Pokemon Analysis

Rohit Ravikumar

1/9/2021

Getting the Data

We begin by downloading the table of all Pokemon by base stats from Bulbapedia. By base stat total, here are the weakest Pokemon as of Generation VIII:

```
##                     Name HP Atk Def SpA SpD Speed Total Average
## 1 Wishiwashi (Solo Form) 45  20  20  25  25    40   175   29.17
## 2                Sunkern 30  30  30  30  30    30   180   30.00
## 3                Blipbug 25  20  20  25  45    45   180   30.00
## 4                   Snom 30  25  35  45  30    20   185   30.83
## 5                Azurill 50  20  40  20  40    20   190   31.67
## 6              Kricketot 37  25  41  25  41    25   194   32.33
```

And here are the strongest:

```
##                       Name  HP Atk Def SpA SpD Speed Total Average
## 1    Eternatus (Eternamax) 255 115 250 125 250   130  1125  187.50
## 2   Mewtwo (Mega Mewtwo X) 106 190 100 154 100   130   780  130.00
## 3   Mewtwo (Mega Mewtwo Y) 106 150  70 194 120   140   780  130.00
## 4 Rayquaza (Mega Rayquaza) 105 180 100 180 100   115   780  130.00
## 5   Kyogre (Primal Kyogre) 100 150  90 180 160    90   770  128.33
## 6 Groudon (Primal Groudon) 100 180 160 150  90    90   770  128.33
```

We're also interested in the usage statistics of various Pokemon on the competitive scene. As of October 2020, here are the most-used (non-banned) competitive Pokemon:

```
##   Rank      Name UsagePercent    Raw RawPercent   Real RealPercent
## 1    1 Dragapult    0.1659842 606998    0.16598 447357     0.15756
## 2    2   Urshifu    0.1459203 533625    0.14592 406037     0.14301
## 3    3  Clefable    0.1382013 505397    0.13820 395417     0.13927
## 4    4 Rillaboom    0.1344477 491670    0.13445 379709     0.13373
## 5    5 Excadrill    0.1267249 463428    0.12672 369454     0.13012
## 6    6 Regieleki    0.1232854 450850    0.12329 338309     0.11915
```

Finally, we want to know the types of each Pokemon. Below is a table showing the relative frequency of each primary type (ignoring secondary type for the time being).

```
## # A tibble: 18 x 2
##    Type1    Count
##    <chr>    <int>
```

