Final Project

Warren Kennedy

2023-06-09

Application Problems

Question 1

Consider the Carseats data in the ISLR2 package.

Part A

Fit a linear regression model with Sales as the response and all other variables as covariates. Report the coefficient estimates.

```
# Loading Carseat Data
library(ISLR2)
data("Carseats")
```

```
# Fit Linear Regression Model
lm.fit <- lm(Sales ~ ., data = Carseats)

# Report Coefficient Estimates
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Sales ~ ., data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8692 -0.6908  0.0211  0.6636  3.4115
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.6606231  0.6034487   9.380  < 2e-16 ***
## CompPrice        0.0928153  0.0041477  22.378  < 2e-16 ***
## Income           0.0158028  0.0018451   8.565 2.58e-16 ***
## Advertising      0.1230951  0.0111237  11.066  < 2e-16 ***
## Population       0.0002079  0.0003705   0.561    0.575
## Price           -0.0953579  0.0026711 -35.700  < 2e-16 ***
## ShelveLocGood    4.8501827  0.1531100  31.678  < 2e-16 ***
## ShelveLocMedium  1.9567148  0.1261056  15.516  < 2e-16 ***
## Age              -0.0460452  0.0031817 -14.472  < 2e-16 ***
## Education        -0.0211018  0.0197205  -1.070    0.285    
## UrbanYes          0.1228864  0.1129761   1.088    0.277    
## USYes            -0.1840928  0.1498423  -1.229    0.220    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 1.019 on 388 degrees of freedom
## Multiple R-squared:  0.8734, Adjusted R-squared:  0.8698 
## F-statistic: 243.4 on 11 and 388 DF,  p-value: < 2.2e-16
```

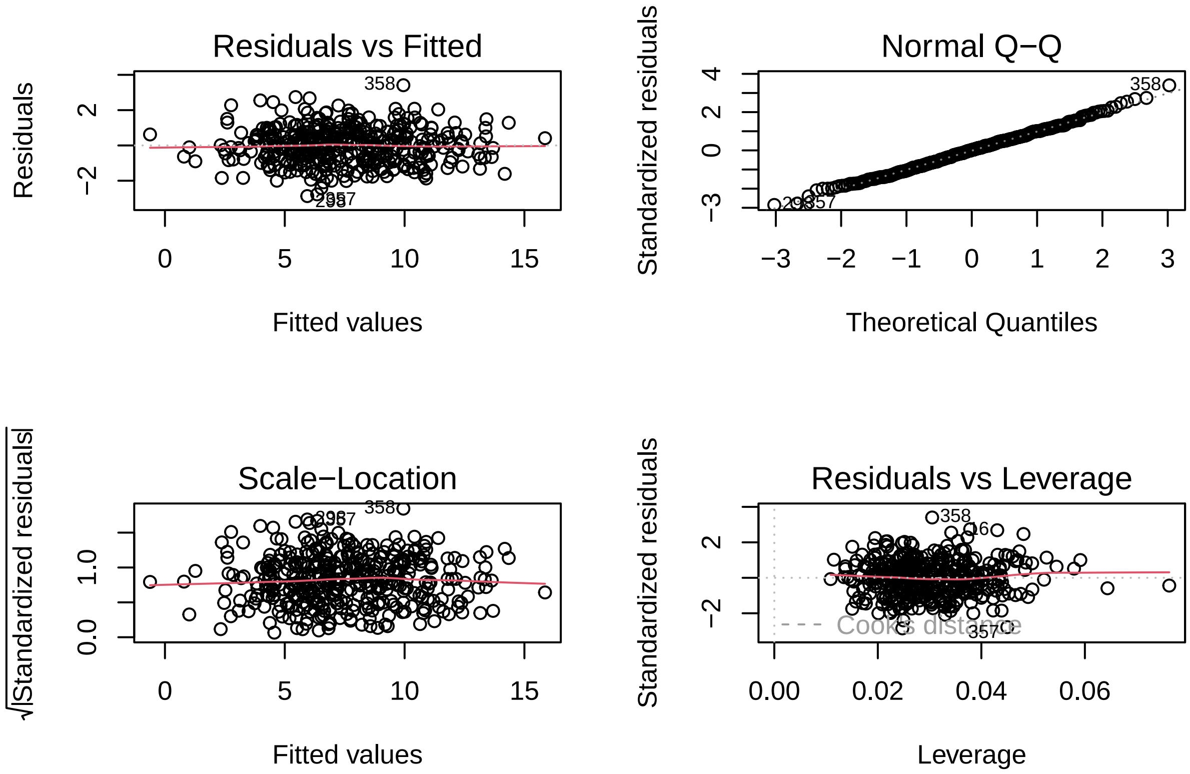
```
coefficients(lm.fit)
```

```
##     (Intercept)       CompPrice          Income     Advertising      Population 
##    5.6606230631    0.0928153421    0.0158028363    0.1230950886    0.0002078771 
##           Price   ShelveLocGood ShelveLocMedium             Age       Education 
##   -0.0953579188    4.8501827110    1.9567148062   -0.0460451630   -0.0211018389 
##        UrbanYes           USYes 
##    0.1228863965   -0.1840928246
```

Part B

Determine whether the linear model is appropriate.

```
# Linear Model Assumption Testing
par(mfrow = c(2, 2))
plot(lm.fit)
```



We can determine the fitness of applying a linear model to our data by observing the residual plots in order to test the assumptions for linear regression. In our the above visualization, we plot our residuals against the fitted values. In a residual plot for a linear model, you typically look for certain patterns or characteristics that can help assess the adequacy of the model. The spread of the residuals should remain roughly constant across the range of predicted values. The residuals should appear randomly scattered around the horizontal axis. The residuals should be independent of each other, meaning that the value of one residual should not provide any information about the value of another residual. The residuals should not exhibit any systematic curvature or nonlinear patterns. In our case, we see that none of these conditions are violated, meaning that the data points do not seem to show any systematic deviations from randomness.

