equation: $\mathbb{P}[\text{HD} = 1|\text{SBP}] = \textit{logistic}(-10.14 + 0.0737 \times \text{SBP})$. **Given a value of** `SBP`**,** x**, the estimated probability is** $\frac{e^{-10.14+0.0737\times x}}{1+e^{-10.14+0.0737\times x}}$.

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

Solution.

The fitted coefficients in part iii were obtained using the formula in part ii and the 5 observations in `hd_train_subset`. Given a value of `HD` (either 1 or 0) and a value of `SBP` from `hd_train_subset`, one must first express the probability that the patient has heart disease: $\mathbb{P}[\text{HD} = 1]$. This can be expressed by: $\frac{e^{\beta_0+\beta_1\times\text{SBP}}}{1+e^{\beta_0+\beta_1\times\text{SBP}}}$. Next, $\mathbb{P}[\text{Observed}]$ is can be expressed by $1 - \frac{e^{\beta_0+\beta_1\times\text{SBP}}}{1+e^{\beta_0+\beta_1\times\text{SBP}}} = \frac{1}{1+e^{\beta_0+\beta_1\times\text{SBP}}}$ if $\mathbb{P}[\text{HD} = 0]$ and $\frac{e^{\beta_0+\beta_1\times\text{SBP}}}{1+e^{\beta_0+\beta_1\times\text{SBP}}}$ otherwise. Now find β_0 and β_1 that minimize the product of all $\mathbb{P}[\text{Observed}]$.

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.]

Solution.

```
# find mean of SBP in training subset
subset_mean_sbp = hd_train_subset %>% pull(SBP) %>% mean()
# define likelihood as a function of beta_1
likelihood = function(beta_1) {
  (exp(-10.1427 + beta_1*subset_mean_sbp))/(1+exp(-10.1427 + beta_1*subset_mean_sbp))
}
# plot
ggplot() +
  stat_function(fun = likelihood, xlim = c(0,0.1)) +
  geom_vline(xintercept = coef(subset_glm_fit)[2],
             linetype = "dashed", color = "red") +
  labs(x = "beta(1)", y = "Likelihood") +
  theme_bw()
```

See Figure 3. In this plot we see that the likelihood increases exponentially as β_1 increases, until reaching the the fitted value of β_1 where the likelihood begins to flatten out as it approaches 1.

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?

Solution.

```
# univariate logistic regression
train_glm_fit = glm(HD ~ SBP,
                    family = "binomial",
                    data = hd_train)
# estimated coefficient of SBP
coef(train_glm_fit)[2]
```

The estimated coefficient of `SBP` is 0.017. The odds of heart disease increase by $e^{0.0166} = 1.0167$ times when `SBP` increases by 1.

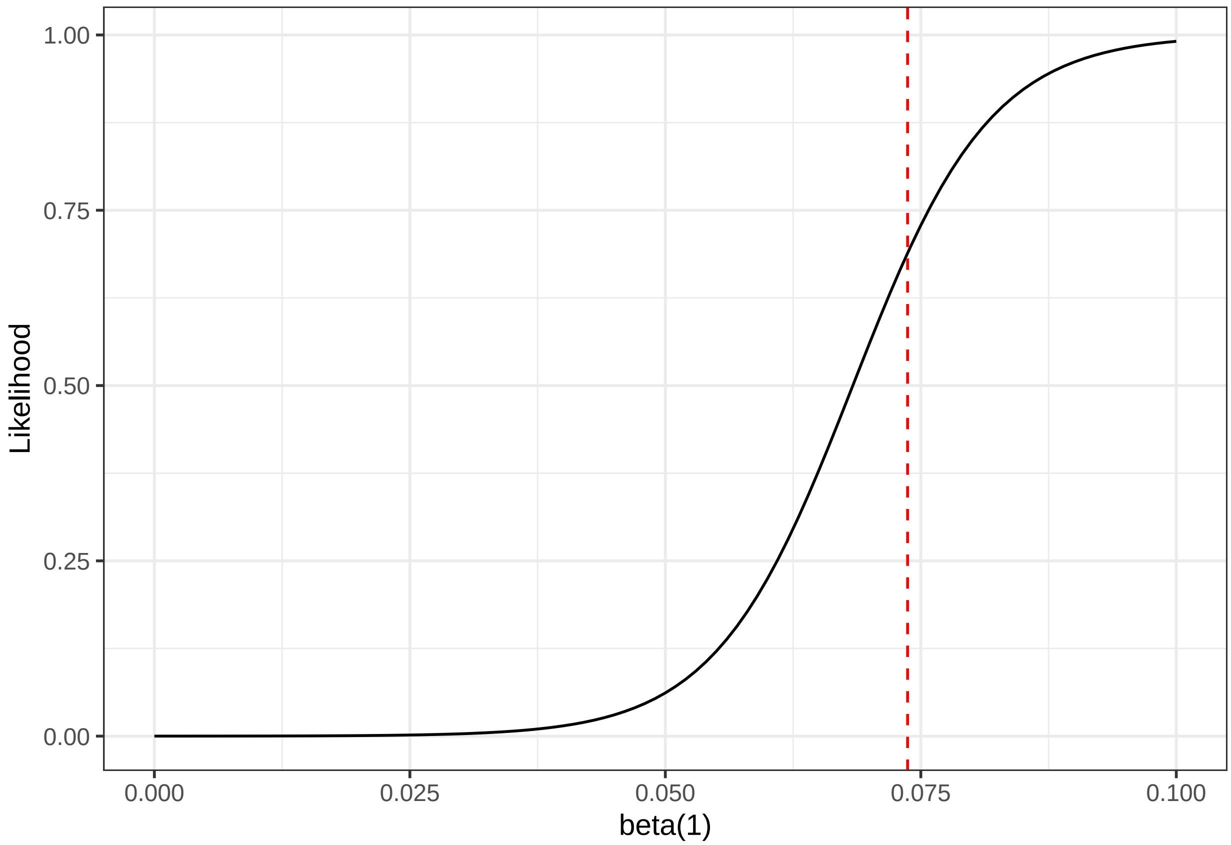


Figure 3: Likelihood as a function of beta(1) with the fitted value of beta(1).

