Homework 2

Amy Nguyen

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ISLR)
library(ROCR)
```

Linear Regression

1.

```
car_lm <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + origin, Auto)
summary(car_lm)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
```

```
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

Displacement, weight, and year are statistically significant to mpg within a 0.01 threshold, so we can reject the null hypothesis that there is no linear association between mpg and any of the predictors

2.

```
MSE <- mean((car_lm$residuals)^2)
MSE
```

```
## [1] 10.84748
```

The training MSE of this model is 10.85.

3.

```
predict(car_lm, data.frame(origin = 2, cylinders = 4, displacement = 122, horsepower = 105, weight = 31
```

```
##        fit      lwr      upr
## 1 35.13695 27.70563 42.56826
```

The predicted gas mileage for a car with these characteristics is 35.14 MPG.

4.

```
origin_lm = lm(mpg ~ origin, Auto)

american = predict(origin_lm, data.frame(origin = 1), interval = 'prediction')
european = predict(origin_lm, data.frame(origin = 2), interval = 'prediction')
japanese = predict(origin_lm, data.frame(origin = 3), interval = 'prediction')

japanese - american
```

```
##        fit      lwr      upr
## 1 10.95309 10.91093 10.99526
```

```
european - american
```

```
##        fit      lwr      upr
## 1 5.476547 5.480364 5.472731
```

There is a 10.95 difference between the MPG of Japanese and American cars. There is a 5.477 difference in MPG between European and American cars.

5.

```
displ_lm = lm(mpg ~ displacement, Auto)
summary(displ_lm)
```

```
##
## Call:
## lm(formula = mpg ~ displacement, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9170  -3.0243  -0.5021   2.3512  18.6128
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.12064    0.49443   71.03   <2e-16 ***
## displacement -0.06005    0.00224  -26.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.635 on 390 degrees of freedom
## Multiple R-squared:  0.6482, Adjusted R-squared:  0.6473
## F-statistic: 718.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

There would be about a 6 unit decrease in mpg associated with a 10 unit increase in displacement.

Algae Classification using Logistic regression

```
algae <- read_table2("algaeBloom.txt", col_names=
c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4',
'oPO4','PO4','Chla','a1','a2','a3','a4','a5','a6','a7'),
na="XXXXXXX")

algae.transformed <- algae %>% mutate_at(vars(4:11), funs(log(.)))
algae.transformed <- algae.transformed %>%
  mutate_at(vars(4:11),funs(ifelse(is.na(.),median(.,na.rm=TRUE),.)))
# a1 == 0 means low
algae.transformed <- algae.transformed %>% mutate(a1 = factor(as.integer(a1 > 5), levels = c(0, 1)))

calc_error_rate <- function(predicted.value, true.value){
return(mean(true.value!=predicted.value))
}

set.seed(1)
test.indices = sample(1:nrow(algae.transformed), 50)
algae.train=algae.transformed[-test.indices,]
algae.test=algae.transformed[test.indices,]
```

1.

$$p(z) = \frac{e^z}{1+e^z}$$
$$p(1+e^z) = e^z$$
$$p + pe^z = e^z$$

$$e^z - pe^z = p$$
$$e^z = \frac{p}{1-p}$$
$$z(p) = log(\frac{p}{1-p})$$
$$z(p) = ln(\frac{p}{1-p})$$

2. Increasing x_1 by two will change the odds of the outcome by $e^{2\beta_1}$. As x1 approaches positive infinity, p approaches infinity, and as x_1 approaches negative infinity, p approaches 1.

3.

```
log_algae <- glm(a1 ~ . , data = algae.train, family = "binomial")

# training error
train_prob <- predict(log_algae, type = "response")
algae.train = algae.train %>%
  mutate(predval=as.factor(ifelse(train_prob <= 0.5, "0", "1")))
algae.train$predval <- factor(algae.train$predval)
train_error <- calc_error_rate(algae.train$predval, algae.train$a1)

# test error
test_prob <- predict(log_algae, algae.test, type = "response")
algae.test = algae.test %>%
  mutate(predval = as.factor(ifelse(test_prob<=0.5, "0", "1")))
algae.test$predval <- factor(algae.test$predval)
test_error <- calc_error_rate(algae.test$predval, algae.test$a1)

head(train_prob)
```

```
##         1         2         3         4         5         6
## 0.1096502 0.4216051 0.2299598 0.1455352 0.6928225 0.5275182
```

```
head(test_prob)
```

```
##           1           2           3           4           5           6
## 0.999888917 0.004707483 0.204884531 0.002800264 0.041924393 0.997138302
```

