POKEMON

Youhee Kil

12/1/2017

Description of the data

- isLegendary: Boolean indicating whether the Pokémon is legendary or not. Legendary Pokémon tend to be stronger, to have unique abilities, to be really hard to find, and to be even harder to catch.
- hasMegaEvolution: Boolean indicating whether a Pokémon can mega-evolve or not. Mega-evolving is property that some Pokémon have and allows them to change their appearance, types, and stats during a combat into a much stronger form.
- Catch_Rate: Numerical variable indicating how easy is to catch a Pokémon when trying to capture it to make it part of your team. It is bounded between 3 and 255. The number of different values it takes is not too high notwithstanding, we can consider it is a continuous variable.

Questions

Can we successfully predict the catch rate (how easy to catch a Pokemon when trying to capture it to make it part of your team) with Decision Tree ?

Load data

```
library(readr)
pokemon <- read_csv("~/Downloads/pokemon/pokemon_alopez247.csv")
```

Libraries

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggvis)
```

```
##
## Attaching package: 'ggvis'
```

```
## The following object is masked from 'package:ggplot2':
##
##     resolution
```

```
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

Detail

```
head(pokemon)
```

```
## # A tibble: 6 × 23
##   Number       Name Type_1 Type_2 Total    HP Attack Defense Sp_Atk Sp_Def
##    <int>      <chr>  <chr>  <chr> <int> <int>  <int>   <int>  <int>  <int>
## 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   <NA>   309    39     52      43     60     50
## 5      5 Charmeleon   Fire   <NA>   405    58     64      58     80     65
## 6      6  Charizard   Fire Flying   534    78     84      78    109     85
## # ... with 13 more variables: Speed <int>, Generation <int>,
## #   isLegendary <chr>, Color <chr>, hasGender <chr>, Pr_Male <dbl>,
## #   Egg_Group_1 <chr>, Egg_Group_2 <chr>, hasMegaEvolution <chr>,
## #   Height_m <dbl>, Weight_kg <dbl>, Catch_Rate <int>, Body_Style <chr>
```

```
pokemon_m <- pokemon[, c(2, 13, 3:12, 14:23)]
head(pokemon_m)
```

```
## # A tibble: 6 × 22
##         Name isLegendary Type_1 Type_2 Total    HP Attack Defense Sp_Atk
##        <chr>       <chr>  <chr>  <chr> <int> <int>  <int>   <int>  <int>
## 1  Bulbasaur       False  Grass Poison   318    45     49      49     65
## 2    Ivysaur       False  Grass Poison   405    60     62      63     80
## 3   Venusaur       False  Grass Poison   525    80     82      83    100
## 4 Charmander       False   Fire   <NA>   309    39     52      43     60
## 5 Charmeleon       False   Fire   <NA>   405    58     64      58     80
## 6  Charizard       False   Fire Flying   534    78     84      78    109
## # ... with 13 more variables: Sp_Def <int>, Speed <int>, Generation <int>,
## #   Color <chr>, hasGender <chr>, Pr_Male <dbl>, Egg_Group_1 <chr>,
## #   Egg_Group_2 <chr>, hasMegaEvolution <chr>, Height_m <dbl>,
## #   Weight_kg <dbl>, Catch_Rate <int>, Body_Style <chr>
```

```
write.csv(pokemon_m, file = "pokemon_m.csv")
```

Check NA

As we see the graph from the below, Egg group2, Type 2 and Pr_Male has missing values so much. We will discover more about it whether those columns are valuable to include.

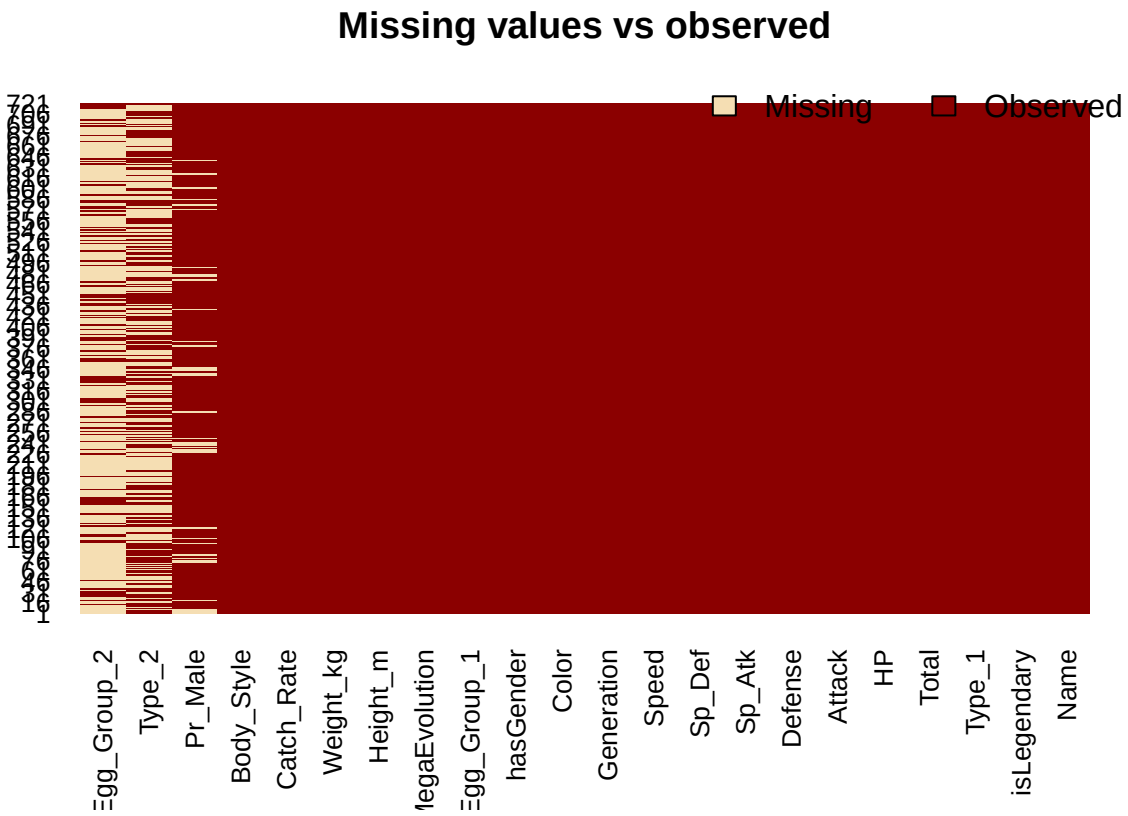
```
missmap(pokemon_m, main = "Missing values vs observed")
```

```
## Warning in if (class(obj) == "amelia") {: the condition has length > 1 and
## only the first element will be used
```

```
## Warning: Unknown column 'arguments'
```

```
## Warning: Unknown column 'arguments'
```

```
## Warning: Unknown column 'imputations'
```



Data

```
str(pokemon_m)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    721 obs. of  22 variables:
##  $ Name            : chr  "Bulbasaur" "Ivysaur" "Venusaur" "Charmander" ...
##  $ isLegendary     : chr  "False" "False" "False" "False" ...
##  $ Type_1          : chr  "Grass" "Grass" "Grass" "Fire" ...
##  $ Type_2          : chr  "Poison" "Poison" "Poison" NA ...
##  $ Total           : int  318 405 525 309 405 534 314 405 530 195 ...
##  $ HP              : int  45 60 80 39 58 78 44 59 79 45 ...
##  $ Attack          : int  49 62 82 52 64 84 48 63 83 30 ...
```

```
##  $ Defense         : int  49 63 83 43 58 78 65 80 100 35 ...
##  $ Sp_Atk          : int  65 80 100 60 80 109 50 65 85 20 ...
##  $ Sp_Def          : int  65 80 100 50 65 85 64 80 105 20 ...
##  $ Speed           : int  45 60 80 65 80 100 43 58 78 45 ...
##  $ Generation      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Color           : chr  "Green" "Green" "Green" "Red" ...
##  $ hasGender       : chr  "True" "True" "True" "True" ...
##  $ Pr_Male         : num  0.875 0.875 0.875 0.875 0.875 0.875 0.875 0.875 0.875 0.5 ...
##  $ Egg_Group_1     : chr  "Monster" "Monster" "Monster" "Monster" ...
##  $ Egg_Group_2     : chr  "Grass" "Grass" "Grass" "Dragon" ...
##  $ hasMegaEvolution: chr  "False" "False" "True" "False" ...
##  $ Height_m        : num  0.71 0.99 2.01 0.61 1.09 1.7 0.51 0.99 1.6 0.3 ...
##  $ Weight_kg       : num  6.9 13 100 8.5 19 90.5 9 22.5 85.5 2.9 ...
##  $ Catch_Rate      : int  45 45 45 45 45 45 45 45 45 255 ...
##  $ Body_Style      : chr  "quadruped" "quadruped" "quadruped" "bipedal_tailed" ...
```

Changing data type

```
pokemon_m$isLegendary <- as.factor(pokemon_m$isLegendary)
pokemon_m$Type_1 <- as.factor(pokemon_m$Type_1)
pokemon_m$Type_2 <- as.factor(pokemon_m$Type_2)
pokemon_m$Generation <- as.factor(pokemon_m$Generation)
pokemon_m$Color <- as.factor(pokemon_m$Color)
pokemon_m$hasGender <- as.factor(pokemon_m$hasGender)
pokemon_m$Egg_Group_1 <- as.factor(pokemon_m$Egg_Group_1)
pokemon_m$Egg_Group_2 <- as.factor(pokemon_m$Egg_Group_2)
pokemon_m$hasMegaEvolution <- as.factor(pokemon_m$hasMegaEvolution)
pokemon_m$Body_Style <- as.factor(pokemon_m$Body_Style)

str(pokemon_m)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    721 obs. of  22 variables:
##  $ Name            : chr  "Bulbasaur" "Ivysaur" "Venusaur" "Charmander" ...
##  $ isLegendary     : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Type_1          : Factor w/ 18 levels "Bug","Dark","Dragon",..: 10 10 10 7 7 7 18 18 18 1 ...
##  $ Type_2          : Factor w/ 18 levels "Bug","Dark","Dragon",..: 14 14 14 NA NA 8 NA NA NA NA ...
##  $ Total           : int  318 405 525 309 405 534 314 405 530 195 ...
##  $ HP              : int  45 60 80 39 58 78 44 59 79 45 ...
##  $ Attack          : int  49 62 82 52 64 84 48 63 83 30 ...
##  $ Defense         : int  49 63 83 43 58 78 65 80 100 35 ...
##  $ Sp_Atk          : int  65 80 100 60 80 109 50 65 85 20 ...
##  $ Sp_Def          : int  65 80 100 50 65 85 64 80 105 20 ...
##  $ Speed           : int  45 60 80 65 80 100 43 58 78 45 ...
##  $ Generation      : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Color           : Factor w/ 10 levels "Black","Blue",..: 4 4 4 8 8 8 2 2 2 4 ...
##  $ hasGender       : Factor w/ 2 levels "False","True": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Pr_Male         : num  0.875 0.875 0.875 0.875 0.875 0.875 0.875 0.875 0.875 0.5 ...
##  $ Egg_Group_1     : Factor w/ 15 levels "Amorphous","Bug",..: 11 11 11 11 11 11 11 11 11 2 ...
##  $ Egg_Group_2     : Factor w/ 13 levels "Amorphous","Bug",..: 7 7 7 3 3 3 11 11 11 NA ...
##  $ hasMegaEvolution: Factor w/ 2 levels "False","True": 1 1 2 1 1 2 1 1 2 1 ...
##  $ Height_m        : num  0.71 0.99 2.01 0.61 1.09 1.7 0.51 0.99 1.6 0.3 ...
##  $ Weight_kg       : num  6.9 13 100 8.5 19 90.5 9 22.5 85.5 2.9 ...
##  $ Catch_Rate      : int  45 45 45 45 45 45 45 45 45 255 ...
```

```
##  $ Body_Style      : Factor w/ 14 levels "bipedal_tailed",..: 10 10 10 1 1 1 1 1 1 8 ...
```

Aggregate the power by total and type1 (total_mean)

```
type_power <- aggregate(pokemon_m$Total, by=list(pokemon_m$Type_1), FUN=mean)  # aggregate
head(type_power)
```

```
##    Group.1        x
## 1      Bug 365.1270
## 2     Dark 434.7500
## 3   Dragon 501.9583
## 4 Electric 420.6944
## 5    Fairy 413.1765
## 6 Fighting 404.3600
```

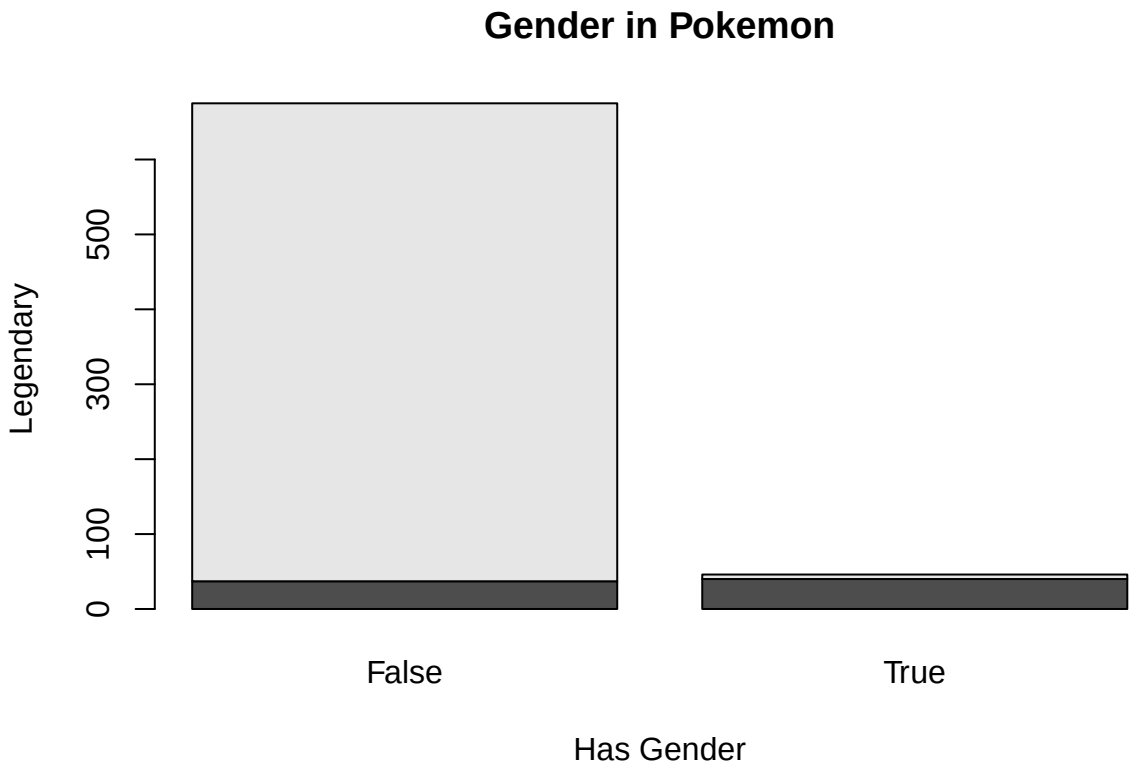
HasGender

As I show earlier, Pre_Male has some missing values.

```
gender = table(pokemon_m$hasGender, pokemon_m$isLegendary)
gender
```

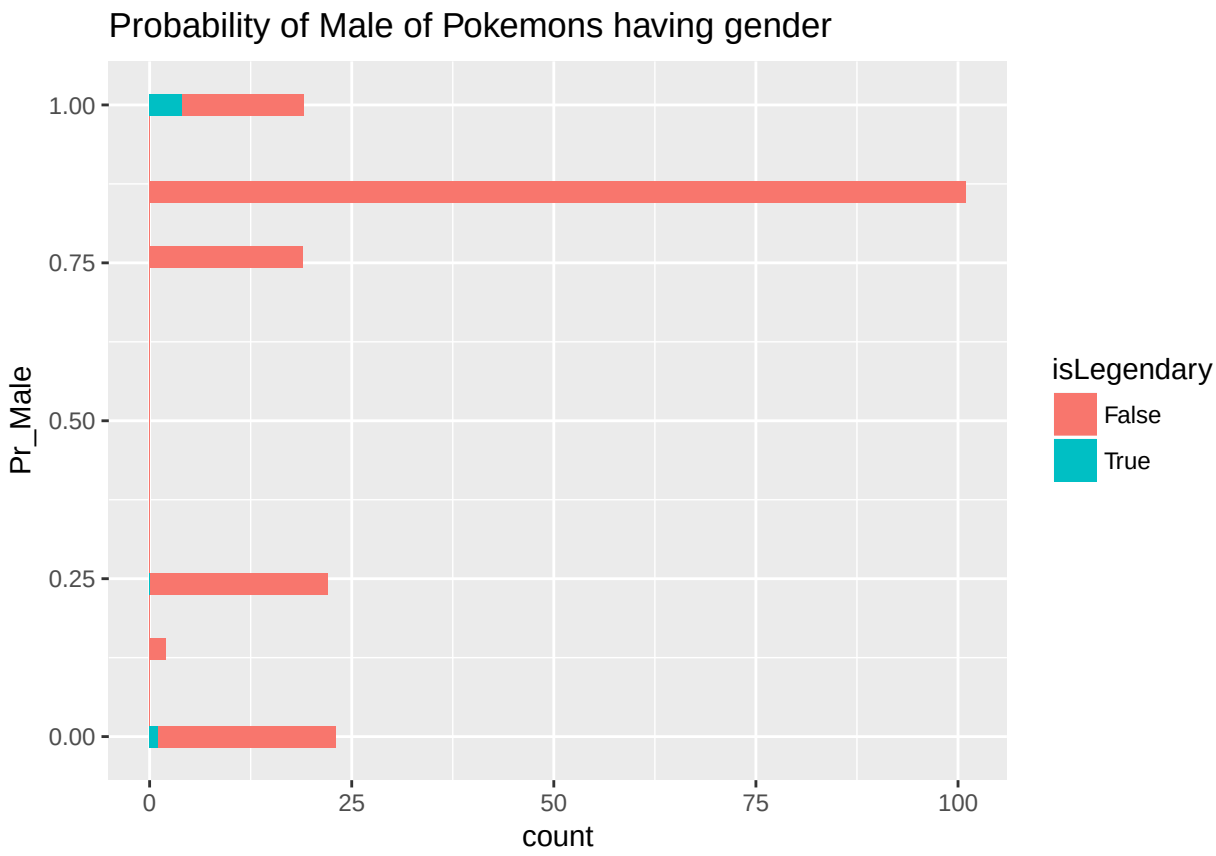
```
##
##         False True
##   False    37   40
##   True    638    6
```

```
barplot(gender,main="Gender in Pokemon",xlab="Has Gender",ylab="Legendary")
```



```
pokemon_has_gender = pokemon_m %>% filter(hasGender=="True" & Pr_Male != 0.5)
# I will consider pr_male = 0.5 as no gender indicated.
ggplot(pokemon_has_gender , aes(x=Pr_Male, fill = isLegendary)) +
  geom_histogram() +
  coord_flip() +
  labs(title ="Probability of Male of Pokemons having gender")
```

