Pokemon - Gender things

Barret Jackson

March 22, 2019

Creating a subset with only pokemon that have a gender

```
pokemon<-read.csv("pokemon_alopez247.csv")
poke<-data.frame(pokemon)
attach(poke)
poke1<-poke[which(hasGender=='True'),]
head(poke1)
```

```
##   Number       Name Type_1 Type_2 Total HP Attack Defense Sp_Atk Sp_Def
## 1      1  Bulbasaur  Grass Poison   318 45     49      49     65     65
## 2      2    Ivysaur  Grass Poison   405 60     62      63     80     80
## 3      3   Venusaur  Grass Poison   525 80     82      83    100    100
## 4      4 Charmander   Fire          309 39     52      43     60     50
## 5      5 Charmeleon   Fire          405 58     64      58     80     65
## 6      6  Charizard   Fire Flying   534 78     84      78    109     85
##   Speed Generation isLegendary Color hasGender Pr_Male Egg_Group_1
## 1    45          1       False Green      True   0.875     Monster
## 2    60          1       False Green      True   0.875     Monster
## 3    80          1       False Green      True   0.875     Monster
## 4    65          1       False   Red      True   0.875     Monster
## 5    80          1       False   Red      True   0.875     Monster
## 6   100          1       False   Red      True   0.875     Monster
##   Egg_Group_2 hasMegaEvolution Height_m Weight_kg Catch_Rate
## 1       Grass            False     0.71       6.9         45
## 2       Grass            False     0.99      13.0         45
## 3       Grass             True     2.01     100.0         45
## 4      Dragon            False     0.61       8.5         45
## 5      Dragon            False     1.09      19.0         45
## 6      Dragon             True     1.70      90.5         45
##       Body_Style
## 1      quadruped
## 2      quadruped
## 3      quadruped
## 4 bipedal_tailed
## 5 bipedal_tailed
## 6 bipedal_tailed
```

```
length(poke1[,1])
```

```
## [1] 644
```

Trees on the new data set

```
attach(poke1)
```

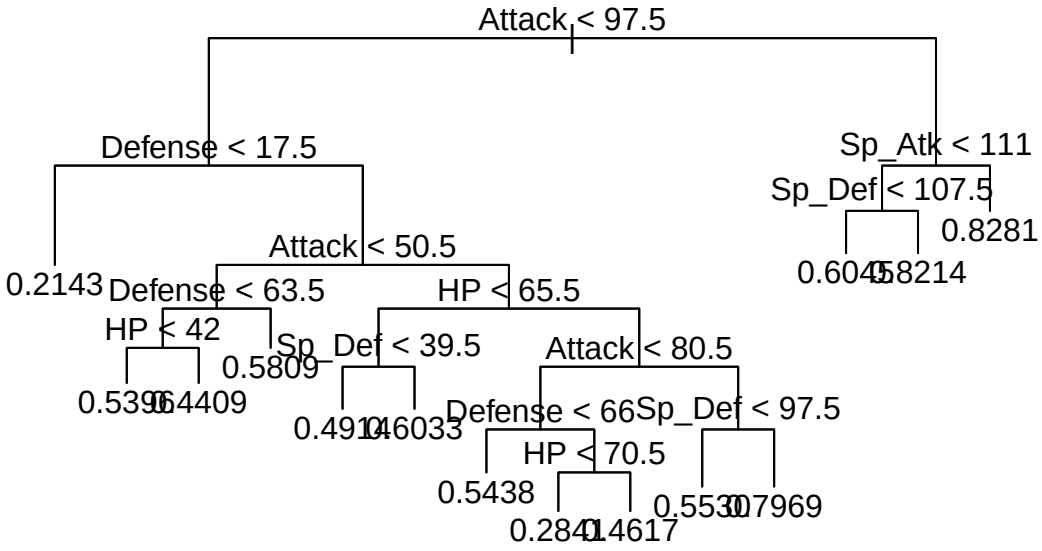
```
## The following objects are masked from poke:
##
##     Attack, Body_Style, Catch_Rate, Color, Defense, Egg_Group_1,
```

```
##     Egg_Group_2, Generation, hasGender, hasMegaEvolution,
##     Height_m, HP, isLegendary, Name, Number, Pr_Male, Sp_Atk,
##     Sp_Def, Speed, Total, Type_1, Type_2, Weight_kg
```

```
library(tree)
```