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 $= 100$?

Solution.

```
# plot logistic regresssion fit with scatter plot
hd_train %>%
  ggplot(aes(x = SBP, y = HD)) +
  geom_jitter(height = 0.05) +
  geom_smooth(method = "glm",
              formula = "y ~ x",
              method.args = list(family = "binomial"),
              se = FALSE) +
  labs(x = "Systolic blood pressure", y = "Probability of Heart Disease") +
  theme_bw()
```

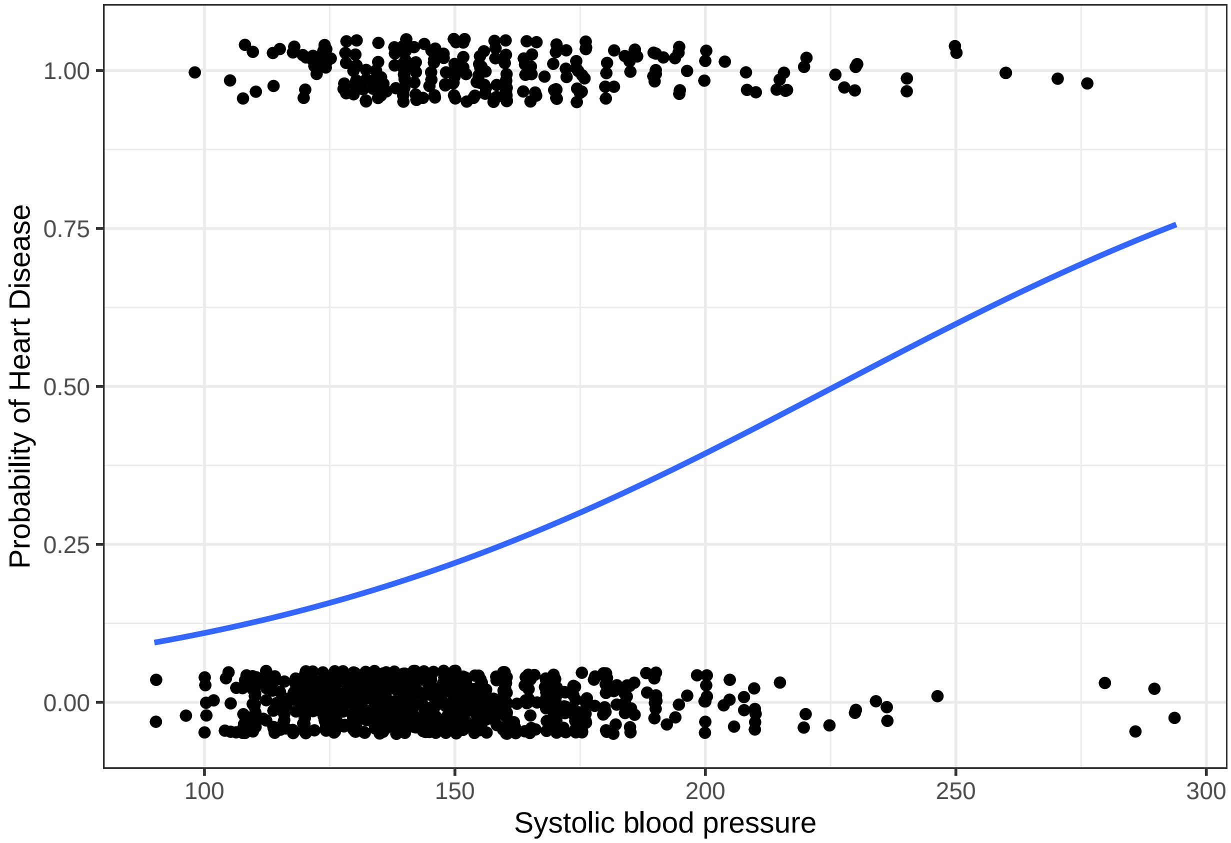


Figure 4: Logistic regression fit with scatter plot of the data.

See Figure 4. The rough estimate of heart disease for someone with SDP $= 100$ is 0.125.

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)?

Solution.

```
# multi-variable logistic regression
glm_fit = glm(HD ~ .,
              family = "binomial",
              data = hd_train)
# estimated coefficients
coef(glm_fit)
```

Other things being equal, the estimated coefficient suggests that males are more prone to heart disease because the `SEXFEMALE` coefficient is negative, meaning that being a female decreases the likelihood of having a heart disease (other things being equal). Other things being equal the impact of an increase in `AGE` by 10 on the odds of heart disease is $e^{0.06151\times 10} = 1.8498$ times, according to the estimated coefficient of about 0.062.

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?