Next we consider the Normal Q-Q plot. In a normal Q-Q plot, the residuals are plotted against the quantiles of a theoretical normal distribution. If the residuals are normally distributed, the points on the plot should roughly follow a straight line. Departures from normality can impact the accuracy of statistical tests and confidence intervals associated with the model. In our case, our normality assumption is met, given that our residuals follow the line denoting the quantiles of a theoretical normal distribution. We conclude that fitting a linear model is indeed appropriate.

Part C

Let beta-1 and beta-2 be the coefficients for CompPrice and Income, respectively. Test the hypothesis that beta-1 = beta-2 = 0. State your hypothesis, test statistic, and test statistic's distribution clearly. Choose an alpha you feel is appropriate.

```
### ANOVA METHOD

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data.subset = Carseats[, -c(2,3)]
lmfit1 = lm(Sales~., data=data.subset)
lmfit2 = lm(Sales~., data=Carseats)
anova(lmfit1,lmfit2)
```

```
## Analysis of Variance Table
##
## Model 1: Sales ~ Advertising + Population + Price + ShelveLoc + Age +
##     Education + Urban + US
## Model 2: Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education + Urban + US
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    390 977.31
## 2    388 402.83  2    574.48 276.66 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Null hypothesis: There is no linear relationship between the predictor variables and the response variable. In other words, the coefficients for both CompPrice and Income are equal to zero.

Alternative hypothesis: There is a linear relationship between the predictor variables and the response variable. At least one of the coefficients for CompPrice or Income is not equal to zero.

The test statistics for each coefficient are represented by their respective t-values. Meaning that our test statistic for CompPrice is 1.548, and 3.187 for Income.

The distribution of the test statistics is the t-distribution, which is used in hypothesis testing for linear regression.

Since we are testing multiple hypotheses, we need to account for the family-wise error rate. We do so by applying the Bonferroni-Holm corrrection to our p-values.

Our resulting p-values are 0.1222 and 0.0023, for Carprice and Income, respectively. Comparing our adjusted p-values to 95% significance level, α=0.05, we can conclude that we should reject our null hypothesis that both covariates are jointly equal to zero.

Question 2

Consider the Carseats data again.

Part A

Split the data into a training set and a validation set. State the proportions of your training/validation split.

```
# Define Variables
x <- model.matrix(Sales ~ ., Carseats)[, -1]
y <- Carseats$Sales

# Train/Validation Split
set.seed(1)
train <- sample(1:nrow(x), nrow(x) *0.8)
test <- (-train)
y.test <- y[test]
```

We will train our model on 80 percent of our data. The validation(test) set will be comprised of the remaining 20 percent of unseen data.

Part B

Fit a ridge regression model on the training data, choosing the lambda by cross-validation and reporting the final coefficients. Choose an appropriate value for K when doing cross-validation.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
rr.fit <- cv.glmnet(x[train, ], y[train], alpha = 0, nfolds = 10)
```

```
# Report best lambda
bestLambda <- rr.fit$lambda.min
bestLambda
```

```
## [1] 0.1395915
```

```
#Run ridge regression with best lambda, report coefficients
ridge.mod <- glmnet(x, y, alpha = 0, lambda = bestLambda)
coef(ridge.mod)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                             s0
## (Intercept)      6.3017064310
## CompPrice        0.0808046540
## Income           0.0146693896
## Advertising      0.1117261077
## Population       0.0001902178
## Price           -0.0861049868
## ShelveLocGood    4.4232806820
## ShelveLocMedium  1.6747315661
## Age             -0.0434522076
## Education       -0.0209803748
## UrbanYes         0.0972233371
## USYes           -0.0760578845
```

Part C

Report the RMSE using the validation set on the model from 2b.

```
# Predict values using new X's
ridge.pred <- predict(ridge.mod, s = bestLambda, newx = x[test, ])

# Calculating RMSE Root Mean Square Error
sqrt(mean((ridge.pred - y.test)^2))
```

```
## [1] 0.9883017
```

Part D

Fit a random forest model on the training data, and report the RMSE on the validation set.

```
#fit using training
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: ’randomForest’
```

```
## The following object is masked from ’package:ggplot2’:
##
##     margin
```

```
## The following object is masked from ’package:dplyr’:
##
##     combine
```

```
set.seed(1)

rf.carseats <- randomForest(Sales ~ ., data = Carseats, subset = train, mtry = 4, importance = TRUE)
rf.predict <- predict(rf.carseats, newdata = Carseats[test, ])

#get RMSE using test
sqrt(mean((rf.predict - y.test)^2))
```

```
## [1] 1.78617
```

Part E

For both of the models you fit in (b) and (d), give an example why a marketing team would prefer one model over the other.

