HW 7

Enter your name and EID here: Erik Mercado, emm4376

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)
```

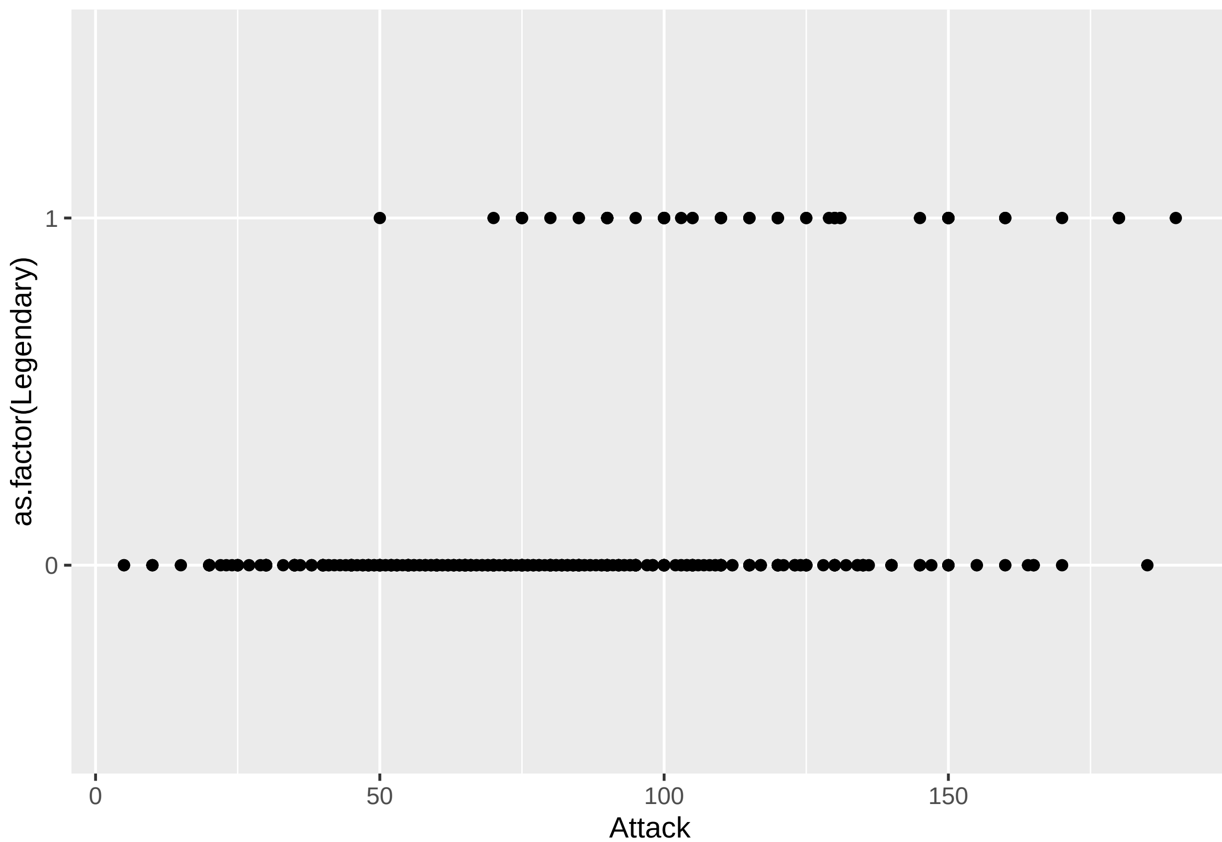
```
## # A tibble: 6 x 13
##   Number Name   Type1 Type2 Total    HP Attack Defense SpAtk SpDef Speed Gener~1
##    <dbl> <chr>  <chr> <chr> <dbl> <dbl>  <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1      1 Bulba~ Grass Pois~   318    45     49      49    65    65    45       1
## 2      2 Ivysa~ Grass 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`.