Solution.

```
# get the probability Mary has heart disease
marys_prob = predict(glm_fit,
                     newdata = tibble(AGE=50, SEX=factor("FEMALE"), SBP=110,
                                      DBP=80, CHOL=180, FRW=105, CIG=0),
                     type = "response")
```

According to the model, the estimated probability that Mary has heart disease is 0.05.

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?

Solution.

```
# get probabilities of model on hd_test
fitted_probs = predict(glm_fit,
                       newdata = hd_test,
                       type = "response")
# get predictions based on probability threshold of 0.5
predictions = as.numeric(fitted_probs > 0.5)
# add predictions to hd_test
hd_test = hd_test %>%
  mutate(predicted_hd = predictions)
# confusion matrix
confusion_matrix = hd_test %>%
  select(HD, predicted_hd) %>%
```

Table 2: Misclassification rate, false positive rate, and false negative rate of the logistic regression classifier based on the probability threshold of 0.5

Type	Rate
Missclassification rate	0.201
False positive rate	0.023
False negative rate	0.895

```
  table()
# get FP, TN, FN, TP from confusion matrix
FN = confusion_matrix[2,1]
FP = confusion_matrix[1,2]
TP = confusion_matrix[2,2]
TN = confusion_matrix[1,1]
# false positive rate
fp_rate = FP / (FP + TN)
# false negative rate
fn_rate = FN / (FN + TP)
# misclassification rate
mc_rate = hd_test %>%
  summarise(mean(HD != predicted_hd)) %>%
  pull()
# table
rate_table = tibble(
  Type = c("Missclassification rate", "False positive rate", "False negative rate"),
  Rate = c(mc_rate, fp_rate, fn_rate)
) %>%
  kable(format = "latex", row.names = NA, booktabs = TRUE,
        col.names = NA,
        digits = 3,
        caption = "Misclassification rate, false positive rate, and false negative rate of the logistic
  kable_styling(position = "center")
# display
rate_table
```

```
# plot ROC curve
roc_data = roc(hd_test %>% pull(HD),
               fitted_probs)

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 = fp_rate, y = 1-fn_rate, colour = "red") +
  theme_bw()
```

```
# AUC
roc_data$auc
```

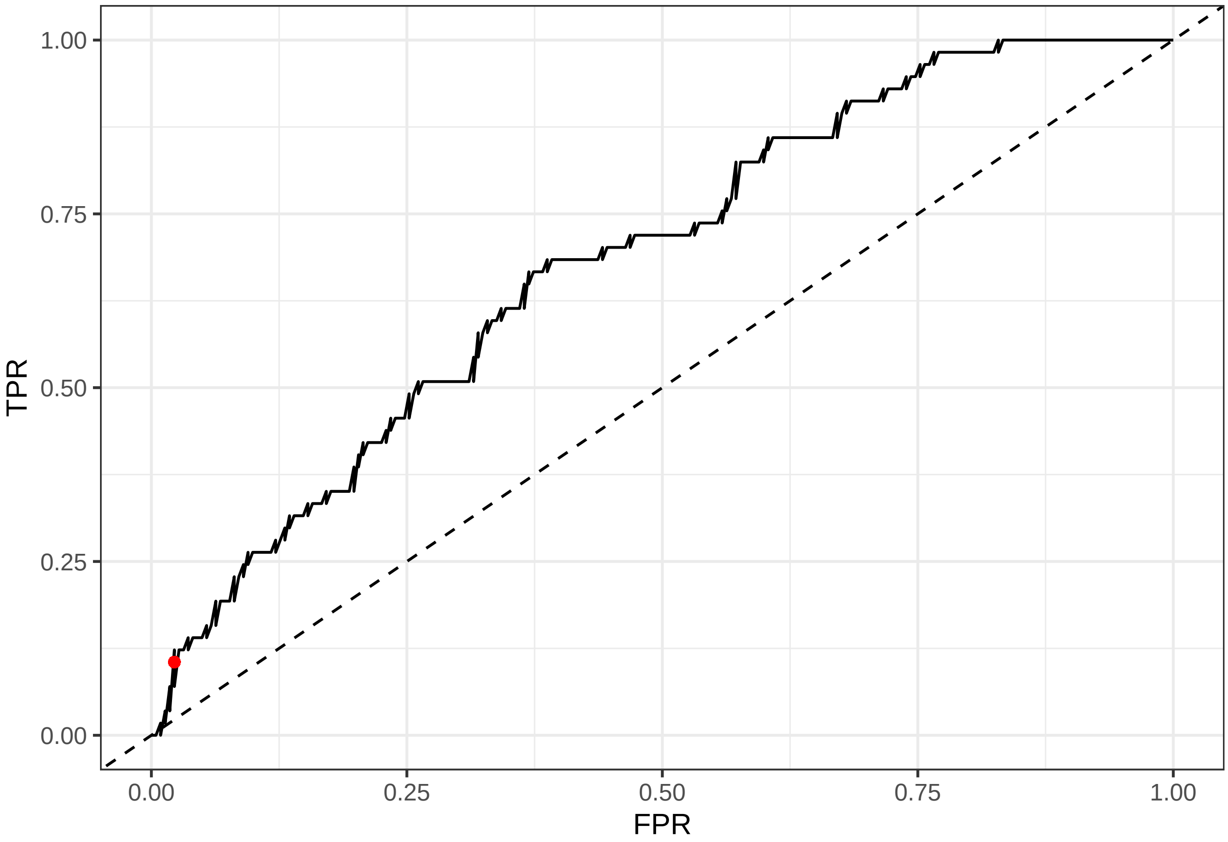


Figure 5: ROC curve logistic regression classifier.