Suppose a marketing team seeks a model that balances accurate predictions with interpretability. Ridge regression meets their requirements by providing interpretable coefficient estimates, allowing them to assess the impact of predictors on customer satisfaction. This aids in identifying influential variables and making informed decisions. Additionally, ridge regression is a straightforward and transparent linear model, simplifying communication with stakeholders. Moreover, in our case, the ridge regression model performs better, exhibiting the lowest root mean square error compared to our random forest model. Hence, based on its simplicity and accuracy, we believe the marketing team is likely to prefer the ridge regression model.

Question 3

In this question, you will simulate data to perform regression between X and Y.

Part A

Use the rt() function to generate a predictor X of length n=200. Set df=15 for X

```
set.seed(1)
X <- rt(n=200, df=15)
sorted_idx = order(X)
X = X[sorted_idx]
```

Part B

Use rt() to generate a noise vector epsilon. Set df=5.

```
epsilon <- rt(n=200, df=5)
```

Part C

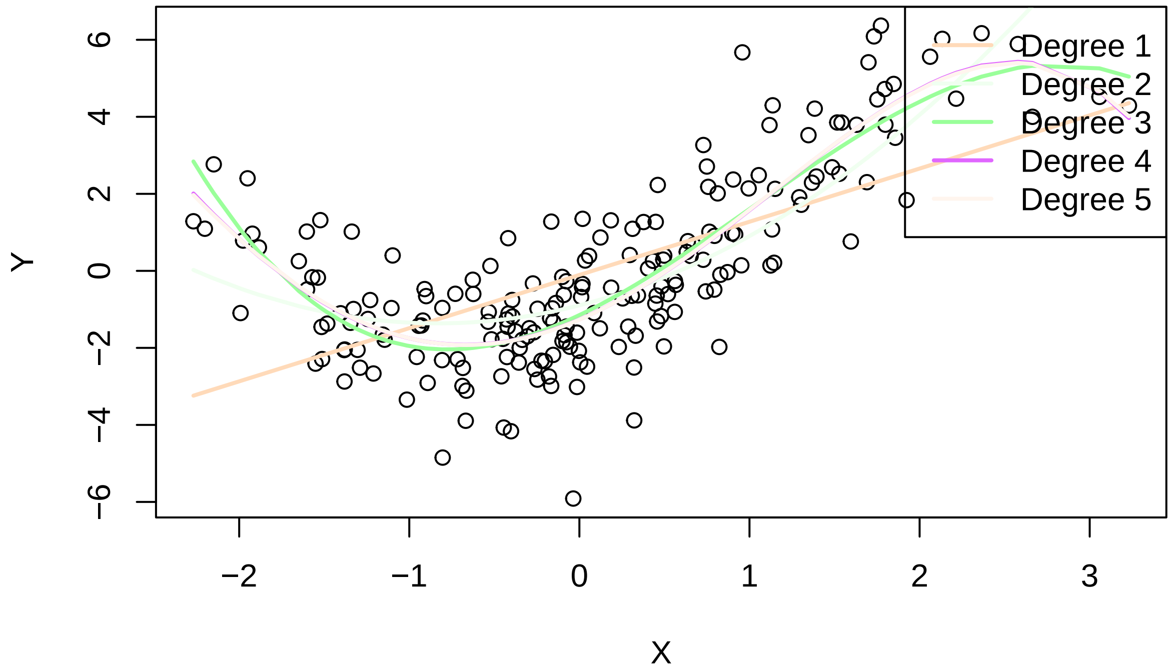
Generate a response vector Y of length n

```
Y <- 5+2*sin(X)-(7*(exp(2*cos(X))/(1+exp(2*cos(X))))) + epsilon
```

Part D

Fit polynomial regression for Y on X with the order of X ranging from 1 to 5 and plot each of the five model fits, in different colors and with a legend, on top of your simulated data.

```
plot(X, Y)
colors <- sample(colors(), 5)
legend_labels <- c()

for (p in 1:5) {
  fit <- lm(Y ~ poly(X, p))
  val <- predict(fit, data.frame(X = X))
  lines(X, val, col = colors[p], lwd = 2)
  legend_labels <- c(legend_labels, paste0("Degree ", p))
}

legend("topright", legend = legend_labels, col = colors, lwd = 2)
```



Part E

Which one of these models do you prefer? Justify your answer.

We utilize 10-fold cross validation to determine the most suitable polynomial for our data. Through this process, we identify that the polynomial with degree 5 yields the smallest validation/prediction error. The corresponding visualization of this final polynomial can be observed in the plot below.

```
library(boot)
```

```
##
## Attaching package: 'boot'

## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
set.seed (1)
df = cbind.data.frame(X,Y)
predErr <- rep (0, 5)
for (i in 1:5) {
  glm.fit <- glm(Y~poly(X,i), data = df)
  predErr[i] <- cv.glm (df , glm.fit , K = 10)$delta[1]
}
predErr
```

```
## [1] 3.416248 2.375047 1.724267 1.688903 1.681169
```

```
min(predErr)
```

```
## [1] 1.681169
```

Part F

For the model, compute a 90% confidence interval at X=1 using least squares theory. Provide an interpretation for this interval.

```
fit = lm(Y ~ poly(X,2))
ci = predict(fit, data.frame(X=1), level = 0.90, interval = "confidence")
ci
```

```
##         fit       lwr      upr
## 1 0.9029684 0.6640983 1.141839
```

90% CI: [0.6640983, 1.141839]

Our Confidence Interval indicates that in around 90% of cases, the true value of the response variable (when X=1) would lie within this interval if we were to replicate the data collection and modeling process multiple times.