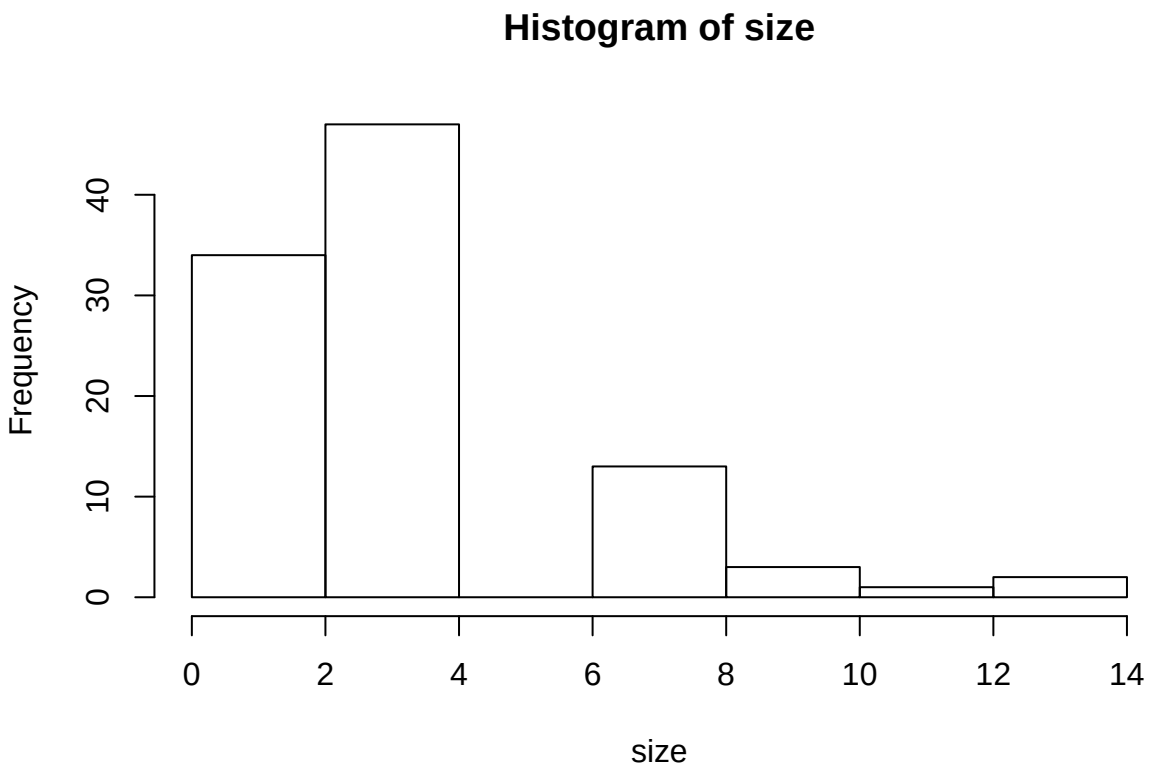
```
## Warning: package 'tree' was built under R version 3.5.2
```

```
pocl<-tree(Pr_Male~HP+Attack+Defense+Sp_Atk+Sp_Def+Speed,data=poke1)
plot(pocl)
text(pocl)
```



Ok, let's prune this tree down now...