See Table 2 for the missclassification rates, false positive rate, and false negative rate. See Figure 5. The AUC is 0.681. This is higher than the AUC of the classifier that guesses randomly, 0.5, by 0.181.

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
```

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:

```
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.

Solution.

```
# plot histogram
college_train %>%
  ggplot(aes(Accept)) +
  geom_histogram() +
  geom_vline(xintercept = college_train %>% pull(Accept) %>% median(),
             linetype = "dashed",
             color = "blue") +
  labs(x = "acceptance rate") +
  theme_bw()
```

```
# college with the smallest acceptance rate
smallest_accept = college_train %>%
  arrange(Accept) %>%
  slice_head(n = 1)
```

See Figure 6. The median value of acceptance rate is 0.779. The college that has the smallest acceptance rate is Harvard University and the rate is 0.156. This acceptance rate is a lot higher than the current acceptance rate of Harvard which is around 5.2%.

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

Solution.

```
# scatter plot with LSE for `Accept` by `Grad.Rate`
accept_grad_plot = college_train %>%
  ggplot(aes(x = `Grad.Rate`, y = Accept)) +
  geom_point() +
  geom_smooth(method = "lm",
              formula = y ~ x,
              se = FALSE) +
  labs(x = "Graduation rate", y = "Acceptance rate") +
  theme_bw()
# scatter plot with LSE for `Accept` by `Top10perc`
accept_top10_plot = college_train %>%
  ggplot(aes(x = `Top10perc`, y = Accept)) +
  geom_point() +
```

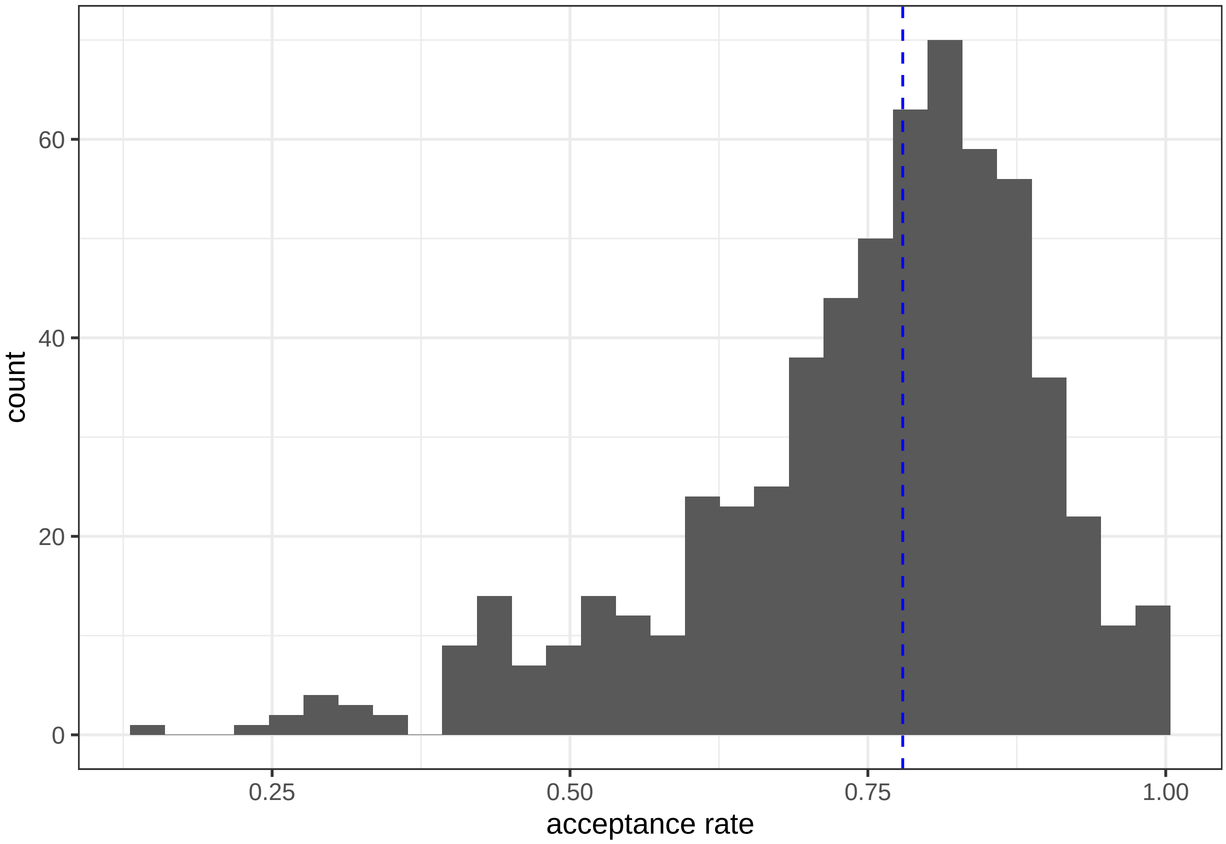


Figure 6: Histogram of the acceptance rates of the training data

```
  geom_smooth(method = "lm",
              formula = y ~ x,
              se = FALSE) +
  labs(x = "Percent of students from top 10% of class",
       y = "Acceptance rate") +
  theme_bw()
# scatter plot with LSE for `Accept` by `Room.Board`
accept_roomboard_plot = college_train %>%
  ggplot(aes(x = `Room.Board`, y = Accept)) +
  geom_point() +
  geom_smooth(method = "lm",
              formula = y ~ x,
              se = FALSE) +
  labs(x = "Cost of room and board ($)",
       y = "Acceptance rate") +
  theme_bw()
# plot
plot_grid(accept_grad_plot, accept_top10_plot, accept_roomboard_plot, rows = 2)
```