Part G

For the model, compute a 90% confidence interval at X=1 using a bootstrap. Provide an interpretation for this interval.

```
df=data.frame(Y=Y,X=X)

# Set the number of bootstrap samples
B <- 200

# Vector to store bootstrap estimates
bootstrap_estimates <- numeric(B)

#Bootstrp Procedure

for (b in 1:B) {
  # Sample with replacement from the original dataset
  bootstrap_data <- df[sample(nrow(df), replace = TRUE), ]

  # Fit the model on the bootstrap sample
  fit <- lm(Y ~ poly(X, 2), data = bootstrap_data)

  # Predict the response at X = 1
  bootstrap_estimates[b] <- predict(fit, newdata = data.frame(X = 1))
}

confidence_interval <- quantile(bootstrap_estimates, c(0.05, 0.95))
confidence_interval
```

```
##        5%       95%
## 0.6498244 1.1400490
```

The confidence interval provides valuable insights into the likely range of the model's prediction at X = 1. It takes into consideration the data's variability and the estimation process through bootstrapping. Upon repeating the bootstrapping procedure multiple times, approximately 90% of the calculated confidence intervals would encompass the true value of the model's prediction at X = 1. The lower bound of the interval represents the estimated minimum value, while the upper bound represents the estimated maximum value. In simpler terms, the confidence interval offers a reasonable range within which we can anticipate the model's prediction at X = 1 to fall.

Question 4

Consider the College data set in the ISLR2 package.

Part A

Split the data set into a training and validation set.

```
data(College)
```

```
set.seed(4)
train = sample(nrow(College), nrow(College)*.8)
train_set <- College[train, ]
valid_set <- College[-train, ]
```

Part B

Perform logistic regression on the training data to predict the variable Private using all other variables. Provide an interpretation of the coefficient for Top10Perc.

```
logRegMulti <- glm(Private ~ ., data=College, family="binomial")
summary(logRegMulti)
```

```
##
## Call:
## glm(formula = Private ~ ., family = "binomial", data = College)
```

```
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.7673  -0.0318   0.0502   0.1717   4.2070  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -2.574e-02  1.860e+00  -0.014  0.98896    
## Apps        -5.138e-04  2.284e-04  -2.249  0.02452 *  
## Accept       9.328e-05  4.382e-04   0.213  0.83144    
## Enroll       1.331e-03  8.487e-04   1.568  0.11687    
## Top10perc    8.451e-03  2.841e-02   0.297  0.76614    
## Top25perc    7.305e-03  1.895e-02   0.385  0.69993    
## F.Undergrad -4.168e-04  1.472e-04  -2.832  0.00462 ** 
## P.Undergrad  1.836e-05  1.348e-04   0.136  0.89164    
## Outstate     6.822e-04  1.099e-04   6.207  5.4e-10 ***
## Room.Board   1.901e-04  2.575e-04   0.738  0.46053    
## Books        2.059e-03  1.318e-03   1.562  0.11837    
## Personal    -3.283e-04  2.700e-04  -1.216  0.22395    
## PhD         -6.027e-02  2.665e-02  -2.262  0.02371 *  
## Terminal    -3.590e-02  2.580e-02  -1.392  0.16402    
## S.F.Ratio   -8.461e-02  6.076e-02  -1.393  0.16372    
## perc.alumni  4.782e-02  2.097e-02   2.280  0.02260 *  
## Expend       2.077e-04  1.207e-04   1.721  0.08529 .  
## Grad.Rate    1.634e-02  1.171e-02   1.395  0.16294    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 910.75  on 776  degrees of freedom
## Residual deviance: 239.50  on 759  degrees of freedom
## AIC: 275.5
## 
## Number of Fisher Scoring iterations: 8
```

```
exp(coefficients(logRegMulti))
```

```
## (Intercept)        Apps      Accept      Enroll   Top10perc   Top25perc 
##   0.9745854   0.9994864   1.0000933   1.0013317   1.0084863   1.0073315 
## F.Undergrad P.Undergrad    Outstate  Room.Board       Books    Personal 
##   0.9995833   1.0000184   1.0006824   1.0001901   1.0020609   0.9996717 
##         PhD    Terminal   S.F.Ratio perc.alumni      Expend   Grad.Rate 
##   0.9415102   0.9647323   0.9188665   1.0489822   1.0002077   1.0164721 
```

$\beta_4 =$ 8.451e-03 corresponding to Top10perc

The coefficient for the variable Top10perc suggests that a one-unit increase in Top10perc is expected to result in a logarithmic odds increase of around 0.008451 for the outcome variable. However, it is crucial to highlight that this estimate does not achieve statistical significance. This lack of significance implies that the observed relationship between Top10perc and the outcome variable may not hold any meaningful or distinguishable pattern from random variation.

