Homework 8

Tobias Boggess

2022-04-13

Chapter 11 Problems

Problem 6: Do the following.

a) For each of the following models: • Build a classifier for SleepTrouble • Report its effectiveness on the NHANES training data • Make an appropriate visualization of the model • Interpret the results. What have you learned about people's sleeping habits?

```
nhanes <-
  NHANES %>% select(
    SleepTrouble,
    Age,
    HHIncomeMid,
    Poverty,
    HomeRooms,
    Weight,
    Height,
    BMI,
    Pulse:BPDiaAve,
    DirectChol,
    TotChol,
    UrineVol1,
    SleepHrsNight
  )

nhanes <- na.omit(nhanes)

nhanes.nn <- nnet(as.factor(SleepTrouble) ~ Age + Poverty + Weight + BMI,
                  data = nhanes,
                  size = 9,
                  maxit = 3000,
                  trace = FALSE)
```

```
summary(nhanes.nn)
```

```
## a 4-9-1 network with 55 weights
## options were - entropy fitting
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1
##   -0.57   -0.64   -0.54   -0.54   -0.48
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2
##    0.70   -0.29   -0.62    1.60    0.27
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3
##   -0.48   -0.97   -0.51   -0.12   -0.03
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4
## -171.48   47.67  104.69 -274.38  464.85
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5
##    1.27  129.90  -68.18 -196.06  -48.54
##   b->h6  i1->h6  i2->h6  i3->h6  i4->h6
##   -0.01    0.26    0.02   -0.70   -0.61
##   b->h7  i1->h7  i2->h7  i3->h7  i4->h7
##   -2.11   -0.05    0.07    0.02   -0.06
##   b->h8  i1->h8  i2->h8  i3->h8  i4->h8
##    0.12   -0.09    0.69   -0.55   -0.19
##   b->h9  i1->h9  i2->h9  i3->h9  i4->h9
##    0.19    0.13   -0.27    0.52    0.33
##    b->o   h1->o   h2->o   h3->o   h4->o   h5->o   h6->o   h7->o   h8->o   h9->o
##   -0.07    0.52    0.28   -0.49   -2.05  -18.57   -0.42  -28.20   -0.05   -0.82
```

```
nhanes.nn.preds <- predict(nhanes.nn)
nhanes.nn.preds <- case_when(nhanes.nn.preds < 0.2 ~ "No",
                             nhanes.nn.preds >= 0.2 ~ "Yes")
nhanes1 <- mutate(.data = nhanes, predType = nhanes.nn.preds)


accuracy(
  data = nhanes1,
  truth = as.factor(SleepTrouble),
  estimate = as.factor(predType)
)
```

