HW 6

Enter your name and EID here: Arsh Ali & asa3683

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)
```

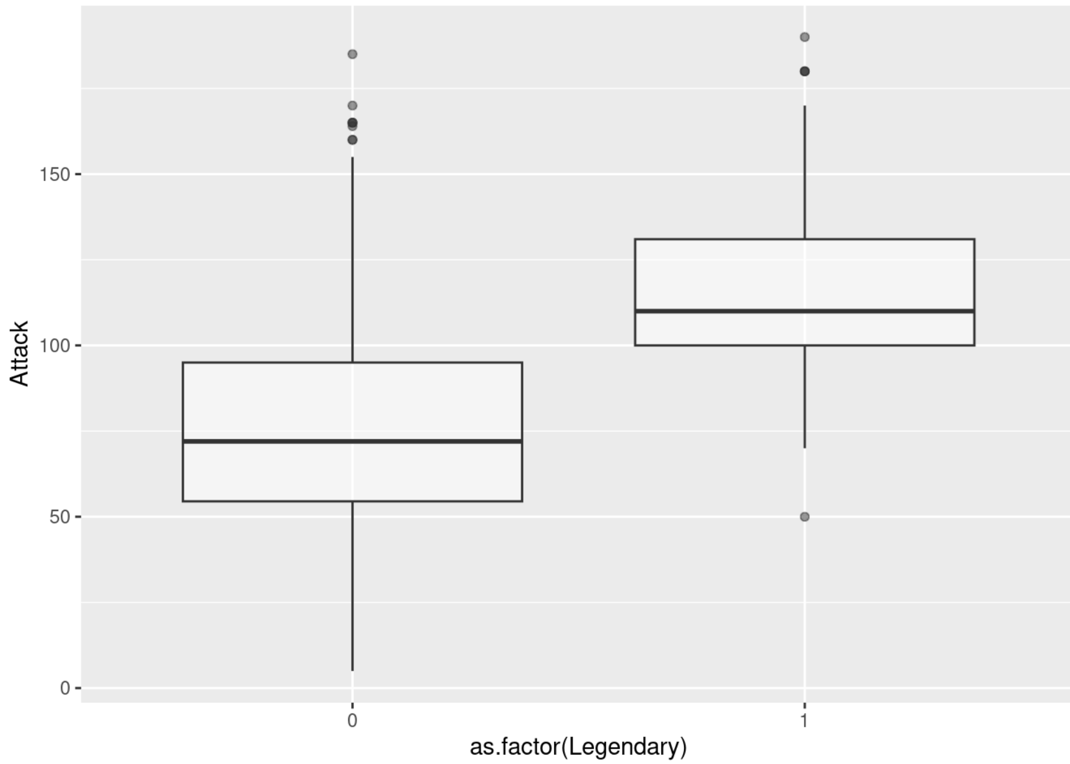
```
## # A tibble: 6 × 13
##   Number Name   Type1 Type2 Total    HP Attack Defense SpAtk SpDef Speed Gener…¹
##    <dbl> <chr>  <chr> <chr> <dbl> <dbl>  <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1      1 Bulba… Grass Pois…   318    45     49      49    65    65    45       1
## 2      2 Ivysa… Grass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

