Pokémon Data Analysis

Christian Stochholm

2023-11-27

Introduction

Pokemon is a game that has been built on for more than 20 years and therefore have a lot of data on individuals Pokemon. In this project I will dive into different aspect of Pokémon through visually presenting the data.

Project Setup

The file for the project was retrieved here: GitRepo

Install tidyverse package (incl. ggplot2 for more graphical plots)

```r
#install.packages("tidyverse")
library(tidyverse)
```

```
## Warning: pakke ’tidyverse’ blev bygget under R version 4.3.2

## Warning: pakke ’ggplot2’ blev bygget under R version 4.3.2

## Warning: pakke ’tibble’ blev bygget under R version 4.3.2

## Warning: pakke ’tidyr’ blev bygget under R version 4.3.2

## Warning: pakke ’readr’ blev bygget under R version 4.3.2

## Warning: pakke ’purrr’ blev bygget under R version 4.3.2

## Warning: pakke ’dplyr’ blev bygget under R version 4.3.2

## Warning: pakke ’stringr’ blev bygget under R version 4.3.2

## Warning: pakke ’forcats’ blev bygget under R version 4.3.2

## Warning: pakke ’lubridate’ blev bygget under R version 4.3.2
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

Reads the Pokemon.csv file into the data frame Pokemon.

```
Pokemon <- readr::read_csv("pokemon.csv")
```

```
## Rows: 800 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Name, Type 1, Type 2
## dbl (9): #, Total, HP, Attack, Defense, Sp. Atk, Sp. Def, Speed, Generation
## lgl (1): Legendary
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Dataframe General Information

The data frame includes each Pokemon number, name, type, stats, generation and whether they are legendary or not.

Data frame column names & summary

```
colnames(Pokemon)
```

```
##  [1] "#"          "Name"       "Type 1"     "Type 2"     "Total"
##  [6] "HP"         "Attack"     "Defense"    "Sp. Atk"    "Sp. Def"
## [11] "Speed"      "Generation" "Legendary"
```

```
summary(Pokemon)
```