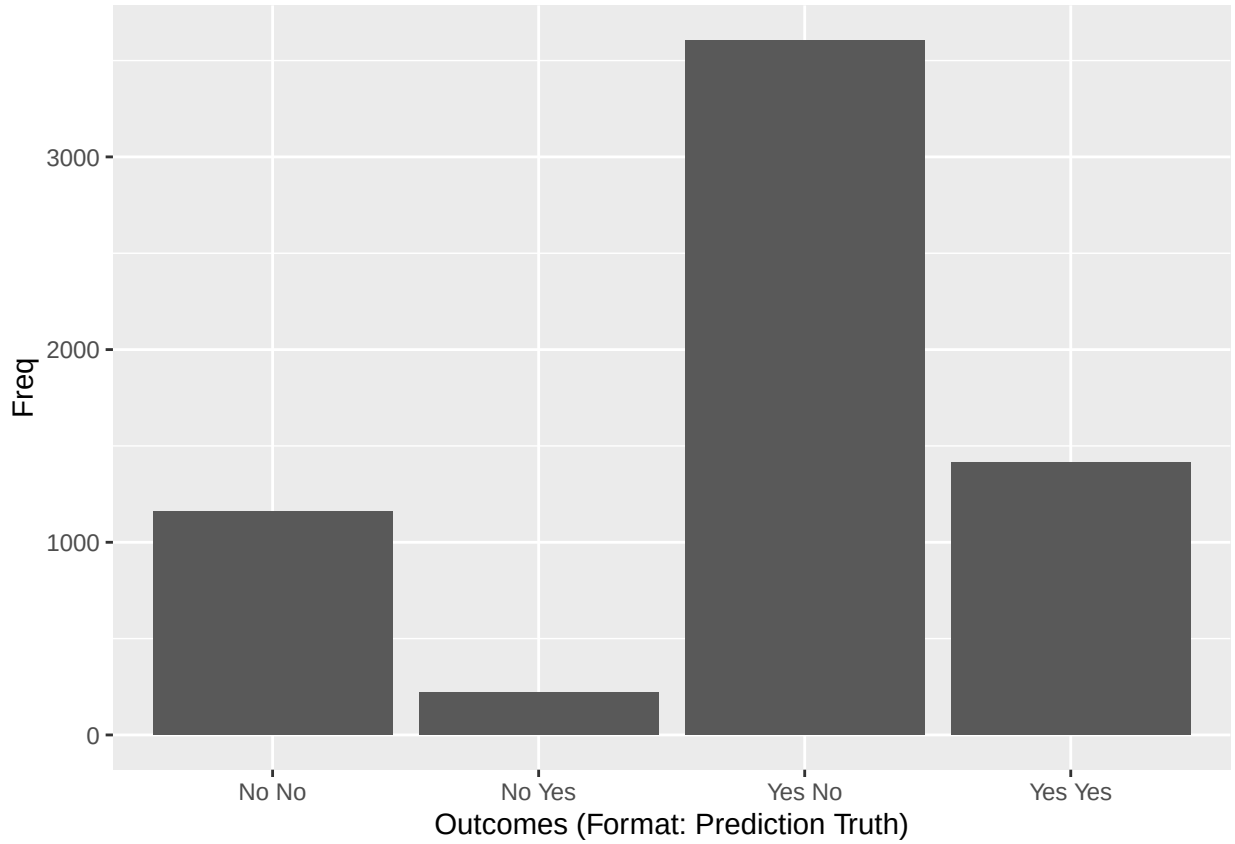
```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.402
```

```
nn.conf_mat <- conf_mat(data = nhanes1, truth = SleepTrouble, estimate = predType)
```

```
## Warning in vec2table(truth = truth, estimate = estimate, dnn = dnn, ...):
## `estimate` was converted to a factor
```

```
nn.conf_mat <- data.frame(nn.conf_mat$table)
nn.conf_mat <- unite(data = nn.conf_mat, col = "Combo", Prediction, Truth, sep = " ")
```

```
ggplot(data = nn.conf_mat) +
  geom_col(mapping = aes(x = Combo, y = Freq)) +
  xlab("Outcomes (Format: Prediction Truth)")
```



The model I was able to create did not have a great accuracy to the model in which age, BMI, and poverty calculations probably aren't the best at predicting the outcomes of whether someone has trouble sleeping or not.