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

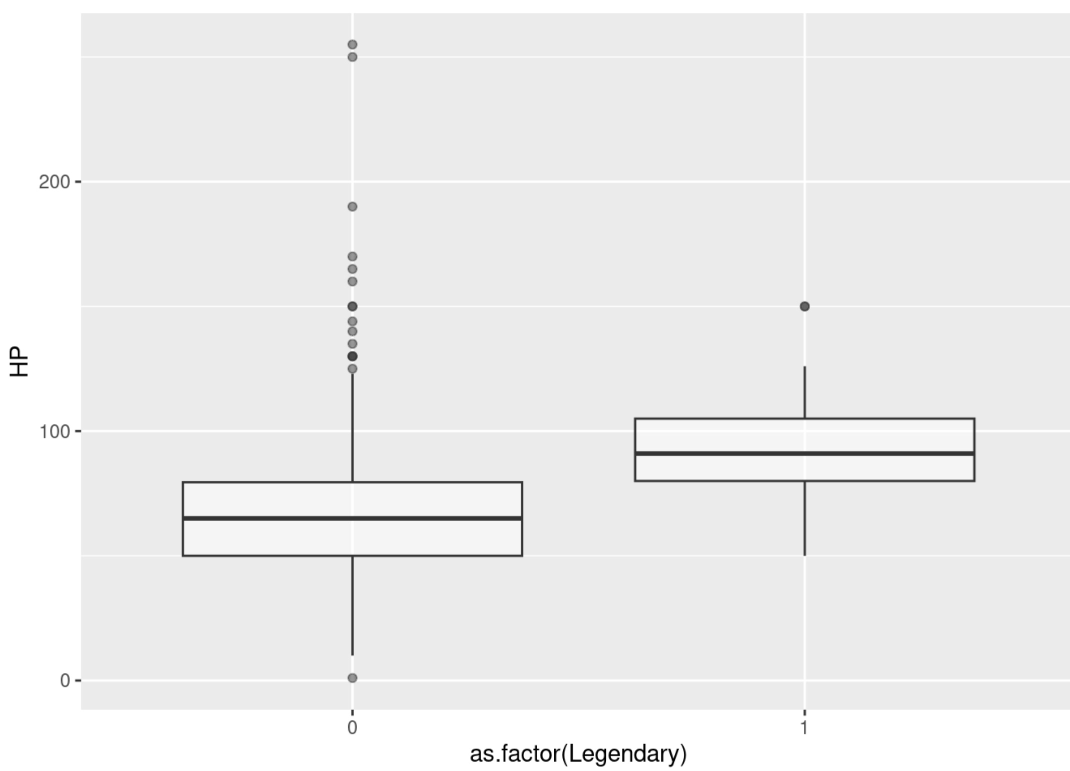
```
# mutate values
my_pokemon <- pokemon %>%
  mutate(Legendary = ifelse(Legendary == TRUE, 1, 0))
```

Let's visualize how the features of `Attack` and `HP` impact the legendary status. First, visualize the distribution of `Attack` for legendary pokemons vs those that are not. Also visualize the distribution of `HP` for these two groups. *Note: consider the binary variable as a factor for your `ggplot` using `as.factor()`.* Comment with what you see in these visualizations.

```
#create a ggplot of the distribution of Attack for legendary vs non-legendary pokemons
ggplot(my_pokemon, aes(x = as.factor(Legendary), y = Attack)) +
  geom_boxplot(alpha = 0.5) +
  labs(y = "Attack", fill = "Legendary")
```



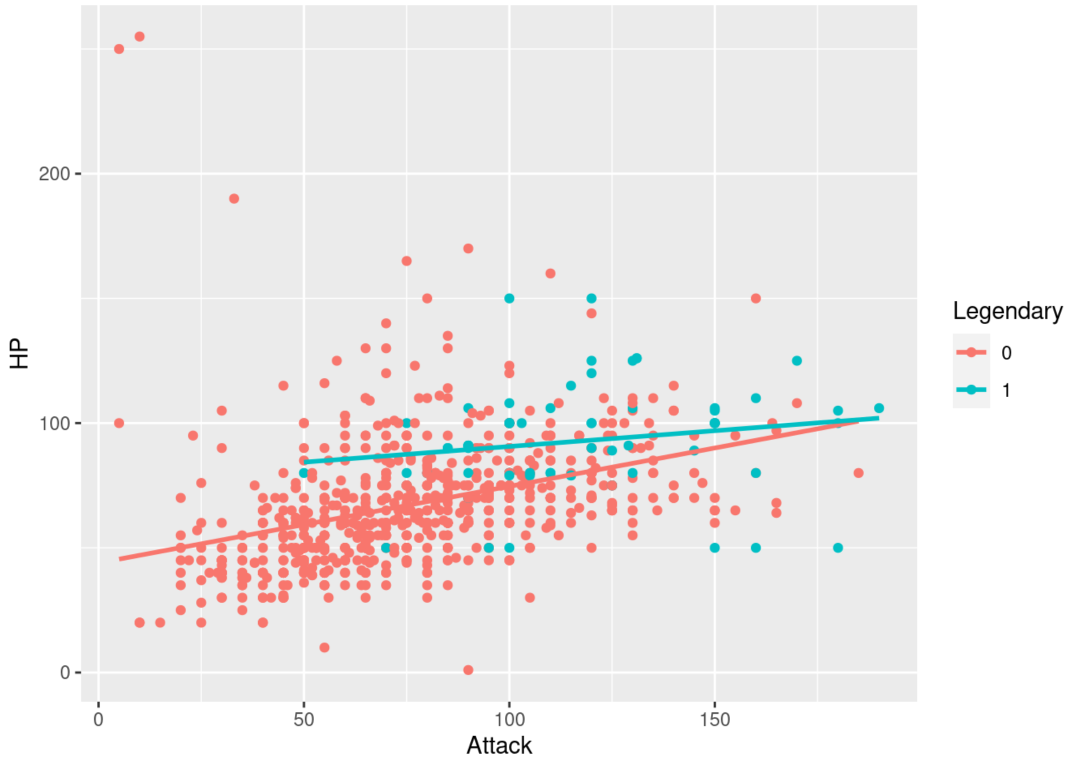

```
# create a ggplot of the distribution of HP for legendary vs non-legendary pokemons
ggplot(my_pokemon, aes(x = as.factor(Legendary), y = HP)) +
  geom_boxplot(alpha = 0.5) +
  labs(y = "HP", fill = "Legendary")
```



