The Myths and The Legends

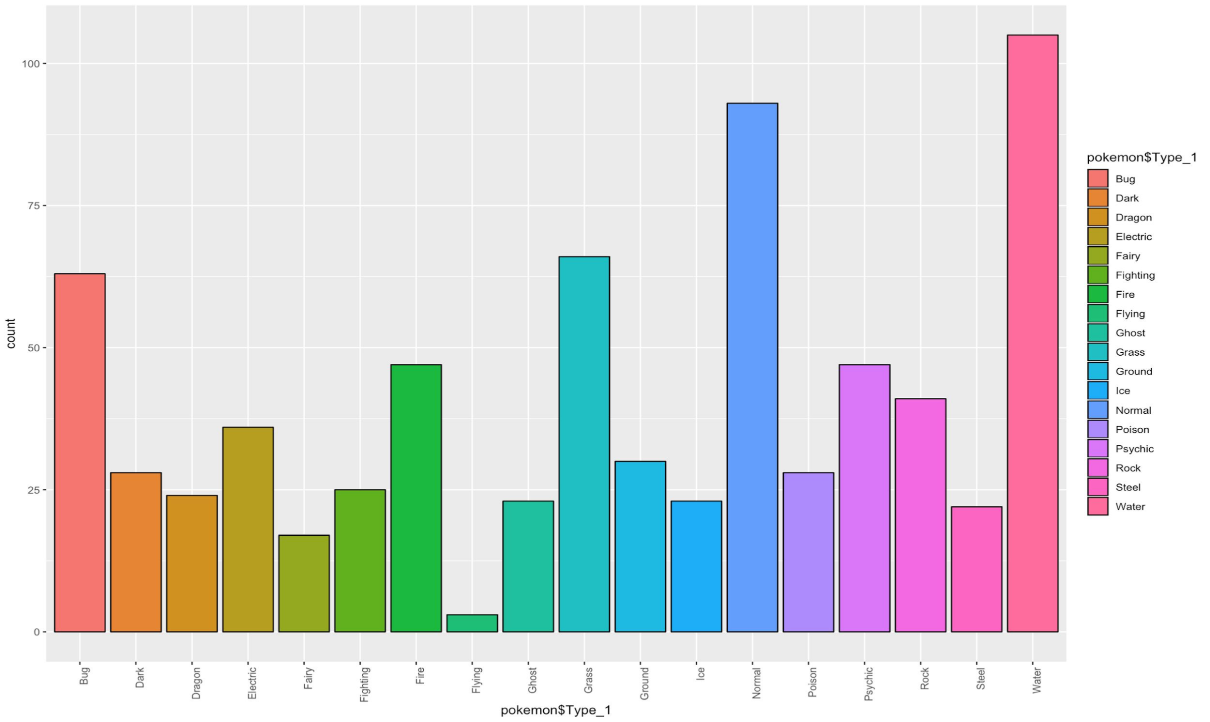
Can you catch them all?

Emily Medema, Kathryn Lecha, Barret Jackson, Lauren St. Clair

Overview Of Data

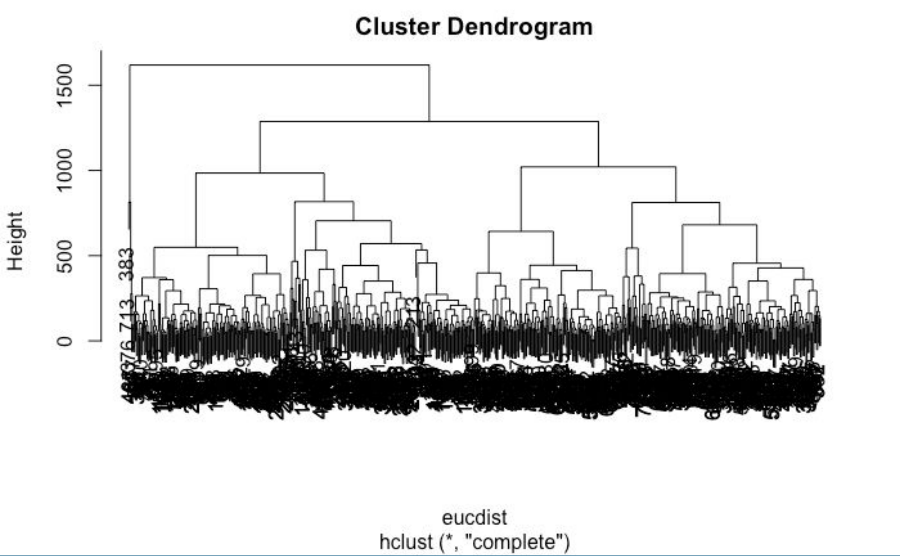
- Dataset of Pokemon, **721 observations** of **23 variables**
  - Accessed through Kaggle
  - Categorical predictors such as **isLegendary**, **Name**, **Type_1** & **Type_2** , and **hasMegaEvolution**
  - Numerical predictors such as **Total**, **HP**, **Attack**, and **Defense**
- Focus on predictability and relationship between **isLegendary** and predictors, with an emphasis on **hasGender** and **Pr_Male** predictors

