

Pokémon Power Creep

Christian Stochholm

2023-11-27

Introduction

Pokemon is a game much like any other that suffers from power creep that steadily increases stats over time. Besides a general power creep, I believe the game also leans towards making things more awesome and thereby having more so-called “Legendary” Pokemon.

Using a datasheet that covers six generations of Pokemon, I will visually demonstrate these two phenomenon.

Project Setup

The file for the project was retrieved here: <https://gist.github.com/armgilles/194bcff35001e7eb53a2a8b441e8b2c6>

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

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

```
## -- 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 errors
```

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)
```

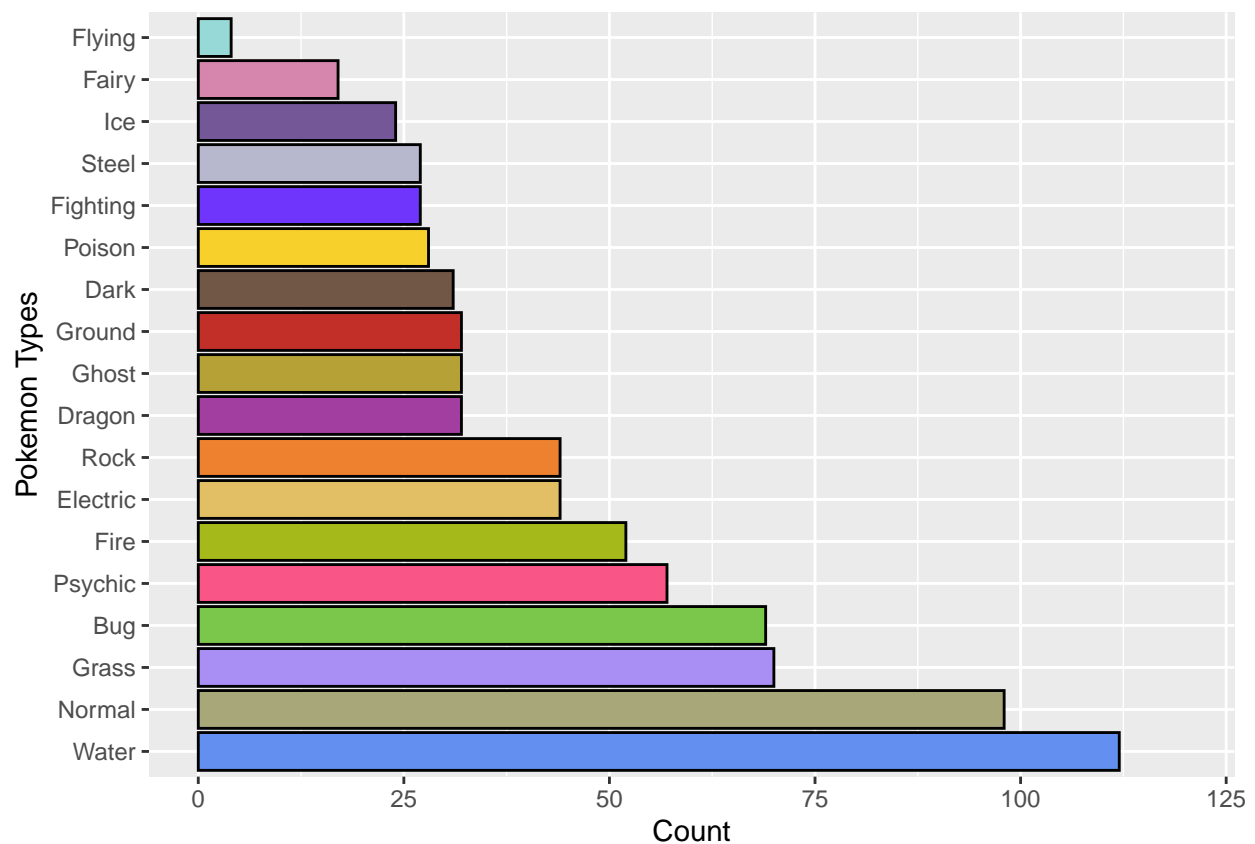
```
##      #           Name           Type 1           Type 2
## Min.   : 1.0   Length:800      Length:800      Length:800
## 1st Qu.:184.8   Class :character   Class :character   Class :character
## Median :364.5   Mode  :character   Mode  :character   Mode  :character
## Mean    :362.8
## 3rd Qu.:539.2
## 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