```
train_error
```

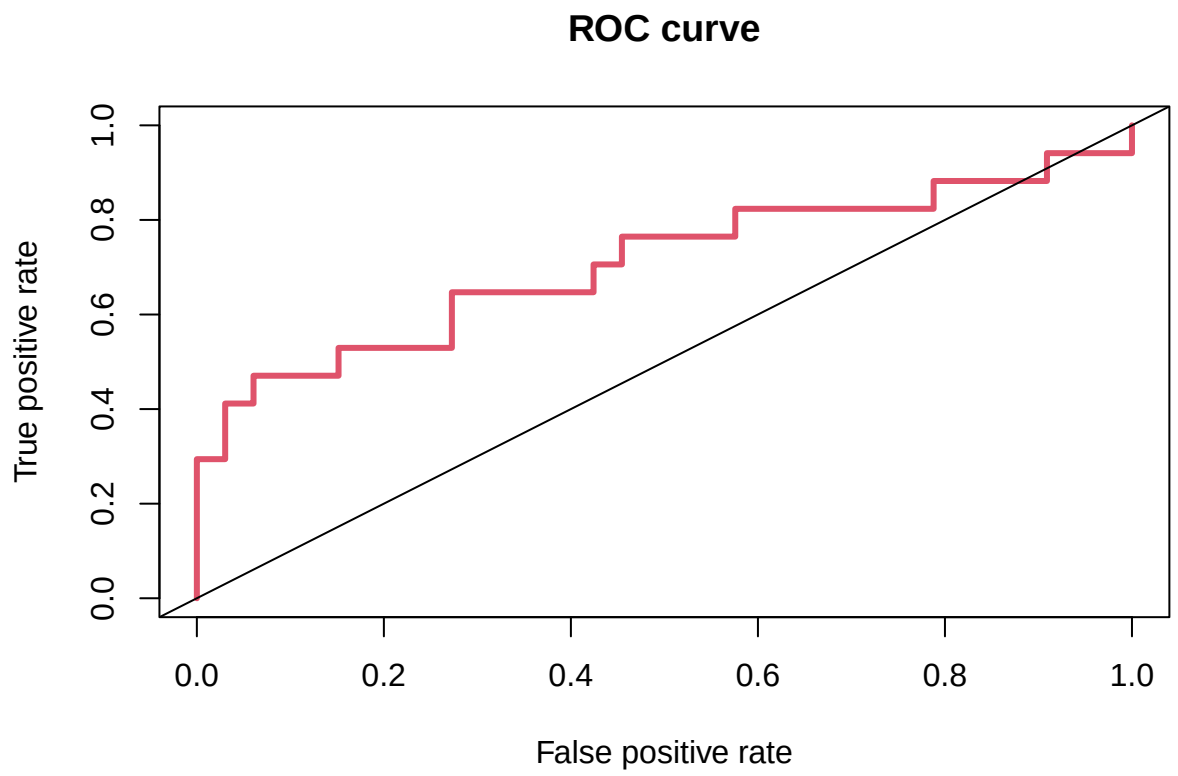
```
## [1] 0.16
```

```
test_error
```

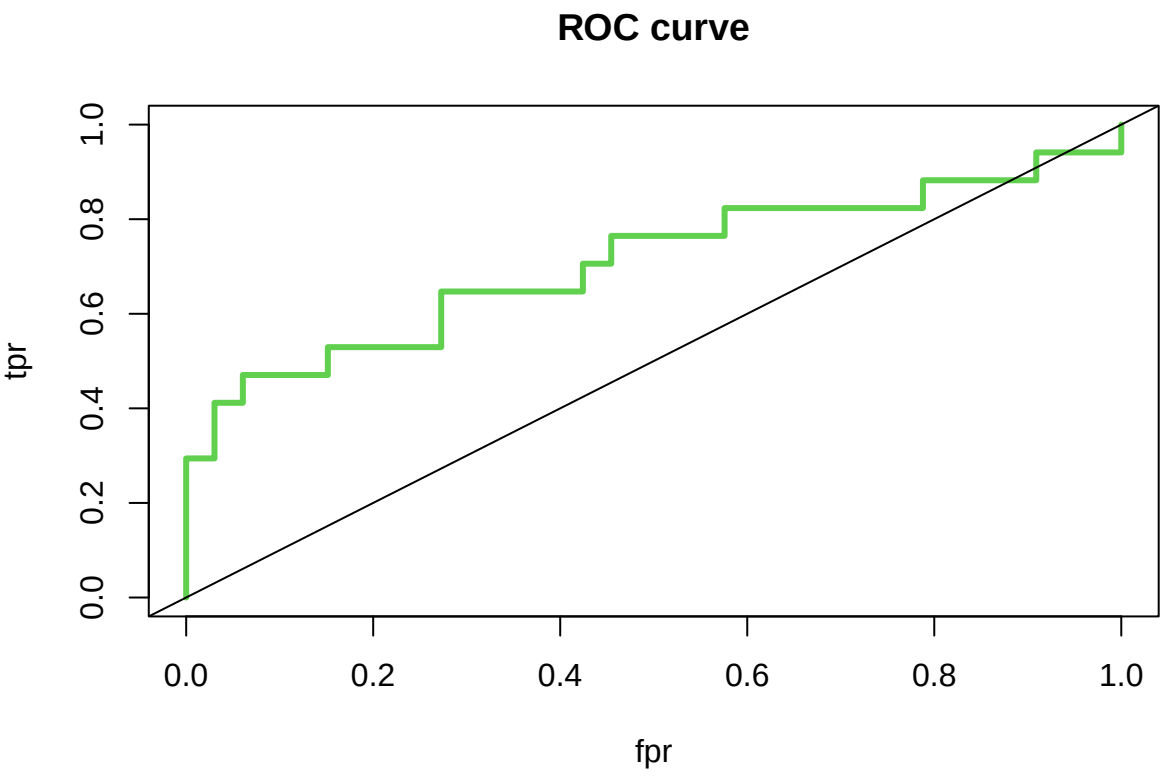
```
## [1] 0.34
```

4.