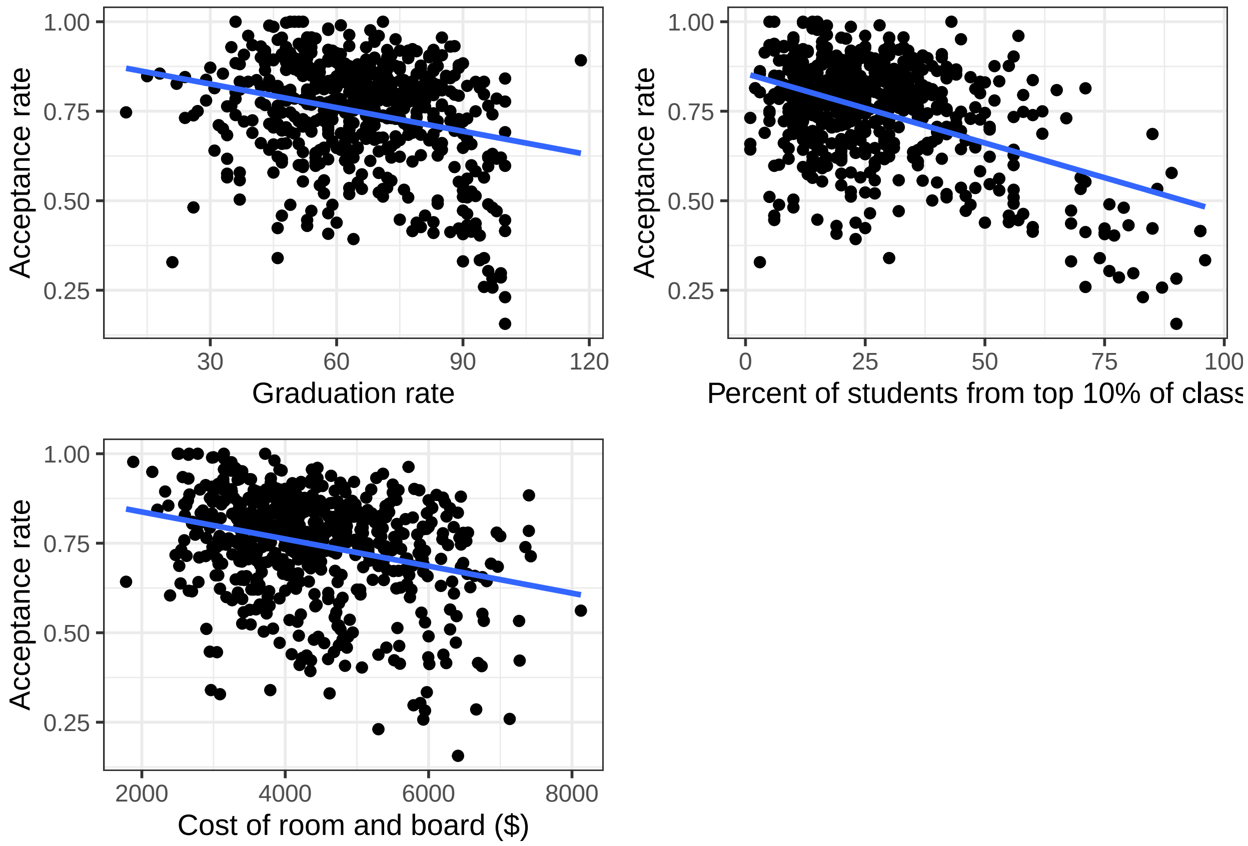


Figure 7: Plots of acceptance rate by graduation rate, percent of students in top 10 percent of class, and the cost of room and board.

See Figure 7.

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?

Solution.

```
# most selective college, fraction of new students in top 10% of high school class
smallest_accept %>% pull(Top10perc)
# colleges with the largest fraction fo new students in the top 10%
top10 = college_train %>%
  arrange(desc(Top10perc)) %>%
  slice_head(n = 3)
```

For Harvard, the most selective college in the training data, the fraction fo new students that were in the top 10% of their high school class is 90. The college with the largest fraction of new students in the top 10% of their high school class is 0.334. The college is Massachusetts Institute of Technology.

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:

```
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?

Solution.

```
# run linear regression
lm_fit = lm(Accept ~ ., data = college_train)
# get r^2
summary(lm_fit)$r.squared
```

The fraction of the variation in the response that is explained by the features is 0.32.

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?

Solution.

```
# view summary of results to see fitted coefficients
summary(lm_fit)
```

The signs of the fitted oefficients for `Grad.Rate`, `Top10perc`, and `Room.Board` DO align with the directions of the univariate relationships observed in part iii of the EDA section - they are all 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?

Solution.