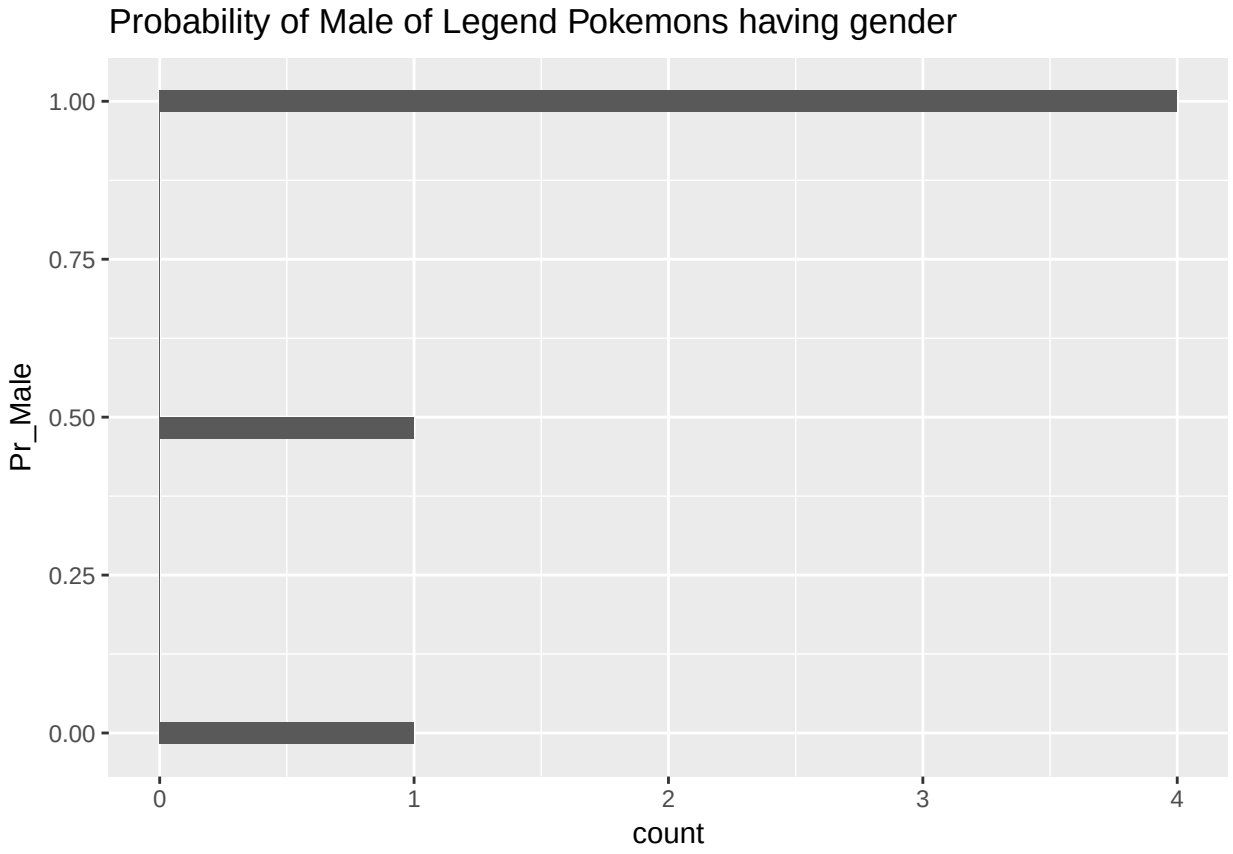
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
legend=pokemon_m %>% filter(isLegendary =="True")

ggplot(legend, aes(x=Pr_Male)) +
  geom_histogram() +
  coord_flip() +
  labs(title = "Probability of Male of Legend Pokemons having gender")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 40 rows containing non-finite values (stat_bin).
```



MegaEvolution percentage

```
pokemon_m %>% group_by(Type_1, isLegendary) %>% tally() %>%  arrange(desc(n))
```

```
## Source: local data frame [33 x 3]
## Groups: Type_1 [18]
##
##      Type_1 isLegendary     n
##      <fctr>      <fctr> <int>
## 1     Water       False   102
## 2    Normal       False    91
## 3     Grass       False    64
## 4       Bug       False    63
## 5      Fire       False    42
## 6   Psychic       False    39
## 7      Rock       False    38
## 8  Electric       False    33
## 9    Ground       False    28
## 10   Poison       False    28
## # ... with 23 more rows
```

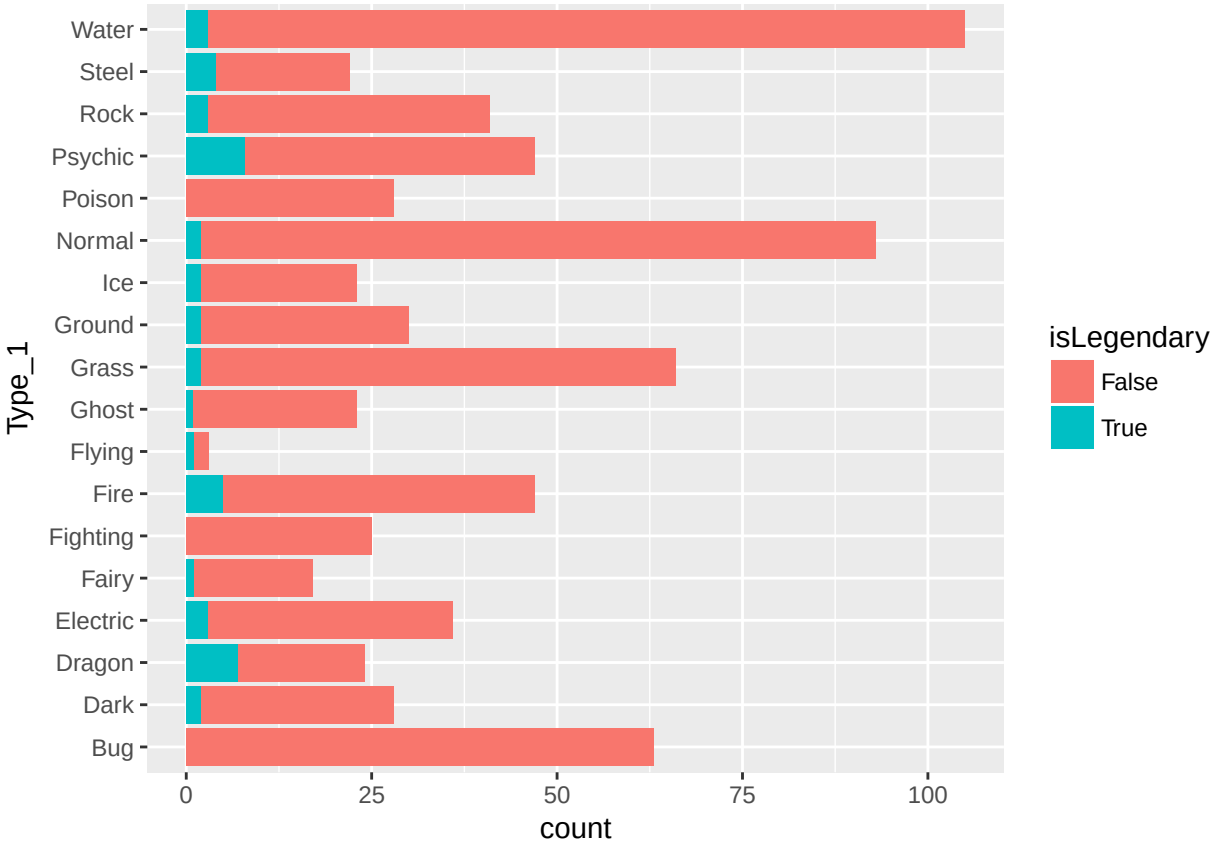
```
pokemon_m %>% group_by(hasMegaEvolution, isLegendary) %>% tally() %>% mutate(perc = n/sum(n)) %>% arrang
```

```
## Source: local data frame [4 x 4]
## Groups: hasMegaEvolution [2]
##
```

```
##   hasMegaEvolution isLegendary     n       perc
##              <fctr>      <fctr> <int>      <dbl>
## 1            False        True    41 0.06074074
## 2             True        True     5 0.10869565
## 3             True       False    41 0.89130435
## 4            False       False   634 0.93925926
```

Type_1

```
type_1 <- ggplot(pokemon_m, aes(x=Type_1))
 type_1+geom_bar(aes(fill=isLegendary)) +
   coord_flip()
```

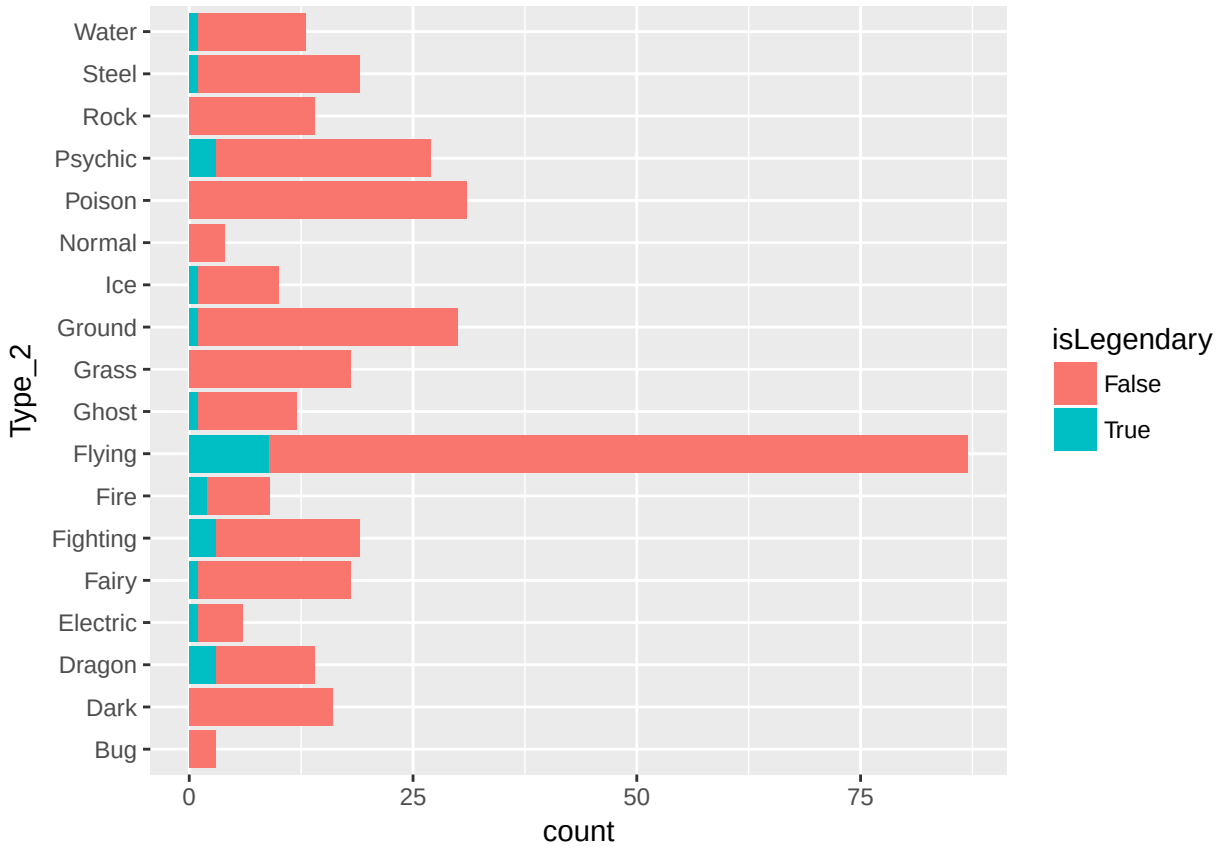


Type_2

```
sum(is.na(pokemon_m$Type_2))
```

```
## [1] 371
```

```
pokemon_type2<- pokemon %>% filter(!is.na(Type_2))


type_2 <- ggplot(pokemon_type2, aes(x=Type_2))
 type_2+geom_bar(aes(fill=isLegendary)) +
   coord_flip()
```



more details about legendary group

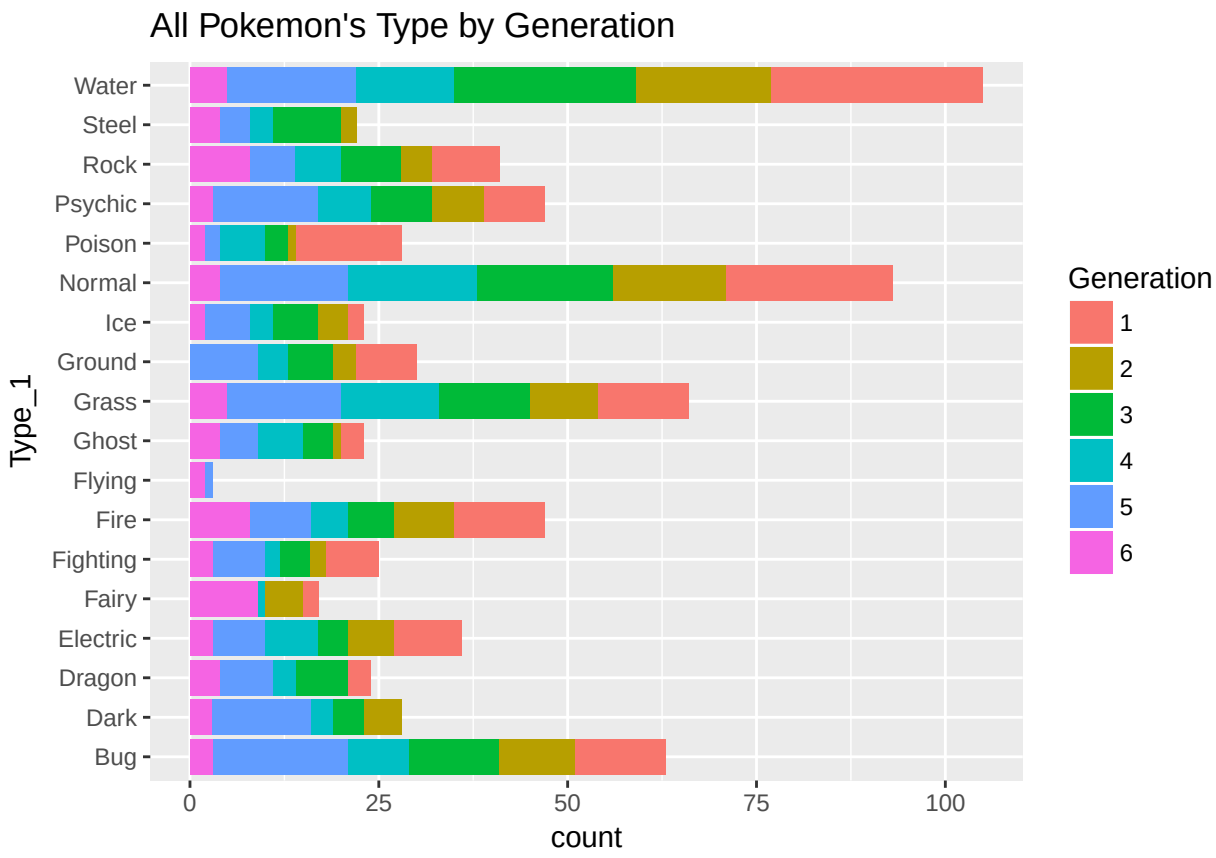
```
legend %>% group_by(Type_1)%>% tally() %>% arrange(desc(n))  # Psychic has the highest number of legend
```