Type 1 Count

```
colors <- c('#6390F0', '#A8A77A', '#A98FF3', '#7AC74C', '#F95587', '#A6B91A', '#E2BF65', '#EE8130', '#A3392B')

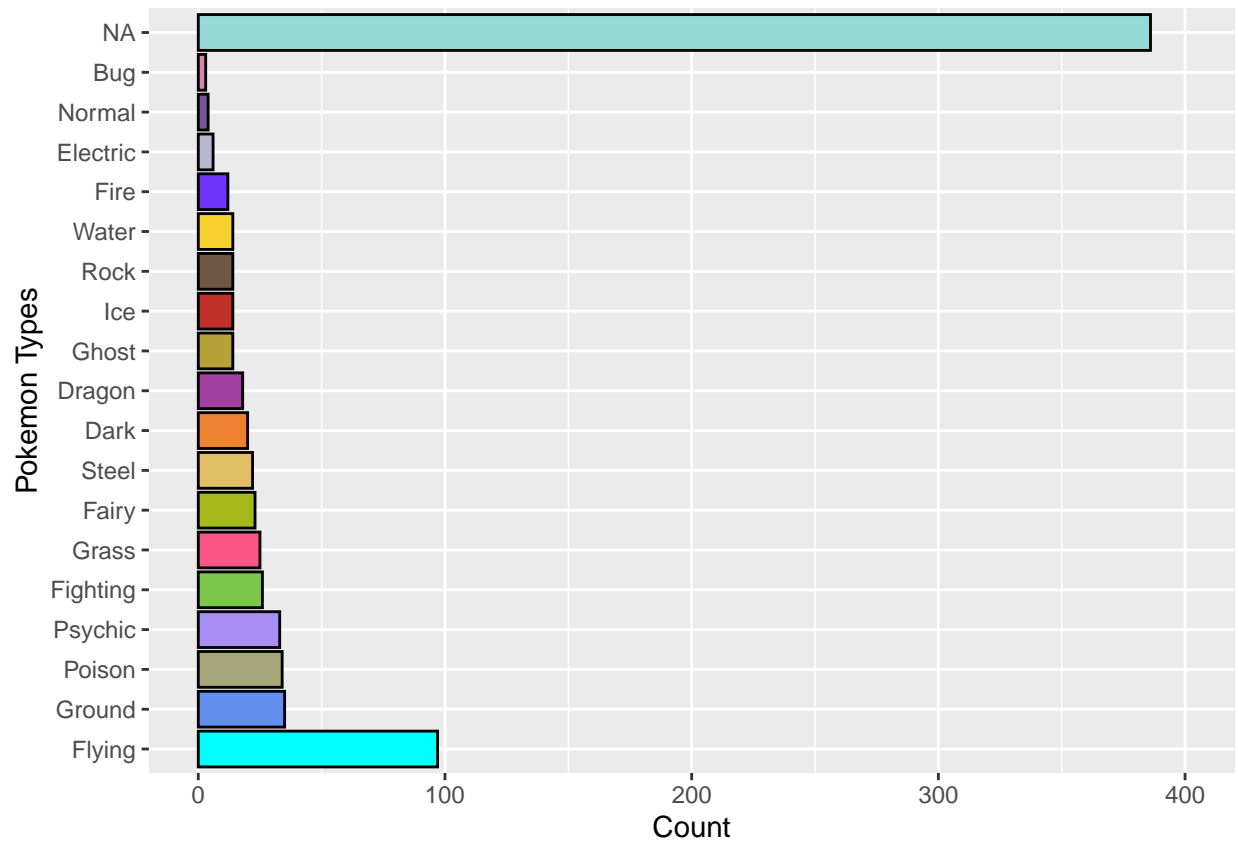
ggplot(Pokemon, aes(y=reorder(`Type 1`, `Type 1`,
                              function(y) -length(y)))) +
  geom_bar(fill=colors, col='black') +
  scale_x_continuous(limits=c(0,120)) +
  xlab("Count") +
  ylab("Pokemon Types")
```