c) Repeat either of the previous exercises, but this time first separate the NHANES data set uniformly at random into 75% training and 25% testing sets. Compare the effectiveness of each model on training vs. testing data
Code:

```
data1 <- sort(sample(nrow(nhanes), nrow(nhanes) * 0.75))

nhanes.train <- nhanes[data1,]
nhanes.test <- nhanes[-data1,]

nhanes.nn1 <- nnet(SleepTrouble ~ Age + Poverty + Weight + BMI,
                   data = nhanes.train,
                   size = 9,
                   maxit = 3000,
                   trace = FALSE)
```

```
summary(nhanes.nn1)
```

```
## a 4-9-1 network with 55 weights
## options were - entropy fitting
##      b->h1     i1->h1     i2->h1     i3->h1     i4->h1
##       0.36       0.53       0.14       0.36       0.23
##      b->h2     i1->h2     i2->h2     i3->h2     i4->h2
##  -23948.63  -54908.34  -38859.05   51705.23  -39077.17
##      b->h3     i1->h3     i2->h3     i3->h3     i4->h3
## -170882.73    -354.82  -10945.04    -555.05   11918.02
##      b->h4     i1->h4     i2->h4     i3->h4     i4->h4
##     867.96   -4023.96   19634.13   12023.91  -14812.66
##      b->h5     i1->h5     i2->h5     i3->h5     i4->h5
##       0.51       0.15      -0.40       0.69      -0.52
##      b->h6     i1->h6     i2->h6     i3->h6     i4->h6
##      -0.40       0.01      -0.02       0.59       0.47
##      b->h7     i1->h7     i2->h7     i3->h7     i4->h7
##    6902.04    -139.20    4754.62     604.31   -1968.46
##      b->h8     i1->h8     i2->h8     i3->h8     i4->h8
##  -99071.91   -2945.89  -36677.86    8874.42  -10780.81
##      b->h9     i1->h9     i2->h9     i3->h9     i4->h9
##  -58642.06   28443.37  -13739.21   -3723.12  -18672.16
##       b->o       h1->o      h2->o      h3->o      h4->o      h5->o      h6->o
##   -3597.37   -3597.22      -0.45      -0.85   14388.65   -3597.13   -3597.70
##      h7->o      h8->o      h9->o
##      -0.61       0.94       0.68
```

```
nhanes.nn.preds1 <- predict(nhanes.nn1, newdata = nhanes.test)
nhanes.nn.preds1 <- case_when(nhanes.nn.preds1 < 0.25 ~ "No",
                              nhanes.nn.preds1 >= 0.25 ~ "Yes")
nhanes.test <- mutate(.data = nhanes.test, predType = nhanes.nn.preds1)


accuracy(
  data = nhanes.test,
  truth = as.factor(SleepTrouble),
  estimate = as.factor(predType)
)
```