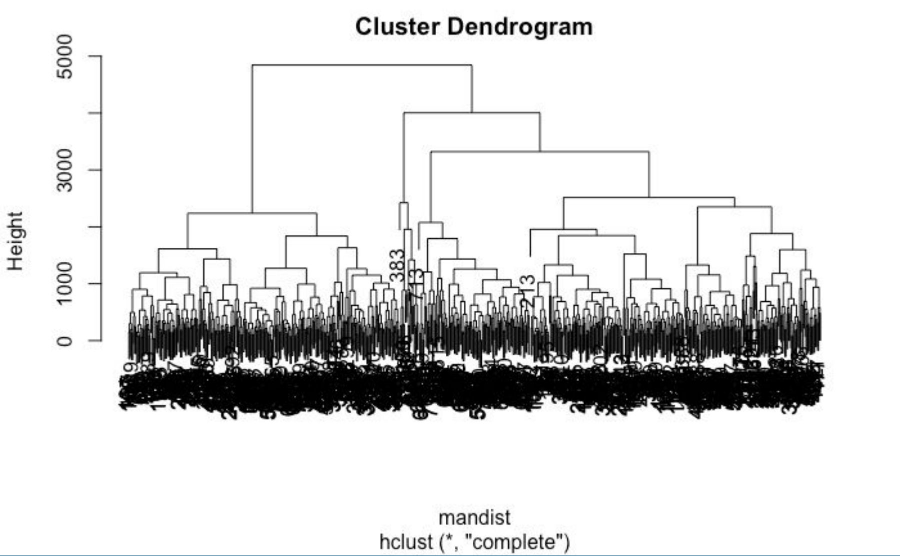
General Analyses and Plots



Number of Pokemon per type, using **Type_1** as the predictor

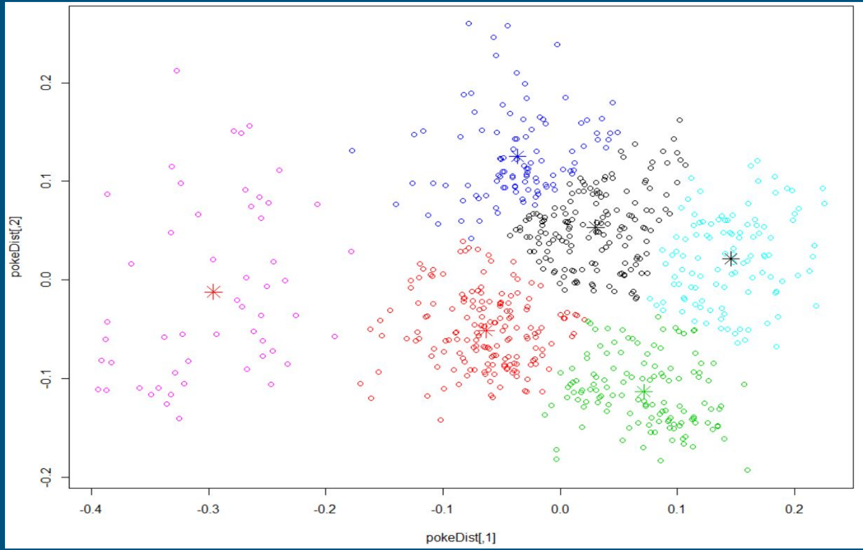
Clustering - **Complete Linkage**





k-Means Clustering

Visualization



Code

```{r}
pokeNum<-select_if(pokemon, is.numeric)
distPoke<-daisy(pokemon)
summary(distPoke)
pokeDist<-cmdscale(distPoke)
plot(pokeDist, type = "n")
text(pokeDist, rownames(pokeDist))
set.seed(413)
kPoke2<-kmeans(pokeDist, 6, nstart=25)
plot(pokeDist, col = kPoke2$cluster)
points(kPoke2$centers, col = 1:4, pch=8, cex=2)
out <- cbind(pokemon, clusterNum = kPoke2$cluster)
clusterGroups<-order(out$clusterNum, decreasing = TRUE)
out[clusterGroups,]
```

*We found that 6 clusters gave the best representation of the data and a reasonably low WSS

Who’s in these clusters?

	Total	HP	Attack	Defense	Sp_Atk	Sp_Def
Clus1	403.86	69.13	71.39	67.31	66.36	65.23
Clus2	482.66	78.38	90.04	82.46	78.62	79.73
Clus3	322.46	53.66	56.39	57.23	49.58	53.83
Clus4	497.04	78.11	90.05	81.89	83.90	84.95
Clus5	281.89	48.27	49.58	48.81	43.54	45.20
Clus6	615.94	93.13	106.98	101.51	113.98	105.55

	Speed	Height_m	Weight_kg	Catch_Rate	Pr_Male
Clus1	64.44	1.13	39.45	79.50	0.56
Clus2	73.43	1.29	73.54	54.88	0.58
Clus3	51.76	0.56	15.63	172.38	0.51
Clus4	78.14	1.65	81.33	50.73	0.61
Clus5	46.50	0.61	16.00	205.45	0.49
Clus6	94.79	2.26	191.63	7.47	0.75