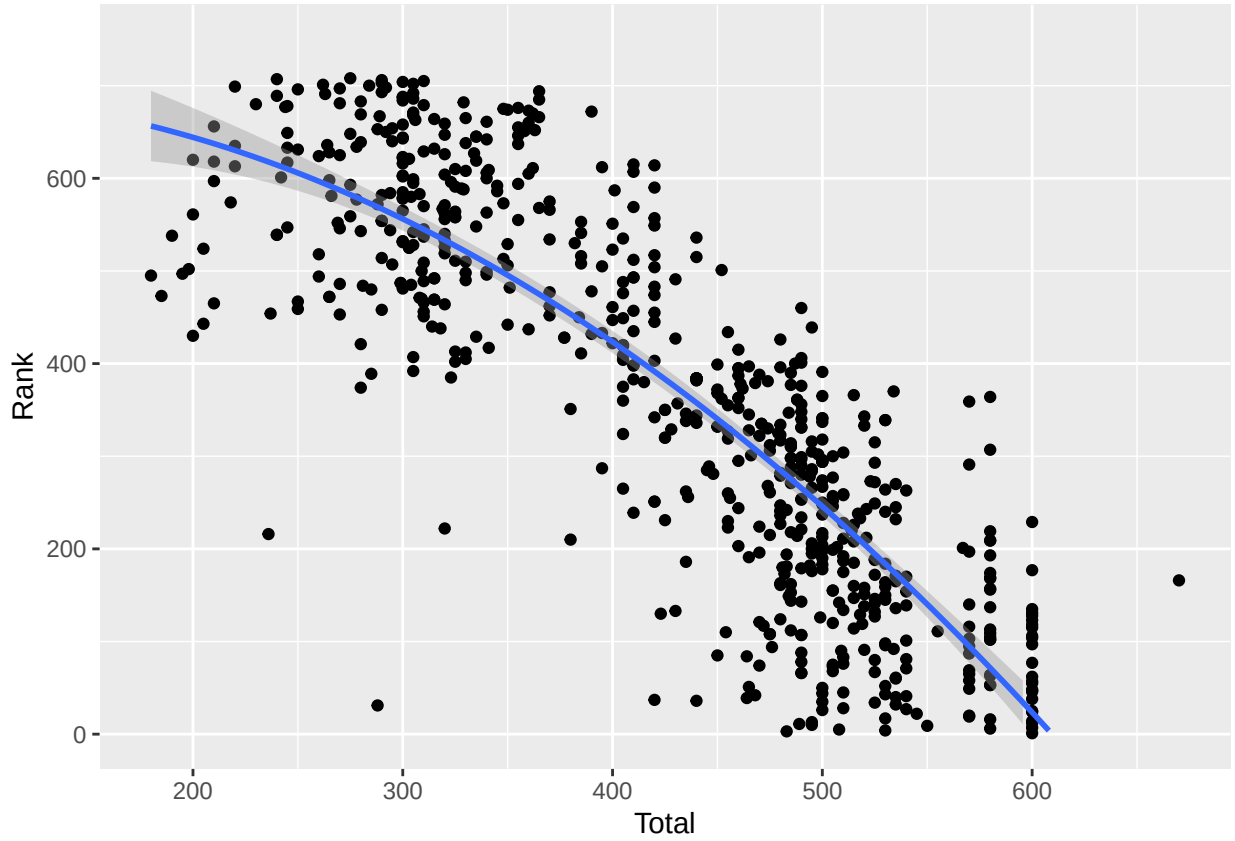
```
##  1 Water      129
##  2 Normal     111
##  3 Grass       88
##  4 Bug         79
##  5 Psychic     69
##  6 Fire        61
##  7 Electric    56
##  8 Rock        53
##  9 Dark        43
## 10 Fighting    40
## 11 Poison      40
## 12 Ground      39
## 13 Ice         37
## 14 Ghost       34
## 15 Dragon      31
## 16 Steel       31
## 17 Fairy       22
## 18 Flying       7
```

Given that certain Pokemon are not allowed in the competitive scene, and Mega Evolutions are double-counting certain Pokemon, combining the three sets of data by Pokemon name reduces our data set to 669 Pokemon. The first few are listed alphabetically here:

```
##         Name HP Atk Def SpA SpD Speed Total Average   Type1  Type2 Rank
## 1  Abomasnow 90  92  75  92  85    60   494   82.33   Grass    Ice  278
## 2       Abra 25  20  15 105  55    90   310   51.67 Psychic   <NA>  451
## 3      Absol 65 130  60  75  60    75   465   77.50    Dark   <NA>  191
## 4   Accelgor 80  70  40 100  60   145   495   82.50     Bug   <NA>  176
## 5 Aerodactyl 80 105  65  60  75   130   515   85.83    Rock Flying  147
## 6     Aggron 70 110 180  60  60    50   530   88.33   Steel   Rock  164
##   UsagePercent   Raw RawPercent  Real RealPercent
## 1    0.0016380  5990    0.00164  4777     0.00168
## 2    0.0000785   287    0.00008   210     0.00007
## 3    0.0041909 15326    0.00419 10856     0.00382
## 4    0.0047925 17526    0.00479 15771     0.00555
## 5    0.0071253 26057    0.00713 21631     0.00762
## 6    0.0058363 21343    0.00584 16666     0.00587
```

Analyzing the Data

When looking through this data, the most obvious question to ask is whether Pokemon usage correlates with their stats in any meaningful way. Let's start by looking at the effect of base stat total on rank (we would expect to see a strong negative correlation here, as more powerful Pokemon are naturally more useful competitively):