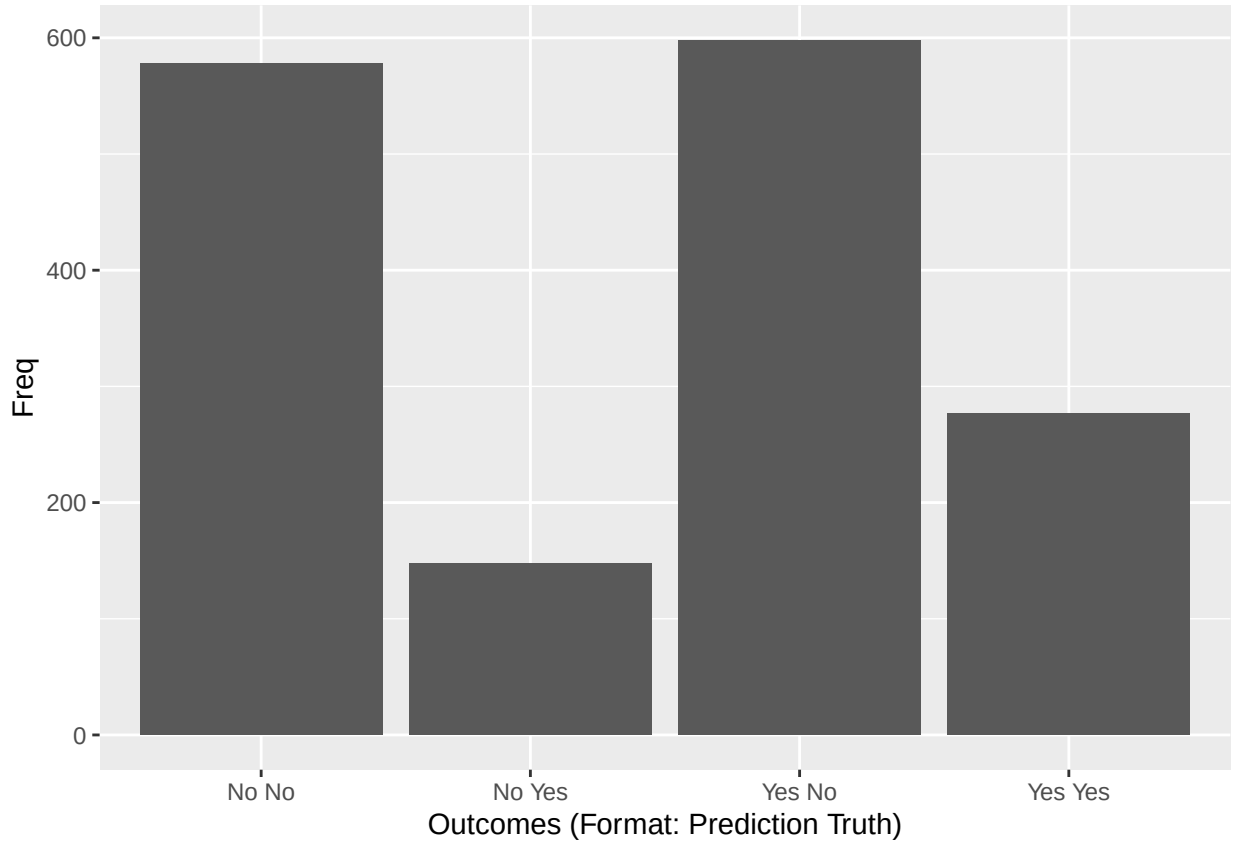
```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.534
```

```
nn.conf_mat1 <- conf_mat(data = nhanes.test, truth = SleepTrouble, estimate = predType)
```

```
## Warning in vec2table(truth = truth, estimate = estimate, dnn = dnn, ...):
## `estimate` was converted to a factor
```

```
nn.conf_mat1 <- data.frame(nn.conf_mat1$table)
nn.conf_mat1 <- unite(data = nn.conf_mat1, col = "Combo", Prediction, Truth, sep = " ")
```

```
ggplot(data = nn.conf_mat1) +
  geom_col(mapping = aes(x = Combo, y = Freq)) +
  xlab("Outcomes (Format: Prediction Truth)")
```



This model does slightly worse than the previous model with the full data set being used to train the model. Also, the variables chosen are the same and allow me to see the effect of using less data points decreases the accuracy of the model.

Chapter 12 Problems

Problem 6 (part a only): Use the kmeans function to perform a cluster analysis on these players. Describe the properties that seem common to each cluster.

Code:

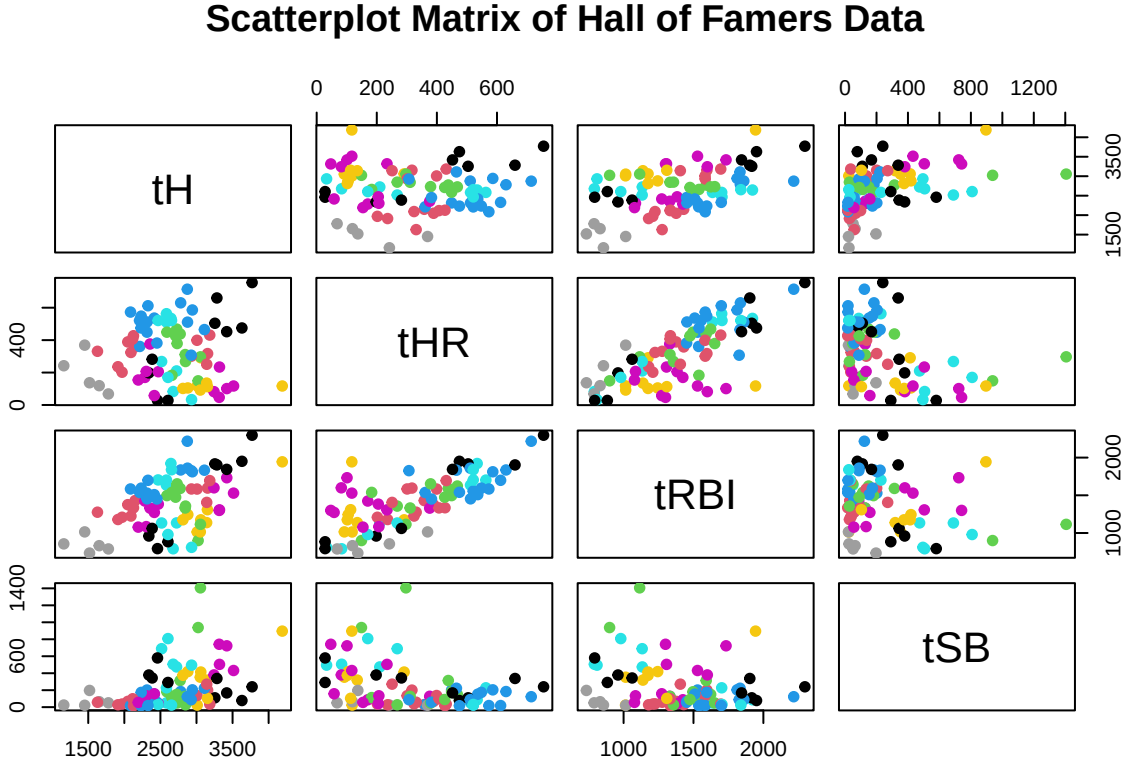
```
hof <- Batting %>% group_by(playerID) %>%
  inner_join(HallOfFame, by = c("playerID" = "playerID")) %>%
  filter(inducted == "Y" & votedBy == "BBWAA") %>%
  summarize(
    tH = sum(H),
    tHR = sum(HR),
    tRBI = sum(RBI),
    tSB = sum(SB)
  ) %>%
  filter(tH > 1000)

set.seed(21)
hof.kclust <- kmeans(select(hof, -playerID), centers = 15)
hof.kclust
```

```
## K-means clustering with 15 clusters of sizes 5, 5, 2, 10, 5, 6, 1, 5, 4, 8, 8, 6, 5, 5, 7
##
## Cluster means:
##          tH      tHR     tRBI       tSB
## 1  3471.600 569.2000 1982.400  186.8000
## 2  3083.400 339.8000 1572.800  156.0000
## 3  3039.000 223.0000 1007.500 1172.0000
## 4  2280.100 496.9000 1545.200   76.6000
## 5  2587.000 530.2000 1760.600   84.2000
## 6  2337.500 195.0000 1262.167  100.8333
## 7  4189.000 117.0000 1944.000  896.0000
## 8  1512.200 187.0000  845.200   71.8000
## 9  2447.750 134.0000  924.500  398.5000
## 10 2005.250 334.2500 1295.375   47.0000
## 11 2735.625 365.2500 1508.125  131.5000
## 12 2918.333 535.5000 1897.500  151.8333
## 13 2691.000 152.8000  969.600  594.4000
## 14 3362.200 116.2000 1493.600  556.0000
## 15 3015.429 136.5714 1152.857  286.8571
##
## Clustering vector:
##  [1]  1 13 13  4  5 10 10 15 15  8  2  3  8 15 10  7  8 14  6 11 10  4  6  5 15
## [26] 10 12 11 15 10 11  3  2  5 11  2 13  4  8 14  9  4  9  4  1  4  6 14 13  1
## [51]  1 12 11 10  6 13  4  2 11 12  8 11 12  9  4 12 15  9 10 14  4  6  5  4  6
## [76] 14 15 11  5 12  1  2
##
## Within cluster sum of squares by cluster:
##  [1] 443086.0 143614.8 144088.5 255265.8 106310.8 224749.2      0.0 340766.4
##  [9] 175446.8 313358.9 278366.2 302637.2 331071.2 317898.8 348011.1
##  (between_SS / total_SS =  91.2 %)
##
## Available components:
```

```
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
clusters <- hof.kclust$cluster
pairs(select(hof, -playerID),
      col = clusters,
      main = "Scatterplot Matrix of Hall of Famers Data",
      pch = 19)
```



The closest thing I can imagine with 15 clusters is the number of votes acquired to get into the hall of fame. Other than that, I have no idea what else would cause the grouping the way they did.

