HW 7

Enter your name and EID here: Harini Shanmugam

You will submit this homework assignment as a pdf file on Gradescope.

For all questions, include the R commands/functions that you used to find your answer (show R chunk). Answers without supporting code will not receive credit. Write full sentences to describe your findings.

We will use the packages `tidyverse` and `plotROC` for this assignment.

```
# Load packages
library(tidyverse)
library(plotROC)
```

Question 1: (4 pts)

We will use the `pokemon` dataset for this assignment:

```
# Upload data from GitHub
pokemon <- read_csv("https://raw.githubusercontent.com/laylaguyot/datasets/main//pokemon.csv")

# Take a look
head(pokemon)
```

```
## # A tibble: 6 x 13
##   Number Name   Type1 Type2 Total    HP Attack Defense SpAtk SpDef Speed Gener~1
##    <dbl> <chr>  <chr> <chr> <dbl> <dbl>  <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1      1 Bulba~ Grass Pois~   318    45     49      49    65    65    45       1
## 2      2 Ivysa~ Grass Pois~   405    60     62      63    80    80    60       1
## 3      3 Venus~ Grass Pois~   525    80     82      83   100   100    80       1
## 4      3 Venus~ Grass Pois~   625    80    100     123   122   120    80       1
## 5      4 Charm~ Fire  <NA>    309    39     52      43    60    50    65       1
## 6      5 Charm~ Fire  <NA>    405    58     64      58    80    65    80       1
## # ... with 1 more variable: Legendary <lgl>, and abbreviated variable name
## #   1: Generation
```

Recode the variable `Legendary`, taking a value of 0 if a Pokemon is not legendary and a value of 1 if it is. Save the resulting data as `my_pokemon`.

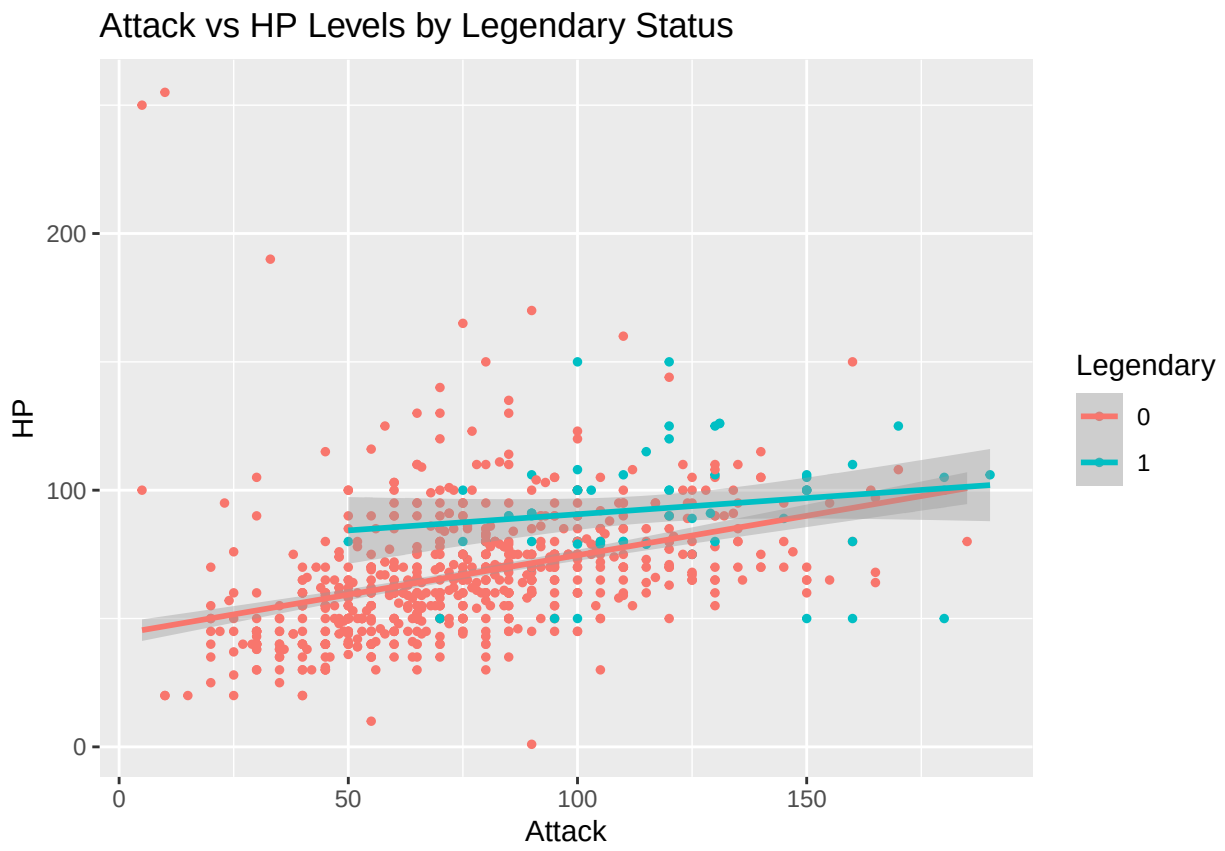
```
# Save to new object
 my_pokemon <- pokemon %>%
  # If Legend is TRUE, then rewrite as 1. Otherwise, rewrite as 0.
  mutate(Legendary=ifelse(Legendary=='TRUE',1,0))
```

```
# View first 6 rows of data set
head(my_pokemon)
```

```
## # A tibble: 6 x 13
##   Number Name   Type1 Type2 Total    HP Attack Defense SpAtk SpDef Speed Gener~1
##    <dbl> <chr>  <chr> <chr> <dbl> <dbl>  <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1      1 Bulba~ Grass Pois~   318    45     49      49    65    65    45       1
## 2      2 Ivysa~ Grass Pois~   405    60     62      63    80    80    60       1
## 3      3 Venus~ Grass Pois~   525    80     82      83   100   100    80       1
## 4      3 Venus~ Grass Pois~   625    80    100     123   122   120    80       1
## 5      4 Charm~ Fire  <NA>    309    39     52      43    60    50    65       1
## 6      5 Charm~ Fire  <NA>    405    58     64      58    80    65    80       1
## # ... with 1 more variable: Legendary <dbl>, and abbreviated variable name
## #   1: Generation
```