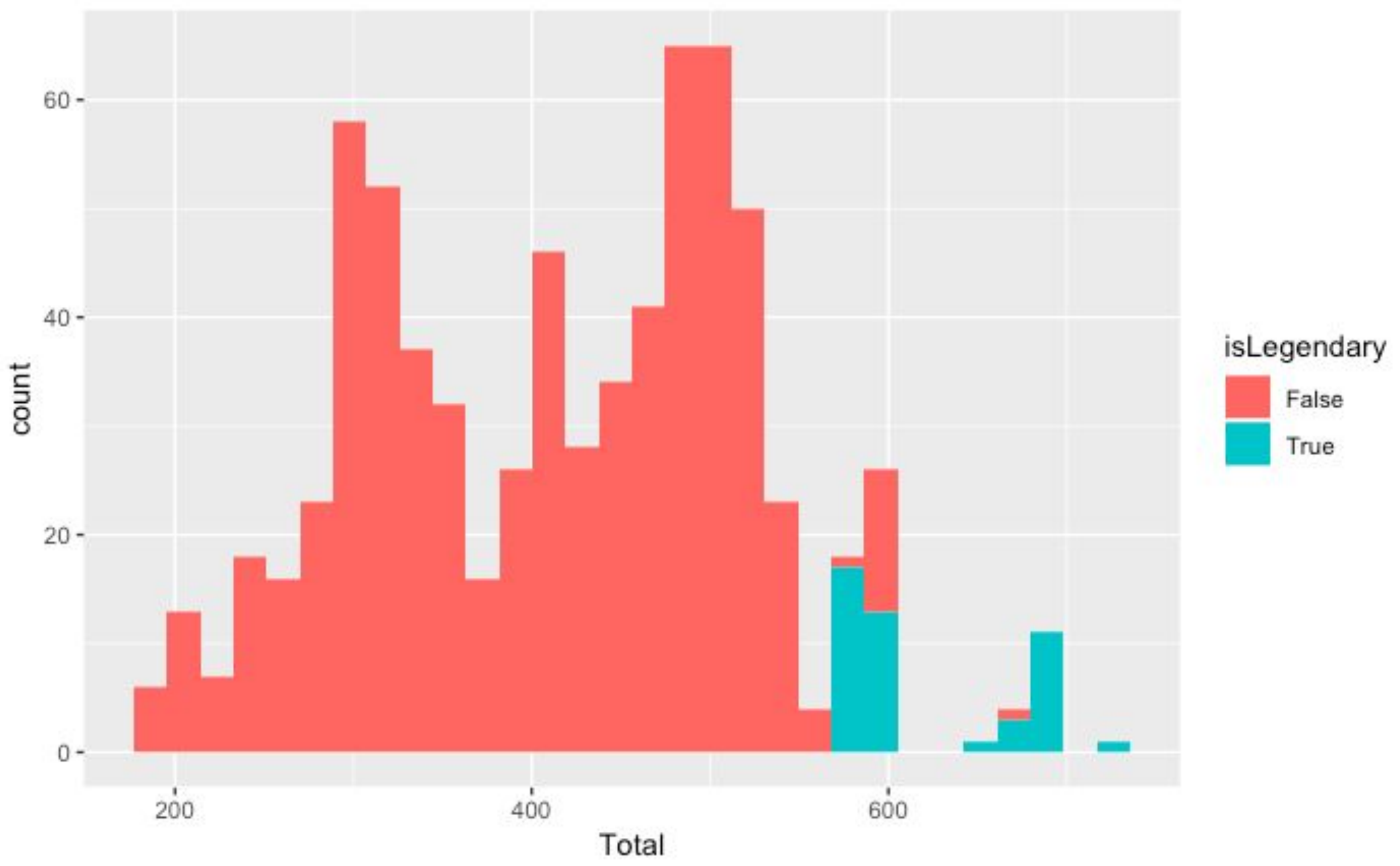
	Legend	!Legend
Clus1	0	157
Clus2	0	169
Clus3	0	127
Clus4	0	103
Clus5	0	112
Clus6	46	7

- Cluster 3 contains the short and lightweight Pokemon
- Cluster 5 contains the lowest stat/easiest to catch Pokemon
  - These Pokemon are also the least likely to be Male
- Cluster 6 contains every legendary Pokemon
  - As predicted, their stats are the highest and catch rate the lowest
  - They also happen to be the heaviest and tallest on average
  - The are also the most likely to be Male

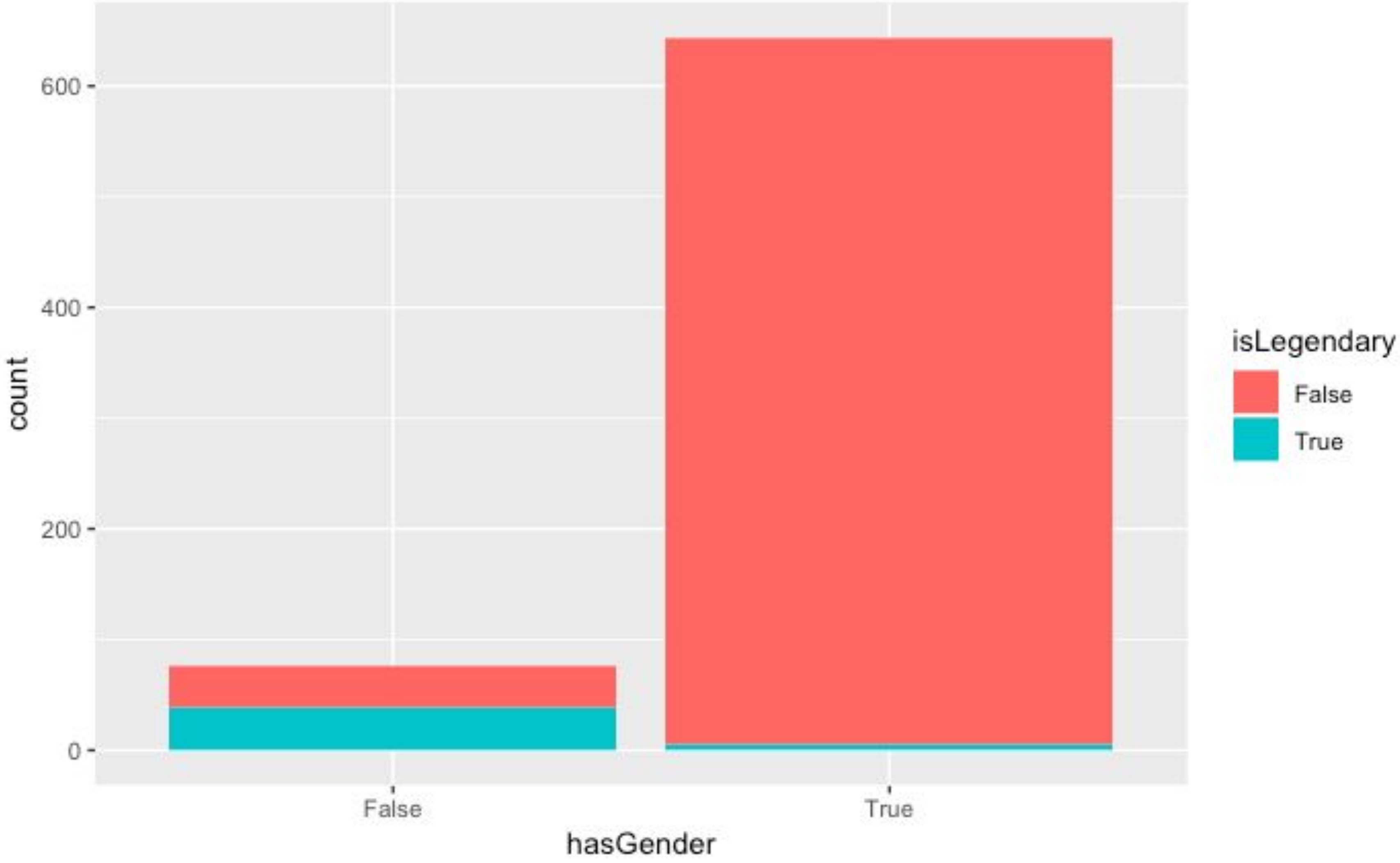
Is your Pokémon Legendary?