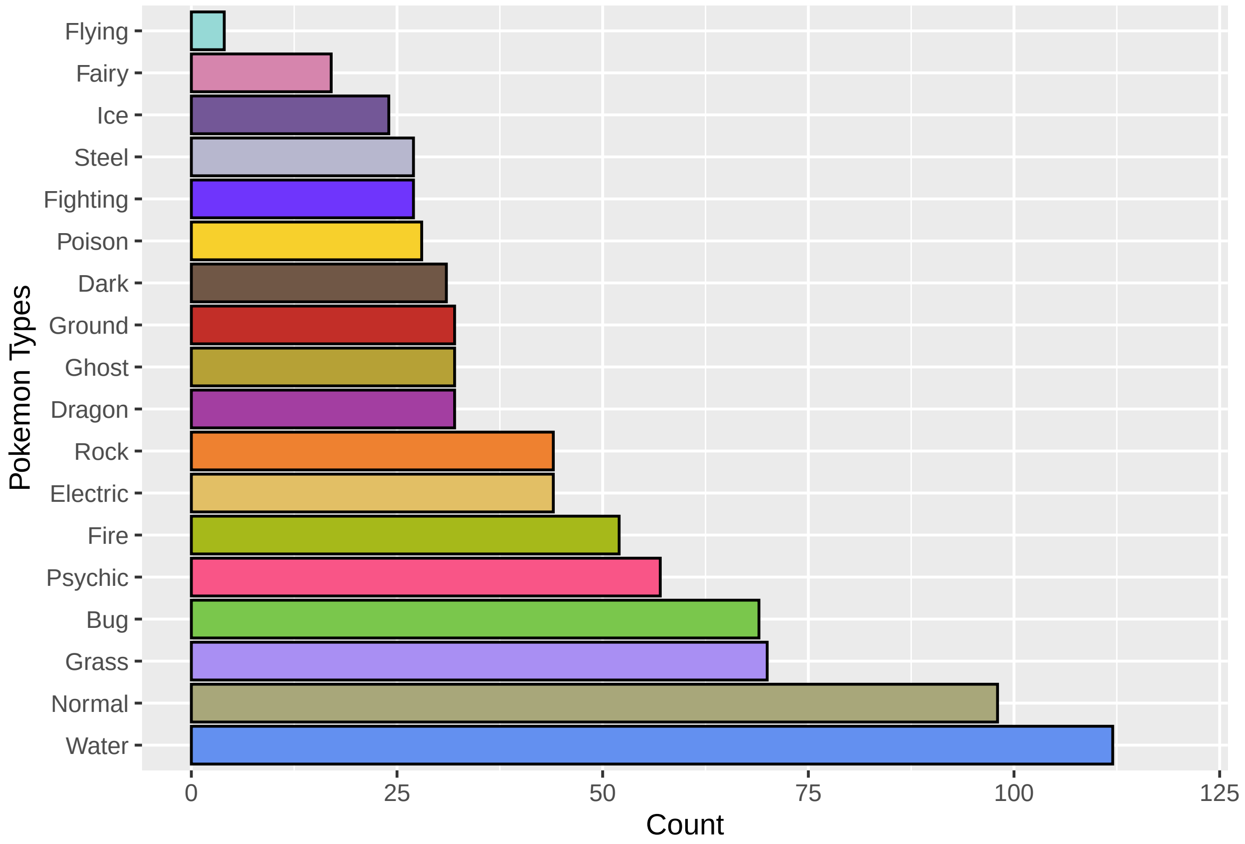
```
##        #              Name              Type 1             Type 2
##  Min.   :  1.0   Length:800         Length:800         Length:800
##  1st Qu.:184.8   Class :character   Class :character   Class :character
##  Median :364.5   Mode  :character   Mode  :character   Mode  :character
##  Mean   :362.8
##  3rd Qu.:539.2
##  Max.   :721.0
##      Total             HP             Attack       Defense
##  Min.   :180.0   Min.   :  1.00   Min.   :  5   Min.   :  5.00
##  1st Qu.:330.0   1st Qu.: 50.00   1st Qu.: 55   1st Qu.: 50.00
##  Median :450.0   Median : 65.00   Median : 75   Median : 70.00
##  Mean   :435.1   Mean   : 69.26   Mean   : 79   Mean   : 73.84
##  3rd Qu.:515.0   3rd Qu.: 80.00   3rd Qu.:100   3rd Qu.: 90.00
##  Max.   :780.0   Max.   :255.00   Max.   :190   Max.   :230.00
##     Sp. Atk          Sp. Def          Speed          Generation
##  Min.   : 10.00   Min.   : 20.0   Min.   :  5.00   Min.   :1.000
##  1st Qu.: 49.75   1st Qu.: 50.0   1st Qu.: 45.00   1st Qu.:2.000
##  Median : 65.00   Median : 70.0   Median : 65.00   Median :3.000
##  Mean   : 72.82   Mean   : 71.9   Mean   : 68.28   Mean   :3.324
##  3rd Qu.: 95.00   3rd Qu.: 90.0   3rd Qu.: 90.00   3rd Qu.:5.000
##  Max.   :194.00   Max.   :230.0   Max.   :180.00   Max.   :6.000
##  Legendary
##  Mode :logical
##  FALSE:735
##  TRUE :65
##
##
##
```

Type Demographic

Pokemon types are separated into 18 categories, each Pokemon has at least one type which represents their primary type and a potential secondary type but that is not always the case.
I believe that the Normal Type Pokemon are the most common type of Pokemon, to visualize this we have to understand that the two types (Type 1 & Type 2) are two separate data fields.
To visually display this we will use a bar graph to count the amount (X Axis) of each type (Y Axis)