Part C

What is the test error for the logistic regression (justify your selection of your threshold)?

```
# Set the number of folds for cross-validation
num_folds <- 10

accuracy <- numeric(num_folds)
set.seed(123)  # For reproducibility
fold_indices <- sample(rep(1:num_folds, length.out = nrow(College)))

# Perform cross-validation
for (i in 1:num_folds) {
  # Split the data into training and test sets based on the fold indices
  train_data <- College[fold_indices != i, ]
  test_data <- College[fold_indices == i, ]

  # Fit logistic regression model on training data
  logreg_model <- glm(Private ~ ., data = train_data, family = "binomial")

  # Predict on test data
  test_preds <- predict(logreg_model, newdata = test_data, type = "response")
  test_class_preds <- ifelse(test_preds > 0.65, "Yes", "No")

  accuracy[i] <- mean(test_class_preds == test_data$Private)
}
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#average test error
test_error <- 1 - mean(accuracy)
print(test_error)
```

```
## [1] 0.06182151
```

```
library(pROC)
```

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

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

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Fit logistic regression model on the entire dataset
logreg_model <- glm(Private ~ ., data = College, family = "binomial")
pred_probs <- predict(logreg_model, type = "response")

# Create a ROC object
roc.info <- roc(College$Private, pred_probs, legacy.axes = TRUE)
```

```
## Setting levels: control = No, case = Yes
```

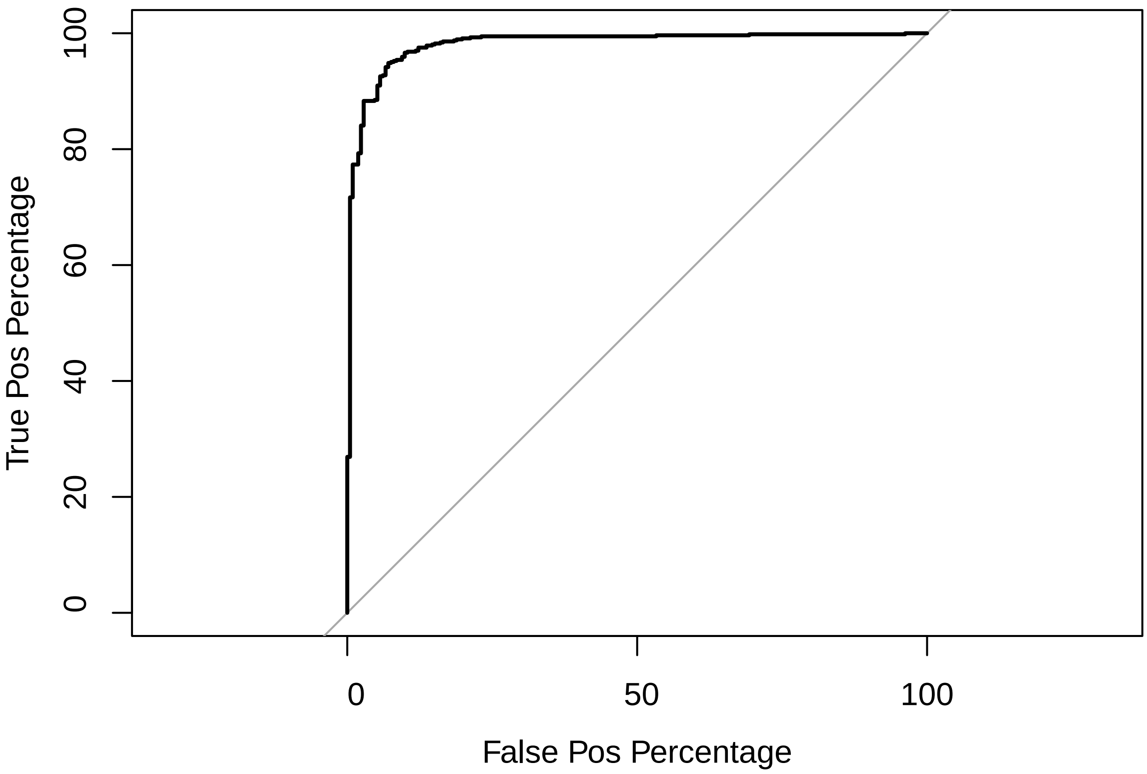
```
## Setting direction: controls < cases
```

```
roc.df <- data.frame(tpp = roc.info$sensitivities*100, fpp = (1 - roc.info$specificities)*100, threshol

# Get the coordinates of the point with the highest AUC
best_coords <- coords(roc.info, "best")

# Extract the best threshold and corresponding AUC
best_threshold <- best_coords$threshold
best_threshold
```

```
## [1] 0.6538537
```

```
plot.roc(College$Private, predict(logreg_model, type = "response"), legacy.axes = TRUE, percent = TRUE,
```

```
## Setting levels: control = No, case = Yes
## Setting direction: controls < cases
```