In evaluating the relationship between predictors and **isLegendary**, we used a variety of predictors in combination in order to approximate the importance of their interactions,

- **Total** (values of 550-625 for legendary type)
- **Pr_Male**, **hasGender**
- **Attack**, **Defense**, **HP**, **Sp_Atk**, **Sp_Def**, **Speed**



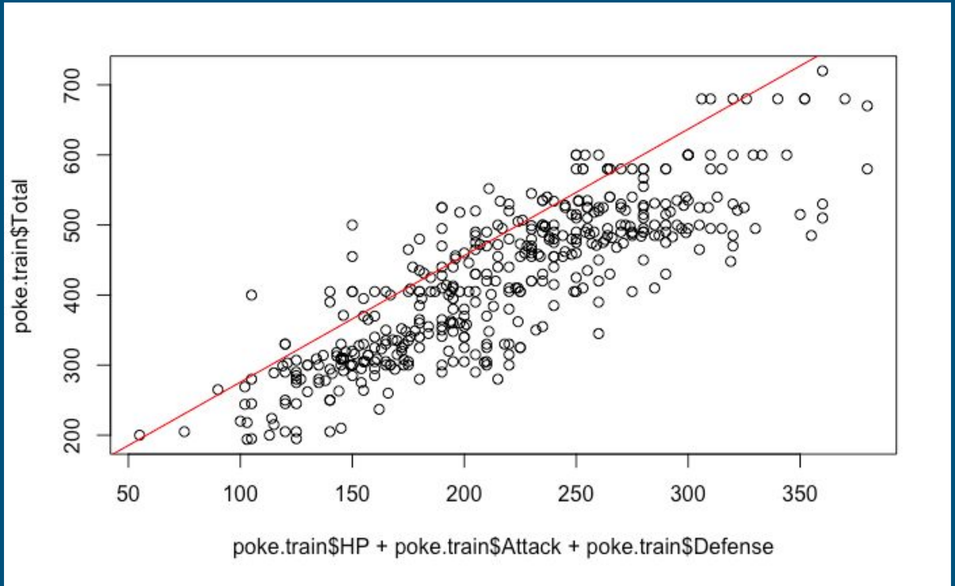
count
60
40
20
0
200
400
600
Total
isLegendary
False
True



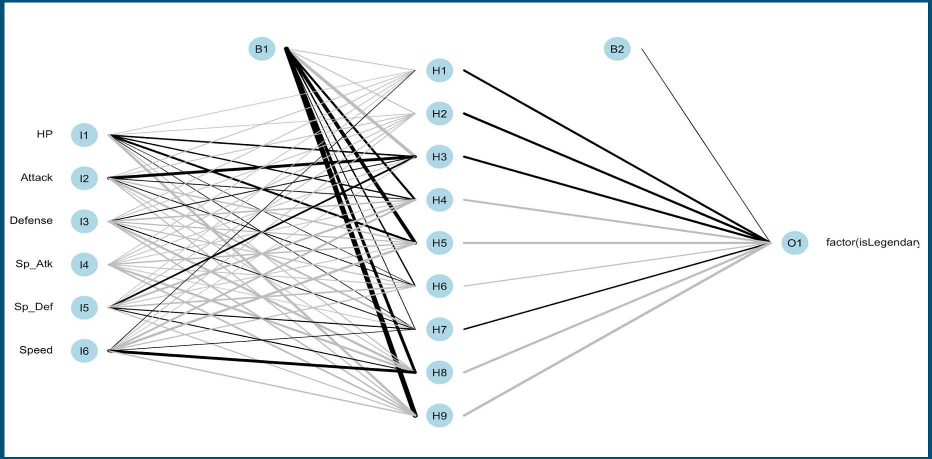
600
400
200
0
count
False
True
hasGender
isLegendary
False
True

Models Used to Predict Legendary Status

- **Linear Model**
  - Total~HP + Attack + Defense
- **LDA**
  - isLegendary~hasGender + Total
    - Misclassification Rate: 0.06
- **QDA**
  - isLegendary~hasGender + Total
    - Misclassification Rate: 0.038
- **LogReg**
  - isLegendary~hasGender + Total
    - Misclassification Rate: 0.016
- **KNN Classification**
  - Misclassification Rate: 0.009
- **Random Forest**
  - Misclassification Rate: 0.012
- **Neural Net**