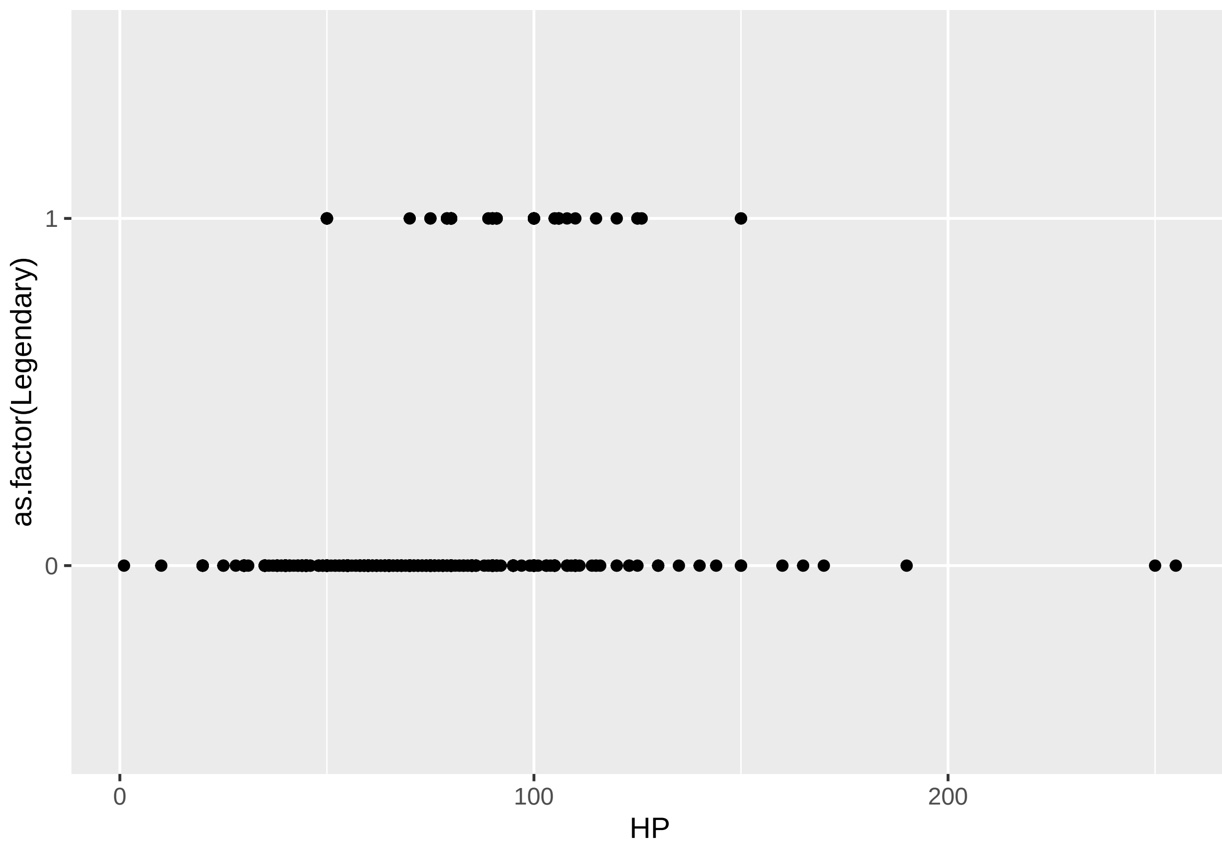
```
# recode legendary as a binary variable
 my_pokemon <- pokemon |>
  mutate(Legendary = ifelse(Legendary == FALSE, 0, 1))
```

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.

```
# plots for attack and hp against legendary status as a factor variable.
 my_pokemon |>
  ggplot(aes(x = Attack, y = as.factor(Legendary))) +
  geom_point()
```




```
my_pokemon |>
  ggplot(aes(x = HP, y = as.factor(Legendary))) +
  geom_point()
```



Attack and HP do not seem to be predict Legendary status as both legendary and non-legendary pokemon seem to have parts of their populations within the same ranges of both.

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.

```
# regress legendary. on attack and hp
pokemon_lin <- glm(Legendary ~ Attack + HP, data = my_pokemon)
```

The regression equation I end up with is Legendary = -0.2201775 + 0.002356294(Attack) + 0.001664444(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" (multiplying the predictors with the estimated coefficients):

```r
# create object for entei and then predicting outcome by hand
Entei <- my_pokemon |>
  filter(Name == "Entei")
Entei |>
  summarize(Entei_Legendary_Chance = -0.2201775 + 0.002356294*Attack + 0.001664444*HP)
```

```
## # A tibble: 1 x 1
##   Entei_Legendary_Chance
##                    <dbl>
## 1                  0.242
```

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

```r
# predict the outcome of entei using previous linear model
 predict(pokemon_lin, newdata = Entei)
```

```
##         1
## 0.2422074
```

Was your pokemon predicted to be legendary (i.e. is the prediction close to 0 or 1)? Why or why not? Does it match character's Legendary status in dataset?

My pokemon was not predicted to be legendary. This is contradictory to what is shown in the dataset, as it shows as legendary in the dataset.

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. What is the sum of all the residuals. Why does it make sense?

```r
# set residuals to a vector and then summing up the residuals
pokemon_resids <- resid(pokemon_lin)
sum(pokemon_resids)
```

```
## [1] 4.850668e-13
```

The sum of all the residuals is 4.850668e-13. This does make sense because the residual sum of a linear regression will always be practically zero because least squares reduces the sum of the squared residuals.

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.*

```r
# log. regression of legenary on attack and hp
 pokemon_log <- glm(Legendary ~ Attack + HP, data = my_pokemon, family = binomial)
```

The equation I end up with after running the logistic regression is Legendary = -7.659078 + 0.03290057*Attack + 0.02592296*HP.