```
## # A tibble: 15 × 2
##      Type_1     n
##      <fctr> <int>
## 1   Psychic     8
## 2    Dragon     7
## 3      Fire     5
## 4     Steel     4
## 5  Electric     3
## 6      Rock     3
## 7     Water     3
## 8      Dark     2
## 9     Grass     2
## 10   Ground     2
## 11      Ice     2
## 12   Normal     2
## 13    Fairy     1
## 14   Flying     1
## 15    Ghost     1
```

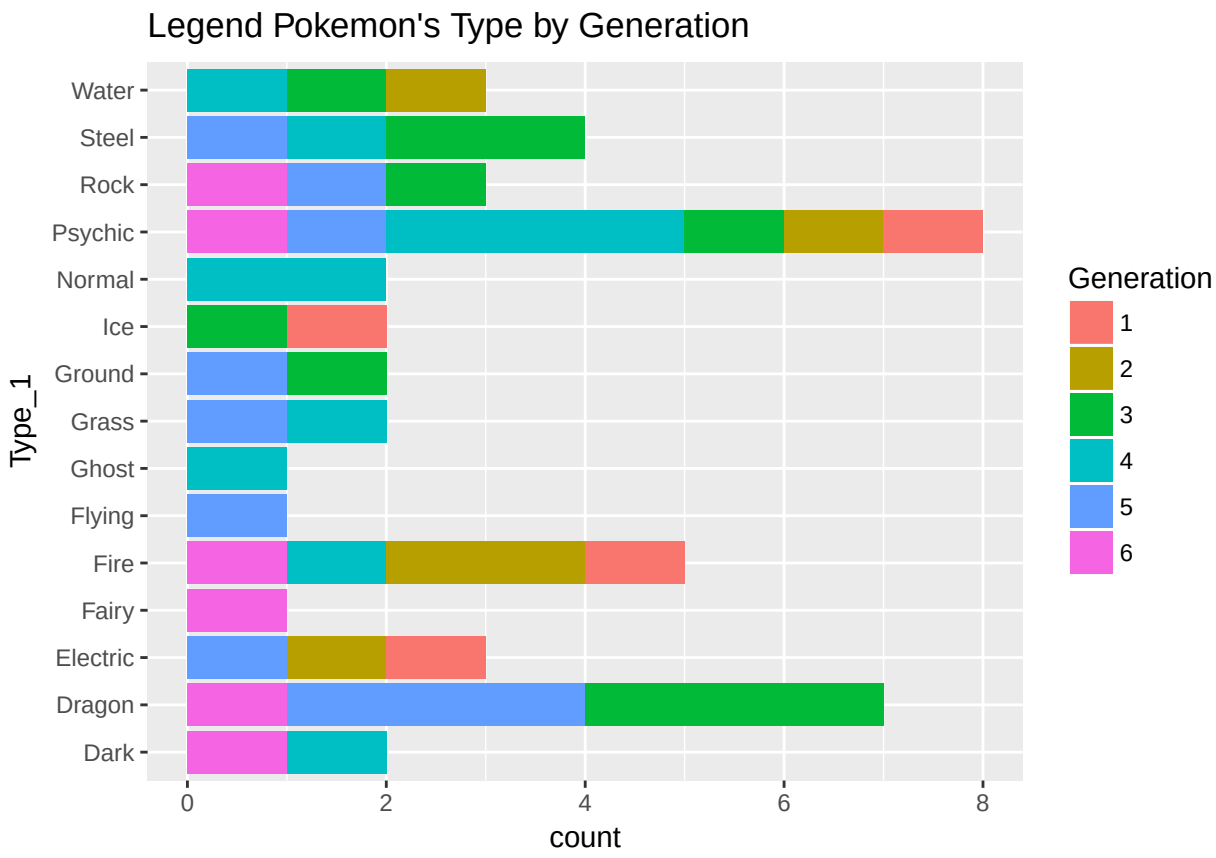
```
#Psychi, Dragon, Fire, Steel
```

Generation

```
ggplot(data = pokemon_m) +
  geom_bar(mapping = aes(x = Type_1, fill = Generation)) +
  labs(title = "All Pokemon's Type by Generation") +
  coord_flip()
```

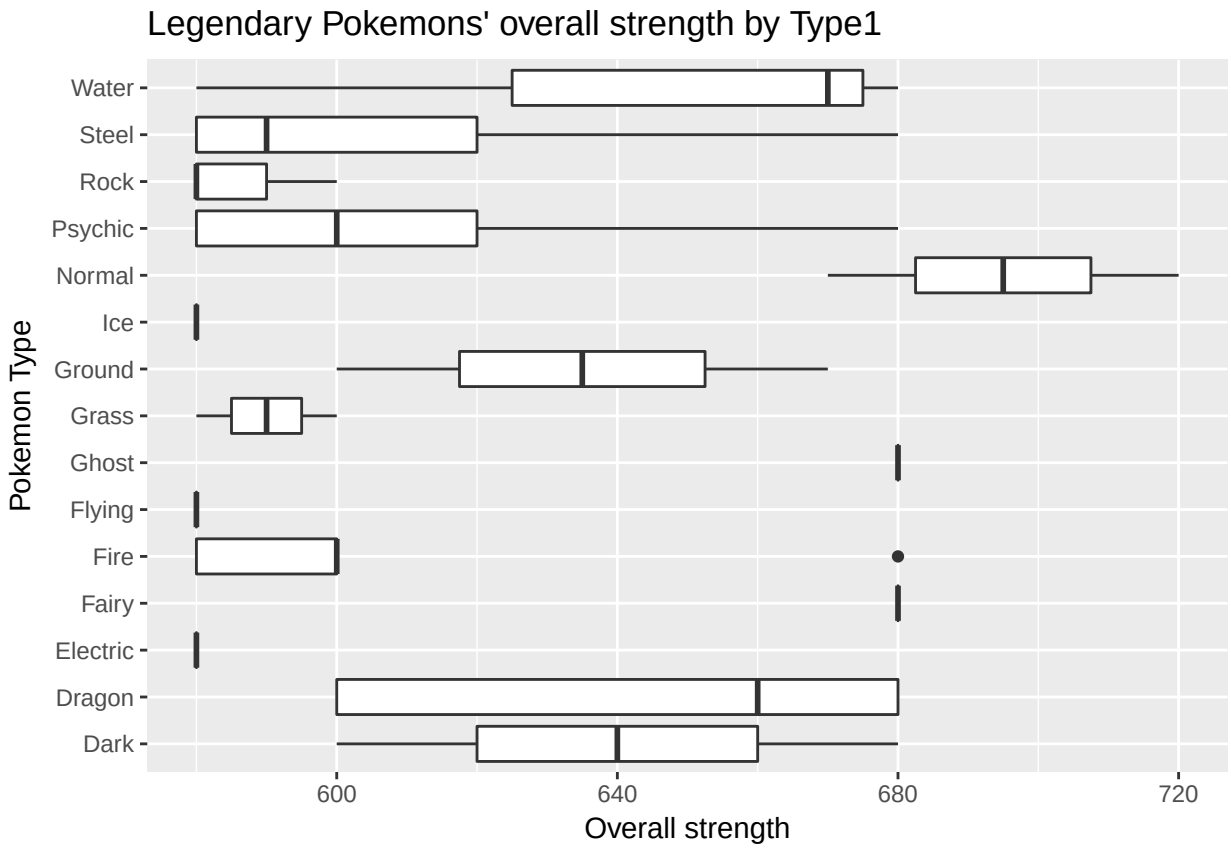



```
ggplot(data = legend) +
  geom_bar(mapping = aes(x = Type_1, fill = Generation)) +
  labs(title = "Legend Pokemon's Type by Generation") +
  coord_flip()
```

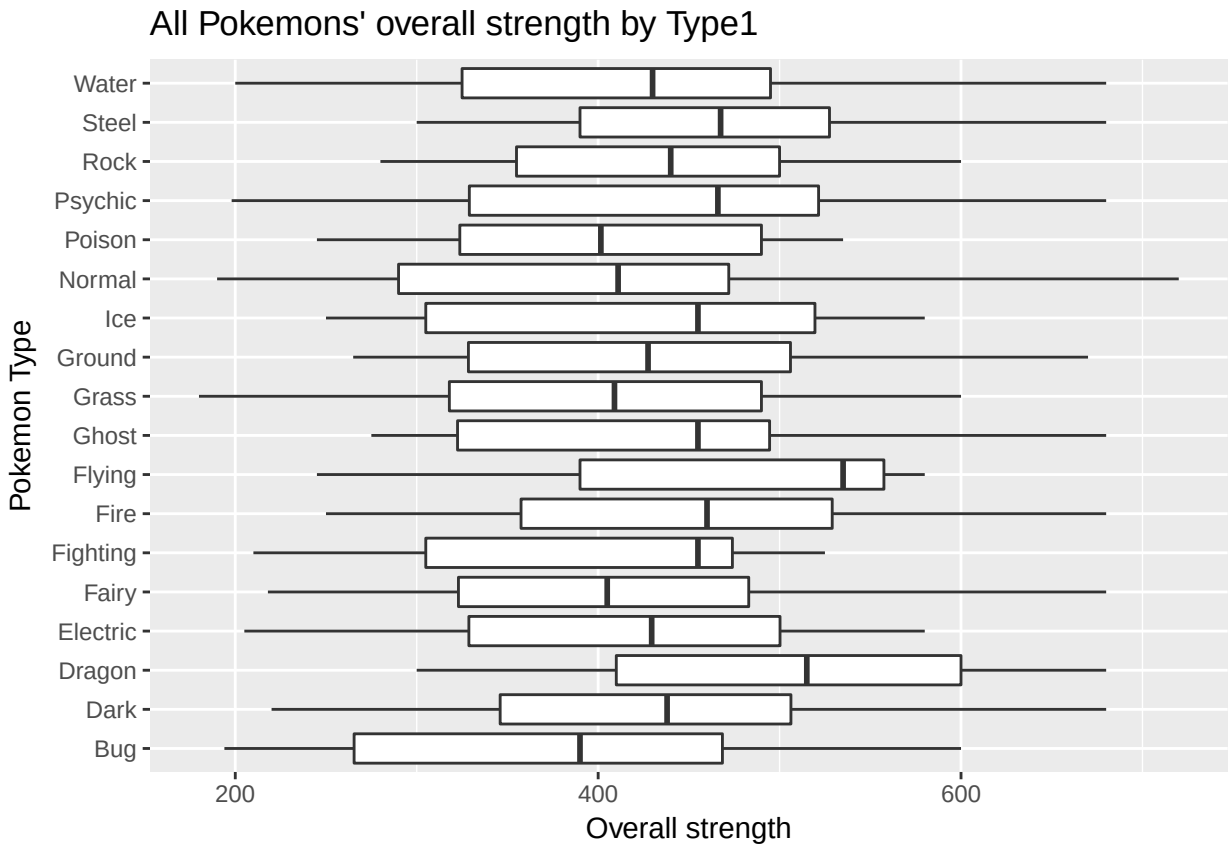


Total - Overall Strength

```
ggplot(data = legend, mapping = aes(x = Type_1, y = Total)) +
  geom_boxplot() +
  coord_flip() +
  labs( x = "Pokemon Type", y = "Overall strength",
 title ="Legendary Pokemons' overall strength by Type1")
```



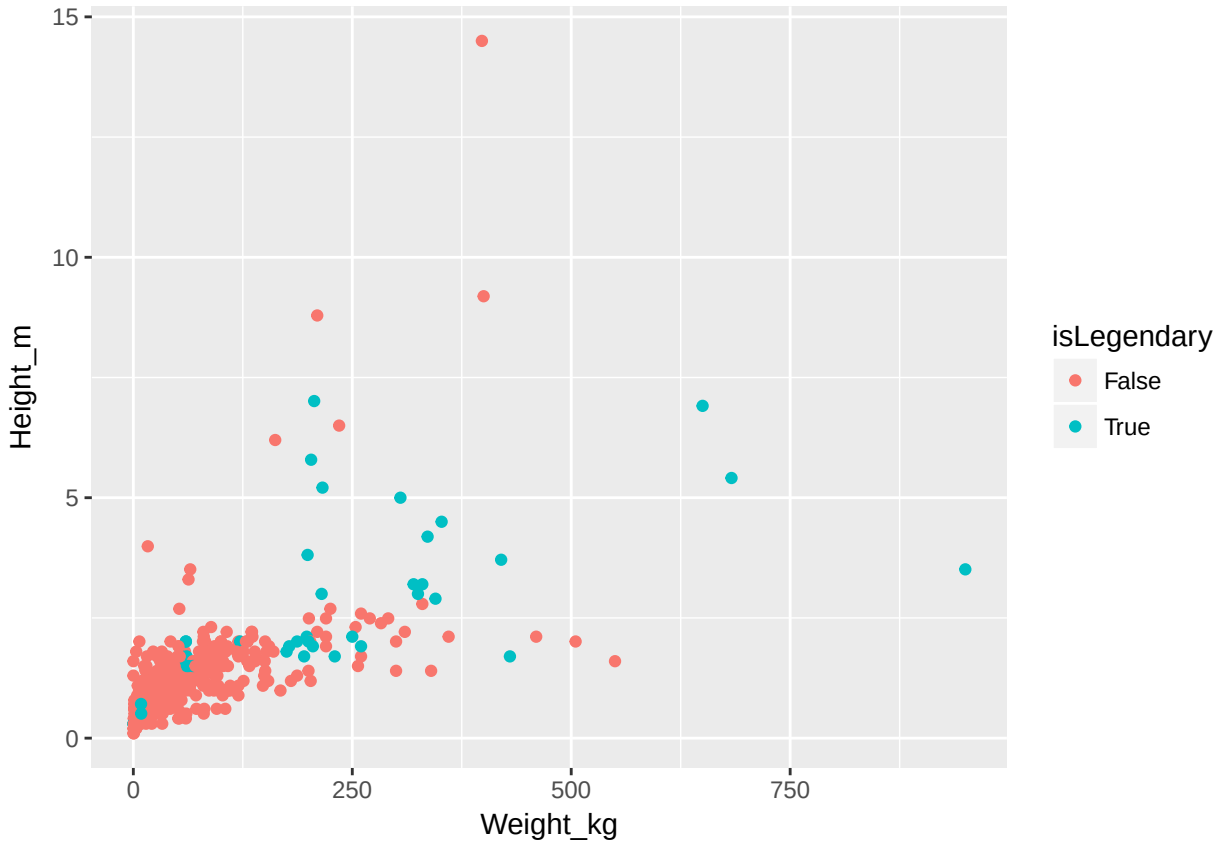

```
ggplot(data = pokemon_m, mapping = aes(x = Type_1, y = Total)) +
  geom_boxplot() +
  coord_flip() +
  labs( x = "Pokemon Type", y = "Overall strength",
 title ="All Pokemons' overall strength by Type1")
```



height and Weight

Heavier pokemons are more likely to be a legendary pokemon.

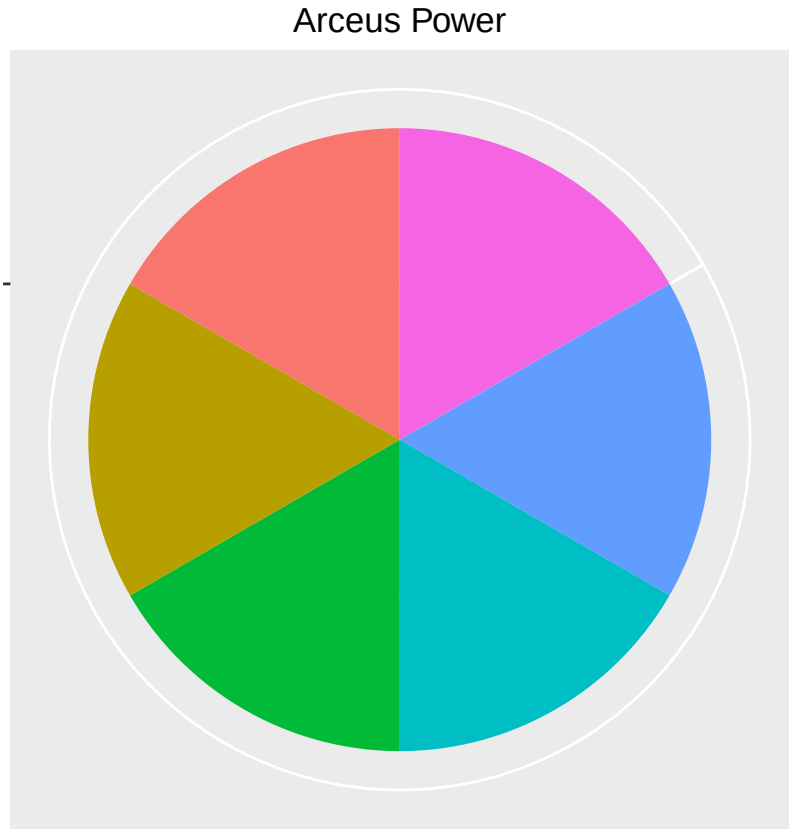
```
ggplot(pokemon_m, aes(x=Weight_kg, y=Height_m)) +
  geom_point(aes(color = isLegendary))
```



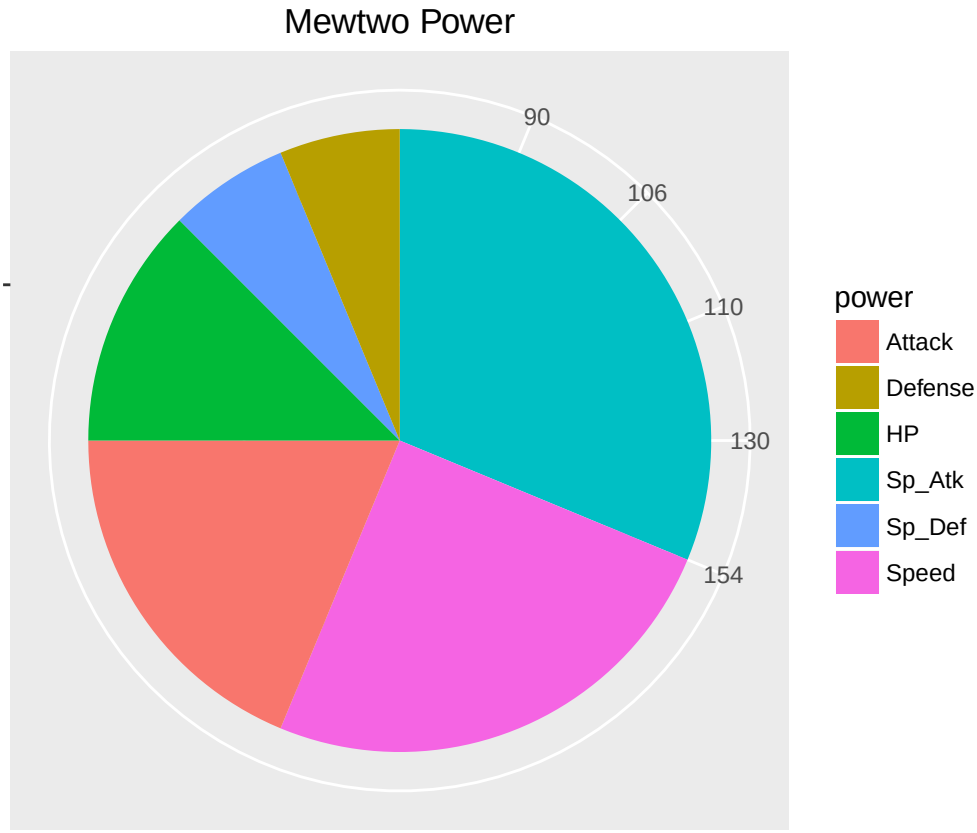
Top 3 legend pokemon by Power - Pie Chart