```
#####################
```

We ended up choosing **.65** as our threshold by selecting the threshold corresponding to the best sum of sensitivity + specificity respectively.

Part D

Fit an LDA to the same model, and report the test error.

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```r
num_folds <- 10

accuracy <- numeric(num_folds)

set.seed(123)
fold_indices <- sample(rep(1:num_folds, length.out = nrow(College)))

#cross-validation
for (i in 1:num_folds) {
  train_data <- College[fold_indices != i, ]
  test_data <- College[fold_indices == i, ]

  # Fit LDA model on training data
  lda_model <- lda(Private ~ ., data = train_data)

  # Predict on test data
  test_preds <- predict(lda_model, newdata = test_data)
  test_class_preds <- test_preds$class

  accuracy[i] <- mean(test_class_preds == test_data$Private)
}

test_error <- 1 - mean(accuracy)
print(test_error)
```

```
## [1] 0.06438561
```

Part E

Fit an QDA to the same model, and report the test error.

```r
library(MASS)
num_folds <- 10
accuracy <- numeric(num_folds)
```

```r
set.seed(123)
fold_indices <- sample(rep(1:num_folds, length.out = nrow(College)))

for (i in 1:num_folds) {
  train_data <- College[fold_indices != i, ]
  test_data <- College[fold_indices == i, ]

  # Fit QDA model on training data
  qda_model <- qda(Private ~ ., data = train_data)

  # Predict on test data
  test_preds <- predict(qda_model, newdata = test_data)
  test_class_preds <- test_preds$class

  accuracy[i] <- mean(test_class_preds == test_data$Private)
}

test_error <- 1 - mean(accuracy)
print(test_error)
```

```
## [1] 0.1004163
```

Part F

Fit an SVM to the same model, and report the test error.

```r
library(e1071)
num_folds <- 10
accuracy <- numeric(num_folds)

set.seed(123)
fold_indices <- sample(rep(1:num_folds, length.out = nrow(College)))

for (i in 1:num_folds) {
  train_data <- College[fold_indices != i, ]
  test_data <- College[fold_indices == i, ]

  # Fit SVM model on training data
  svm_model <- svm(Private ~ ., data = train_data)

  # Predict on test data
  test_preds <- predict(svm_model, newdata = test_data)

  accuracy[i] <- mean(test_preds == test_data$Private)
}
test_error <- 1 - mean(accuracy)
print(test_error)
```

```
## [1] 0.06828172
```

Part G

Pick which model you think is the best and explain your choice. Among the models considered, Logistic Regression demonstrates the lowest prediction error, indicating its superior performance in predicting whether a school is private or not. Therefore, Logistic Regression is the recommended choice for this particular prediction task. On the other hand, Linear Discriminant Analysis (LDA) is known to be highly sensitive to non-Gaussian data. Given this characteristic, it becomes apparent that LDA may not be the optimal choice for the task at hand, as it may struggle to effectively handle datasets that deviate from Gaussian distribution assumptions. Thus, based on the performance and sensitivity considerations, Logistic Regression emerges as the preferable model, while LDA may not be suitable given its limitations with non-Gaussian data.

Question 5

For this problem use the protein.csv file which contains protein consumption in twenty-five European countries for nine food groups. It is available in the MultBiplotR R package.

Part A

Perform principal component analysis on these data (omitting variables Comunist and Region). Report the proportion of variance and cumulative proportion of variance explained by the first 5 principal components.

```r
protein <- read.csv("C:/Users/oklah/OneDrive/Spring23/Math 189/protein.csv")
# Remove Comunist column.
protein <- protein[, -1]
# Remove Region column.
protein <- protein[, -1]
# Perform PCA.
pca <- prcomp(protein, scale = TRUE)
```

Proportion of variance:

```r
pov <- (pca$sdev^2) / sum(pca$sdev^2)
pov
```

```
## [1] 0.44515973 0.18166661 0.12532439 0.10607377 0.05153760 0.03612566 0.03017848
## [8] 0.01292132 0.01101243
```

Cumulative proportion of variance explained by the first 5 principal components:

```r
cpov <- cumsum(pov)[1:5]
cpov
```

```
## [1] 0.4451597 0.6268263 0.7521507 0.8582245 0.9097621
```

Part B

Provide an interpretation of the first two principal components.

```
pca$rotation
```

```
##                          PC1         PC2         PC3          PC4         PC5
## Red_Meat          -0.3026094 -0.05625165 -0.29757957 -0.646476536  0.32216008
## White_Meat        -0.3105562 -0.23685334  0.62389724  0.036992271 -0.30016494
## Eggs              -0.4266785 -0.03533576  0.18152828 -0.313163873  0.07911048
## Milk              -0.3777273 -0.18458877 -0.38565773  0.003318279 -0.20041361
## Fish              -0.1356499  0.64681970 -0.32127431  0.215955001 -0.29003065
## Cereal             0.4377434 -0.23348508  0.09591750  0.006204117  0.23816783
## Starch            -0.2972477  0.35282564  0.24297503  0.336684733  0.73597332
## Nuts               0.4203344  0.14331056 -0.05438778 -0.330287545  0.15053689
## Fruits_Vegetables  0.1104199  0.53619004  0.40755612 -0.462055746 -0.23351666
##                           PC6         PC7         PC8        PC9
## Red_Meat          -0.45986989  0.15033385 -0.01985770  0.2459995
## White_Meat        -0.12100707 -0.01966356 -0.02787648  0.5923966
## Eggs               0.36124872 -0.44327151 -0.49120023 -0.3333861
## Milk               0.61843780  0.46209500  0.08142193  0.1780841
## Fish              -0.13679059 -0.10639350 -0.44873197  0.3128262
## Cereal             0.08075842  0.40496408 -0.70299504  0.1522596
## Starch             0.14766670  0.15275311  0.11453956  0.1218582
## Nuts               0.44701001 -0.40726235  0.18379989  0.5182749
## Fruits_Vegetables  0.11854972  0.44997782  0.09196337 -0.2029503
```