```
pred = prediction(test_prob, algae.test$a1)
perf= performance(pred, measure = 'tpr', x.measure = 'fpr')
plot(perf, col = 2, lwd = 3, main = "ROC curve")
abline(0,1)
```




```
tpr = performance(pred,'tpr')@y.values[[1]]
fpr = performance(pred, 'fpr')@y.values[[1]]
plot(fpr,tpr,type='l',col=3,lwd=3,main="ROC curve")
abline(0,1)
```



```
auc = performance(pred, 'auc')@y.values
auc
```

```
## [[1]]
## [1] 0.7076649
```

The AUC is 0.713.

Fundamentals of the bootstrap

1.

$$(1 - \frac{1}{n})^n$$

2.

```
(1 - 1/1000)^1000
```

```
## [1] 0.3676954
```

The probability for n=1000 is 0.3677

3.

```
obs <- sample(1:1000, size=1000, replace=TRUE)
missing <- 1000 - length(unique(obs))
missing/1000
```

```
## [1] 0.371
```

The ratio of missing observations is 0.362 which is very close to 0.3677, thus we can consider that our calculation is reasonable.

Cross-validation estimate of test error

1.

```
set.seed(123)
dat = subset(Smarket, select = -c(Year,Today))
dat$Direction = ifelse(dat$Direction == "Up", 1, 0)
train = dat[1:700,]
test = dat[701:nrow(dat),]
train$Direction <- factor(train$Direction)

train_fit = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data =
                  train, family = "binomial")
summary(train_fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = "binomial", data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.266  -1.157  -1.046   1.191   1.397
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20472    0.38137   0.537    0.591
## Lag1        -0.05028    0.05457  -0.921    0.357
## Lag2        -0.03988    0.05459  -0.731    0.465
## Lag3         0.01074    0.05443   0.197    0.844
## Lag4         0.02243    0.05448   0.412    0.681
## Lag5        -0.01552    0.05386  -0.288    0.773
## Volume      -0.19096    0.27713  -0.689    0.491
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 969.94  on 699  degrees of freedom
## Residual deviance: 967.71  on 693  degrees of freedom
## AIC: 981.71
##
## Number of Fisher Scoring iterations: 3
```

```
test_prob = predict(train_fit, test, type="response")

test = test %>%
  mutate(predval=as.factor(ifelse(test_prob <= 0.5, "0", "1")))
calc_error_rate(test$predval, test$Direction)
```

```
## [1] 0.5436364
```

The test error rate is 0.5436. ### 2.

```
set.seed(123)
do.chunk <- function(chunkid, folddef, dat, ...){
  # Get training index
  train = (folddef!=chunkid)
  # Get training set and validation set
  dat.train = dat[train, ]
  dat.val = dat[-train, ]
  # Train logistic regression model on training data
  fit.train = glm(Direction ~ ., family = binomial, data = dat.train)
  # get predicted value on the validation set
  pred.val = predict(fit.train, newdata = dat.val, type = "response")
  pred.val = ifelse(pred.val > .5, 1,0)
  data.frame(fold = chunkid, val.error = mean(pred.val != dat.val$Direction))

}
nfold = 10
folds = cut(1:nrow(train), breaks=nfold, labels=FALSE) %>% sample()
error.folds = NULL
tmp = do.chunk(chunkid = 10, folddef=folds, dat = dat)
error.folds = rbind(error.folds, tmp)
error = error.folds$val.error
error
```

```
## [1] 0.4747798
```

Using the 10-fold cross-validation approach, the test error is estimated to be 0.4748.