```
set.seed(3) # set seed before cross-validation for reproducibility
# 10-fold cross-validated ridge regression
ridge_fit = cv.glmnet(Accept ~ .,
                      alpha = 0,
                      nfolds = 10,
                      data = college_train)
# value of lambda selecting according to the one-standard-error rule
ridge_fit$lambda.1se
# plot cv
plot(ridge_fit)
```

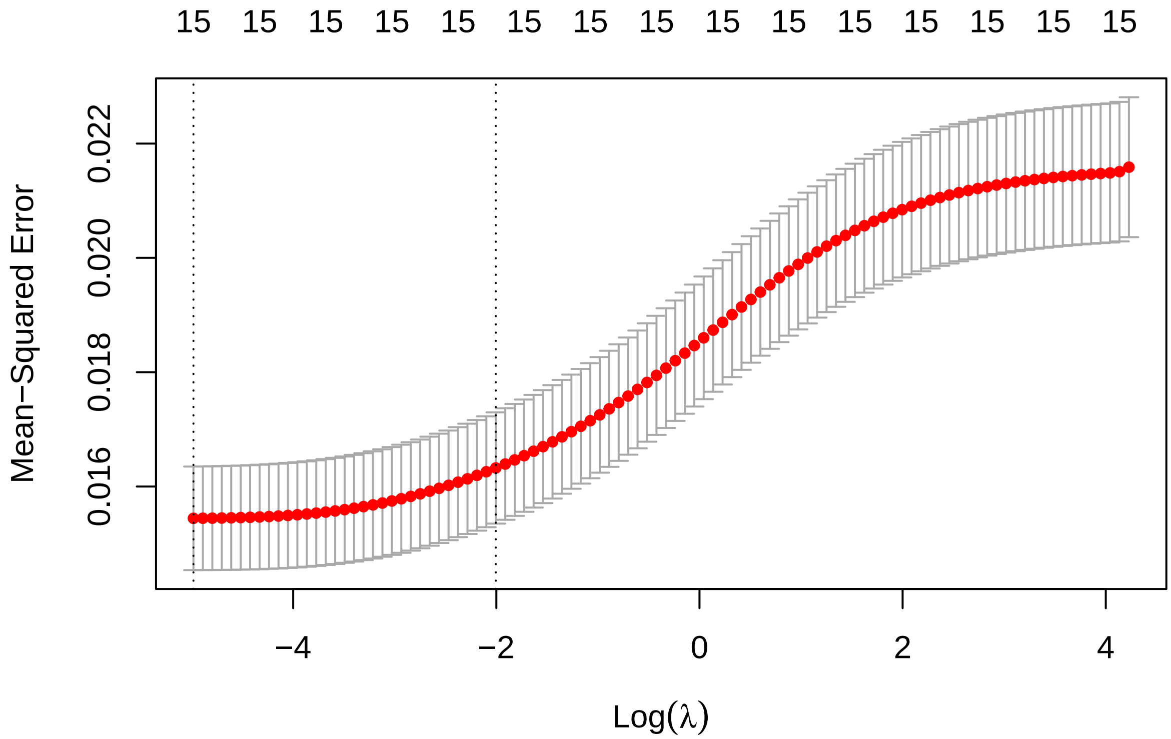


Figure 8: CV plot of ridge

See Figure 8. The value of lambda selecting 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?

Solution.

9 ridge regressions were fit on data that included UPenn.

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?

Solution.

```
# plot ridge regression fitted coefficients
plot_glmnet(ridge_fit, college_train, features_to_plot = 6)
```

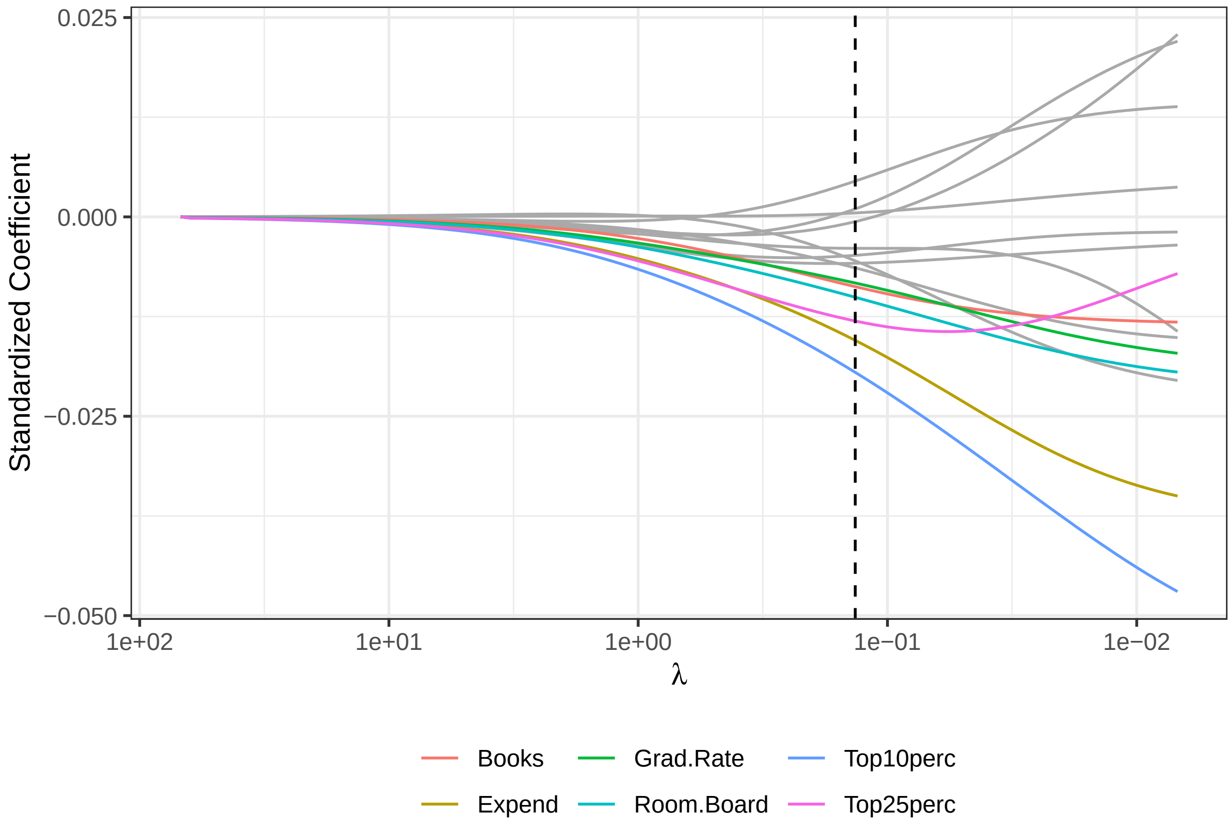


Figure 9: Ridge regression 6 largest fitted coefficients.

See Figure 9. None of the highlighted coefficients change sign as lamda increases. `Top25percs`'s coefficient does no increase monotonically as lamda decreases.

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?

Solution.