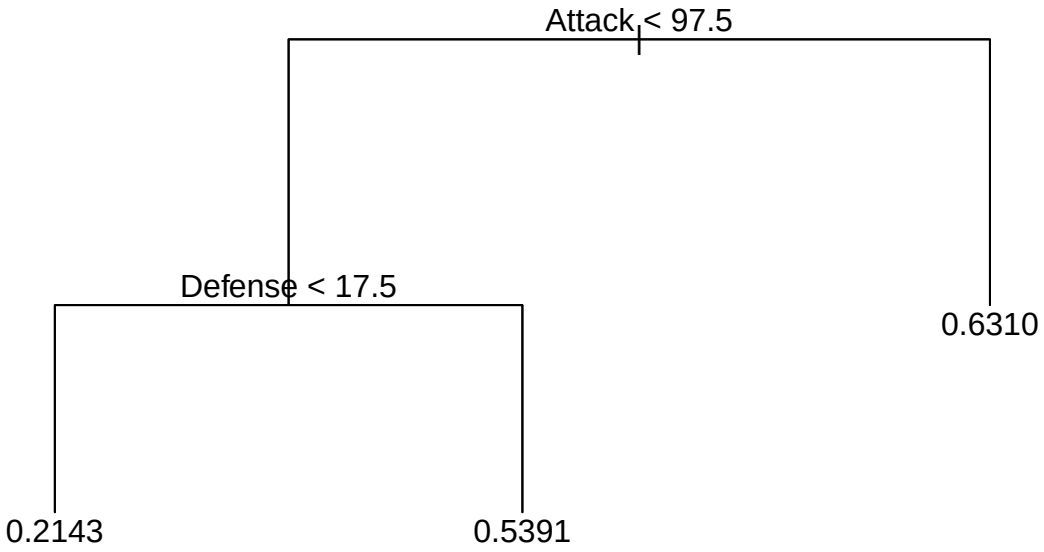
```
j<-sample(0,10000,100)
size<-{}
for(i in 1:100) {
  set.seed(i)
  cv.pocl<-cv.tree(pocl, FUN=prune.tree)
  thing<-cv.pocl$size[which.min(cv.pocl$dev)]
  size[i]<-thing
}
hist(size)
```




```
sort(table(size),decreasing=TRUE)[1:3]
```

```
## size
##  3  2  1
## 47 22 12
```

```
p.pocl<-prune.tree(pocl,best=3)
plot(p.pocl)
text(p.pocl)
```




```
summary(p.pocl)
```

```
##
## Regression tree:
## snip.tree(tree = pocl, nodes = c(3L, 5L))
## Variables actually used in tree construction:
## [1] "Attack"  "Defense"
## Number of terminal nodes:  3
## Residual mean deviance:  0.03752 = 24.05 / 641
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.53910 -0.03906 -0.03906  0.00000 -0.03906  0.53570
```

Alright, let's use bagging now...