Visualize the linear relationship between `Attack` and `HP` (hit points) for each legendary status. *Hint: consider the binary variable as a factor using* ***`as.factor()`***. Do `Attack` and `HP` seem to predict Legendary status? Comment with what you see in the visualization.

```
# Attack vs HP plot colored by Legendary Status
 ggplot(my_pokemon, aes(x=Attack, y=HP, color=as.factor(Legendary)))+
  # Decrease plot point size to 1
  geom_point(size=1) +
  labs(title="Attack vs HP Levels by Legendary Status",
       # Change legend title
       color="Legendary") +
   # Linear model line
   geom_smooth(method="lm")
```



The slopes for both linear models are close to 0, indicating no strong correlation between Attack and HP levels. So it is fair to say that Attack and HP levels do not predict Legendary status.

Question 2: (2 pt)

Let's predict `Legendary` status using a linear regression model with `Attack` and `HP` in `my_pokemon`. Fit this model, call it `pokemon_lin`, and write its equation.

```
# Linear regression model using Attack and HP to predict Legendary status
pokemon_lin <- lm(Legendary ~ Attack+HP, data = my_pokemon)

# Take a look at the model summary
summary(pokemon_lin)
```

```
##
## Call:
## lm(formula = Legendary ~ Attack + HP, data = my_pokemon)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40650 -0.12385 -0.05025  0.01914  0.97201
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.2201775  0.0289417  -7.608 7.88e-14 ***
## Attack       0.0023563  0.0003054   7.715 3.61e-14 ***
## HP           0.0016644  0.0003882   4.288 2.03e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.254 on 797 degrees of freedom
## Multiple R-squared:  0.1392, Adjusted R-squared:  0.137
## F-statistic: 64.42 on 2 and 797 DF,  p-value: < 2.2e-16
```

Legend Status = - 0.220178 + 0.002356*Attack + 0.00166*HP

Question 3: (3 pts)