```
# number of features' whos least squares and ridge regression coefficients
#that have different signs
num_1 = coeffs %>%
  summarise(diff_signs = (lm_coef > 0 & ridge_coef < 0) |
              (lm_coef < 0 & ridge_coef > 0))
num_1 %>% sum()
# number features' whos least squares coefficient is smaller in magnitude
#than their ridge regression coefficient
num_2 = coeffs %>%
  summarise(smaller_mag = abs(lm_coef) < abs(ridge_coef))
num_2 %>% sum()
```

The number of features' whos least squares and ridge regression coefficients that have different signs is 2. The number features' whos least squares coefficient is smaller in magnitude than their ridge regression coefficient is 3

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)

Solution.

In this case, having a feature's ridge regression coefficient change signs based on lambda would have been possible.

2.2.3 Lasso regression

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

Solution

```
set.seed(5) # set seed before cross-validation for reproducibility
# 10-fold cross-validated ridge regression
lasso_fit = cv.glmnet(Accept ~ .,
                      alpha = 1,
                      nfolds = 10,
                      data = college_train)
# plot cv
plot(lasso_fit)
```

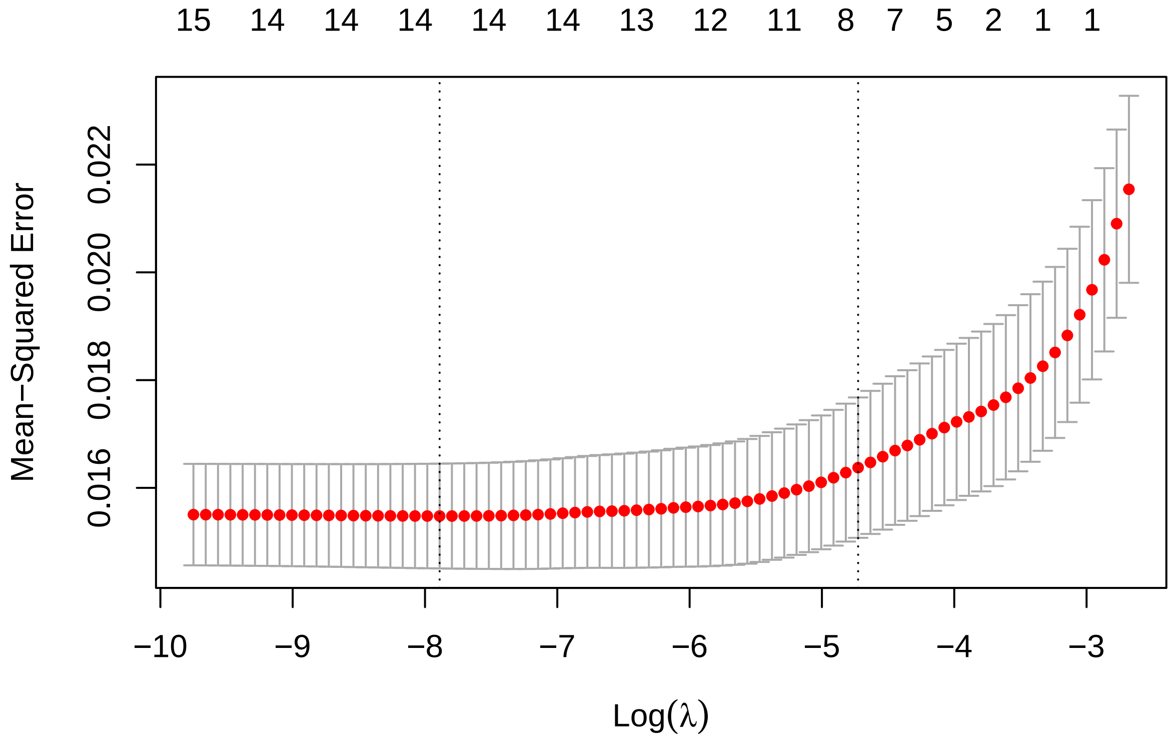


Figure 10: CV plot of lasso.

See Figure 10.

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

Solution.

8 features are re selected if lambda is 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?

Solution.