Worksheet Problems

Problem 1: Repeat Problem 6 (part a only) from Ch 12, but now use hclust() (instead of kmeans()) to perform hierarchical clustering.

Code:

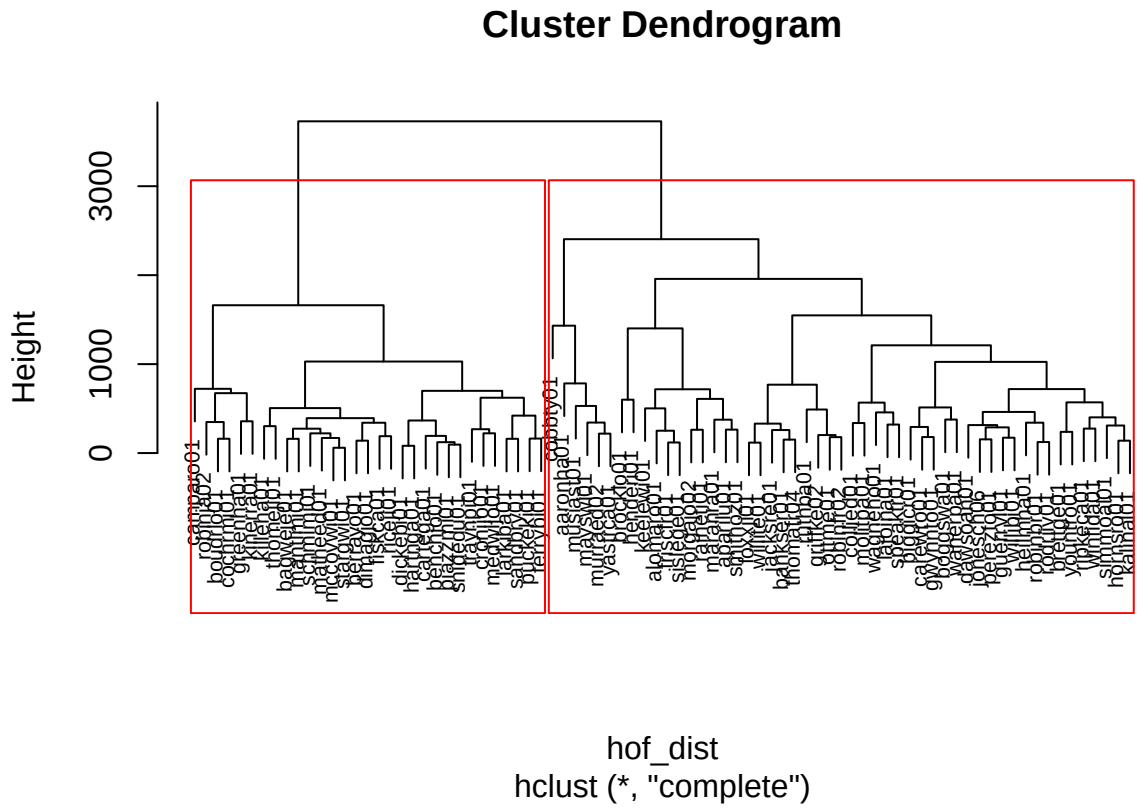
```
rownames(hof) <- hof$playerID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
hof_dist <- dist(hof)
```

```
## Warning in dist(hof): NAs introduced by coercion
```

```
hof_hclust <- hclust(hof_dist)
plot(hof_hclust, cex = 0.7)
rect.hclust(hof_hclust, k = 2, border = "red")
```



Based on the cluster diagram, I imagine the clusters are used to represent the amount of hits the selected hall of famers' acquired throughout their career. This is just speculation and my best guess, though.