Choose a Pokemon whose name starts with the same letter as yours. Take a look at its stats and, using the equation of your model from the previous question, predict the legendary status of this Pokemon, "by hand":

```
# Find and select a pokemon that starts with the letter H
my_pokemon %>%
  # Filter names that start with the letter H
  filter(str_detect(Name, '^H')) %>%
  # Filter for just Hypno
  filter(Name == "Hypno")
```

```
## # A tibble: 1 x 13
##   Number Name  Type1  Type2 Total    HP Attack Defense SpAtk SpDef Speed Gener~1
##    <dbl> <chr> <chr>  <chr> <dbl> <dbl>  <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1     97 Hypno Psych~ <NA>    483    85     73      70    73   115    67       1
## # ... with 1 more variable: Legendary <dbl>, and abbreviated variable name
## #   1: Generation
```

Legend Status = - 0.220178 + 0.002356*Attack + 0.001664*HP

Legend Status= - 0.220178 + 0.002356*73 + 0.001664*85

Legend Status = 0.09325 ~ 0

Check your answer by using `predict()` with the argument `newdata =`:

```
# Predicted Legendary value for Hypno using linear regression model
predict(pokemon_lin, newdata=my_pokemon %>%
  filter(Name == "Hypno"))
```

```
##          1
## 0.09330972
```

Was your Pokemon predicted to be legendary? Why or why not? Does it match the reality?

It was not predicted to be legendary as the predicted value was 0.09330972 which is close to 0 (not being legendary). Yes, this matches the reality that Hypno is not legendary.

Question 4: (2 pts)

We can measure how far off our predictions are from reality with residuals. Use `resid()` to find the residuals of each Pokemon in the dataset then find the sum of all residuals. Why does it make sense?

```
# Caluclate residuals
resid(pokemon_lin) %>%
  # Sum all residuals
  sum
```

```
## [1] 2.775558e-15
```

The sum of all residuals is 2.775558e-15 which is almost 0. This makes sense because the linear model is trying to best fit the data/ be in the center thus having positive and negative residuals. So the sum of all of those residuals should even out and be close to 0.

Question 5: (2 pts)

A logistic regression would be more appropriate to predict `Legendary` status since it can only take two values. Fit this new model with `Attack` and `HP`, call it `pokemon_log`, and write its equation. *Hint: the logit form is given by the R output.*

```
# Fit the model
pokemon_log <- glm(Legendary ~ Attack+HP, data = my_pokemon, family = "binomial")

# Take a look at the model summary
summary(pokemon_log)
```

```
##
## Call:
## glm(formula = Legendary ~ Attack + HP, family = "binomial", data = my_pokemon)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8418  -0.3693  -0.2204  -0.1334   2.8555
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.659078   0.680595 -11.253  < 2e-16 ***
## Attack       0.032901   0.004431   7.425 1.12e-13 ***
## HP           0.025923   0.004982   5.203 1.96e-07 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 450.90  on 799  degrees of freedom
## Residual deviance: 340.34  on 797  degrees of freedom
## AIC: 346.34
##
## Number of Fisher Scoring iterations: 6
```

ln(p/1-p) = - 7.659078 + 0.032901*Attack + 0.025923*HP where p is the probability of a "success" for when Legendary is 1 or True.

Question 6: (2 pts)

According to this new model, is the Pokemon you chose in question 3 predicted to be legendary? Why or why not? *Hint: you can use predict() with the arguments* ***newdata =*** *and* ***type = "response"****.*

```
# Predicted Legendary value for Hypno using logistic regression model
predict(pokemon_log, newdata=my_pokemon %>% filter(Name == "Hypno"), type="response")
```