```r
# lasso fitted coefficients
plot_glmnet(lasso_fit, data = college_train)
```

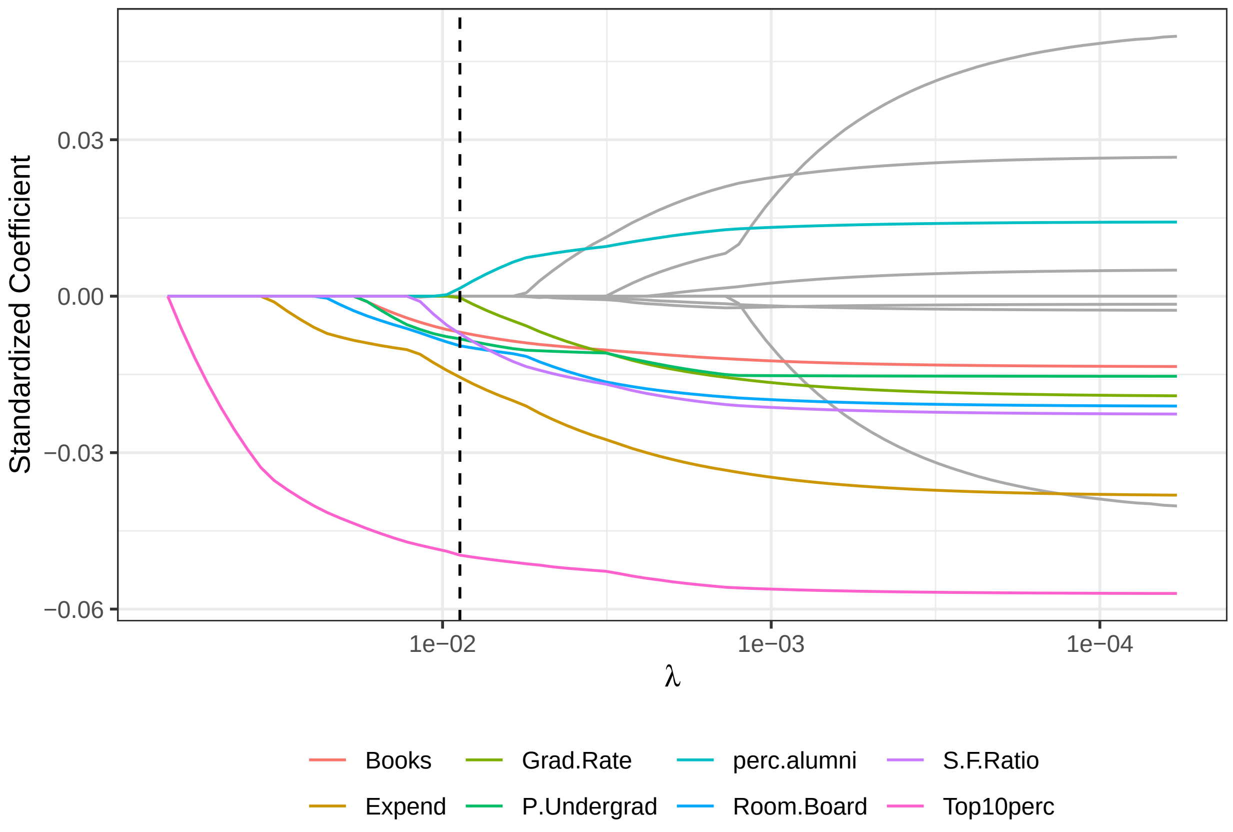


Figure 11: Fitted coefficients of lasso.

See Figure 11. The first Feature to enter the model as lamda descreases is `Top10perc`. `Top10perc` also has the largest absolute coefficient for the most flexibly fitted lasso model.

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?

Table 3: The root mean squared test errors of the linear model, ridge regression, and lasso regression on test data.

Method	RMSE
Linear	0.116
Ridge	0.122
Lasso	0.122

Solution.

```r
# linear predictions
lm_predictions = predict(lm_fit,
                         newdata = college_test) %>% as.numeric()
lm_rmse = sqrt(mean((lm_predictions - college_test$Accept)^2))
# ridge predictions
ridge_predictions = predict(ridge_fit,
                            newdata = college_test,
                            s = "lambda.1se") %>% as.numeric()
ridge_rmse = sqrt(mean((ridge_predictions - college_test$Accept)^2))
# lasso predictions
lasso_predictions = predict(lasso_fit,
                            newdata = college_test,
                            s = "lambda.1se") %>% as.numeric()
lasso_rmse = sqrt(mean((lasso_predictions - college_test$Accept)^2))
# display in table
tibble(Method = c("Linear", "Ridge", "Lasso"),
       RMSE = c(lm_rmse, ridge_rmse, lasso_rmse)) %>%
  kable(format = "latex", row.names = NA, booktabs = TRUE,
        col.names = NA,
        digits = 3,
        caption = "The root mean squared test errors of the linear model, ridge regression, and lasso r
  kable_styling(position = "center")
```

See Table 3. The linear model has the least test error.

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?

Solution.

We suspect that bias is the dominant force in driving the test error in this data because both of our models that reduced the number or weight of the coefficients did worse, suggesting that the model with all the coefficients is more accurate to the actual underlying model than one with less coefficients. Meaning when we cut coefficients we are getting further off from the real model. This model makes sense because the number of features is way less than the sample size.