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.2
```

```
## randomForest 4.6-14
```

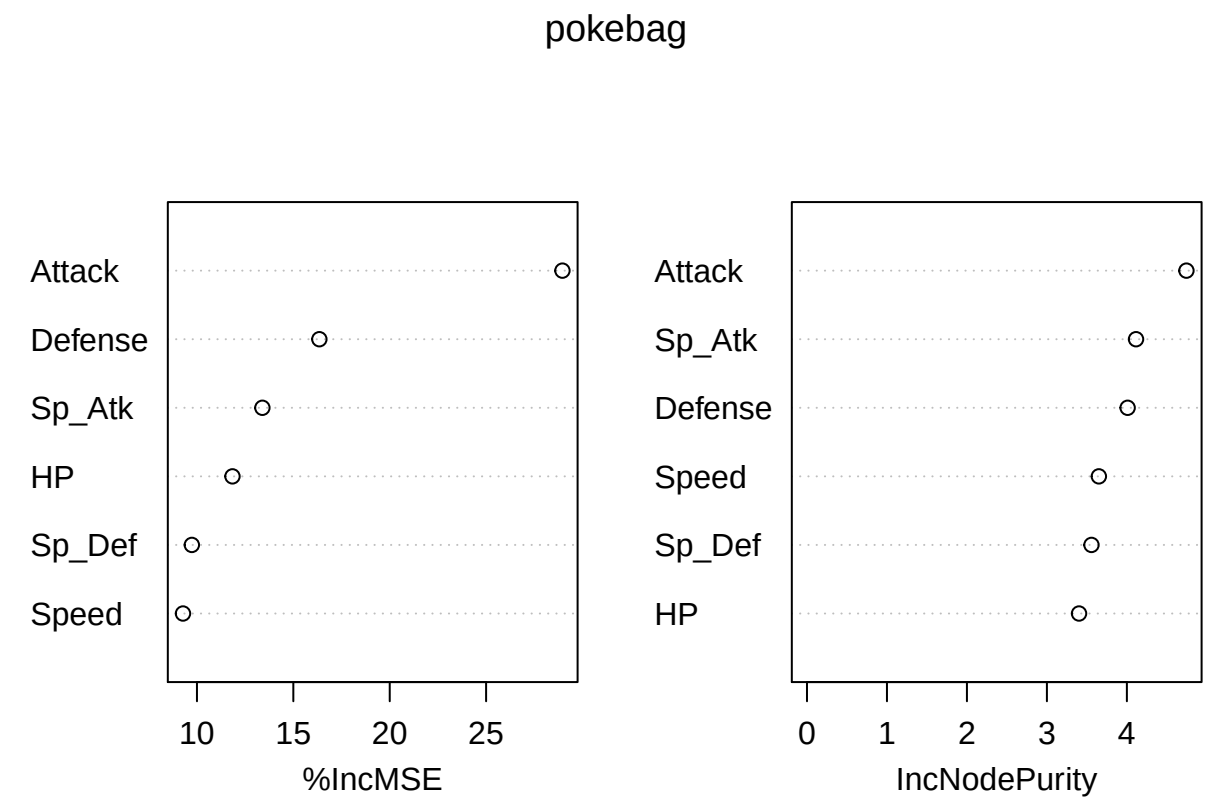
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(1995)
pokebag<-randomForest(Pr_Male~HP+Attack+Defense+Sp_Atk+Sp_Def+Speed,data=poke1,mtry=6,importance=TRUE)
pokebag
```

```
##
## Call:
```