Question 6: (2 pts)

According to this new model, is the pokemon you chose in question 3 predicted to be legendary (i.e. probability is greater than 0.5)? Why or why not? *Hint: you can use predict() with the arguments `newdata =` and `type = "response"`.*

```r
# predict entei's legendary status with the logistic model
predict(pokemon_log, newdata = Entei, type = "response")
```

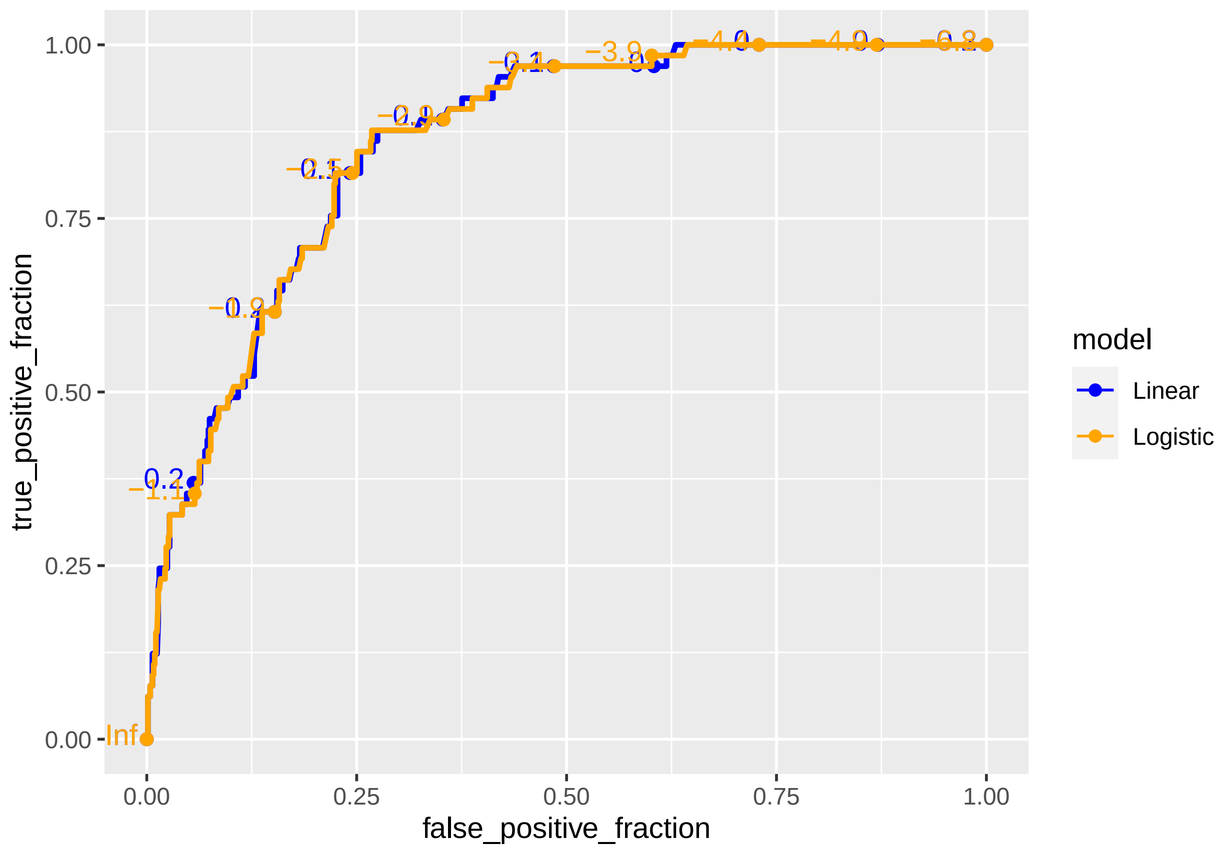
```
##         1
## 0.2902084
```

According to the new model, Entei is not predicted to be legendary. This is because the probability of entei being legendary is less than 0.5.

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.

```r
# create predictions for the linear model
lin_pred <- my_pokemon |>
  select(Legendary) |>
  mutate(predictions = predict(pokemon_lin, my_pokemon),
         predicted = ifelse(predictions > 0.5, 1, 0))
# create predictions for the logistic model
log_pred <- my_pokemon |>
  select(Legendary) |>
  mutate(predictions = predict(pokemon_log, my_pokemon, response = "response"),
         predicted = ifelse(predictions > 0.5, 1, 0))
# create object for the roc curves
roc_model <- bind_rows(lin_pred, log_pred, .id = "model") |>
  mutate(model = ifelse(model == "1", "Linear", "Logistic"))
# plotting roc curves
roc_model |>
   ggplot(aes(d = Legendary, m = predictions, color = model)) +
    geom_roc() +
    scale_color_manual(values = c("blue", "orange"))
```



How do these two models compare?

Both models seems to have the same levels of performance.

Formatting: (2 pts)

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

```
##                                                                                      sysname
##                                                                                     "Darwin"
##                                                                                      release
##                                                                                     "22.4.0"
##                                                                                      version
## "Darwin Kernel Version 22.4.0: Mon Mar  6 21:00:17 PST 2023; root:xnu-8796.101.5~3/RELEASE_X86_64"
##                                                                                     nodename
##                                                                           "Eriks-MBP-2424.lan"
##                                                                                      machine
##                                                                                     "x86_64"
##                                                                                        login
##                                                                                       "root"
```

```
##                                                                    user
##                                                                  "erik"
##                                                          effective_user
##                                                                  "erik"
```