```
top10=legend %>% select(Name, Type_1, Total, HP, Attack, Defense, Sp_Atk, Sp_Def, Speed, Catch_Rate) %>%

# Simple Pie Chart
t_top10 = data.frame(t(top10))
t_top10 = t_top10[c(4:9),]

power = c("HP", "Attack", "Defense", "Sp_Atk", "Sp_Def", "Speed")

t_top10[, 11] <- power
names(t_top10) <- c("Arceus", "Mewtwo","Lugia", "Ho-Oh", "Rayquaza", "Dialga", "Palkia", "Giratina", "R

pie_arceus <- ggplot(t_top10, aes(x = "", y=Arceus, fill = factor(power))) +
  geom_bar(width = 1, stat = "identity") +
  theme(axis.line = element_blank(),
        plot.title = element_text(hjust=0.5)) +
  labs(fill="power",
       x=NULL,
       y=NULL,
       title="Arceus Power") +
  coord_polar(theta = "y", start=0)

pie_Mewtwo <- ggplot(t_top10, aes(x = "", y=Mewtwo, fill = factor(power))) +
```

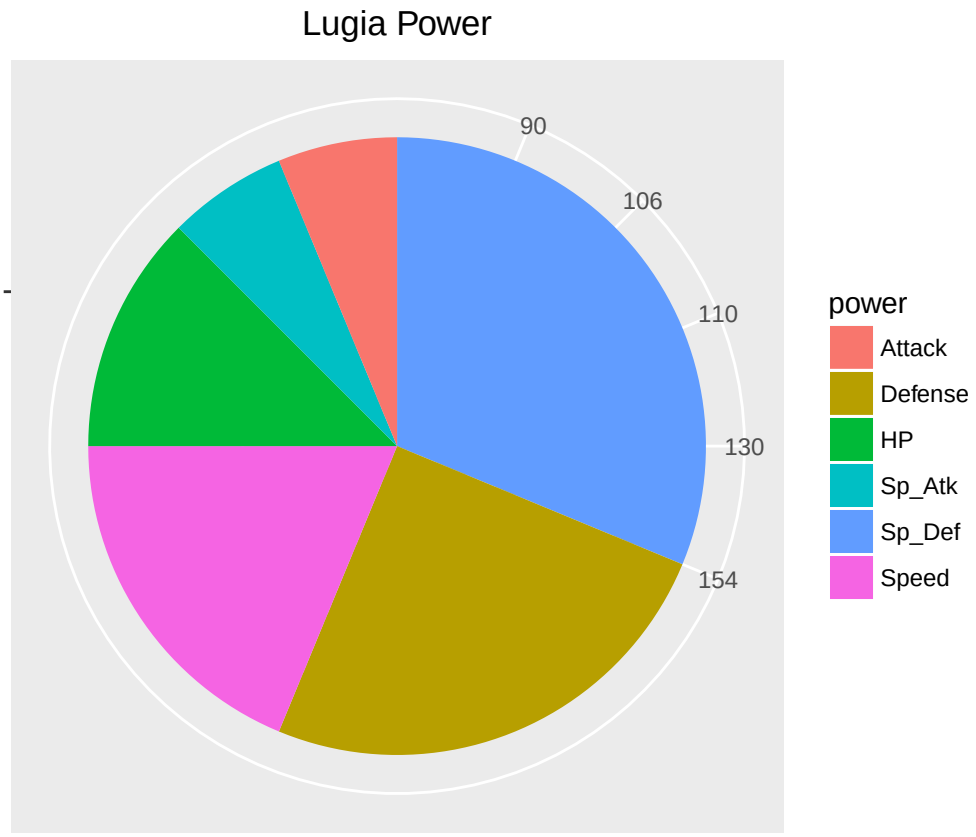
```
  geom_bar(width = 1, stat = "identity") +
  theme(axis.line = element_blank(),
        plot.title = element_text(hjust=0.5)) +
  labs(fill="power",
       x=NULL,
       y=NULL,
       title="Mewtwo Power") +
  coord_polar(theta = "y", start=0)

pie_Lugia <- ggplot(t_top10, aes(x = "", y=Lugia, fill = factor(power))) +
  geom_bar(width = 1, stat = "identity") +
  theme(axis.line = element_blank(),
        plot.title = element_text(hjust=0.5)) +
  labs(fill="power",
       x=NULL,
       y=NULL,
       title="Lugia Power") +
  coord_polar(theta = "y", start=0)


pie_arceus
```




```
pie_Mewtwo
```


Mewtwo Power
90
106
110
130
154
power
Attack
Defense
HP
Sp_Atk
Sp_Def
Speed

```
pie_Lugia
```

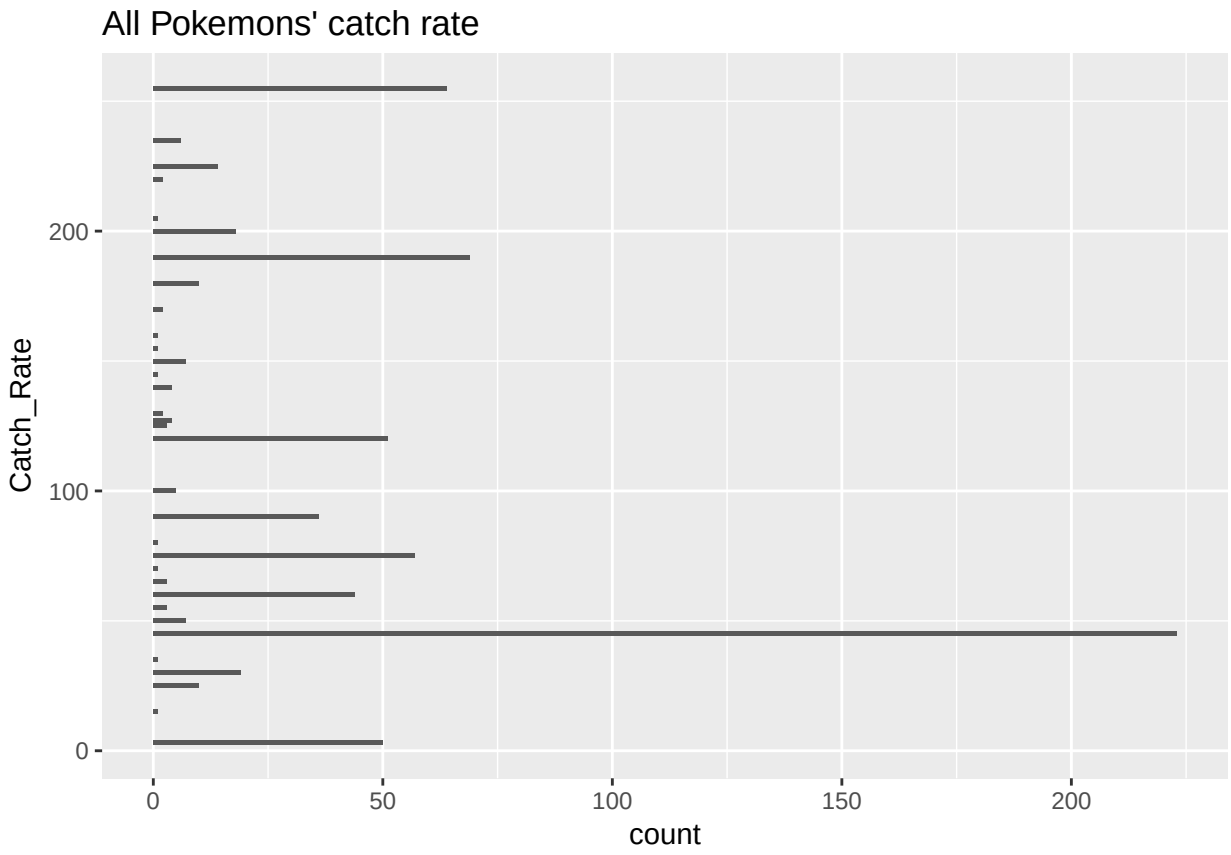

Lugia Power
90
106
110
130
154
power
Attack
Defense
HP
Sp_Atk
Sp_Def
Speed

Catch Rate

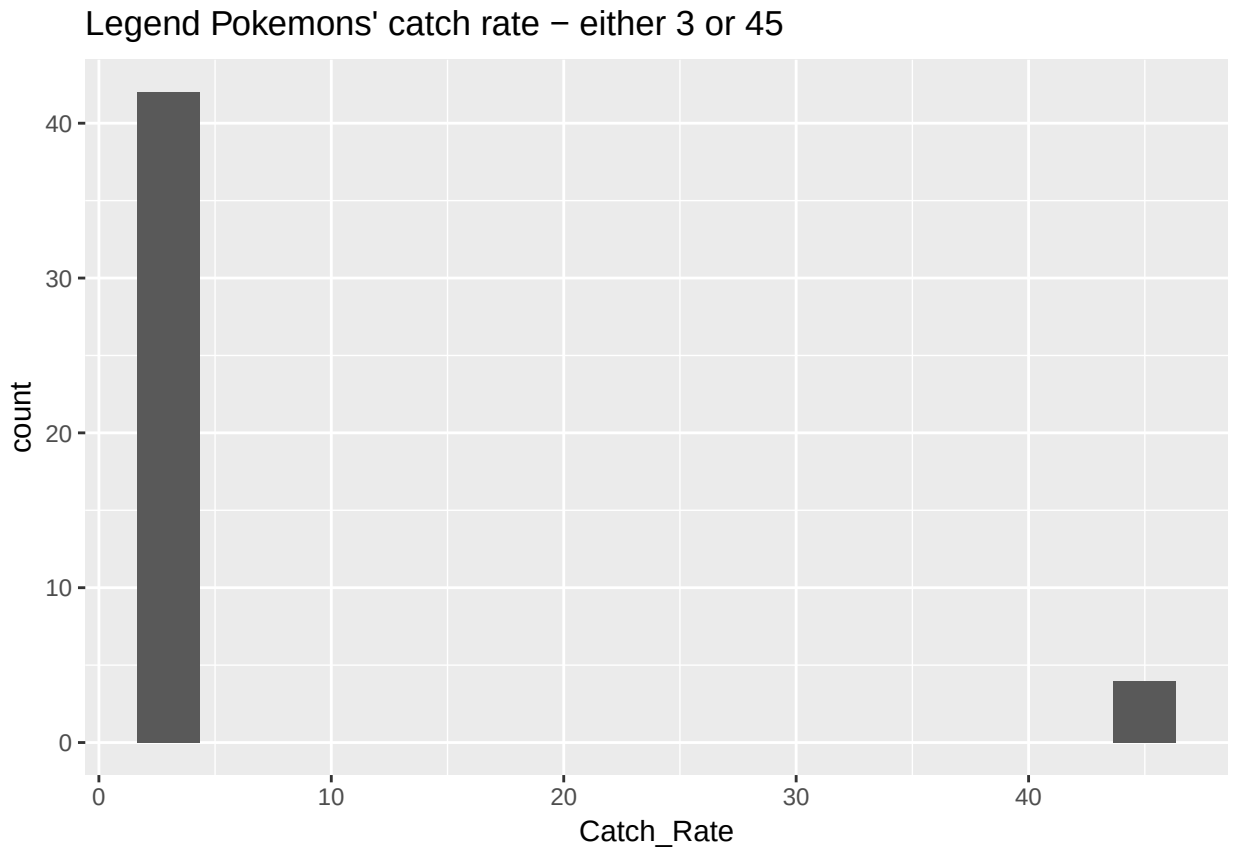
```
pokemon_m %>% select(Catch_Rate) %>%  arrange(Catch_Rate)
```

```
## # A tibble: 721 × 1
##    Catch_Rate
##         <int>
## 1           3
## 2           3
## 3           3
## 4           3
## 5           3
## 6           3
## 7           3
## 8           3
## 9           3
## 10          3
## # ... with 711 more rows
```

```
ggplot(pokemon_m, aes(x=Catch_Rate)) + geom_bar() + labs(title = "All Pokemons' catch rate") +
 coord_flip()
```




```
ggplot(legend, aes(x= Catch_Rate)) + geom_bar() + labs(title ="Legend Pokemons' catch rate - either 3 o
```




```
pokemon_m %>% group_by(Body_Style) %>% tally() %>% arrange(desc(n)) %>% slice(1:10)
```

```
## # A tibble: 10 × 2
##          Body_Style     n
##              <fctr> <int>
## 1    bipedal_tailed   158
## 2         quadruped   135
## 3  bipedal_tailless   109
## 4         two_wings    63
## 5         head_arms    39
## 6         head_only    34
## 7         with_fins    31
## 8         head_base    30
## 9         insectoid    30
## 10  serpentine_body    29
```

```
# Number of all pokemons by Body Style
legend %>% group_by(Body_Style) %>% tally() %>% arrange(desc(n)) %>% slice(1:10)
```

```
## # A tibble: 8 × 2
##         Body_Style     n
##             <fctr> <int>
## 1        quadruped    12
## 2        two_wings     9
## 3   bipedal_tailed     8
## 4 bipedal_tailless     8
## 5        head_arms     4
## 6  serpentine_body     3
```

```
## 7        head_only     1
## 8        with_fins     1
```

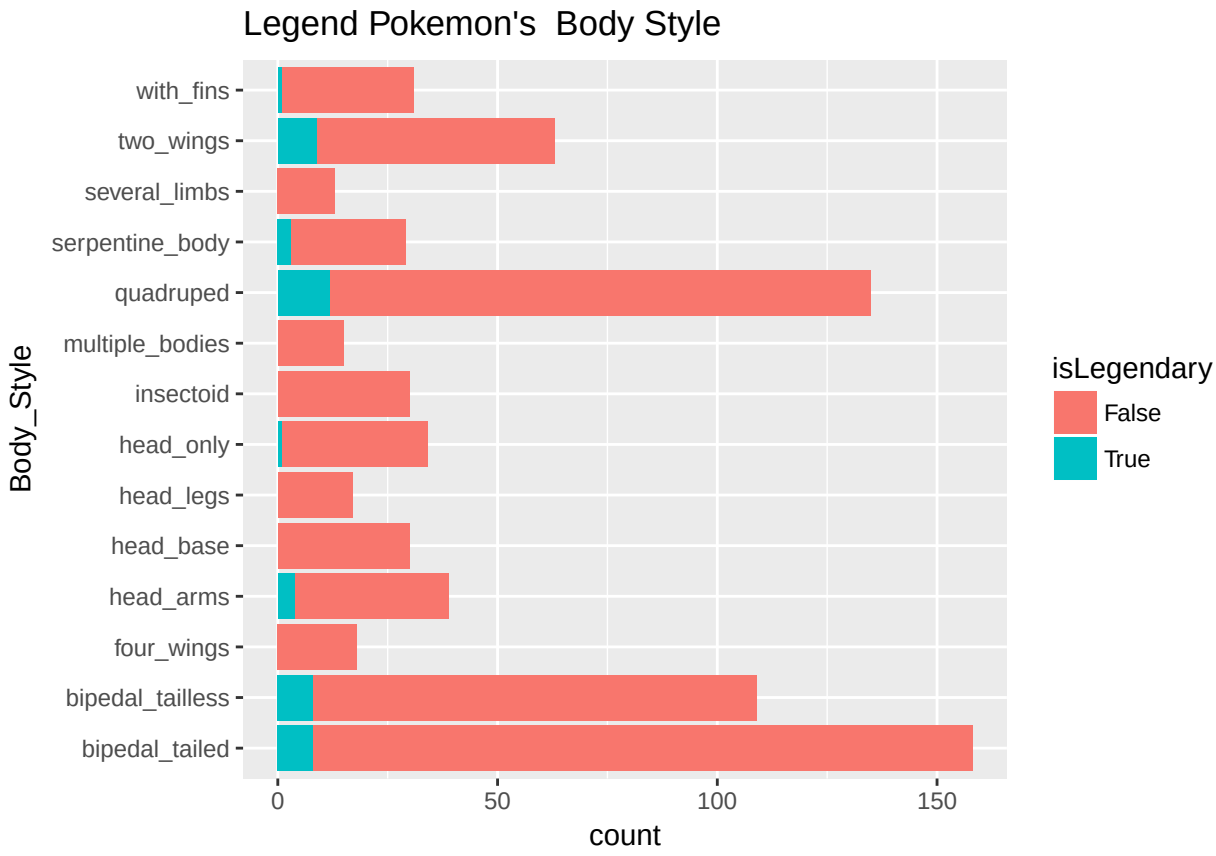
```
# Number of legend pokemons by Body Style

d= pokemon_m %>% select(Body_Style, isLegendary, Catch_Rate) %>% group_by(Body_Style, isLegendary) %>% r

style_catchrate = d %>% group_by(Body_Style, avg_catchrate, isLegendary) %>% tally() %>% arrange(avg_ca
style_catchrate[1:10,]
```