```
##  randomForest(formula = Pr_Male ~ HP + Attack + Defense + Sp_Atk +      Sp_Def + Speed, data = poke1
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 0.03642897
##                     % Var explained: 8.76
```

```
varImpPlot(pokebag)
```



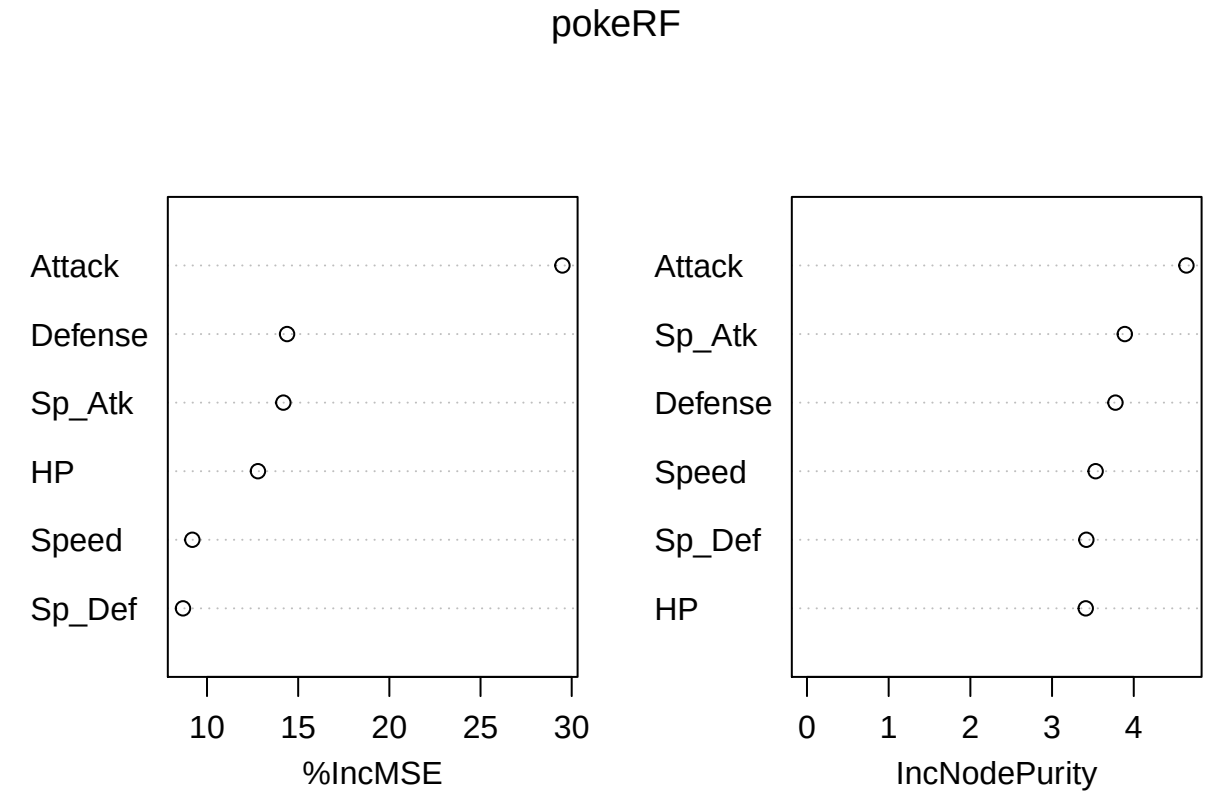
Well that didn't work out very well...

