Homework 8

Eric Zou

Honor code:

"I have neither given nor received unauthorized assistance on this assignment." ez

I receive help from " " and give help to " ".

Problem 1

Suppose we collect data for a group of students in a statistics class with variables X_1 : hours studied, X_2 : undergrad GPA, and Y : receive an A or not. We fit a logistic regression using Y as response and X1, X2 as predictors. Suppose that we have the estimated coefficients, $\hat{\beta}_0 = -6$, $\hat{\beta}_1 = 0.05$, $\hat{\beta}_2 = 1$.

$$P(Y = \text{“recive A”}|X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2}}$$

(a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

(Done in calculator, just plug in values with 40 as x1, 3.5 as x2)

0.37754

(b) How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class?

solve for

$$0.5 = \frac{e^{-6+0.05X_1+3.5}}{1+e^{-6+0.05X_1+3.5}}$$

(c) Calculate the odds for a student who studies for 40 h and has an undergrad GPA of 3.5.

37.7% A, 62.3% not A

Problem 2

In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set. Convert categorical variables to factors.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
auto_dat = read_csv("https://www.statlearning.com/s/Auto.csv",
                    col_types = cols(horsepower=col_double()),na=c("?"))
```

(a) Generate a binary variable named `mpg01` which should be assigned a value of 1 if the "mpg" variable is greater than its median, and 0 if "mpg" is less than or equal to its median. You can compute the median using the `median()` function. Additionally, consider using the `data.frame()` function to create a unified dataset containing both `mpg01` and the other variables present in the "Auto" dataset.

```
auto_dat$mpg01 = ifelse(auto_dat$mpg>median(auto_dat$mpg),1,0)
new_auto = data.frame(auto_dat)
```

(b) Split the data into a training set and a test set.

```
set.seed(100000)
idx = sample(1:397, 397)
train = auto_dat[idx[1:200], ]
test =  auto_dat[idx[201:397], ]
```

(c) Perform logistic regression on the training data with `mpg01` as response and use AIC to select other predictors as a optimal model. Output the predicted probability on the test data set and draw the ROC curve.

```
fit = glm(mpg01 ~ cylinders + horsepower, family = 'binomial', data = train)

summary(fit)
```

```
##
## Call:
## glm(formula = mpg01 ~ cylinders + horsepower, family = "binomial",
##     data = train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 10.23649    1.55387   6.588 4.47e-11 ***
## cylinders   -0.93938    0.24740  -3.797 0.000146 ***
## horsepower  -0.06124    0.01727  -3.546 0.000391 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 268.82  on 196  degrees of freedom
## Residual deviance: 125.82  on 194  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 131.82
##
## Number of Fisher Scoring iterations: 7
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
```

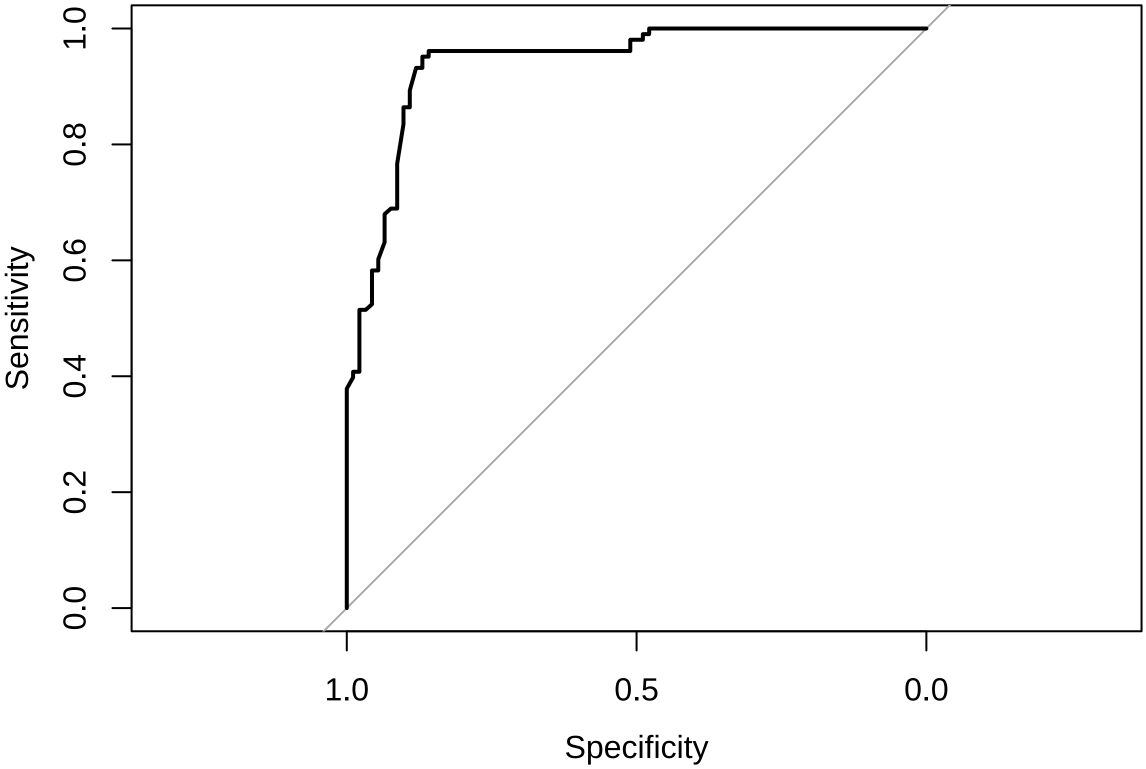
```
##
##     cov, smooth, var
```

```
predictions <- predict(fit, newdata = test, type = "response")
roc_obj <- roc(test$mpg01, predictions)
```

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

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

```
plot(roc_obj,color = "blue")
```


Sensitivity
Specificity