```
## Source: local data frame [10 x 4]
## Groups: Body_Style, avg_catchrate [10]
##
##          Body_Style avg_catchrate isLegendary     n
##              <fctr>         <dbl>      <fctr> <int>
## 1    bipedal_tailed      3.000000        True     8
## 2  bipedal_tailless      3.000000        True     8
## 3         head_arms      3.000000        True     4
## 4         head_only      3.000000        True     1
## 5         with_fins      3.000000        True     1
## 6         two_wings      7.666667        True     9
## 7         quadruped     10.000000        True    12
## 8   serpentine_body     17.000000        True     3
## 9        four_wings     57.222222       False    18
## 10  multiple_bodies     76.866667       False    15
```

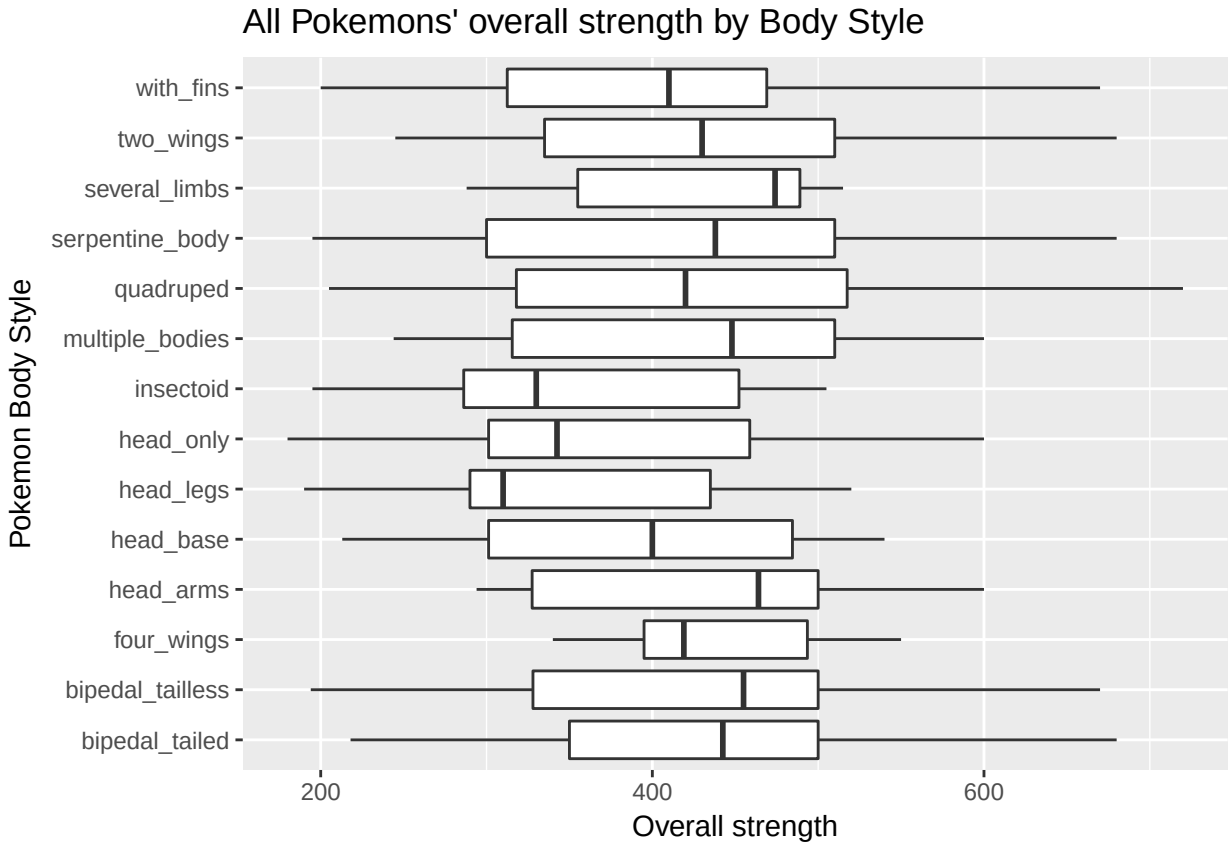
```
# We can see that pokemons who have low average catch rate by body_style are more likley to be a legend
```

Body_Style

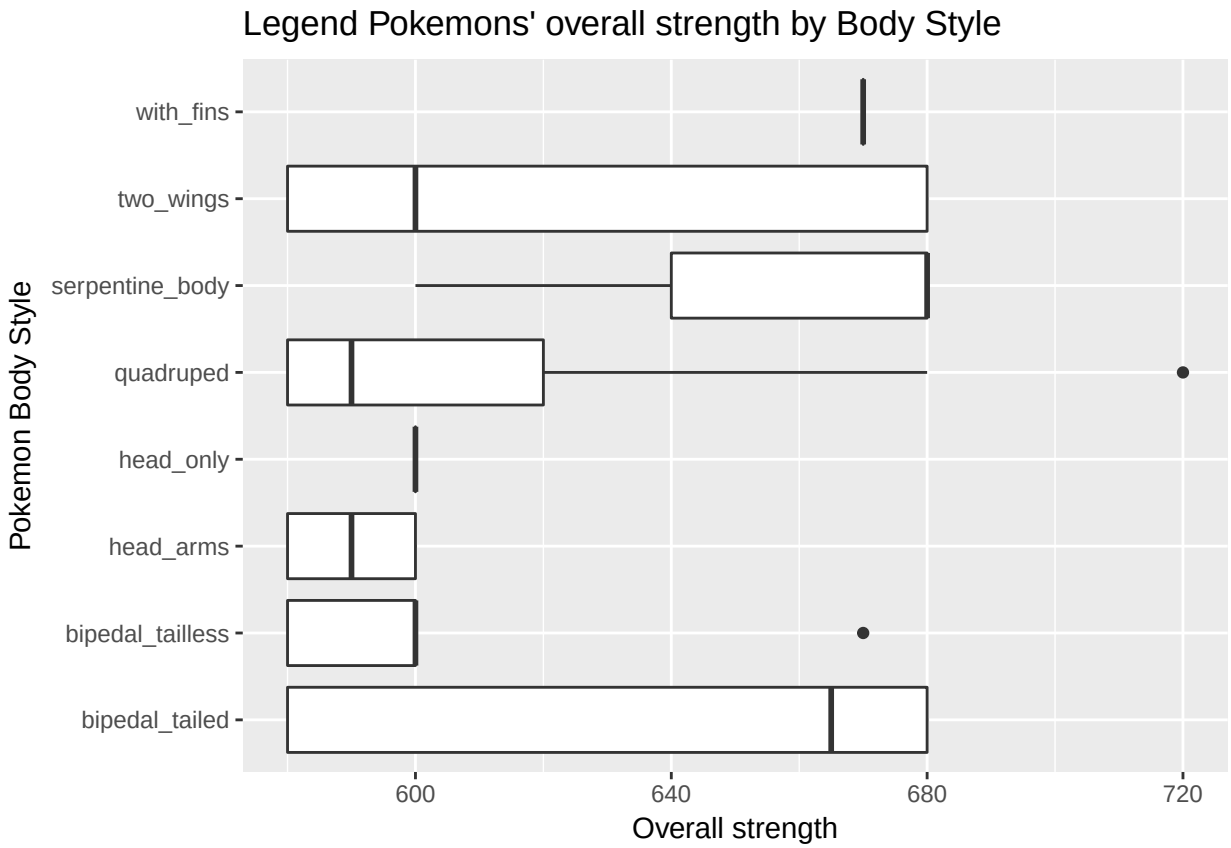
```
ggplot(data = pokemon_m) +
  geom_bar(mapping = aes(x = Body_Style, fill = isLegendary)) +
  labs(title = "Legend Pokemon's  Body Style") +
  coord_flip()
```




```
ggplot(data = pokemon_m, mapping = aes(x = Body_Style, y = Total)) +
  geom_boxplot() +
  coord_flip() +
  labs( x = "Pokemon Body Style", y = "Overall strength",
 title ="All Pokemons' overall strength by Body Style")
```




```
ggplot(data = legend, mapping = aes(x = Body_Style, y = Total)) +
  geom_boxplot() +
  coord_flip() +
  labs( x = "Pokemon Body Style", y = "Overall strength",
 title ="Legend Pokemons' overall strength by Body Style")
```



Variable Selection

```
#create training and validation data from given data
library(caTools)
```

isLegendary ~ Type_1 + Total + HP + Attack + Defense + Sp_Atk + Sp_Def + Speed + Generation + Color + hasGender + Egg_Group_1 + Height_m + Weight_kg + Catch_Rate + Body_Style

Model

- Cross Validation - Variable Selection

```
cross <- lm(Catch_Rate ~isLegendary + Type_1 + Total + hasMegaEvolution + HP + Attack +Defense +Sp_Atk +
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared
## on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 17 in
## model.matrix: no columns are assigned
```

```
step(cross, direction="backward", trace=0)
```

```
## Warning in model.matrix.default(object, data = structure(list(Catch_Rate =
## c(45L, : the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(object, data = structure(list(Catch_Rate =
## c(45L, : problem with term 17 in model.matrix: no columns are assigned
```

```
##
## Call:
## lm(formula = Catch_Rate ~ isLegendary + Type_1 + Total + hasMegaEvolution +
##     Defense + Generation + Color + Egg_Group_1 + Weight_kg, data = pokemon_m)
##
## Coefficients:
##             (Intercept)          isLegendaryTrue               Type_1Dark
##               295.30678                37.36531                10.59729
##            Type_1Dragon           Type_1Electric              Type_1Fairy
##                -0.38965                28.12334                25.88303
##          Type_1Fighting              Type_1Fire             Type_1Flying
##                11.61346                -6.31063                 3.57749
##             Type_1Ghost              Type_1Grass             Type_1Ground
##                17.39704                 2.64692                 8.96040
##               Type_1Ice             Type_1Normal             Type_1Poison
##                 8.48786                20.16227                46.18782
##           Type_1Psychic              Type_1Rock              Type_1Steel
##                28.81990               -16.30070               -20.59420
##             Type_1Water                   Total     hasMegaEvolutionTrue
##                20.55912                -0.58863                11.48419
##                 Defense             Generation2             Generation3
##                 0.20589               -11.27568                 7.63907
##             Generation4             Generation5             Generation6
##                -3.02282                10.83970                 9.32368
##               ColorBlue              ColorBrown              ColorGreen
##                -9.30006                18.76517                 7.34076
##               ColorGrey               ColorPink              ColorPurple
##                 7.01101                -3.32418                 1.69848
##                ColorRed               ColorWhite              ColorYellow
##                 8.17876                 4.75578                15.34485
##          Egg_Group_1Bug         Egg_Group_1Ditto        Egg_Group_1Dragon
##                12.33418              -122.81232               -17.47939
##        Egg_Group_1Fairy         Egg_Group_1Field        Egg_Group_1Flying
##                39.16783                16.51780                14.30235
##        Egg_Group_1Grass    Egg_Group_1Human-Like       Egg_Group_1Mineral
##                40.52211                 1.30733                14.28510
##      Egg_Group_1Monster  Egg_Group_1Undiscovered       Egg_Group_1Water_1
##                -8.51104               -13.62049                15.17999
##      Egg_Group_1Water_2       Egg_Group_1Water_3               Weight_kg
##                28.11618                15.66040                 0.07193
```

```
cross_min <- lm(Catch_Rate ~ isLegendary + Type_1 + Total + hasMegaEvolution + Generation + Defense + C
step(cross_min, scope=list(upper = cross,
lower= cross_min),
direction="both", trace=0)
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##
```

```
## Call:
## lm(formula = Catch_Rate ~ isLegendary + Type_1 + Total + hasMegaEvolution +
##     Generation + Defense + Color + Egg_Group_1 + Weight_kg, data = pokemon_m)
##
## Coefficients:
##             (Intercept)          isLegendaryTrue               Type_1Dark
##               295.30678                37.36531                10.59729
##            Type_1Dragon           Type_1Electric              Type_1Fairy
##                -0.38965                28.12334                25.88303
##          Type_1Fighting              Type_1Fire             Type_1Flying
##                11.61346                -6.31063                 3.57749
##             Type_1Ghost              Type_1Grass             Type_1Ground
##                17.39704                 2.64692                 8.96040
##               Type_1Ice             Type_1Normal             Type_1Poison
##                 8.48786                20.16227                46.18782
##           Type_1Psychic              Type_1Rock              Type_1Steel
##                28.81990               -16.30070               -20.59420
##             Type_1Water                   Total     hasMegaEvolutionTrue
##                20.55912                -0.58863                11.48419
##             Generation2             Generation3             Generation4
##               -11.27568                 7.63907                -3.02282
##             Generation5             Generation6                 Defense
##                10.83970                 9.32368                 0.20589
##               ColorBlue              ColorBrown              ColorGreen
##                -9.30006                18.76517                 7.34076
##               ColorGrey              ColorPink              ColorPurple
##                 7.01101                -3.32418                 1.69848
##                ColorRed              ColorWhite              ColorYellow
##                 8.17876                 4.75578                15.34485
##          Egg_Group_1Bug         Egg_Group_1Ditto        Egg_Group_1Dragon
##                12.33418              -122.81232               -17.47939
##        Egg_Group_1Fairy         Egg_Group_1Field        Egg_Group_1Flying
##                39.16783                16.51780                14.30235
##        Egg_Group_1Grass    Egg_Group_1Human-Like       Egg_Group_1Mineral
##                40.52211                 1.30733                14.28510
##      Egg_Group_1Monster  Egg_Group_1Undiscovered       Egg_Group_1Water_1
##                -8.51104               -13.62049                15.17999
##      Egg_Group_1Water_2       Egg_Group_1Water_3               Weight_kg
##                28.11618                15.66040                 0.07193
```

Multiple linear regressino anova table

```r
library(rpart)
set.seed(3)


train <- sample (1:nrow(pokemon_m), .8*nrow(pokemon_m)) # training row indices

inputData <- pokemon_m[train, ] # training data

testData <- pokemon_m[-train, ] # test data
```

```r
fit <- lm(Catch_Rate ~isLegendary + Type_1 + Total + hasMegaEvolution + HP + Attack +Defense +Sp_Atk +Sp
#summary(fit)
fit1 <- lm(Catch_Rate ~Total*Weight_kg*Egg_Group_1*Type_1*isLegendary, data = inputData)
#summary(fit1)

fit$coefficients["Total"]
```

```
##      Total
## -0.7392571
```

```r
test.pred.lin<- predict(fit1,testData)
```

```
## Warning in predict.lm(fit1, testData): prediction from a rank-deficient fit
## may be misleading
```

```r
test.pred.lin
```

```
##           1           2           3           4           5
##   41.330802   58.571116   56.210501  220.238965  219.506646
##           6           7           8           9          10
##  198.175061  204.846746  154.158474   69.536146  236.426923
##          11          12          13          14          15
##   82.473454   67.029034   85.757301  181.920361  224.164274
##          16          17          18          19          20
##  -17.157857  111.270172  264.482501   35.643351  178.216292
##          21          22          23          24          25
##   43.330638  146.967354  547.786524  248.266836 -635.908084
##          26          27          28          29          30
##   60.584440  323.059804  194.943913  -92.859005   45.302438
##          31          32          33          34          35
##  166.377859   56.146821   55.249108   41.069857   90.609818
##          36          37          38          39          40
##  -58.645213  103.915631  176.359505   86.198633  214.743192
##          41          42          43          44          45
##  158.229098  176.124349   86.776861   42.581386  162.617002
##          46          47          48          49          50
##  -37.873351 1518.135125   60.539683   47.292754 -446.249417
##          51          52          53          54          55
##  139.628065  180.401070  173.476574  -27.432282   67.917979
##          56          57          58          59          60
##  261.860690  108.698568   93.714134   78.765079   91.864206
##          61          62          63          64          65
##  176.987056  237.156668  132.916309   86.176747   83.555756
##          66          67          68          69          70
##   99.962962  120.121984  108.969245  141.982043  308.890407
##          71          72          73          74          75
##   72.165708  191.651933   53.935974   53.374142   45.000000
##          76          77          78          79          80
##   17.052728  610.521003    2.993294 -117.968933   86.020078
##          81          82          83          84          85
##   49.707812   45.000000   72.054929   26.951953   72.508425
##          86          87          88          89          90
##   63.118462   19.972585  293.420549  188.440453   84.113267
##          91          92          93          94          95
##   36.990462  557.287497   10.549258  -79.431855   58.169079
```