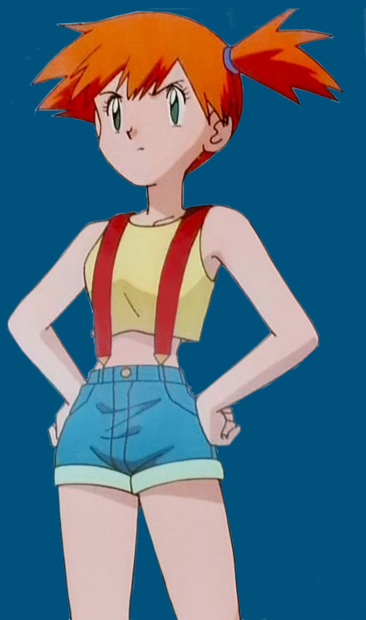
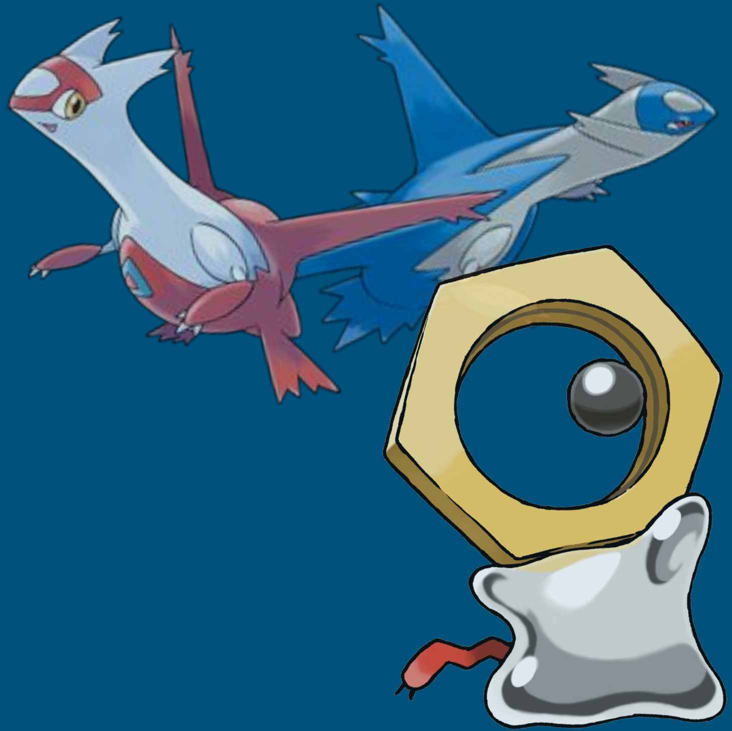
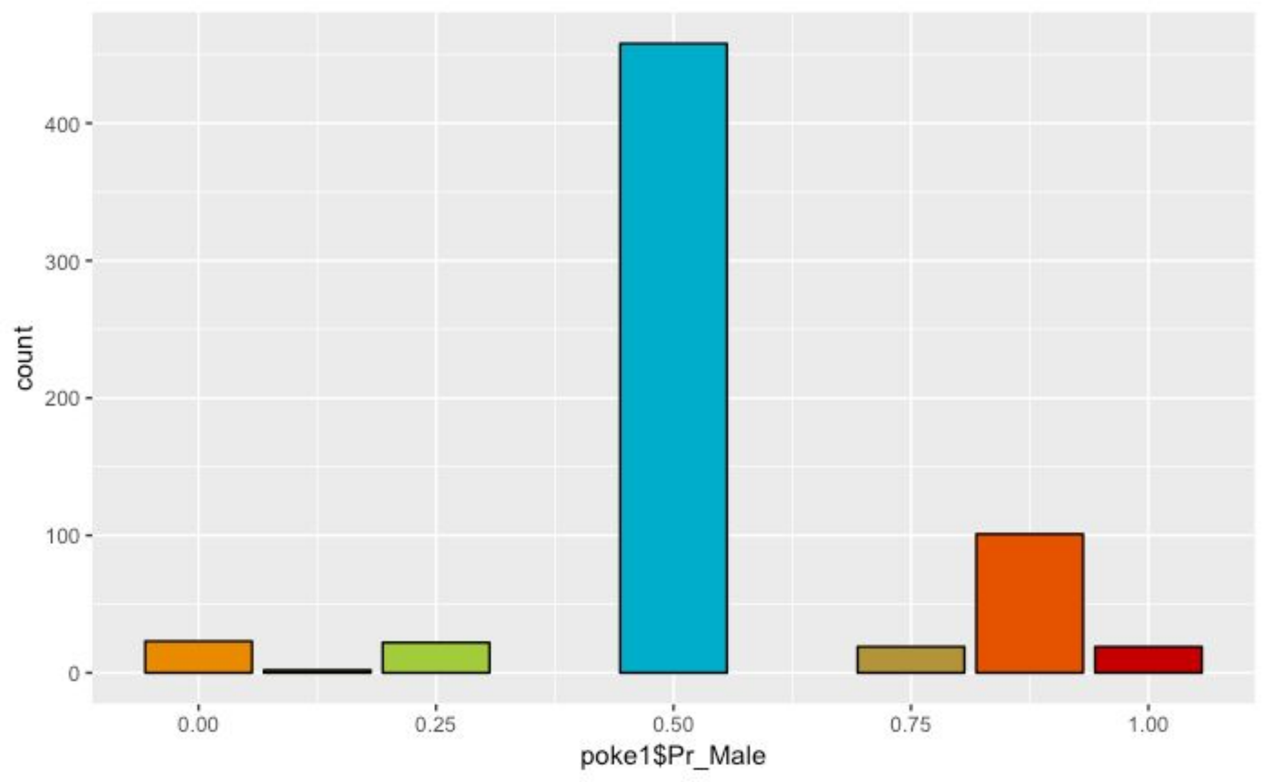
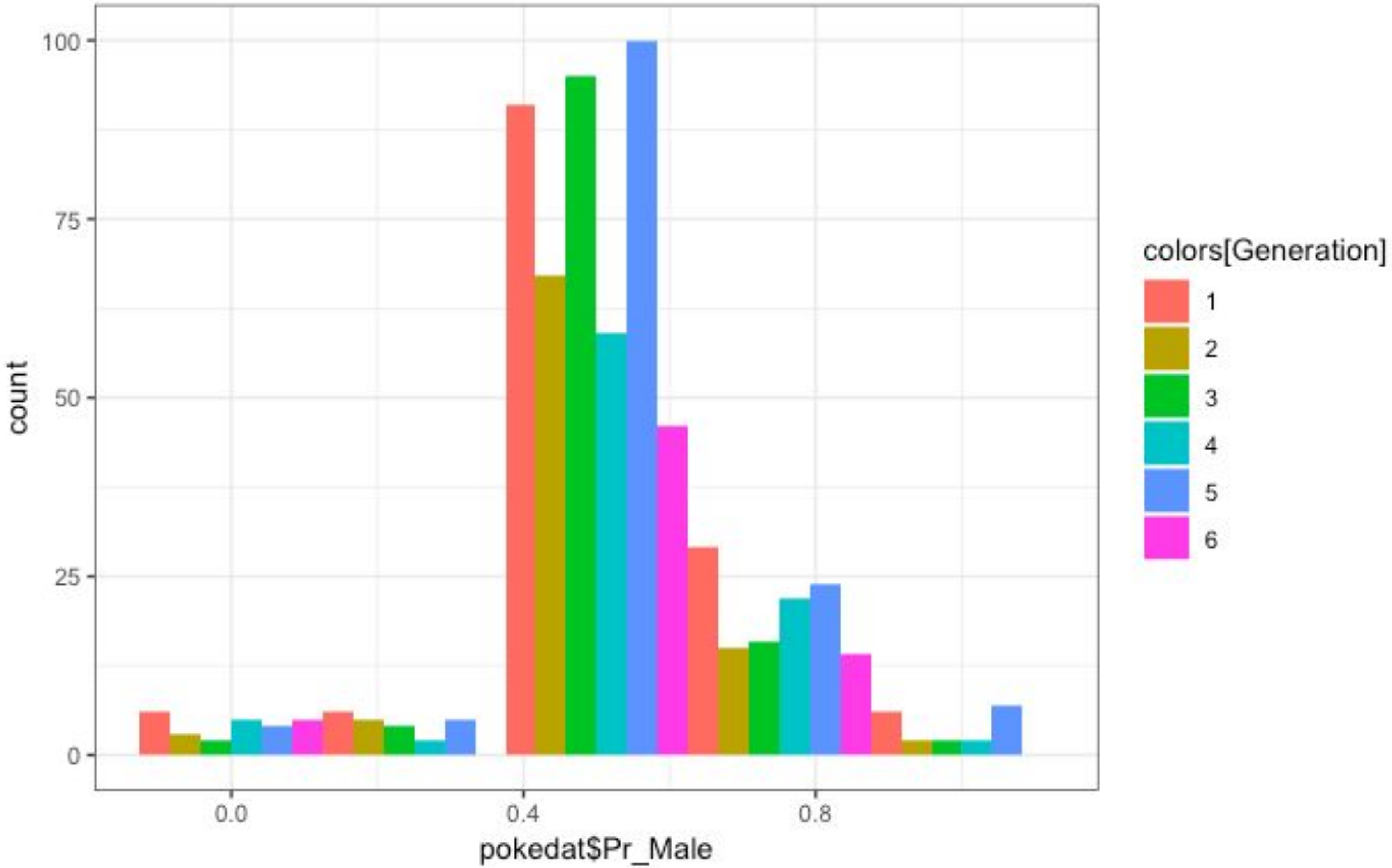
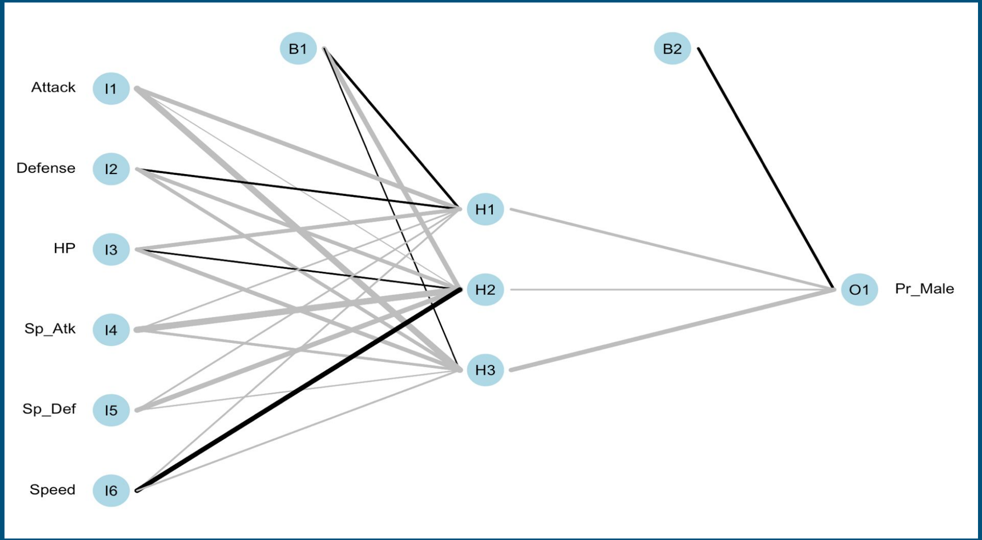