Type 2 Count

```
colors <- c('cyan', '#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")
```

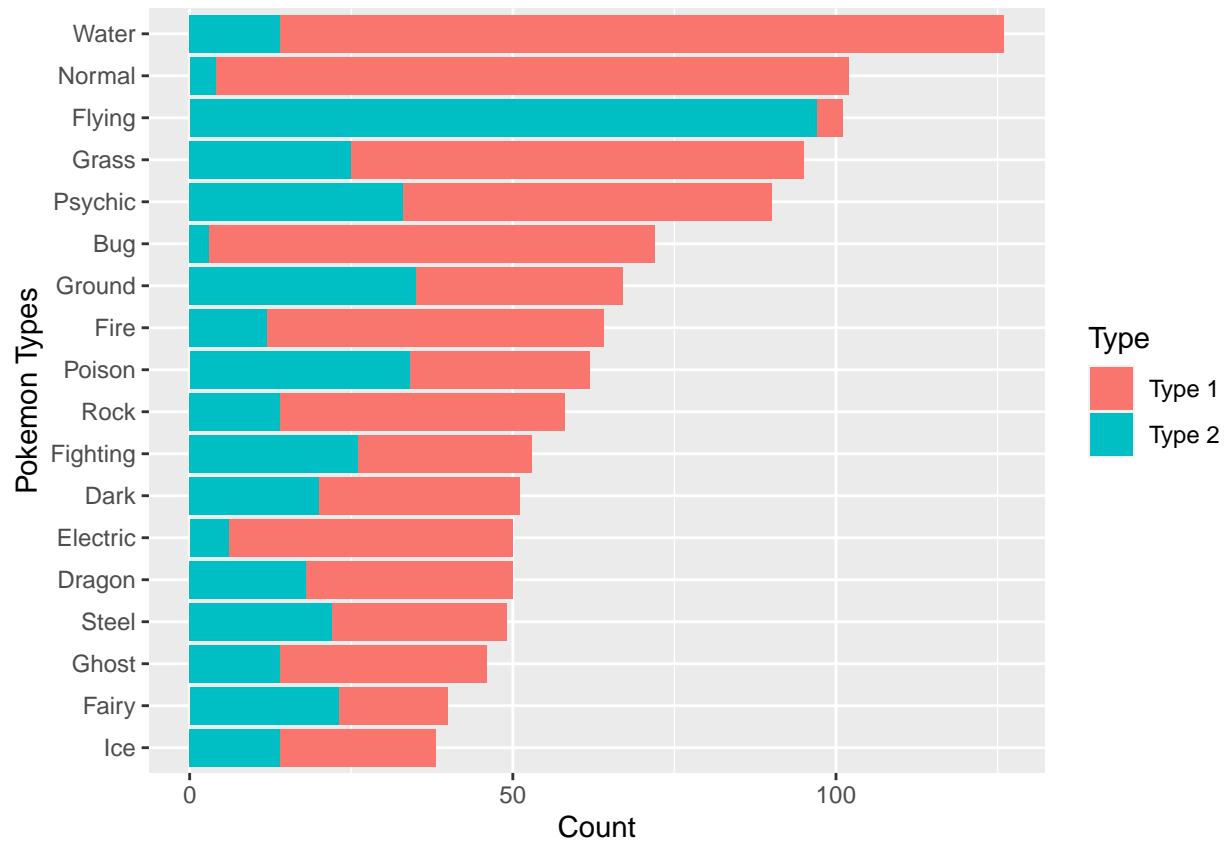


Diving into the the types of Pokemon a weird trend arises, while one might assume the normal type would be the most common type of Pokemon, water appears to be the most common.

```
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")
```