Type 1 Count

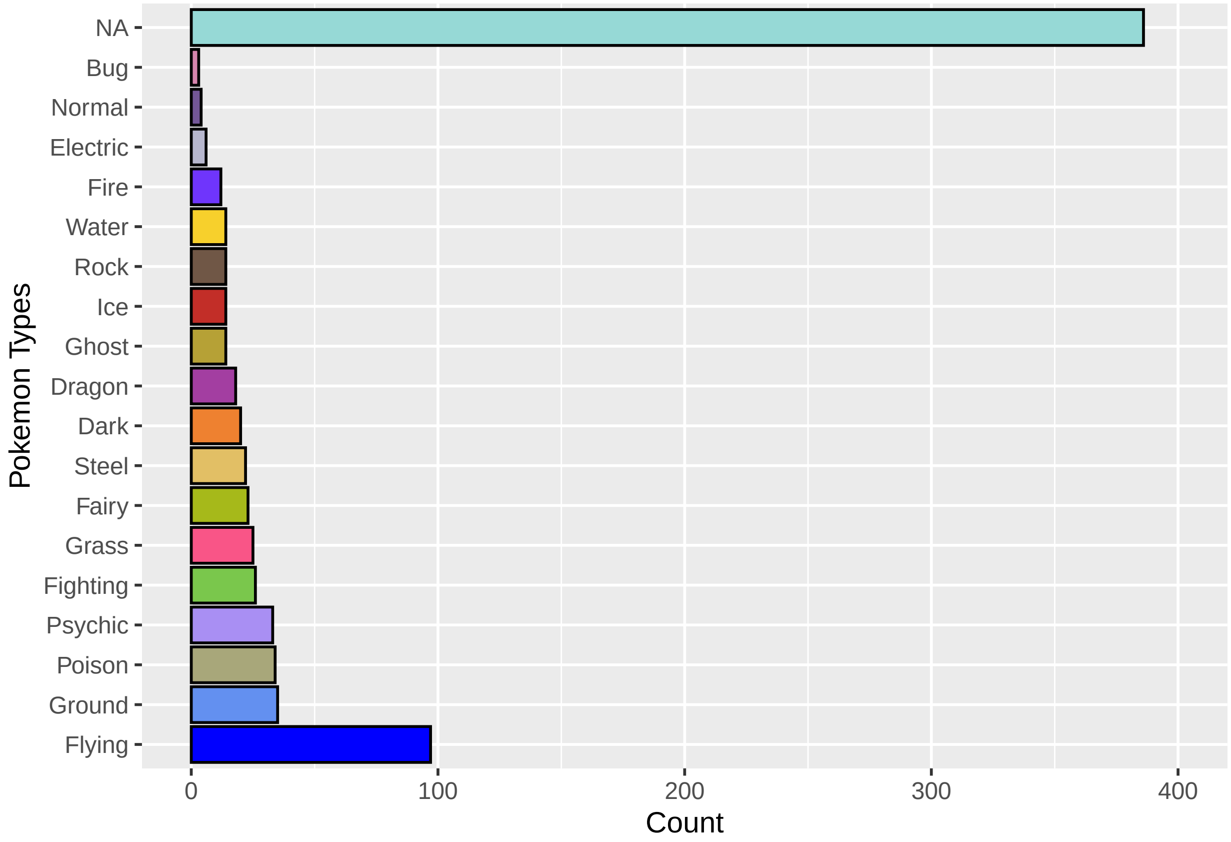
```
colors <- c('#6390F0', '#A8A77A','#A98FF3','#7AC74C', '#F95587', '#A6B91A', '#E2BF65', '#EE8130', '#A33



ggplot(Pokemon, aes(y=reorder(`Type 1`,`Type 1`,
                              function(y)-length(y)))) +
  geom_bar(fill=colors,col='black')+
  scale_x_continuous(limits=c(0,120))+
    xlab("Count") +
  ylab("Pokemon Types")
```



Basing our graph only on “Type 1” shows that Water appears to be the most common type, but this does not represent all the types available. For this we need to show Type 2 as well.