Principal Component 1 captures the largest weights for Eggs, Cereal, and Nuts, indicating their strong influence in shaping this component. This principal component essentially represents a dietary pattern characterized by a high intake of protein and fat (represented by nuts and eggs), as well as carbohydrates (represented by cereals). These three features contribute significantly to the overall variation in the data, accounting for approximately 44.5% of the total variation. Hence, they hold greater importance and exhibit larger magnitudes in the first principal component.

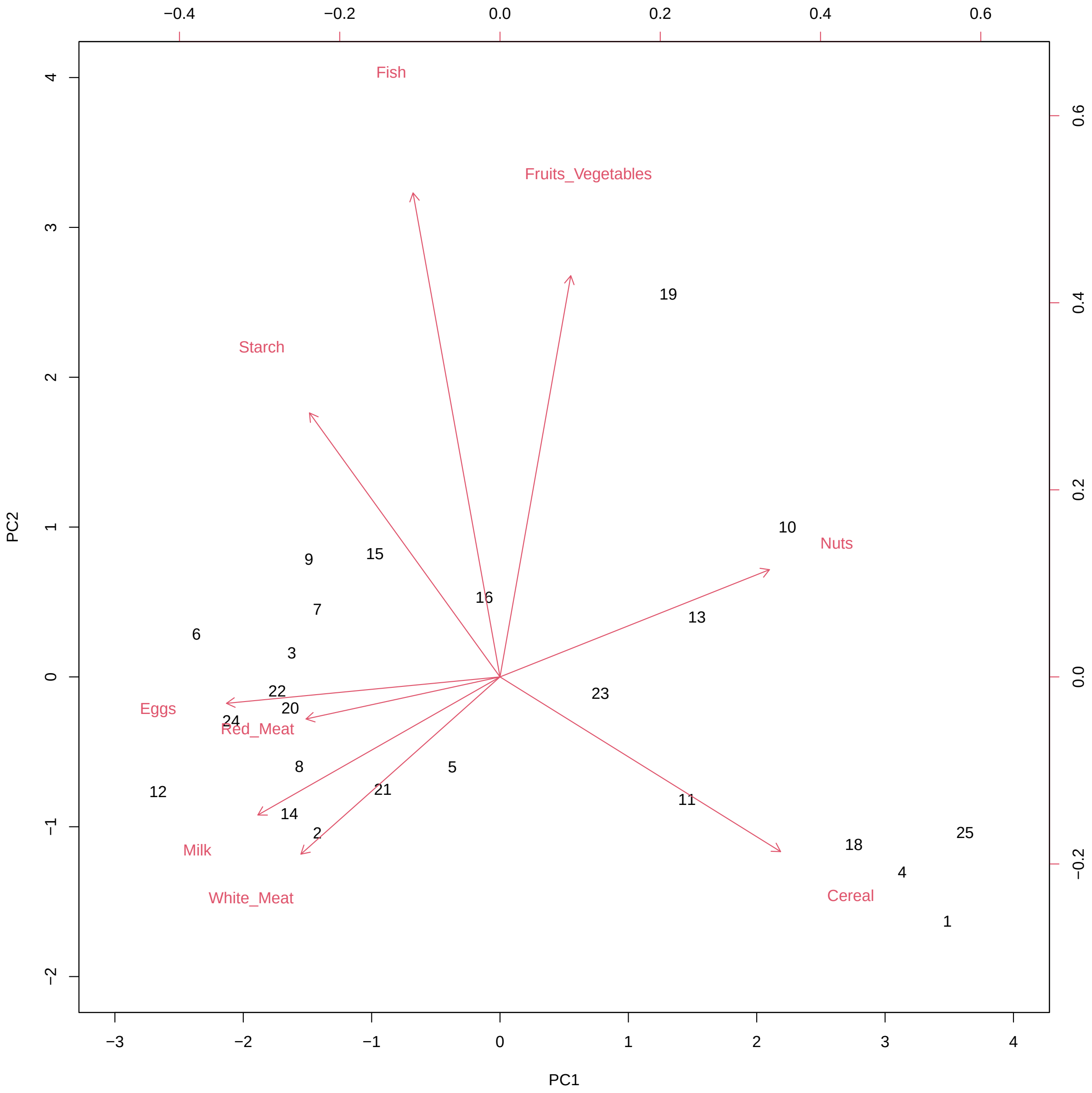
Moving on to Principal Component 2, it assigns considerable weights to fish, fruits/vegetables, and potentially starch, while placing less emphasis on the remaining features. This particular principal component appears to describe a dietary pattern that emphasizes the consumption of natural sources of essential nutrients, such as vitamins, minerals, and fiber. The three mentioned features play a significant role in shaping this component. Collectively, they contribute to around 18.2% of the total variation in the data, indicating their moderate impact.