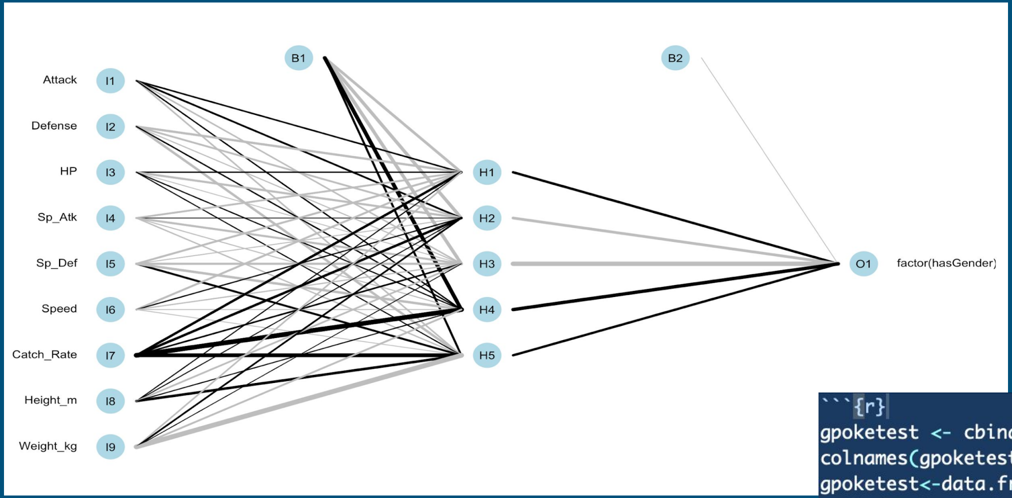
Neural nets, predicting for **isLegendary**




```
# weights:  73
initial  value 230.803477
iter  10 value 21.386442
iter  20 value 10.225302
iter  30 value 5.443762
iter  40 value 5.124080
iter  50 value 5.117497
iter  60 value 5.096526
iter  70 value 5.032397
iter  80 value 4.957708
iter  90 value 4.956312
iter 100 value 4.953087
final  value 4.953087
stopped after 100 iterations

          1   2
  False 479   1
  True    1  24
```

```
```{r}
spoketest <- cbind(scale(testset[,6:11]), factor(testset$isLegendary))
colnames(spoketest)[7] <- "isLegendary"
spoketest<-data.frame(spoketest)
table(spoketest$isLegendary, predict(nnpoke, newdata=spoketest, type="class"))
```
```

```
      1   2
  1 193   2
  2  15   6
```