How about random forest...

```
pokeRF<-randomForest(Pr_Male~HP+Attack+Defense+Sp_Atk+Sp_Def+Speed,data=poke1,mtry=3,importance=TRUE)
pokeRF
```

```
##
## Call:
##  randomForest(formula = Pr_Male ~ HP + Attack + Defense + Sp_Atk +      Sp_Def + Speed, data = poke1
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 0.03577835
##                     % Var explained: 10.39
```

```
varImpPlot(pokeRF)
```



Very slightly better, still not a lot of evidence that this model is any good.

PCA

Alright, let's check out PCA on Pr_Male response with . . . stats as predictors

```
head(poke)
```

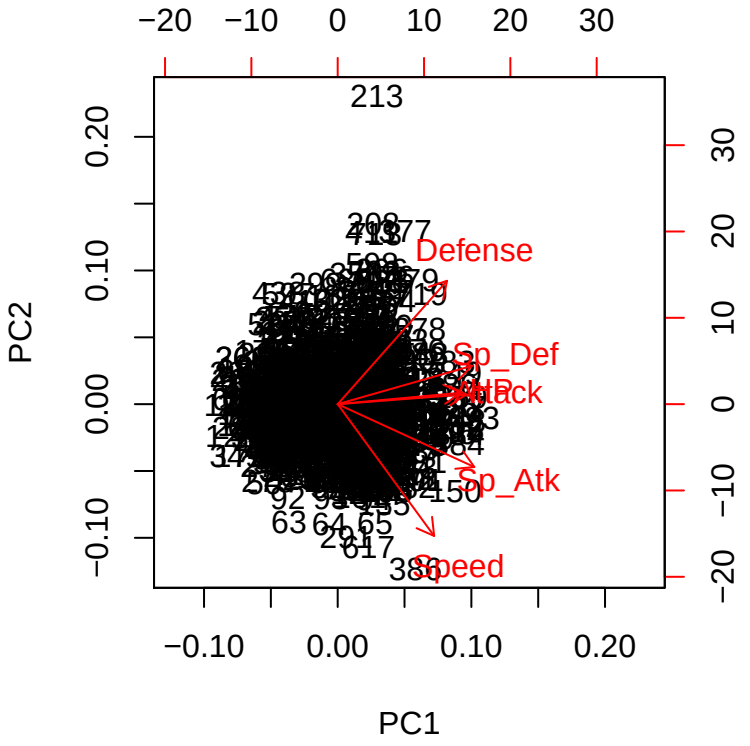
```
##   Number       Name Type_1 Type_2 Total HP Attack Defense Sp_Atk Sp_Def
## 1      1  Bulbasaur  Grass Poison   318 45     49      49     65     65
## 2      2    Ivysaur  Grass Poison   405 60     62      63     80     80
## 3      3   Venusaur  Grass Poison   525 80     82      83    100    100
## 4      4 Charmander   Fire          309 39     52      43     60     50
## 5      5 Charmeleon   Fire          405 58     64      58     80     65
## 6      6  Charizard   Fire Flying   534 78     84      78    109     85
##   Speed Generation isLegendary Color hasGender Pr_Male Egg_Group_1
## 1    45          1       False Green      True   0.875     Monster
## 2    60          1       False Green      True   0.875     Monster
## 3    80          1       False Green      True   0.875     Monster
## 4    65          1       False   Red      True   0.875     Monster
## 5    80          1       False   Red      True   0.875     Monster
## 6   100          1       False   Red      True   0.875     Monster
##   Egg_Group_2 hasMegaEvolution Height_m Weight_kg Catch_Rate
## 1       Grass            False     0.71       6.9         45
## 2       Grass            False     0.99      13.0         45
```

```
## 3       Grass             True     2.01      100.0         45
## 4      Dragon            False     0.61        8.5         45
## 5      Dragon            False     1.09       19.0         45
## 6      Dragon             True     1.70       90.5         45
##       Body_Style
## 1      quadruped
## 2      quadruped
## 3      quadruped
## 4 bipedal_tailed
## 5 bipedal_tailed
## 6 bipedal_tailed
```

```
pcapoke <- prcomp(as.matrix(poke[,6:11]), scale.=TRUE)
summary(pcapoke)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6
## Standard deviation     1.6145 1.0545 0.9201 0.8507 0.66506 0.51875
## Proportion of Variance 0.4344 0.1853 0.1411 0.1206 0.07372 0.04485
## Cumulative Proportion  0.4344 0.6197 0.7608 0.8814 0.95515 1.00000
```

```
biplot(pcapoke)
```



Ok cool, two principal components satisfy the Kaiser criterion. Let's take a look at which predictors influence these components...

```
round(pcapoke$rotation[,1:2], 2)
```

```
##           PC1   PC2
## HP      0.41  0.06
## Attack  0.43  0.05
## Defense 0.37  0.63
## Sp_Atk  0.46 -0.32
## Sp_Def  0.45  0.20
## Speed   0.32 -0.67
```

Ok, so looks like PC1 refers to kind of all around, balanced pokemon, and PC2 refers to slow defenders with bad HP? I don't think this model is all that great... But, let's see which pokemon each component is referring to.

```
poke[order(pcapoke$x[,1], decreasing=TRUE)[1:4] , 1:11]
```

```
##     Number     Name Type_1 Type_2 Total  HP Attack Defense Sp_Atk Sp_Def
## 493    493   Arceus Normal          720 120    120     120    120    120
## 250    250    Ho-Oh   Fire Flying   680 106    130      90    110    154
## 643    643 Reshiram Dragon   Fire   680 100    120     100    150    120
## 487    487 Giratina  Ghost Dragon   680 150    120     100    120    100
##     Speed
## 493   120
## 250    90
## 643    90
## 487    90
```

```
poke[order(pcapoke$x[,2], decreasing=TRUE)[1:4] , 1:11]
```

```
##     Number      Name Type_1 Type_2 Total HP Attack Defense Sp_Atk Sp_Def
## 213    213   Shuckle    Bug   Rock   505 20     10     230     10    230
## 208    208   Steelix  Steel Ground   510 75     85     200     55     65
## 377    377  Regirock   Rock          580 80    100     200     50    100
## 411    411 Bastiodon   Rock  Steel   495 60     52     168     47    138
##     Speed
## 213     5
## 208    30
## 377    50
## 411    30
```