Type 2 Count

```
colors <- c('blue','#6390F0', '#A8A77A','#A98FF3','#7AC74C', '#F95587', '#A6B91A', '#E2BF65', '#EE8130'



ggplot(Pokemon, aes(y=reorder(`Type 2`,`Type 2`,
                              function(y)-length(y)))) +
  geom_bar(fill=colors,col='black')+
  scale_x_continuous(limits=c(0,400))+
    xlab("Count") +
  ylab("Pokemon Types")
```

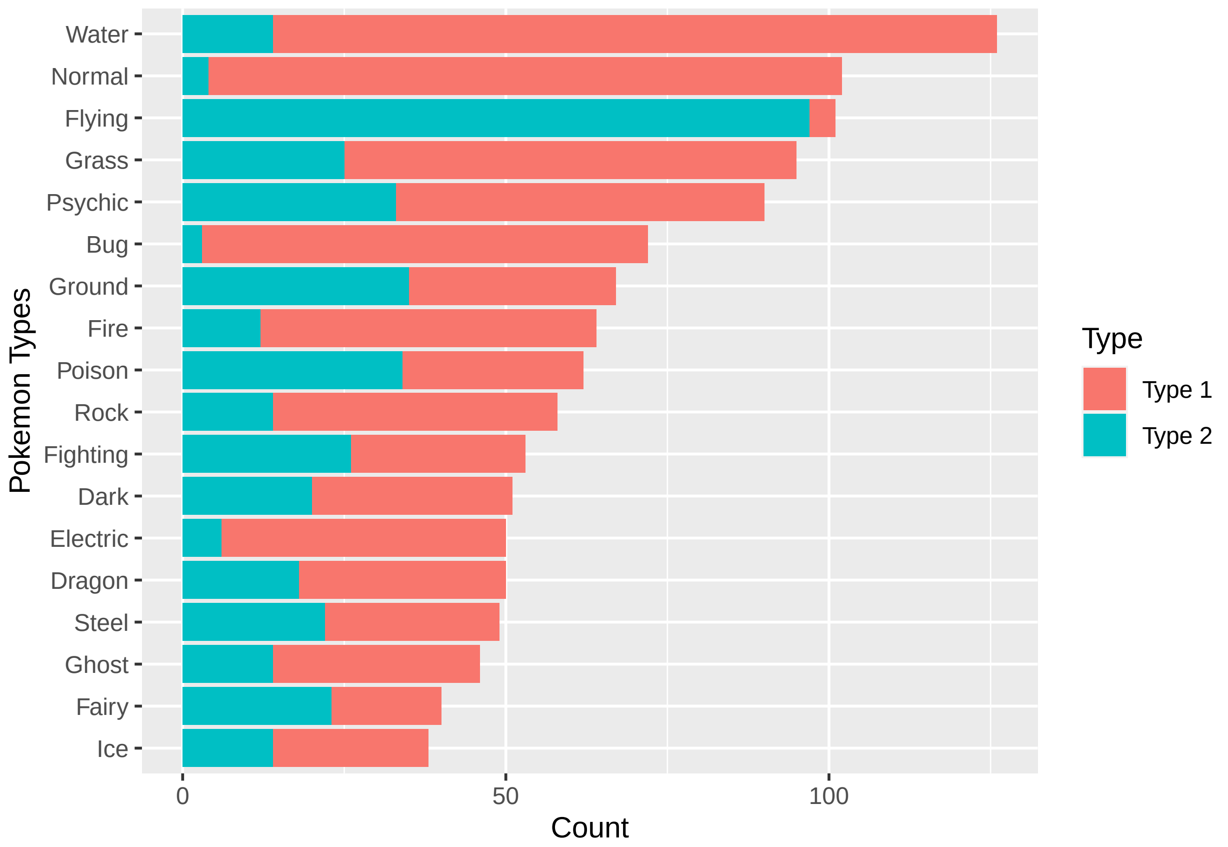


The graph for Type 2 shows a completely different scenario than before with NA being the most common occuring, this can be attributed to the fact that not all Pokemon has a secondary type but rather close to half of them that has a secondary type.