In summary, Principal Component 1 captures a diet with a high protein, fat, and carbohydrate content, while Principal Component 2 represents a diet that prioritizes natural nutrient sources like fish, fruits/vegetables, and potentially starch. These components account for a substantial proportion of the overall data variation, further emphasizing their relevance in understanding dietary patterns.

Part C

Create a biplot for the first two principal components. Based on this plot, which variable(s) is Milk most correlated with? Which variable(s) is Milk most negatively correlated with? Which variables is Milk uncorrelated with?

```
biplot(pca, scale = 0, expand=1, cex=1, xlim=c(-3,4), ylim=c(-2,4))
```



Based on this plot, `Milk` is most correlated with `White Meat`, `Red_Meat`, and `Eggs`, and is most negatively correlated with `Nuts`. `Milk` is uncorrelated with `Starch`, `Fish`, `Fruits_Vegetables`, and `Cereal`.

Part D

Comment on the differences between countries in the North Region and Central Region using only the first two principal components and the respective interpretations of those principal components.

Observations corresponding to the North and Central Region:

- North: 6, 8, 15, 20
- Central: 2, 3, 5, 7, 9, 11, 12, 14, 16, 21, 22, 23, 24

When examining the observations from the North Region, it becomes apparent that they predominantly fall within the negative portion of PC1, indicating a preference for meat and other animal-based proteins.

However, when considering PC2, the observations from the North Region are distributed across both the negative and positive portions. The negative portion of PC1 and PC2 tends to feature more meat and animal-based proteins, suggesting that the North Region favors these protein sources. Additionally, there are two observations located in the negative PC1 and positive PC2 region, indicating a preference for starches and fish-based proteins in the North Region.

Shifting focus to the Central Region, a majority of its observations align with the negative portion of both PC1 and PC2, indicating a similar preference for meat and animal-based proteins as observed in the North Region. Similar to the North Region, the Central Region also contains observations in the negative PC1 and positive PC2 region, suggesting a liking for starches and fish-based proteins. Interestingly, one observation (16) is positioned closer to the principal component vector representing fruits/vegetables, indicating a potential preference for proteins derived from fruits and vegetables in the Central Region. Another distinguishing factor is that the Central Region includes two observations in the positive PC1 and negative PC2 region, which is not observed in any of the North Region observations. This region is primarily influenced by the principal component vector representing `Cereal`, suggesting a preference for protein sources derived from cereals, although to a lesser extent since only two observations are located in this region.

Conceptual Problems

Question 6

Explain why the bootstrap may be more beneficial for random forest than it would be for linear regression.

The bootstrap resampling technique provides unique benefits for random forest models but has limited impact on linear regression models. Random forest combines multiple decision trees, each trained on a bootstrap sample created by randomly selecting data from the original dataset. This process allows random forest to create diverse decision trees with different training sets, capturing more variability in the data and improving model performance.

On the other hand, linear regression models, which assume a linear relationship between predictors and the response variable, do not gain significant advantages from the bootstrap. While the bootstrap can help estimate the uncertainty of regression coefficients, it does not fundamentally change the underlying linear relationship assumed by the model. The bootstrap's benefits are more evident in complex models with overfitting issues or a large number of parameters, which are less prominent in linear regression.

To summarize, the bootstrap is particularly beneficial for random forest models as it enables aggregation of diverse models, enhances robustness, and handles high-dimensional data. In contrast, the impact of the bootstrap on linear regression models is less pronounced due to the absence of model aggregation, the linearity assumption, and the relative simplicity of the model.

Question 7

Give an example of a scenario where you test multiple hypotheses but would not want to corect for FEWR or FDR.

During the early stages of a data research project, it is often best to avoid correcting for error rates like FWER or FDR. This phase is focused on exploring the data and identifying patterns, rather than testing specific hypotheses. The goal is to generate multiple hypotheses and decide which ones are worth investigating further. At this stage, strict control of error rates is unnecessary as it can restrict the flexibility needed to fully explore and understand the data. By not correcting for FWER or FDR, we have more freedom to generate hypotheses and analyze the data, allowing for a broader exploration of potential relationships and patterns. Correcting for error rates assumes that we already have well-defined hypotheses and specific claims to test. However, during the initial stages, we are still in the process of formulating these claims. It would be premature to confine our conclusions to specific error rates when we haven't established the hypotheses we want to test.

In summary, during the early stages of a data research project, it is often better to prioritize exploratory analysis and hypothesis generation over strict control of error rates. This approach allows for a more comprehensive exploration of the data and the identification of promising avenues for further investigation.

Question 8

Why is it necessary to be aware of a model's assumptions, and check those assumptions before using the trained model for inference or prediction?

It is crucial to meet the assumptions of statistical models to ensure accurate results, as violating these assumptions can lead to biased estimates, incorrect hypothesis tests, and unreliable confidence intervals. These assumptions are made to ensure the validity of the model and its predictions, and when they are violated, the model fails to accurately represent the data, resulting in biased parameter estimates and potentially incorrect conclusions. Violating assumptions also affects hypothesis tests and confidence intervals, rendering them inaccurate or unreliable. Additionally, prediction accuracy is compromised when assumptions are violated, as models are built based on certain assumptions about variables and data distribution.