```
##           96           97           98           99          100
##     3.000000   111.868376   119.016156    83.234556   137.451651
##          101          102          103          104          105
##   140.749819    41.884006   117.878564   204.349049   210.570587
##          106          107          108          109          110
##    58.965041    42.190745   131.305629    51.128439   721.619459
##          111          112          113          114          115
##  -908.805442   689.390010   176.844252   214.561153     8.224028
##          116          117          118          119          120
##  -182.201954   173.886579   152.117528  -582.220494    71.032173
##          121          122          123          124          125
##   260.543499   154.403882   176.110138    -6.928850    56.416222
##          126          127          128          129          130
##     3.000000 -1141.648777   189.554390   207.685643    78.421400
##          131          132          133          134          135
##   100.306991   117.432565 -6219.818459    45.000000   221.990983
##          136          137          138          139          140
##    45.000000   148.575299  -192.827045    45.000000    45.000000
##          141          142          143          144          145
##   280.922687  -321.678153    47.032287   461.940734    -7.648674
```

```
RMSE.lin.reg <- sqrt(mean((test.pred.lin-testData$Catch_Rate)^2))
RMSE.lin.reg
```

```
## [1] 581.1608
```

```
MAE.lin.reg <- mean(abs(test.pred.lin-testData$Catch_Rate))
MAE.lin.reg
```

```
## [1] 149.543
```

- Regression Tree

```
library(rpart)
library(rpart.plot)
set.seed(1)
tree <- rpart(Catch_Rate ~Total+Weight_kg +Egg_Group_1, data = pokemon_m, method="anova")

tree2<- rpart(Catch_Rate ~ isLegendary + Type_1 + Total + hasMegaEvolution + Generation + Defense + Col

tree3 <- rpart( Catch_Rate ~ Total + Egg_Group_1 + isLegendary +  Weight_kg + Generation + Type_1 + Def
    data = pokemon_m)

#summary(tree)
printcp(tree) # display the results
```

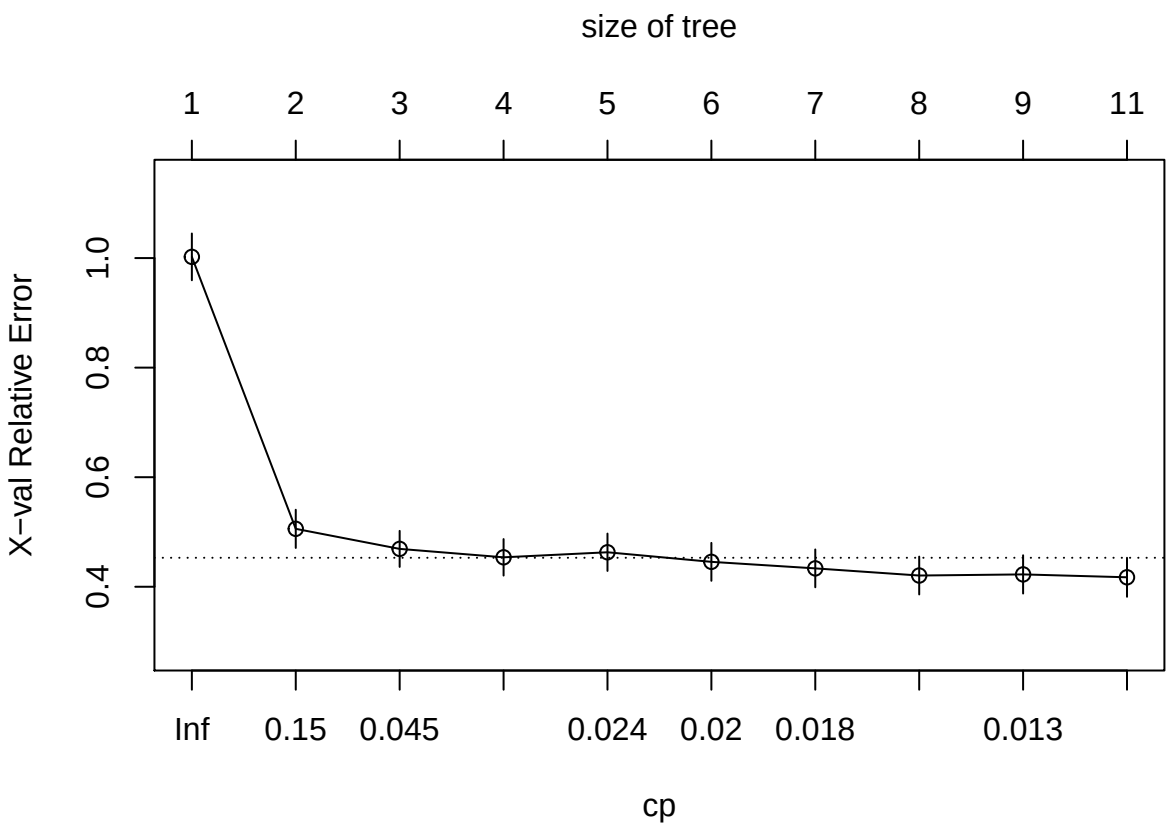
```
##
## Regression tree:
## rpart(formula = Catch_Rate ~ Total + Weight_kg + Egg_Group_1,
##     data = pokemon_m, method = "anova")
##
## Variables actually used in tree construction:
## [1] Egg_Group_1 Total       Weight_kg
##
## Root node error: 4221722/721 = 5855.4
##
## n= 721
```

```
##
##         CP nsplit rel error  xerror     xstd
## 1 0.496939      0   1.00000 1.00061 0.042629
## 2 0.047157      1   0.50306 0.51630 0.035226
## 3 0.042519      2   0.45590 0.48942 0.034162
## 4 0.027637      3   0.41338 0.44867 0.032285
## 5 0.017398      4   0.38575 0.43052 0.031715
## 6 0.016006      5   0.36835 0.42924 0.032115
## 7 0.015675      6   0.35234 0.42671 0.032467
## 8 0.012142      7   0.33667 0.40943 0.032306
## 9 0.010000      9   0.31238 0.39270 0.031278
```

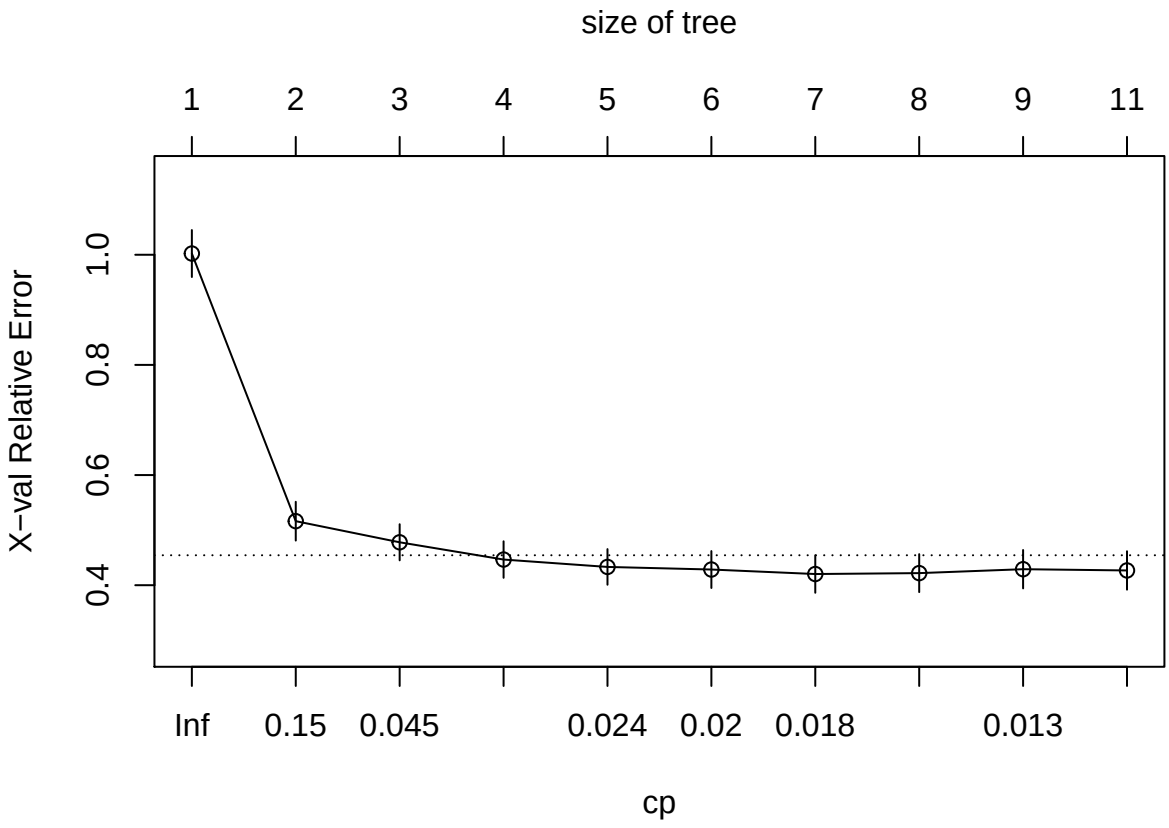
```
printcp(tree) # visualize cross-validation results
```

```
##
## Regression tree:
## rpart(formula = Catch_Rate ~ Total + Weight_kg + Egg_Group_1,
##     data = pokemon_m, method = "anova")
##
## Variables actually used in tree construction:
## [1] Egg_Group_1 Total       Weight_kg
##
## Root node error: 4221722/721 = 5855.4
##
## n= 721
##
##         CP nsplit rel error  xerror     xstd
## 1 0.496939      0   1.00000 1.00061 0.042629
## 2 0.047157      1   0.50306 0.51630 0.035226
## 3 0.042519      2   0.45590 0.48942 0.034162
## 4 0.027637      3   0.41338 0.44867 0.032285
## 5 0.017398      4   0.38575 0.43052 0.031715
## 6 0.016006      5   0.36835 0.42924 0.032115
## 7 0.015675      6   0.35234 0.42671 0.032467
## 8 0.012142      7   0.33667 0.40943 0.032306
## 9 0.010000      9   0.31238 0.39270 0.031278
```

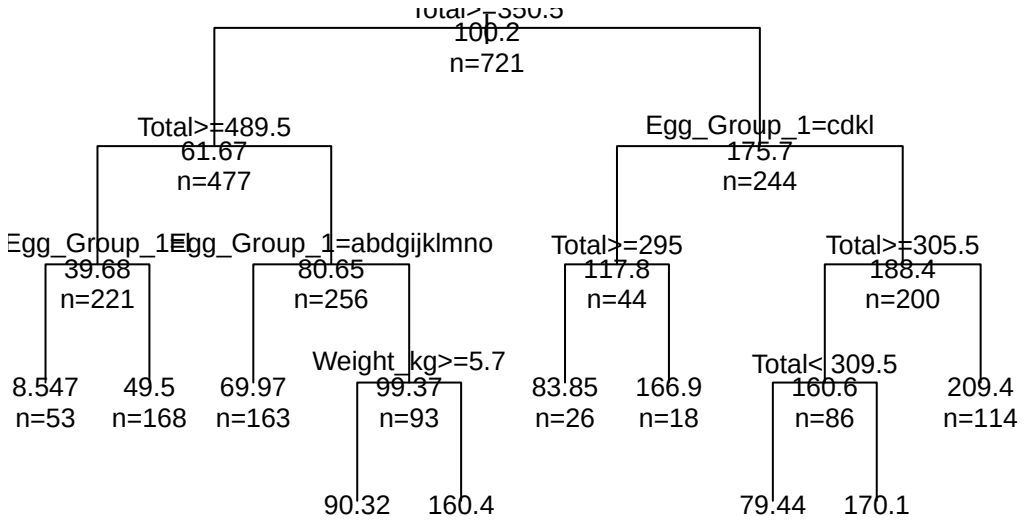
```
#summary(tree2)
plotcp(tree2)
```



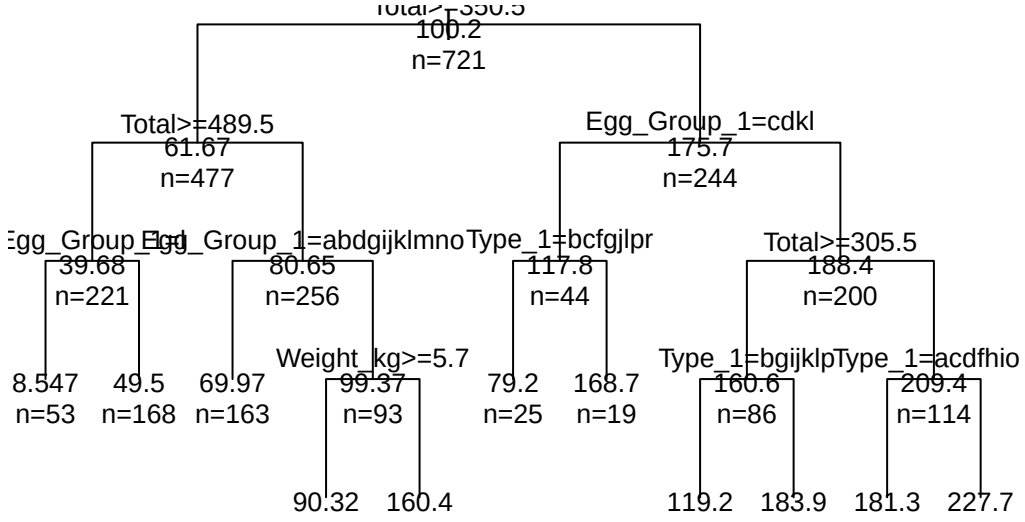

```
#summary(tree3)
plotcp(tree3)
```



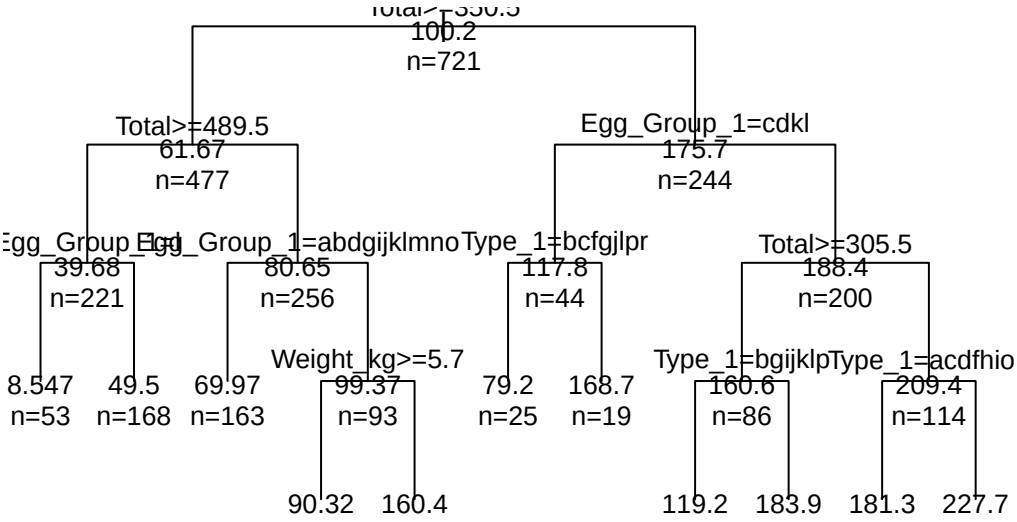
```
# plot tree
plot(tree, uniform=TRUE, main="Regression Tree for catch rate ")
text(tree, use.n=TRUE, all=TRUE, cex=.8)
```




```
plot(tree2, uniform=TRUE, main="Regression Tree for catch rate ")
text(tree2, use.n=TRUE, all=TRUE, cex=.8)
```




```
plot(tree3, uniform=TRUE, main="Regression Tree for catch rate ")
text(tree3,use.n=TRUE, all=TRUE,cex=.8)
```




```
summary(tree3)
```

```
## Call:
## rpart(formula = Catch_Rate ~ Total + Egg_Group_1 + isLegendary +
##     Weight_kg + Generation + Type_1 + Defense + Color + hasMegaEvolution,
##     data = pokemon_m, method = "anova")
##   n= 721
##
##            CP nsplit rel error    xerror       xstd
## 1  0.49693938      0 1.0000000 1.0022316 0.04265813
## 2  0.04715740      1 0.5030606 0.5162477 0.03522533
## 3  0.04251929      2 0.4559032 0.4779241 0.03281770
## 4  0.02763709      3 0.4133839 0.4465848 0.03315623
## 5  0.02047595      4 0.3857468 0.4331731 0.03237365
## 6  0.01966770      5 0.3652709 0.4284493 0.03351063
## 7  0.01600557      6 0.3456032 0.4202908 0.03406352
## 8  0.01388540      7 0.3295976 0.4219365 0.03446653
## 9  0.01214231      8 0.3157122 0.4290574 0.03486185
## 10 0.01000000     10 0.2914276 0.4267827 0.03490950
##
## Variable importance
##       Total     Defense   Weight_kg Egg_Group_1      Type_1 isLegendary
##          45          21          17           8           5           2
##       Color  Generation
##           1           1
##
## Node number 1: 721 observations,    complexity param=0.4969394
##   mean=100.2469, MSE=5855.37
##   left son=2 (477 obs) right son=3 (244 obs)
##   Primary splits:
##       Total       < 350.5 to the right, improve=0.49693940, (0 missing)
##       Weight_kg   < 21.55 to the right, improve=0.23078640, (0 missing)
##       Defense     < 56.5  to the right, improve=0.22515560, (0 missing)
##       isLegendary splits as  RL, improve=0.10195350, (0 missing)
##       Egg_Group_1 splits as  RRLLRRRRRRLLRRR, improve=0.08958795, (0 missing)
```