To add to this, the water type seems to heavily favor Type 1. Researching this trend further on the web led to a great explanation:

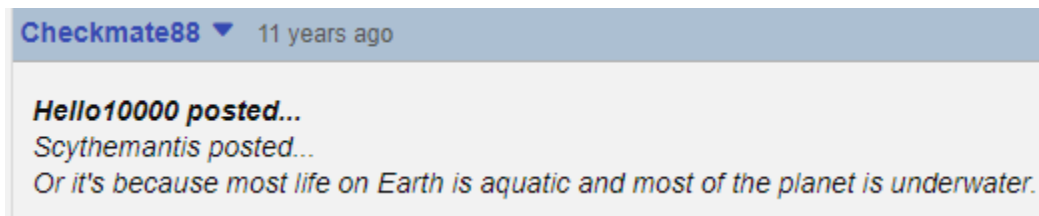
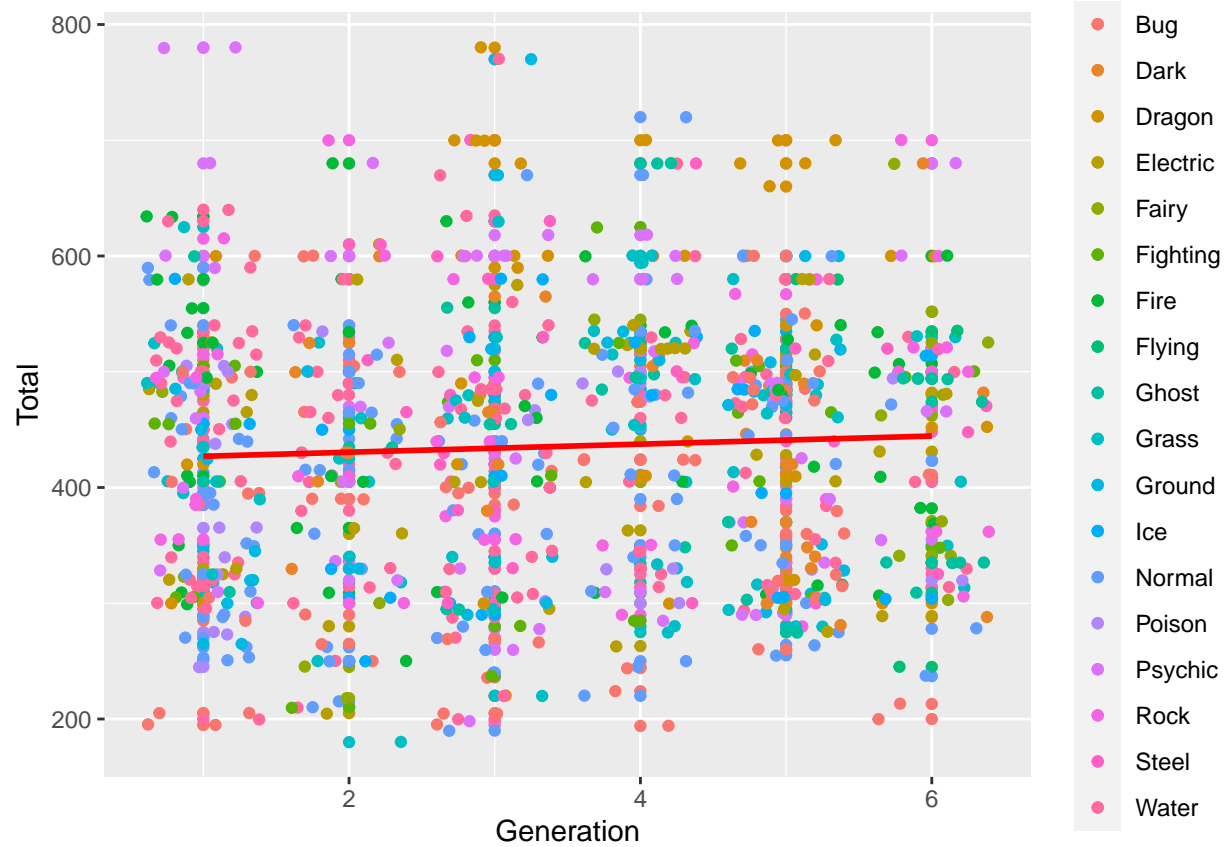