Type 1 & 2 Combined

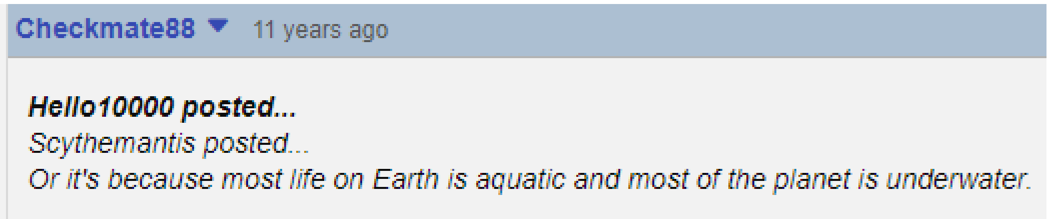
To get a better understanding, the two graphs can be combined.

```
Pokemon$Count <- 1
DF1 <- aggregate(Pokemon$Count,by=list(Pokemon$`Type 1`),FUN=sum)
DF2 <- aggregate(Pokemon$Count,by=list(Pokemon$`Type 2`),FUN=sum)

DF1$Type <- "Type 1"
DF2$Type <- "Type 2"
DF <- rbind(DF1,DF2)
DF$Type <- as.character(DF$Type)


ggplot(DF, aes(fill=Type, y=reorder(Group.1, x), x=x))+
  geom_bar(position="stack", stat="identity")+
  xlab("Count") +
  ylab("Pokemon Types")
```



Conclusion

Combining the two types into one graph shows that the water type seems to still be the most common type with normal coming in at second place and third taken by flying but mostly as a secondary stat. Researching this trend with water types further on the web led to a great explanation:


Checkmate88 11 years ago

Hello10000 posted...

Scythemantis posted...

Or it's because most life on Earth is aquatic and most of the planet is underwater.