```
##   Surrogate splits:
##       Defense     < 50.5  to the right, agree=0.829, adj=0.496, (0 split)
##       Weight_kg   < 12.45 to the right, agree=0.792, adj=0.385, (0 split)
##       Egg_Group_1 splits as  LLRLLLLLLLLLLLL, agree=0.663, adj=0.004, (0 split)
##
## Node number 2: 477 observations,    complexity param=0.0471574
##   mean=61.66667, MSE=1930.885
##   left son=4 (221 obs) right son=5 (256 obs)
##   Primary splits:
##       Total       < 489.5 to the right, improve=0.21615480, (0 missing)
##       Egg_Group_1 splits as  RR-RRRRRRRRLRRR, improve=0.16984620, (0 missing)
##       isLegendary splits as  RL, improve=0.16729330, (0 missing)
##       Weight_kg   < 33.7  to the right, improve=0.08309674, (0 missing)
##       Defense     < 86.5  to the right, improve=0.07375903, (0 missing)
##   Surrogate splits:
##       Weight_kg   < 51.25 to the right, agree=0.700, adj=0.353, (0 split)
##       Defense     < 86.5  to the right, agree=0.683, adj=0.317, (0 split)
##       Egg_Group_1 splits as  RR-RRRRRRRLLRRL, agree=0.658, adj=0.262, (0 split)
##       isLegendary splits as  RL, agree=0.633, adj=0.208, (0 split)
##       Generation  splits as  RRRLRL, agree=0.593, adj=0.122, (0 split)
##
## Node number 3: 244 observations,    complexity param=0.04251929
##   mean=175.668, MSE=4929.304
##   left son=6 (44 obs) right son=7 (200 obs)
##   Primary splits:
##       Egg_Group_1 splits as  RRLLRRRRRRLLRRR, improve=0.14924520, (0 missing)
##       Total       < 299.5 to the right, improve=0.12790120, (0 missing)
##       Type_1      splits as  RRLRRRLRRRRRRRRRRR, improve=0.09248264, (0 missing)
##       Color       splits as  RLRLRRRRRR, improve=0.04569173, (0 missing)
##       Weight_kg   < 5.85  to the right, improve=0.03729783, (0 missing)
##   Surrogate splits:
##       Type_1    splits as  RRLRRRRRRRRRRRRLRR, agree=0.836, adj=0.091, (0 split)
##       Defense   < 17.5  to the left,  agree=0.832, adj=0.068, (0 split)
##       Weight_kg < 97.35 to the right, agree=0.828, adj=0.045, (0 split)
##
## Node number 4: 221 observations,    complexity param=0.01600557
##   mean=39.67873, MSE=598.4805
##   left son=8 (53 obs) right son=9 (168 obs)
##   Primary splits:
##       Egg_Group_1 splits as  RR-RRRRRRRRLRRR, improve=0.51087950, (0 missing)
##       Total       < 573.5 to the right, improve=0.48977230, (0 missing)
##       isLegendary splits as  RL, improve=0.47906720, (0 missing)
##       Weight_kg   < 168.5 to the right, improve=0.11609370, (0 missing)
##       Type_1      splits as  RRLLRRRLRRRLLRLRLR, improve=0.09900055, (0 missing)
##   Surrogate splits:
##       isLegendary splits as  RL, agree=0.968, adj=0.868, (0 split)
##       Total       < 573.5 to the right, agree=0.955, adj=0.811, (0 split)
##       Weight_kg   < 9.25  to the left,  agree=0.792, adj=0.132, (0 split)
##       Type_1      splits as  RRLRRRRRRRRRRLRRR, agree=0.783, adj=0.094, (0 split)
##
## Node number 5: 256 observations,    complexity param=0.01214231
##   mean=80.64844, MSE=2303.447
##   left son=10 (163 obs) right son=11 (93 obs)
##   Primary splits:
```

```
##       Egg_Group_1 splits as  LL-LRRLRLLLLLLL, improve=0.08677543, (0 missing)
##       Type_1      splits as  LLLRRLR-LRLLRRLLLL, improve=0.05957008, (0 missing)
##       Color       splits as  LLRRRLLLRR, improve=0.04883114, (0 missing)
##       Generation  splits as  LLRLRR, improve=0.04594105, (0 missing)
##       Weight_kg   < 9.15  to the right, improve=0.03745817, (0 missing)
##   Surrogate splits:
##       Type_1     splits as  LLLRRLR-LRLLRLLLLL, agree=0.777, adj=0.387, (0 split)
##       Color      splits as  LLLLLLLLRL, agree=0.656, adj=0.054, (0 split)
##       Total      < 486   to the left,  agree=0.645, adj=0.022, (0 split)
##       Weight_kg  < 10.7  to the right, agree=0.645, adj=0.022, (0 split)
##       Generation splits as  LLLLLR, agree=0.645, adj=0.022, (0 split)
##
## Node number 6: 44 observations,    complexity param=0.02047595
##   mean=117.8409, MSE=5504.997
##   left son=12 (25 obs) right son=13 (19 obs)
##   Primary splits:
##       Type_1      splits as  -LLRRLL--LRLRRRLRL, improve=0.35688140, (0 missing)
##       Total       < 295   to the right, improve=0.30322680, (0 missing)
##       Color       splits as  RLRLLRLLRR, improve=0.29264300, (0 missing)
##       Egg_Group_1 splits as  --LL------LR---, improve=0.07833800, (0 missing)
##       Weight_kg   < 16.8  to the right, improve=0.07688658, (0 missing)
##   Surrogate splits:
##       Total       < 289.5 to the right, agree=0.818, adj=0.579, (0 split)
##       Color       splits as  RLLLRRLLRR, agree=0.773, adj=0.474, (0 split)
##       Egg_Group_1 splits as  --RL------LR---, agree=0.682, adj=0.263, (0 split)
##       Weight_kg   < 4.5   to the right, agree=0.682, adj=0.263, (0 split)
##       Defense     < 34    to the right, agree=0.682, adj=0.263, (0 split)
##
## Node number 7: 200 observations,    complexity param=0.02763709
##   mean=188.39, MSE=3905.128
##   left son=14 (86 obs) right son=15 (114 obs)
##   Primary splits:
##       Total      < 305.5 to the right, improve=0.14938840, (0 missing)
##       Type_1     splits as  RRLRRRLRRRRRRRRRLR, improve=0.09430368, (0 missing)
##       Defense    < 39.5  to the right, improve=0.05758924, (0 missing)
##       Weight_kg  < 5.85  to the right, improve=0.03155390, (0 missing)
##       Generation splits as  RLRLRL, improve=0.02422296, (0 missing)
##   Surrogate splits:
##       Type_1      splits as  RLRRLRLRRRRRRLLRRL, agree=0.650, adj=0.186, (0 split)
##       Defense     < 41.5  to the right, agree=0.630, adj=0.140, (0 split)
##       Weight_kg   < 26    to the right, agree=0.620, adj=0.116, (0 split)
##       Generation  splits as  RRRLRL, agree=0.615, adj=0.105, (0 split)
##       Egg_Group_1 splits as  RR--RLRRRR--RLL, agree=0.610, adj=0.093, (0 split)
##
## Node number 8: 53 observations
##   mean=8.54717, MSE=202.21
##
## Node number 9: 168 observations
##   mean=49.5, MSE=321.2857
##
## Node number 10: 163 observations
##   mean=69.96933, MSE=1333.208
##
## Node number 11: 93 observations,    complexity param=0.01214231
```

```
##   mean=99.36559, MSE=3453.759
##   left son=22 (81 obs) right son=23 (12 obs)
##   Primary splits:
##       Weight_kg  < 5.7   to the right, improve=0.15987880, (0 missing)
##       Total      < 463   to the right, improve=0.10788500, (0 missing)
##       Color      splits as  LLRRRRLLRR, improve=0.08206405, (0 missing)
##       Type_1     splits as  -L-RLLL-LLLLLLL-LL, improve=0.07693620, (0 missing)
##       Generation splits as  LLRRRR, improve=0.03559239, (0 missing)
##   Surrogate splits:
##       Type_1 splits as  -L-RLLL-LLLLLLL-LL, agree=0.892, adj=0.167, (0 split)
##
## Node number 12: 25 observations
##   mean=79.2, MSE=4389.36
##
## Node number 13: 19 observations
##   mean=168.6842, MSE=2423.269
##
## Node number 14: 86 observations,    complexity param=0.0196677
##   mean=160.5814, MSE=3763.615
##   left son=28 (31 obs) right son=29 (55 obs)
##   Primary splits:
##       Type_1      splits as  RL-RRRL-LLLLRRRLRR, improve=0.25653070, (0 missing)
##       Total       < 309.5 to the left,  improve=0.20444850, (0 missing)
##       Egg_Group_1 splits as  RR--RLRRRR--RRR, improve=0.10557130, (0 missing)
##       Defense     < 54    to the left,  improve=0.07249819, (0 missing)
##       Color       splits as  RRLLRRRLRR, improve=0.07243646, (0 missing)
##   Surrogate splits:
##       Egg_Group_1 splits as  RR--RLRLRR--RRR, agree=0.756, adj=0.323, (0 split)
##       Total       < 309.5 to the left,  agree=0.698, adj=0.161, (0 split)
##       Color       splits as  RRRLRRRLRR, agree=0.698, adj=0.161, (0 split)
##       Generation  splits as  RLRRRR, agree=0.663, adj=0.065, (0 split)
##       Defense     < 34.5  to the left,  agree=0.663, adj=0.065, (0 split)
##
## Node number 15: 114 observations,    complexity param=0.0138854
##   mean=209.3684, MSE=2988.408
##   left son=30 (45 obs) right son=31 (69 obs)
##   Primary splits:
##       Type_1      splits as  LRLLRLRLLRRRRRLRLR, improve=0.17206920, (0 missing)
##       Color       splits as  LLRLLLLRLL, improve=0.09777059, (0 missing)
##       Egg_Group_1 splits as  LL--RRRRLL--RRL, improve=0.07361456, (0 missing)
##       Defense     < 41.5  to the right, improve=0.05420775, (0 missing)
##       Weight_kg   < 32.5  to the right, improve=0.05397407, (0 missing)
##   Surrogate splits:
##       Egg_Group_1 splits as  LL--RRRRLR--RRR, agree=0.851, adj=0.622, (0 split)
##       Color       splits as  LRRRRRRLRR, agree=0.675, adj=0.178, (0 split)
##       Total       < 207.5 to the left,  agree=0.667, adj=0.156, (0 split)
##       Defense     < 54    to the right, agree=0.658, adj=0.133, (0 split)
##       Weight_kg   < 26    to the right, agree=0.623, adj=0.044, (0 split)
##
## Node number 22: 81 observations
##   mean=90.32099, MSE=2772.514
##
## Node number 23: 12 observations
##   mean=160.4167, MSE=3772.743
```

```
##
## Node number 28: 31 observations
##   mean=119.1935, MSE=3259.834
##
## Node number 29: 55 observations
##   mean=183.9091, MSE=2537.901
##
## Node number 30: 45 observations
##   mean=181.2889, MSE=3769.539
##
## Node number 31: 69 observations
##   mean=227.6812, MSE=1629.406
```

```
# Pick tree3 by cross validation (forward method)
```

- Predict the catch rate, which variables are important to predict the catch rate. - regression tree.
- selecting variables by regression tree.

```
library(caret)
```

```
## Loading required package: lattice
```

```
reg0=glm(Catch_Rate ~1,data=pokemon_m)
summary(reg0)
```

```
##
## Call:
## glm(formula = Catch_Rate ~ 1, data = pokemon_m)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -97.25  -55.25  -35.25   79.75  154.75
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  100.247      2.852   35.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 5863.503)
##
##     Null deviance: 4221722  on 720  degrees of freedom
## Residual deviance: 4221722  on 720  degrees of freedom
## AIC: 8304.9
##
## Number of Fisher Scoring iterations: 2
```

```
reg1=glm(Catch_Rate ~isLegendary + Type_1 + Total + hasMegaEvolution + HP + Attack +Defense +Sp_Atk +Sp
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared
## on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 17 in
## model.matrix: no columns are assigned
```

```
summary(reg1)
```

```
##
```

```
## Call:
## glm(formula = Catch_Rate ~ isLegendary + Type_1 + Total + hasMegaEvolution +
##     HP + Attack + Defense + Sp_Atk + Sp_Def + Speed + Generation +
##     Color + hasGender + Egg_Group_1 + hasMegaEvolution + Height_m +
##     Weight_kg + Catch_Rate + Body_Style, data = inputData)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -148.158   -28.533    -0.458    26.481   148.126
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                270.29360   30.87069   8.756  < 2e-16 ***
## isLegendaryTrue             34.56664   17.85384   1.936 0.053413 .
## Type_1Dark                  17.66778   22.46140   0.787 0.431894
## Type_1Dragon                 0.90540   23.48562   0.039 0.969263
## Type_1Electric              32.70148   21.88849   1.494 0.135798
## Type_1Fairy                 14.94904   25.85309   0.578 0.563366
## Type_1Fighting              14.53667   23.59787   0.616 0.538161
## Type_1Fire                  -9.14503   21.69188  -0.422 0.673505
## Type_1Flying                -2.85619   41.12614  -0.069 0.944659
## Type_1Ghost                 48.73875   25.67731   1.898 0.058248 .
## Type_1Grass                 -8.27856   22.52216  -0.368 0.713345
## Type_1Ground                 7.22522   20.12393   0.359 0.719718
## Type_1Ice                    9.75561   22.72041   0.429 0.667832
## Type_1Normal                25.97430   20.51917   1.266 0.206147
## Type_1Poison                54.32642   21.44574   2.533 0.011603 *
## Type_1Psychic               34.55913   21.85103   1.582 0.114371
## Type_1Rock                 -11.84706   21.60387  -0.548 0.583675
## Type_1Steel                -13.18511   22.91326  -0.575 0.565253
## Type_1Water                 20.44688   20.32065   1.006 0.314795
## Total                       -0.73926    0.10478  -7.056 5.68e-12 ***
## hasMegaEvolutionTrue        17.01259    9.37059   1.816 0.070034 .
## HP                           0.22954    0.15280   1.502 0.133665
## Attack                       0.13968    0.17046   0.819 0.412940
## Defense                      0.24005    0.13956   1.720 0.086035 .
## Sp_Atk                       0.20104    0.17626   1.141 0.254575
## Sp_Def                       0.34768    0.16309   2.132 0.033501 *
## Speed                            NA         NA      NA       NA
## Generation2                -11.33320    7.74530  -1.463 0.144024
## Generation3                  9.85916    7.18658   1.372 0.170707
## Generation4                  0.15527    7.46175   0.021 0.983406
## Generation5                 10.77463    7.28091   1.480 0.139537
## Generation6                 12.96131    8.91246   1.454 0.146486
## ColorBlue                   -2.31168   12.33943  -0.187 0.851469
## ColorBrown                  26.84169   12.54157   2.140 0.032815 *
## ColorGreen                  25.28416   13.36574   1.892 0.059100 .
## ColorGrey                   15.16068   13.16834   1.151 0.250153
## ColorPink                   -2.34095   15.21760  -0.154 0.877804
## ColorPurple                  9.79889   12.77821   0.767 0.443532
## ColorRed                    18.32103   13.34929   1.372 0.170536
## ColorWhite                  12.64223   13.59289   0.930 0.352782
## ColorYellow                 16.90852   13.19426   1.282 0.200603
## hasGenderTrue               -6.73143   12.09386  -0.557 0.578048
```

```
## Egg_Group_1Bug               29.78101   23.17944   1.285 0.199449
## Egg_Group_1Ditto           -114.83219   52.52225  -2.186 0.029247 *
## Egg_Group_1Dragon           -11.31084   24.98303  -0.453 0.650929
## Egg_Group_1Fairy             64.15650   18.04451   3.555 0.000413 ***
## Egg_Group_1Field             28.29072   14.86366   1.903 0.057562 .
## Egg_Group_1Flying            36.66733   19.00934   1.929 0.054301 .
## Egg_Group_1Grass             78.31433   19.67282   3.981 7.87e-05 ***
## Egg_Group_1Human-Like        26.35603   16.91880   1.558 0.119907
## Egg_Group_1Mineral           33.97311   17.22724   1.972 0.049147 *
## Egg_Group_1Monster            7.47578   15.98438   0.468 0.640205
## Egg_Group_1Undiscovered       4.05454   17.52930   0.231 0.817174
## Egg_Group_1Water_1           28.87558   15.71269   1.838 0.066689 .
## Egg_Group_1Water_2           37.53836   23.09395   1.625 0.104687
## Egg_Group_1Water_3           36.49344   21.02629   1.736 0.083242 .
## Height_m                     -1.36157    3.05762  -0.445 0.656291
## Weight_kg                     0.06641    0.03611   1.839 0.066513 .
## Body_Stylebipedal_tailless   -8.94725    8.53616  -1.048 0.295066
## Body_Stylefour_wings        -19.74093   17.28583  -1.142 0.253982
## Body_Stylehead_arms          -0.85929   12.99942  -0.066 0.947322
## Body_Stylehead_base         -12.41350   13.32105  -0.932 0.351848
## Body_Stylehead_legs          36.27195   15.37427   2.359 0.018690 *
## Body_Stylehead_only           1.23034   13.07104   0.094 0.925045
## Body_Styleinsectoid          25.51048   13.72031   1.859 0.063562 .
## Body_Stylemultiple_bodies   -23.77800   17.27462  -1.376 0.169285
## Body_Stylequadruped          10.94467    7.10904   1.540 0.124297
## Body_Styleserpentine_body    -1.95905   13.56911  -0.144 0.885261
## Body_Styleseveral_limbs      15.96979   17.96898   0.889 0.374564
## Body_Styletwo_wings          -2.42524   12.77800  -0.190 0.849544
## Body_Stylewith_fins          16.93319   14.63862   1.157 0.247921
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2316.015)
##
##     Null deviance: 3352708  on 575  degrees of freedom
## Residual deviance: 1171903  on 506  degrees of freedom
## AIC: 6164.6
##
## Number of Fisher Scoring iterations: 2
```