Pokémon and Gender Analyses

Within the Pokemon dataset, the two predictors related to gender are **Pr_Male**, and **hasGender**. **Pr_Male** indicates the probability of male typing, which **hasGender** is a binary indicator.

In examining these predictors we used models such as,

- **Neural networks,**
- **and, Regression Trees**





This is a histogram of Pokemon that have a gender organized by their probability of being male

```
     countM fifty countF
[1,]    139   458     47
```



100
75
50
25
0
count
0.0
0.4
0.8
pokedat$Pr_Male
colors[Generation]
1
2
3
4
5
6

Neural nets, predicting for **Pr_Male**




```
#Neural Net predicting Pr_Male
```{r}
set.seed(906534)
library(nnet)
library(NeuralNetTools)
library(neuralnet)
nnmale <- neuralnet(Pr_Male ~ Attack + Defense + HP + Sp_Atk + Sp_Def + Speed,data=trainsetg, hidden=3,
threshold=0.01)
plotnet(nnmale)
mse<-mean((compute(nnmale, testsetg[,6:11])$net.result-testsetg$Pr_Male)^2)
mse
```
```

```
```{r}
linmod<-lm(Pr_Male ~ Attack + Defense + HP + Sp_Atk + Sp_Def + Speed,data=trainsetg)
mean((predict(linmod,newdata=testsetg)-testsetg$Pr_Male)^2)
```

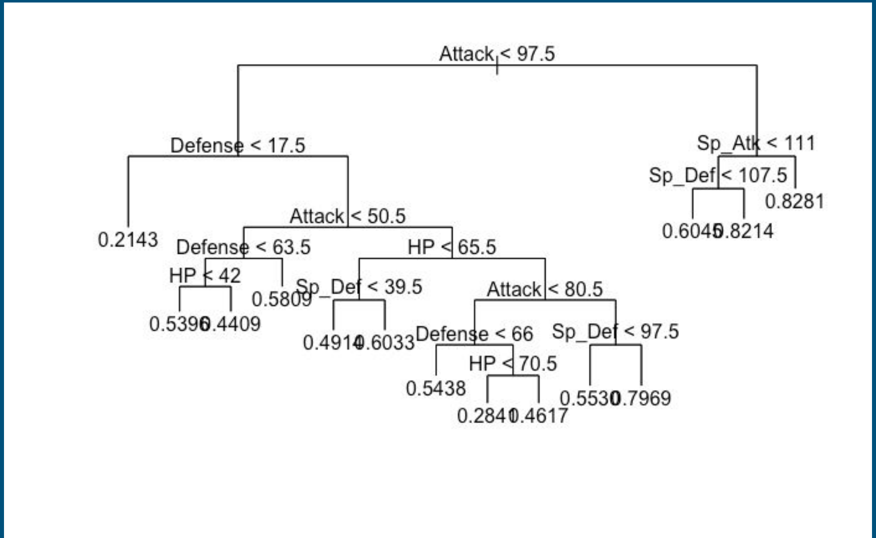
[1] 0.03764972
```

```
MSE = 0.03509201 (neural)
MSE = 0.03764972 (linmod)
```

We can see that our MSE for our neural is fairly close to our calculated MSE for a linear model for **Pr_Male** as a response with the same predictors.