Figure 1: Explanation

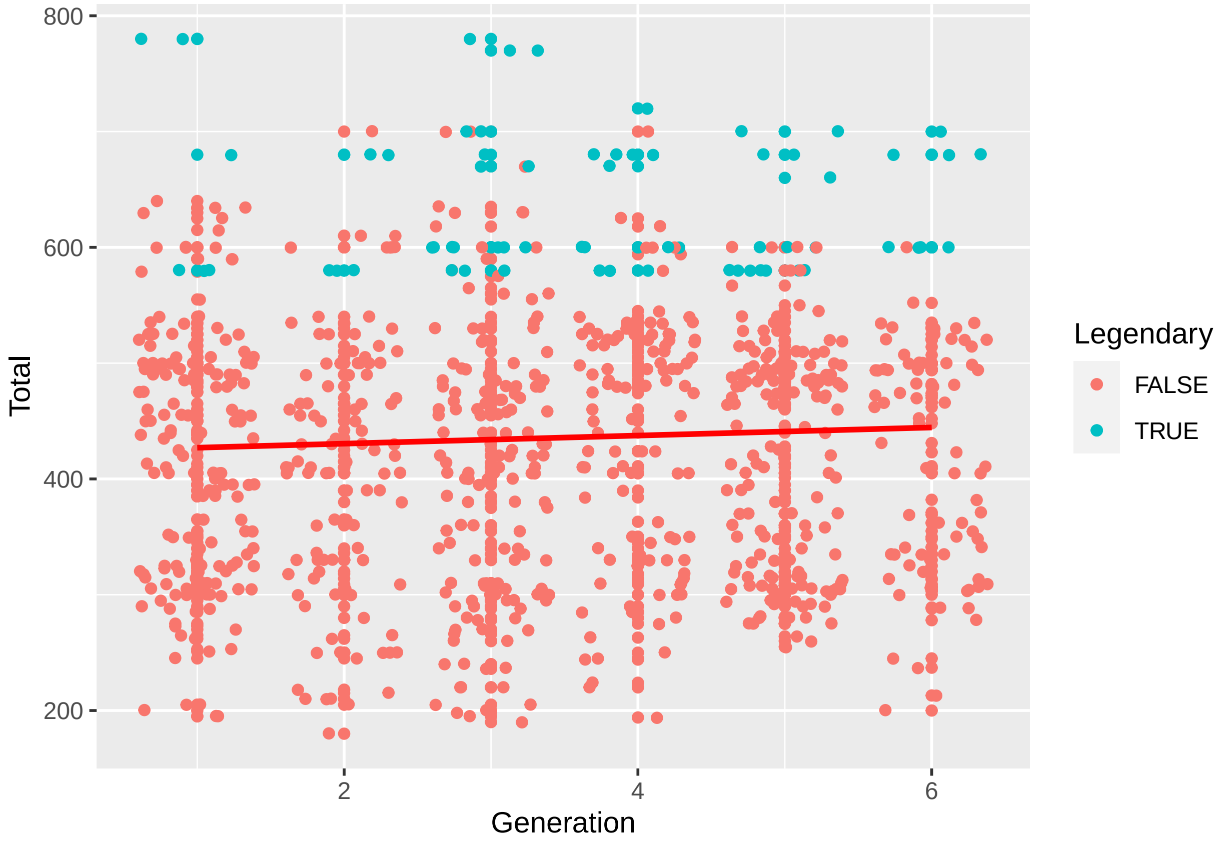
Power Creep

A typical trend in games with a long running service time is power creep, where things in the game will increase in power over time.
To show this, we'll use linear regression by seperate Pokemon into generations on the X axis and plot every Pokemon's Total Stat and look at their stats over time (Generations). Pokemon have many stats so to

simplify we will only look at four graphs that each have one stat: the Total Stat (All stats combined), HP, Attack and Defense, which are all basic stats.

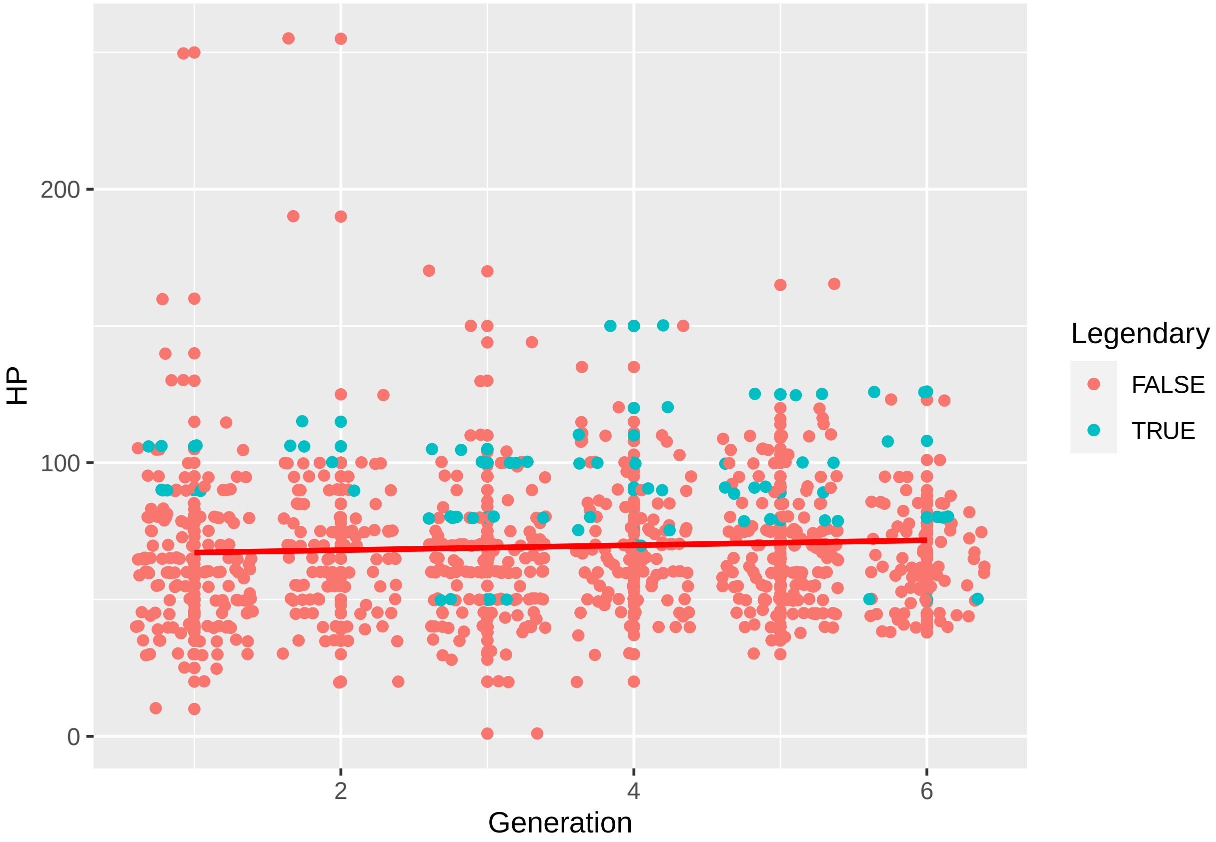
Total Stat per Generation

```r
ggplot(Pokemon, aes(x=`Generation`, y=Total, color=`Legendary`))+
  geom_point()+
  geom_jitter()+
  geom_smooth(formula = y ~ x, method="lm" , color="red", se=FALSE)
```