Figure 1: Explanation

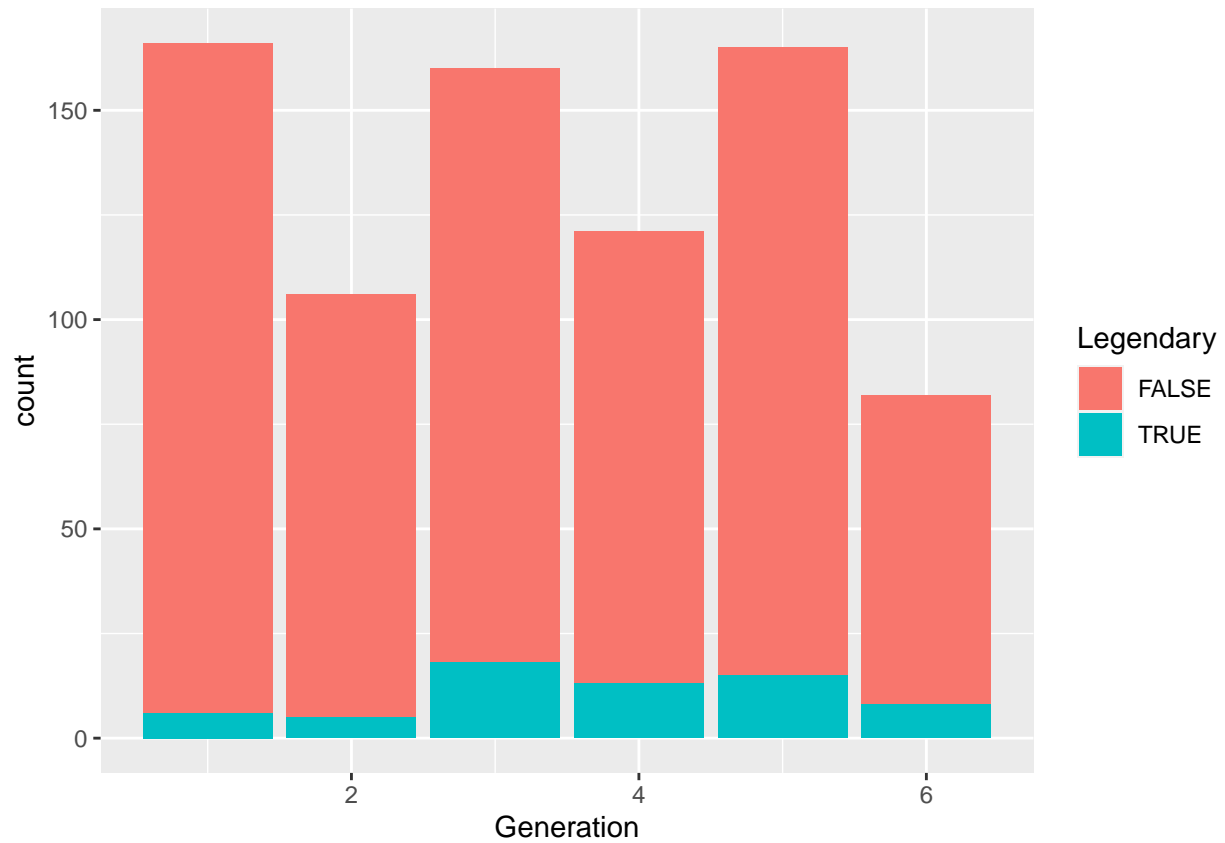
Power Creep

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



Legendaries

```
ggplot(Pokemon, aes(x= Generation, fill = `Legendary`))+
  geom_bar()
```



I believe that normal type pokemon are the most common type of pokemon.

GLEMMER POKEMON UDEN TYPE 2

““

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.