```
step(reg0,scope=formula(reg1),direction="forward",k=2)
```

```
## Start:  AIC=8304.87
## Catch_Rate ~ 1
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##                    Df Deviance    AIC
## + Total             1  1920644 7739.0
## + Sp_Atk            1  2994706 8059.3
## + Attack            1  3057638 8074.3
## + Sp_Def            1  3110634 8086.7
```

```
## + HP                1  3254198 8119.2
## + Defense           1  3417136 8154.4
## + Speed             1  3510119 8173.8
## + Height_m          1  3602886 8192.6
## + Weight_kg         1  3650627 8202.1
## + isLegendary       1  3791303 8229.3
## + Egg_Group_1      14  3758521 8249.1
## + hasGender         1  3908684 8251.3
## + Body_Style       13  3915799 8276.6
## + Type_1           17  3900380 8281.8
## + hasMegaEvolution  1  4094971 8284.9
## + Generation        5  4136015 8300.1
## + Color             9  4110967 8303.7
## <none>                 4221722 8304.9
##
## Step:  AIC=7739.02
## Catch_Rate ~ Total
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##                    Df Deviance    AIC
## + Egg_Group_1      14  1792199 7717.1
## + Generation        5  1881641 7734.2
## + Body_Style       13  1842583 7735.1
## + Color             9  1867565 7736.8
## + isLegendary       1  1913347 7738.3
## <none>                 1920644 7739.0
## + Weight_kg         1  1916035 7739.3
## + hasGender         1  1919849 7740.7
## + Defense           1  1919878 7740.7
## + Sp_Def            1  1920023 7740.8
## + Attack            1  1920414 7740.9
## + Height_m          1  1920428 7740.9
## + Sp_Atk            1  1920440 7740.9
## + Speed             1  1920472 7741.0
## + HP                1  1920511 7741.0
## + hasMegaEvolution  1  1920550 7741.0
## + Type_1           17  1839923 7742.1
##
## Step:  AIC=7717.11
## Catch_Rate ~ Total + Egg_Group_1
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##                    Df Deviance    AIC
## + isLegendary       1  1764364 7707.8
## + Weight_kg         1  1774418 7711.9
## + Generation        5  1757315 7712.9
```

```
## + Body_Style       13  1719847 7713.4
## + Defense           1  1785535 7716.4
## + hasGender         1  1786337 7716.8
## <none>                 1792199 7717.1
## + Speed             1  1787725 7717.3
## + Sp_Atk            1  1790581 7718.5
## + Height_m          1  1790773 7718.5
## + Sp_Def            1  1791250 7718.7
## + HP                1  1791597 7718.9
## + hasMegaEvolution  1  1791621 7718.9
## + Attack            1  1791945 7719.0
## + Color             9  1755023 7720.0
## + Type_1           17  1727508 7724.6
##
## Step:  AIC=7707.83
## Catch_Rate ~ Total + Egg_Group_1 + isLegendary

## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped

## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned

##                    Df Deviance    AIC
## + Weight_kg         1  1753337 7705.3
## + Generation        5  1737304 7706.7
## + Body_Style       13  1699940 7707.0
## + Defense           1  1758484 7707.4
## + Speed             1  1759206 7707.7
## <none>                 1764364 7707.8
## + Sp_Atk            1  1762075 7708.9
## + Sp_Def            1  1762752 7709.2
## + hasMegaEvolution  1  1763553 7709.5
## + Height_m          1  1763930 7709.7
## + hasGender         1  1764212 7709.8
## + Attack            1  1764274 7709.8
## + HP                1  1764293 7709.8
## + Type_1           17  1690422 7711.0
## + Color             9  1730379 7711.8
##
## Step:  AIC=7705.31
## Catch_Rate ~ Total + Egg_Group_1 + isLegendary + Weight_kg

## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped

## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned

##                    Df Deviance    AIC
## + Generation        5  1725566 7703.8
## + Type_1           17  1672171 7705.1
## <none>                 1753337 7705.3
## + Body_Style       13  1691536 7705.4
## + Defense           1  1749847 7705.9
## + Sp_Def            1  1750670 7706.2
```

```
## + Speed             1  1751116 7706.4
## + Height_m          1  1751787 7706.7
## + HP                1  1752434 7706.9
## + Sp_Atk            1  1752510 7707.0
## + hasMegaEvolution  1  1752583 7707.0
## + hasGender         1  1753155 7707.2
## + Attack            1  1753285 7707.3
## + Color             9  1719176 7709.1
##
## Step:  AIC=7703.8
## Catch_Rate ~ Total + Egg_Group_1 + isLegendary + Weight_kg +
##     Generation
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##                    Df Deviance    AIC
## + Type_1           17  1638763 7700.6
## + Sp_Def            1  1720632 7703.7
## <none>                 1725566 7703.8
## + Defense          1  1721661 7704.2
## + Speed            1  1722590 7704.6
## + Body_Style      13  1666750 7704.8
## + Height_m         1  1724212 7705.2
## + Sp_Atk           1  1724306 7705.3
## + hasMegaEvolution  1  1724807 7705.5
## + HP               1  1724990 7705.6
## + Attack           1  1725157 7705.6
## + hasGender        1  1725300 7705.7
## + Color            9  1688326 7706.1
##
## Step:  AIC=7700.58
## Catch_Rate ~ Total + Egg_Group_1 + isLegendary + Weight_kg +
##     Generation + Type_1
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##                    Df Deviance    AIC
## + Defense          1  1624844 7696.4
## + Color            9  1593150 7698.2
## + Speed            1  1631034 7699.2
## + HP               1  1634135 7700.5
## + Sp_Def           1  1634201 7700.6
## <none>                1638763 7700.6
## + hasMegaEvolution  1  1635471 7701.1
## + Sp_Atk           1  1635666 7701.2
## + Height_m         1  1636666 7701.7
## + Body_Style      13  1584486 7702.3
## + Attack           1  1638518 7702.5
```

```
## + hasGender         1  1638748 7702.6
##
## Step:  AIC=7696.43
## Catch_Rate ~ Total + Egg_Group_1 + isLegendary + Weight_kg +
##     Generation + Type_1 + Defense
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##                     Df Deviance    AIC
## + Color              9  1582358 7695.3
## <none>                  1624844 7696.4
## + hasMegaEvolution   1  1621042 7696.7
## + Sp_Def             1  1621874 7697.1
## + Speed              1  1623296 7697.7
## + Height_m           1  1623379 7697.8
## + HP                 1  1623480 7697.8
## + Attack             1  1624265 7698.2
## + Sp_Atk             1  1624807 7698.4
## + hasGender          1  1624811 7698.4
## + Body_Style        13  1577235 7701.0
##
## Step:  AIC=7695.33
## Catch_Rate ~ Total + Egg_Group_1 + isLegendary + Weight_kg +
##     Generation + Type_1 + Defense + Color
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##                     Df Deviance    AIC
## + hasMegaEvolution   1  1577702 7695.2
## <none>                  1582358 7695.3
## + Speed              1  1578853 7695.7
## + Sp_Def             1  1579570 7696.1
## + Height_m           1  1581306 7696.9
## + hasGender          1  1582248 7697.3
## + HP                 1  1582312 7697.3
## + Attack             1  1582318 7697.3
## + Sp_Atk             1  1582339 7697.3
## + Body_Style        13  1534500 7699.2
##
## Step:  AIC=7695.21
## Catch_Rate ~ Total + Egg_Group_1 + isLegendary + Weight_kg +
##     Generation + Type_1 + Defense + Color + hasMegaEvolution
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): the response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object
## $contrasts): problem with term 17 in model.matrix: no columns are assigned
```

```
##              Df Deviance    AIC
## <none>           1577702 7695.2
## + Speed       1  1573952 7695.5
## + Sp_Def      1  1574829 7695.9
## + Height_m    1  1576346 7696.6
## + hasGender   1  1577403 7697.1
## + Attack      1  1577695 7697.2
## + Sp_Atk      1  1577697 7697.2
## + HP          1  1577702 7697.2
## + Body_Style 13  1528344 7698.3
```

```
##
## Call:  glm(formula = Catch_Rate ~ Total + Egg_Group_1 + isLegendary +
##     Weight_kg + Generation + Type_1 + Defense + Color + hasMegaEvolution,
##     data = pokemon_m)
##
## Coefficients:
##             (Intercept)                   Total           Egg_Group_1Bug
##               295.30678                -0.58863                12.33418
##        Egg_Group_1Ditto        Egg_Group_1Dragon         Egg_Group_1Fairy
##              -122.81232                -17.47939                39.16783
##        Egg_Group_1Field        Egg_Group_1Flying         Egg_Group_1Grass
##                16.51780                14.30235                40.52211
##   Egg_Group_1Human-Like       Egg_Group_1Mineral       Egg_Group_1Monster
##                 1.30733                14.28510                -8.51104
## Egg_Group_1Undiscovered       Egg_Group_1Water_1       Egg_Group_1Water_2
##               -13.62049                15.17999                28.11618
##      Egg_Group_1Water_3          isLegendaryTrue               Weight_kg
##                15.66040                37.36531                 0.07193
##             Generation2              Generation3              Generation4
##               -11.27568                 7.63907                -3.02282
##             Generation5              Generation6               Type_1Dark
##                10.83970                 9.32368                10.59729
##            Type_1Dragon           Type_1Electric              Type_1Fairy
##                -0.38965                28.12334                25.88303
##          Type_1Fighting              Type_1Fire             Type_1Flying
##                11.61346                -6.31063                 3.57749
##             Type_1Ghost              Type_1Grass             Type_1Ground
##                17.39704                 2.64692                 8.96040
##               Type_1Ice             Type_1Normal             Type_1Poison
##                 8.48786                20.16227                46.18782
##           Type_1Psychic              Type_1Rock              Type_1Steel
##                28.81990               -16.30070               -20.59420
##             Type_1Water                 Defense                ColorBlue
##                20.55912                 0.20589                -9.30006
##              ColorBrown              ColorGreen               ColorGrey
##                18.76517                 7.34076                 7.01101
##               ColorPink             ColorPurple                ColorRed
##                -3.32418                 1.69848                 8.17876
##              ColorWhite             ColorYellow     hasMegaEvolutionTrue
##                 4.75578                15.34485                11.48419
##
## Degrees of Freedom: 720 Total (i.e. Null);  670 Residual
## Null Deviance:       4222000
```

```
## Residual Deviance: 1578000   AIC: 7695
```

```
strongest_catch_rate <- pokemon_m %>% group_by(Catch_Rate) %>% filter(Catch_Rate==3)
strongest_catch_rate$Name
```

```
##  [1] "Articuno"  "Zapdos"    "Moltres"   "Mewtwo"    "Raikou"
##  [6] "Entei"     "Suicune"   "Lugia"     "Ho-Oh"     "Beldum"
## [11] "Metang"    "Metagross" "Regirock"  "Regice"    "Registeel"
## [16] "Latias"    "Latios"    "Kyogre"    "Groudon"   "Jirachi"
## [21] "Deoxys"    "Uxie"      "Mesprit"   "Azelf"     "Dialga"
## [26] "Palkia"    "Heatran"   "Regigigas" "Giratina"  "Cresselia"
## [31] "Manaphy"   "Darkrai"   "Arceus"    "Victini"   "Cobalion"
## [36] "Terrakion" "Virizion"  "Tornadus"  "Thundurus" "Reshiram"
## [41] "Zekrom"    "Landorus"  "Kyurem"    "Keldeo"    "Meloetta"
## [46] "Genesect"  "Zygarde"   "Diancie"   "Hoopa"     "Volcanion"
```

Predict

```
tree3 <- rpart(Catch_Rate ~Total+Egg_Group_1, data= inputData, method="anova")


 test.pred.rtree <- predict(tree3,testData)
 RMSE.rtree <- sqrt(mean((test.pred.rtree-testData$Catch_Rate)^2))
 RMSE.rtree
```

```
## [1] 43.23274
```

```
 MAE.rtree <- mean(abs(test.pred.rtree-testData$Catch_Rate))
 MAE.rtree
```

```
## [1] 31.39361
```

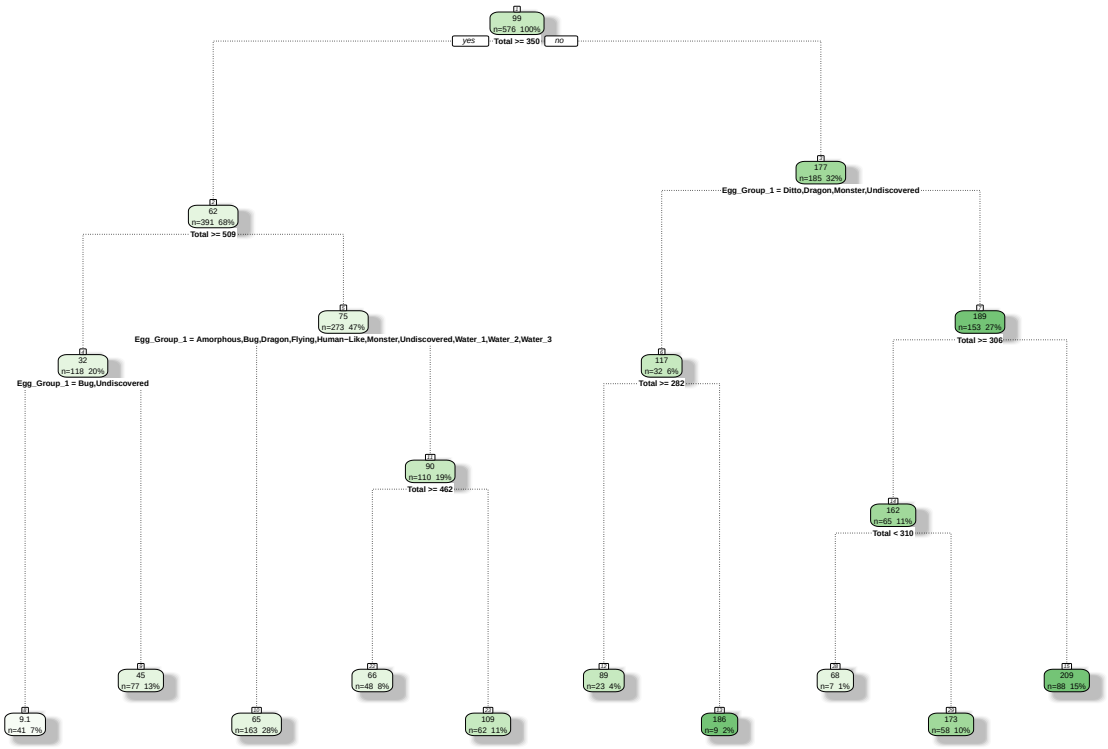
```
 min.xerror <- tree3$cptable[which.min(tree3$cptable[,"xerror"]),"CP"]
 min.xerror
```

```
## [1] 0.01
```

```
 tree3.pruned <- prune(tree3,cp = min.xerror)
 library(rattle)
```

```
## Warning: Failed to load RGtk2 dynamic library, attempting to install it.

## Please install GTK+ from http://r.research.att.com/libs/GTK_2.24.17-X11.pkg

## If the package still does not load, please ensure that GTK+ is installed and that it is on your PATH

## IN ANY CASE, RESTART R BEFORE TRYING TO LOAD THE PACKAGE AGAIN

## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
 fancyRpartPlot(tree3.pruned)
```



Rattle 2018−Jan−09 01:10:55 youheekil

```
 test.pred.rtree.p <- predict(tree3.pruned,testData)
 RMSE.rtree.pruned <- sqrt(mean((test.pred.rtree.p-testData$Catch_Rate)^2))
 RMSE.rtree.pruned
```

```
## [1] 43.23274
```

```
 MAE.rtree.pruned <- mean(abs(test.pred.rtree.p-testData$Catch_Rate))
MAE.rtree.pruned
```

```
## [1] 31.39361
```

```
best.guess <- mean(inputData$Catch_Rate)
RMSE.baseline <- sqrt(mean((best.guess-testData$Catch_Rate)^2))
RMSE.baseline
```

```
## [1] 77.46737
```

```
MAE.baseline <- mean(abs(best.guess-testData$Catch_Rate))
MAE.baseline
```

```
## [1] 66.94239
```

```
# Pruned one and not pruned one are exactly same RMSE
# In the final tree, only Total, Egg group 1 are considered relevant to predict the catch rate, and the

#When total exceeds 509 and egg group is Bug, Undiscovered (7%), then predict Catch rate 9.1.
#When total is lower than 310 and egg group is Ditto, Dragon, Monster, Undiscoverd (1%), then we predic
```