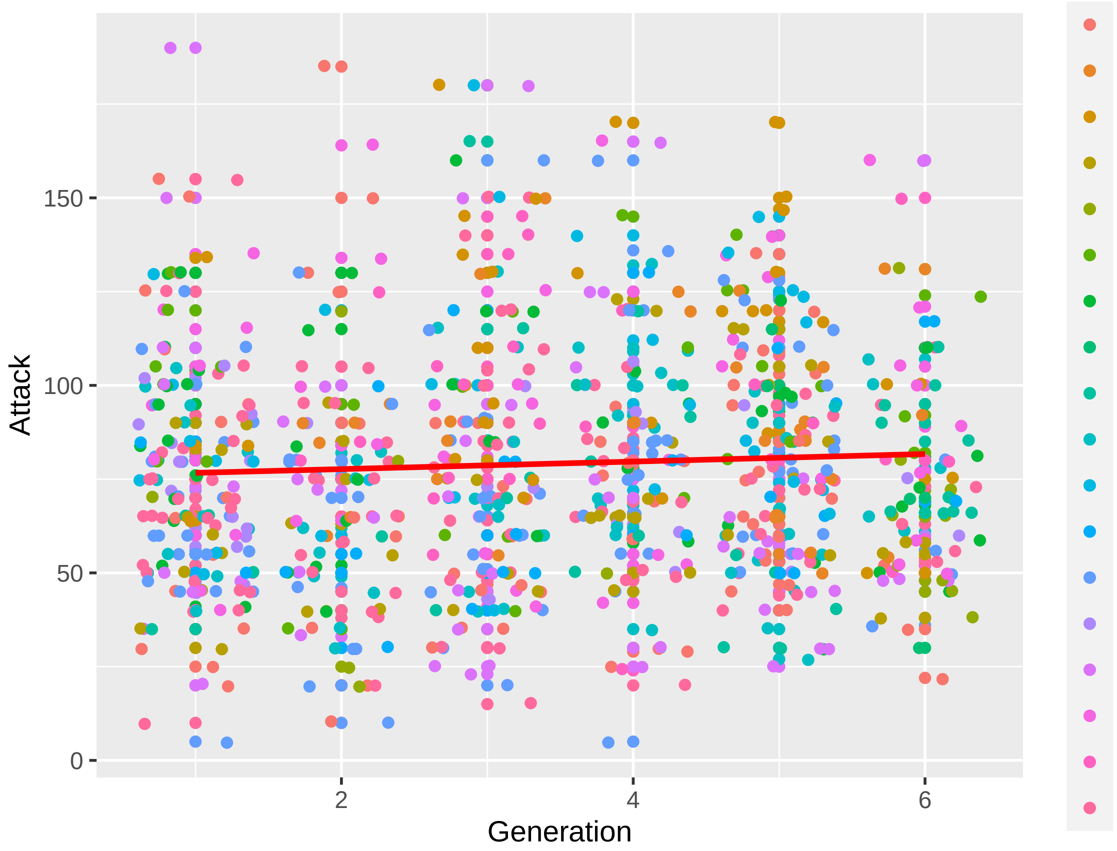
HP Stat per Generation

```r
ggplot(Pokemon, aes(x=`Generation`, y=HP, color=`Legendary`))+
  geom_point()+
  geom_jitter()+
  geom_smooth(formula = y ~ x, method="lm" , color="red", se=FALSE)
```