It seems as though legendary Pokemons tend to have a higher base attack power in comparison to non-legendary Pokemons. Additionally, those with legendary status also seem to have slightly higher HP values than those with non-legendary status.

Then visualize the linear relationship between `Attack` and `HP` (hit points) for each legendary status. *Hint: color the regression lines.* Do `Attack` and `HP` seem to predict Legendary status? Comment with what you see in this visualization.

```r
# create a ggplot between Attack and HP
ggplot(my_pokemon, aes(x = Attack, y = HP, color = as.factor(Legendary))) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Attack", y = "HP", color = "Legendary")
```



There seems to a positive relationship between Attack and HP across Legendary Status. Specifically, as the base power attack increases so does HP - the lines for both legendary statuses increase as HP and Attack increase but it seems as though there is a more defined relationship between HP and Attack for those without legendary status (the lines is less steep).

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.

```r
# create a linear regression model
pokemon_lin <- lm(Legendary ~ Attack + HP, data = my_pokemon)
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
```

$\widehat{LegendaryStatus} = -0.2201 + 0.0023 * Attack + 0.0016 * HP$

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”:

```
# look at stats for Alakazam
my_pokemon %>%
  filter(Name == "Alakazam")
```

```
## # A tibble: 1 × 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     65 Alaka… Psyc… <NA>    500    55     50      45   135    95   120       1
## # … with 1 more variable: Legendary <dbl>, and abbreviated variable name
## #   ¹Generation
```

```
# calculate legendary status of Pokemon 'Alakazam'
-0.2201775 + (0.0023563 * 50) + (0.0016644 * 45)
```

```
## [1] -0.0274645
```

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

```
# Predict the legendary status for Alakazam
Alakazam <- data.frame(HP = 45, Attack = 50)
predict(pokemon_lin, newdata = Alakazam)
```

```
##           1
## -0.02746281
```

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

The Pokemon, Alakazam, was not predicted to be legendary as the status value is very close to zero according to the model. This matches with reality, as the Pokemon in the original dataset (or in the Pokemon universe) is not legendary indicated by ‘FALSE.’

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?

```
# find residual of each pokemon in dataset
my_pokemon %>%
  mutate(residuals = resid(pokemon_lin)) %>%
  select(Name, Attack, HP, residuals)
```

```
## # A tibble: 800 × 4
##    Name                      Attack    HP residuals
##    <chr>                      <dbl> <dbl>     <dbl>
##  1 Bulbasaur                     49    45    0.0298
##  2 Ivysaur                       62    60   -0.0258
##  3 Venusaur                      82    80   -0.106
##  4 VenusaurMega Venusaur        100    80   -0.149
##  5 Charmander                    52    39    0.0327
##  6 Charmeleon                    64    58   -0.0272
##  7 Charizard                     84    78   -0.108
##  8 CharizardMega Charizard X    130    78   -0.216
##  9 CharizardMega Charizard Y    104    78   -0.155
## 10 Squirtle                      48    44    0.0338
## # … with 790 more rows
```

```
# find sum of residuals
sum(resid(pokemon_lin))
```

```
## [1] -3.068726e-15
```

The sum of the residuals is a small number and close to zero, meaning that the linear regression model with 'Attack' and 'HP' as predictors is a good fit for the data. This makes sense because the predicted legendary status/values we calculated are close to the actual values in the dataset.