The first component doesn't really seem to refer to much at all, just kind of all around generalists maybe. The totals are quite high though, so maybe these are the powerhouses? Wait, let's see how many of them are legendary...

```
poke[order(pcapoke$x[,1], decreasing=TRUE)[1:20],]
```

```
##     Number      Name  Type_1   Type_2 Total  HP Attack Defense Sp_Atk
## 493    493    Arceus  Normal            720 120    120     120    120
## 250    250     Ho-Oh    Fire   Flying   680 106    130      90    110
## 643    643  Reshiram  Dragon     Fire   680 100    120     100    150
## 487    487  Giratina   Ghost   Dragon   680 150    120     100    120
## 384    384  Rayquaza  Dragon   Flying   680 105    150      90    150
## 484    484    Palkia   Water   Dragon   680  90    120     100    150
## 716    716   Xerneas   Fairy            680 126    131      95    131
## 717    717   Yveltal    Dark   Flying   680 126    131      95    131
## 483    483    Dialga   Steel   Dragon   680 100    120     120    150
## 382    382    Kyogre   Water            670 100    100      90    150
## 644    644    Zekrom  Dragon Electric   680 100    150     120    120
## 249    249     Lugia Psychic   Flying   680 106     90     130     90
```

```
## 150    150    Mewtwo Psychic            680 106    110      90    154
## 289    289   Slaking  Normal            670 150    160     100     95
## 486    486 Regigigas  Normal            670 110    160     110     80
## 646    646    Kyurem  Dragon      Ice   660 125    130      90    130
## 383    383   Groudon  Ground            670 100    150     140    100
## 720    720     Hoopa Psychic    Ghost   600  80    110      60    150
## 706    706    Goodra  Dragon            600  90    100      70    110
## 648    648  Meloetta  Normal  Psychic   600 100     77      77    128
##     Sp_Def Speed Generation isLegendary  Color hasGender Pr_Male
## 493    120   120          4        True   Grey     False      NA
## 250    154    90          2        True    Red     False      NA
## 643    120    90          5        True  White     False      NA
## 487    100    90          4        True  Black     False      NA
## 384     90    95          3        True  Green     False      NA
## 484    120   100          4        True Purple     False      NA
## 716     98    99          6        True   Blue     False      NA
## 717     98    99          6        True    Red     False      NA
## 483    100    90          4        True  White     False      NA
## 382    140    90          3        True   Blue     False      NA
## 644    100    90          5        True  Black     False      NA
## 249    154   110          2        True  White     False      NA
## 150     90   130          1        True Purple     False      NA
## 289     65   100          3       False  Brown      True     0.5
## 486    110   100          4        True  White     False      NA
## 646     90    95          5        True   Grey     False      NA
## 383     90    90          3        True    Red     False      NA
## 720    130    70          6        True Purple     False      NA
## 706    150    80          6       False Purple      True     0.5
## 648    128    90          5       False  White     False      NA
##      Egg_Group_1 Egg_Group_2 hasMegaEvolution Height_m Weight_kg
## 493 Undiscovered                       False     3.20     320.0
## 250 Undiscovered                       False     3.81     199.0
## 643 Undiscovered                       False     3.20     330.0
## 487 Undiscovered                       False     6.91     650.0
## 384 Undiscovered                        True     7.01     206.5
## 484 Undiscovered                       False     4.19     336.0
## 716 Undiscovered                       False     3.00     215.0
## 717 Undiscovered                       False     5.79     203.0
## 483 Undiscovered                       False     5.41     683.0
## 382 Undiscovered                       False     4.50     352.0
## 644 Undiscovered                       False     2.90     345.0
## 249 Undiscovered                       False     5.21     216.0
## 150 Undiscovered                        True     2.01     122.0
## 289        Field                       False     2.01     130.5
## 486 Undiscovered                       False     3.71     420.0
## 646 Undiscovered                       False     3.00     325.0
## 383 Undiscovered                       False     3.51     950.0
## 720 Undiscovered                       False     0.51       9.0
## 706       Dragon                       False     2.01     150.5
## 648 Undiscovered                       False     0.61       6.5
##     Catch_Rate       Body_Style
## 493          3        quadruped
## 250          3        two_wings
## 643          3        two_wings
```

```
## 487          3  serpentine_body
## 384         45  serpentine_body
## 484          3   bipedal_tailed
## 716         45        quadruped
## 717         45        two_wings
## 483          3        quadruped
## 382          3        with_fins
## 644          3   bipedal_tailed
## 249          3        two_wings
## 150          3   bipedal_tailed
## 289         45 bipedal_tailless
## 486          3 bipedal_tailless
## 646          3   bipedal_tailed
## 383          3   bipedal_tailed
## 720          3        head_only
## 706         45   bipedal_tailed
## 648          3 bipedal_tailless
```