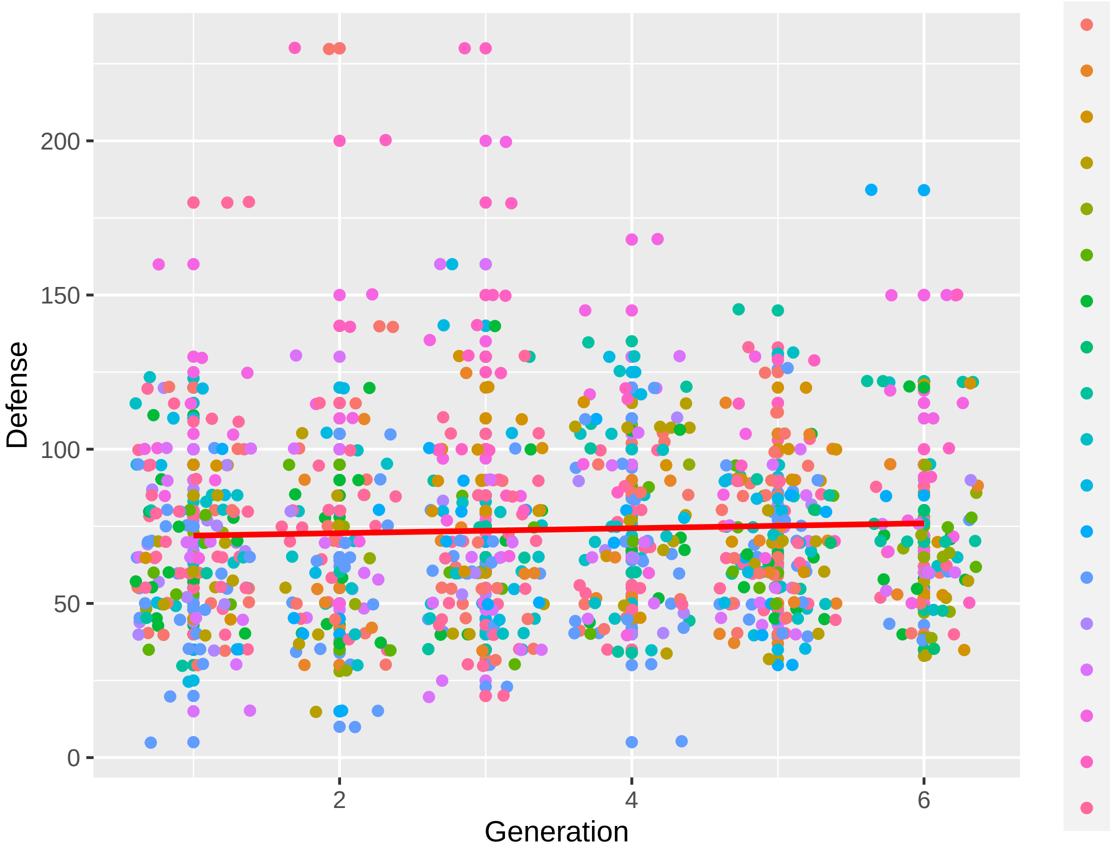
Attack Stat per Generation

```
ggplot(Pokemon, aes(x=`Generation`, y=Attack, color=`Type 1`))+
  geom_point()+
  geom_jitter()+
  geom_smooth(formula = y ~ x, method="lm" , color="red", se=FALSE)
```



Defense Stat per Generation

```
ggplot(Pokemon, aes(x=`Generation`, y=Defense, color=`Type 1`))+
  geom_point()+
  geom_jitter()+
  geom_smooth(formula = y ~ x, method="lm" , color="red", se=FALSE)
```



Conclusion

On all four occasions there is a slight power creep over 6 Generations showing that the trend towards power creep also happens in Pokemon.

Besides the power creep, we also have a display of legendaries vs non-legendaries in the two first graphs (Total and HP) which shows that as expected Legendaries generally have higher Total stats than non-legendaries but their HP stat are not always the highest which means that one or more of their other stats must be higher on average.

The last two shows types instead of legendaries and there is a correlation between rock and steel types having higher defense on average but nothing else besides that without diving further into Types and Stats.