Neural nets, predicting for **hasGender**



When we increased our model to 11 nodes it was overfit, subsequently resulting in a poor misclassification for **gspoktest** (below). Our neural net was instead generated using 1 hidden and 5 nodes.

```
```{r}
gpoketest <- cbind(scale(trainset[,c(6:11, 20:22)]), factor(testset$hasGender))
colnames(gpoketest)[10] <- "hasGender"
gpoketest<-data.frame(gpoketest)
table(gpoketest$hasGender, predict(nngend, newdata=gpoketest, type="class"))
```
```

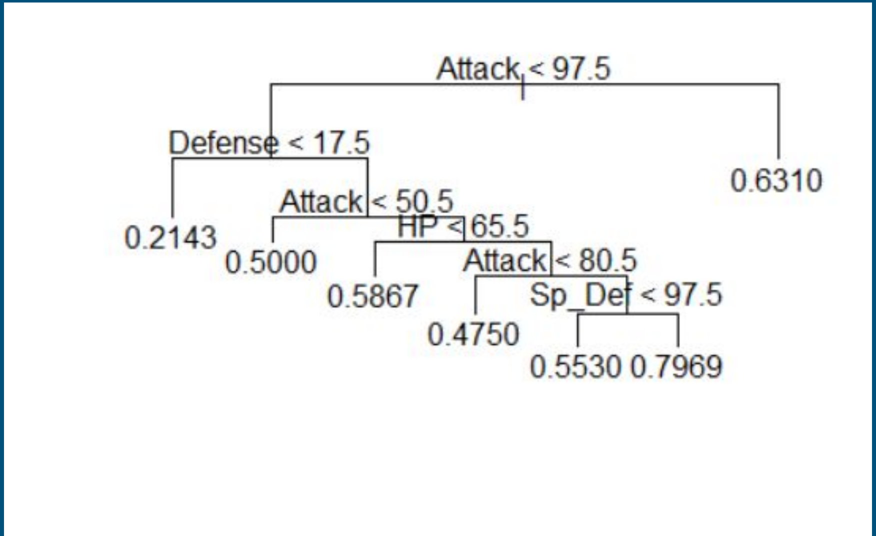
```
number of rows of result is not a multiple of vector length (arg 2)
     1   2
  1  14  57
  2  47 387
```

	1	2
False	36	9
True	25	435

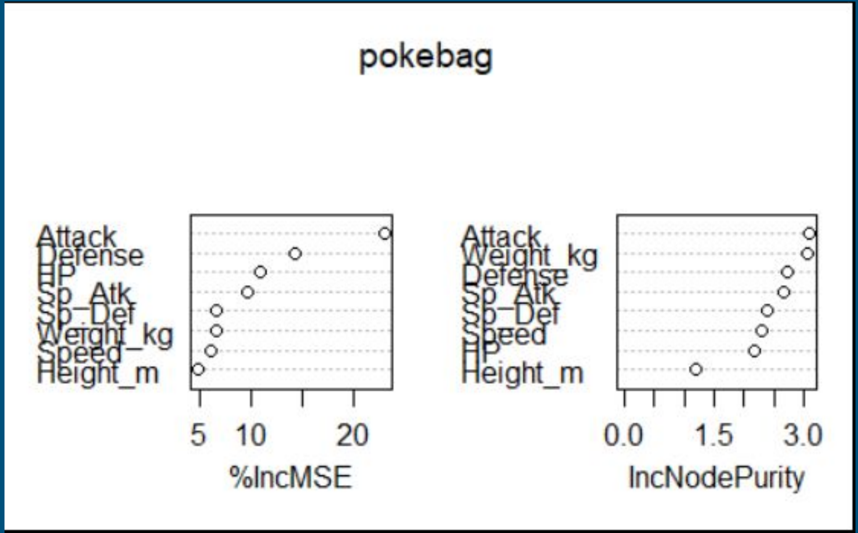
Our neural net appears to be an effective model for predicting **hasGender**.

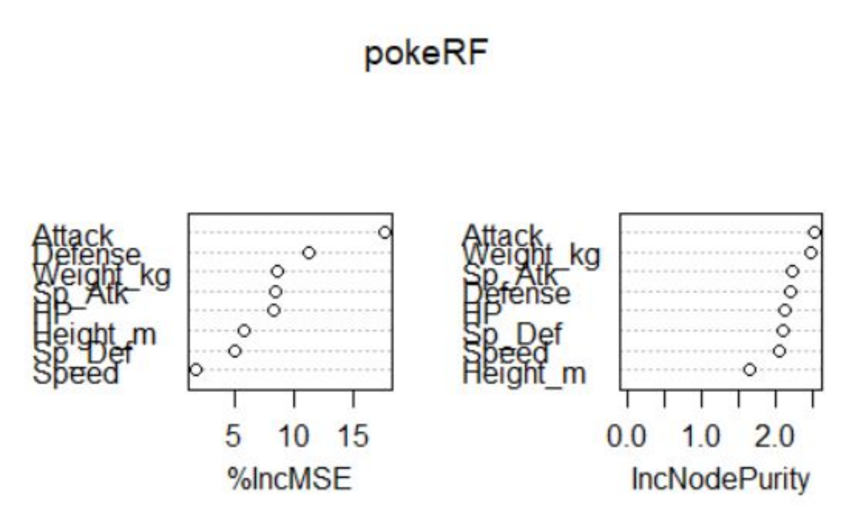
Regression Tree - **Pr_Male**





Variable importance plots - Bagging and Random Forests





Conclusion

Most Useful Models:

- Neural Networks
- Random Forest
- KNN
- K-Means clustering

Most Useful Variables:

- Pr_Male
- isLegendary
- Total
    - Attack
    - Defense
- hasGender
- Weight


ありがとっ♡