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.*

```
# fot a logistic regression model
pokemon_log <- glm(Legendary ~ Attack + HP, data = my_pokemon, family = binomial)
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
```

The equation for this new model is written as below. $\widehat{LegendaryStatus} = -7.659078 + 0.032901 * Attack + 0.025923 * HP$

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"`.

```
Alakazam <- data.frame(HP = 45, Attack = 50)
predict(pokemon_log, newdata = Alakazam, type = "response")
```

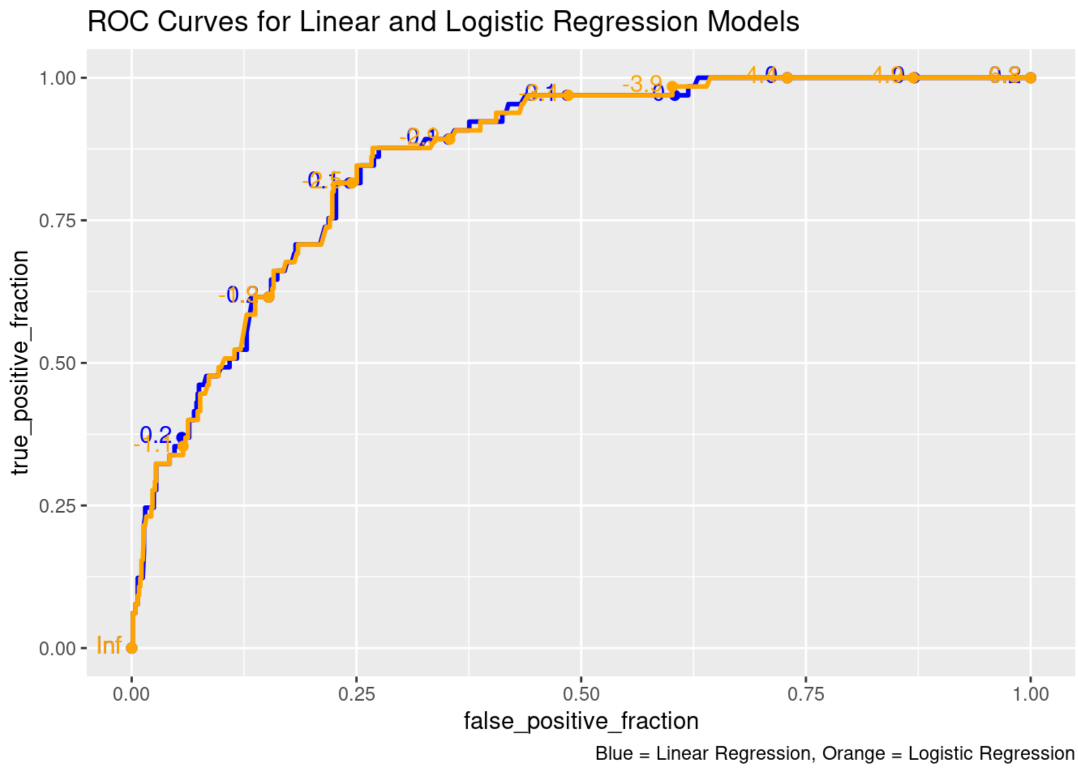
```
##           1
## 0.007786737
```

According to this model, the pokemon, Alakazam, is not predicted to be legendary because it's status value is close to zero in comparison to those with legendary statuses who would have a value of closer to one.

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 regressionin blue and another ROC curve based on the predictions from the logistic regression in orange.

```
# compare performance of the two models using ROC curves
ggplot(my_pokemon) +
  geom_roc(aes(d = Legendary, m = predict(pokemon_lin)), n.cuts = 10, color = "Blue") +
  geom_roc(aes(d = Legendary, m = predict(pokemon_log)),n.cuts = 10, color = "Orange") +
  labs(title = "ROC Curves for Linear and Logistic Regression Models") +
  labs(caption = "Blue = Linear Regression, Orange = Logistic Regression")
```



How do these two models compare?

The linear and logistic regression models seem to be similar to each other based on no differences in the ROC curves, meaning they both can accurately predict the legendary status of a Pokemon.

Formatting: (2 pts)

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

```
##                                                 sysname
##                                                 "Linux"
##                                                 release
##                                    "5.15.0-67-generic"
##                                                 version
## "#74~20.04.1-Ubuntu SMP Wed Feb 22 14:52:34 UTC 2023"
##                                                nodename
##                           "educcomp04.ccbb.utexas.edu"
##                                                 machine
##                                                "x86_64"
##                                                   login
##                                               "unknown"
##                                                    user
##                                               "asa3683"
##                                          effective_user
##                                               "asa3683"
```