The first 13 are legendary, this is a good sign. Let's see how PC1 correlates with isLegendary...

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.5.2
```

```
pcleg<-data.frame(pcapoke$x)
pcleg[1:20,]
```

```
##           PC1         PC2           PC3         PC4         PC5
## 1  -1.4042180 -0.04510528  7.448463e-01  0.06779003  0.40775445
## 2  -0.1345424 -0.11657081  7.624030e-01  0.07154712  0.26850825
## 3   1.6144749 -0.17853200  7.228695e-01  0.03719310  0.09807205
## 4  -1.6221363 -0.72979789  3.098805e-01 -0.35190720  0.18898083
## 5  -0.2118996 -0.82824095  3.663792e-01 -0.22095009  0.13423231
## 6   1.6798096 -0.99070641  4.258421e-01 -0.23359664  0.19747907
## 7  -1.5134958  0.50515323  5.962763e-01 -0.19919501  0.05615033
## 8  -0.1851314  0.46593209  5.888747e-01 -0.22390720 -0.08407012
## 9   1.6468927  0.44021964  6.605840e-01 -0.25299456 -0.31647281
## 10 -3.3206731 -0.20318348 -3.033128e-01  0.25812975 -0.30762801
## 11 -3.1547513  0.57260297  7.752779e-02  0.41566588 -0.09910085
## 12 -0.2717451 -0.78469694  1.256007e+00  0.27553798  0.25698221
## 13 -3.3294498 -0.43707536 -3.809238e-01  0.07267321 -0.31375433
## 14 -3.1635280  0.33871108 -8.314664e-05  0.23020934 -0.10522717
## 15 -0.3048446 -0.53194561 -3.627949e-01  0.06250607 -0.76384672
## 16 -2.4989759 -0.41143630 -1.280428e-01 -0.17821523 -0.16408177
## 17 -1.0599524 -0.43980418 -2.373169e-01  0.04065205 -0.36859899
## 18  0.8071630 -0.74885384 -2.940428e-01 -0.13058498 -0.71696194
## 19 -2.5270694 -0.80639252 -4.267220e-01 -0.73595103 -0.54084354
## 20 -0.1747841 -0.81134653 -2.536535e-01 -0.76836024 -0.87782178
##             PC6
## 1  -0.311506202
## 2  -0.204035100
## 3  -0.083830363
## 4  -0.081154659
## 5   0.131897305
## 6   0.301054413
## 7  -0.093586833
```

```
## 8  -0.023239012
## 9  -0.002062932
## 10  0.417604812
## 11  0.800467276
## 12  0.008943435
## 13  0.231420618
## 14  0.614283082
## 15 -1.117299138
## 16  0.076309914
## 17  0.251314934
## 18  0.464809278
## 19 -0.221233235
## 20 -0.269034628
```

```
leglda <- lda(factor(poke$isLegendary)~PC1+PC2,data=pcleg)
leglda
```

```
## Call:
## lda(factor(poke$isLegendary) ~ PC1 + PC2, data = pcleg)
##
## Prior probabilities of groups:
##      False       True
## 0.93619972 0.06380028
##
## Group means:
##              PC1         PC2
## False -0.2028048  0.01317792
## True   2.9759393 -0.19337164
##
## Coefficients of linear discriminants:
##            LD1
## PC1  0.7038278
## PC2 -0.1072031
```

I might just be high, but I'm pretty sure this indicates PC1 is a pretty good predictor for isLegendary. PC2 doesn't really seem to refer to anything here...