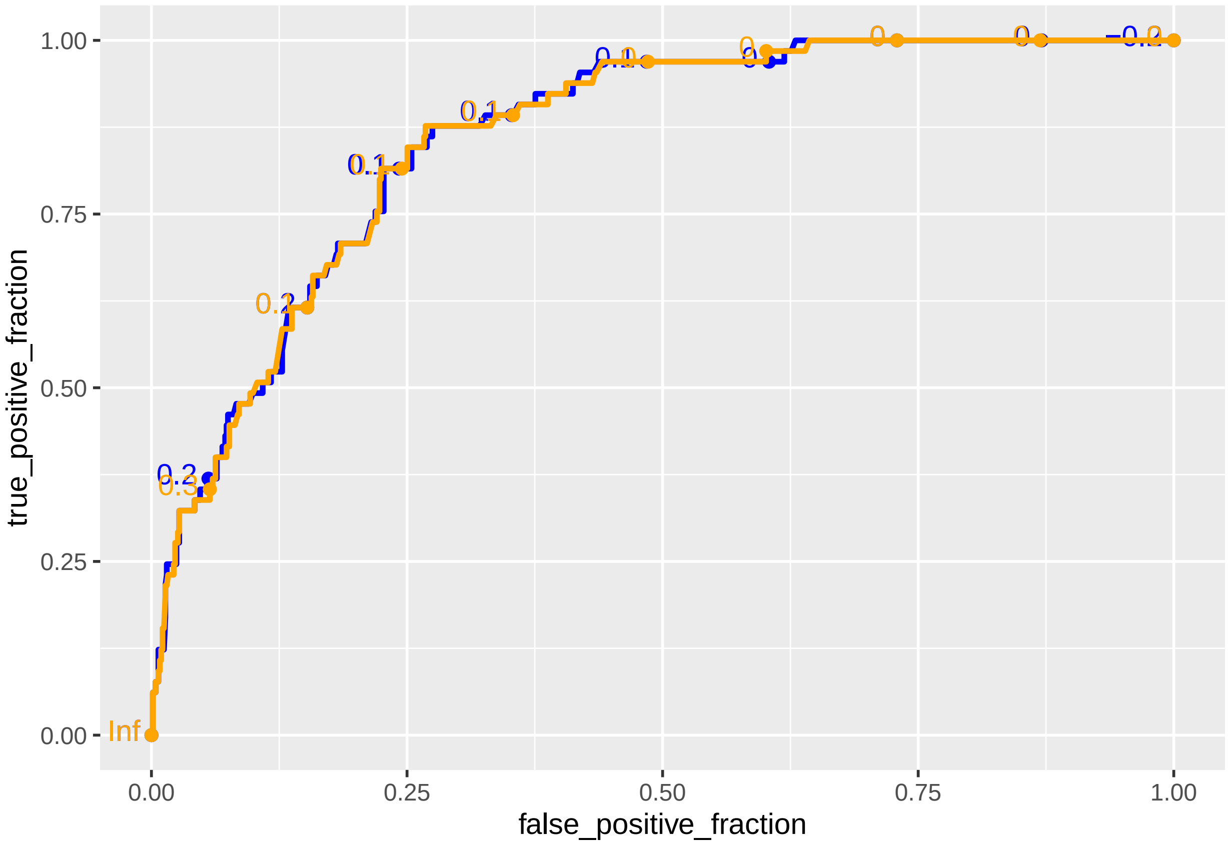
```
##          1
## 0.04505006
```

It was not predicted to be legendary as the predicted value was 0.04505006 which is close to 0 (not being legendary). Yes, this matches the reality that Hypno is not legendary.

Question 7: (3 pts)

Let's compare the performance of these two models using ROC curves. On the same plot, represent the ROC curve for predicting `Legendary` status based on the predictions from the linear regression in blue and another ROC curve based on the predictions from the logistic regression in orange.

```
# ROC curves of linear regression and logistic regression
ggplot(my_pokemon) +
  geom_roc(aes(d = Legendary, # predict legendary 0 = no, 1 = yes
               m = predict(pokemon_lin, my_pokemon)), # use predictions as
           color = "blue",
           n.cuts = 10) +
  geom_roc(aes(d = Legendary,
               m = predict(pokemon_log, my_pokemon, type = "response")),
               color = "orange",
           n.cuts = 10)
```



How do these two models compare?

The two models are very similar. You can conclude the same information from both models.

Formatting: (2 pts)

Comment your code, write full sentences, and knit your file!

```
##                                                                                               sysna
##                                                                                              "Darwi
##                                                                                               relea
##                                                                                              "21.6
##                                                                                               versi
## "Darwin Kernel Version 21.6.0: Wed Aug 10 14:28:35 PDT 2022; root:xnu-8020.141.5~2/RELEASE_ARM64_T81
##                                                                                              nodena
##                                                                            "Harinis-Air.attlocal.ne
##                                                                                               machi
##                                                                                               "arm6
##                                                                                                 log
##                                                                                                "roo
```

```
##                                                                                                    us
##                                                                                         "harinishanmuga
##                                                                                           effective_us
##                                                                                         "harinishanmuga
```