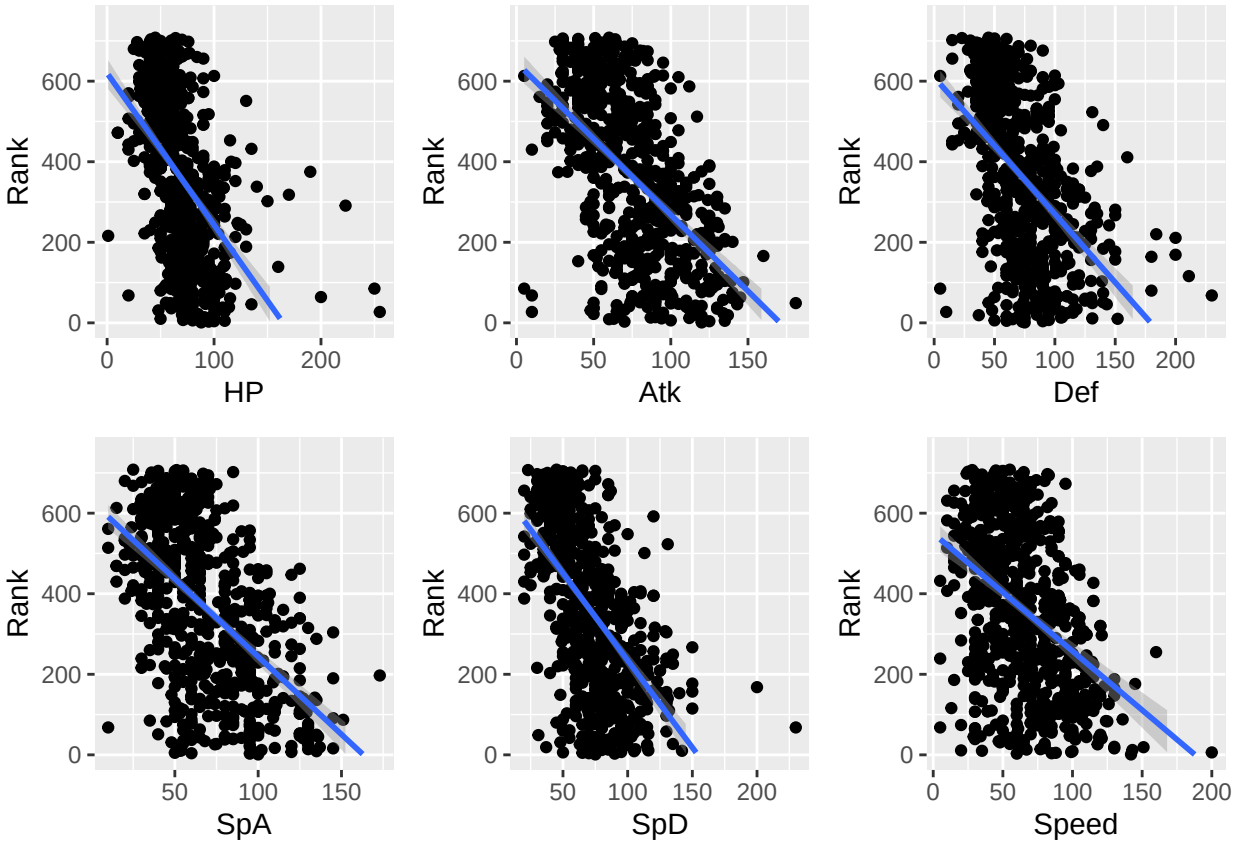


As suspected, there is a strong negative correlation. Notably, the relationship between the two doesn't seem to be entirely linear (which one might expect; jumps in stats seem to have a bigger impact on rank as the former increases). Let's take a look at the results of a quadratic regression here to make sure our findings are statistically significant:

```
##
## Call:
## lm(formula = Rank ~ poly(Total, 2), data = pkmn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -538.12  -64.01    7.74   74.61  324.73
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       360.06       4.17   86.34  < 2e-16 ***
## poly(Total, 2)1 -4363.44     107.87  -40.45  < 2e-16 ***
## poly(Total, 2)2  -621.27     107.87   -5.76 1.29e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 107.9 on 666 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.714
## F-statistic: 834.7 on 2 and 666 DF,  p-value: < 2.2e-16
```

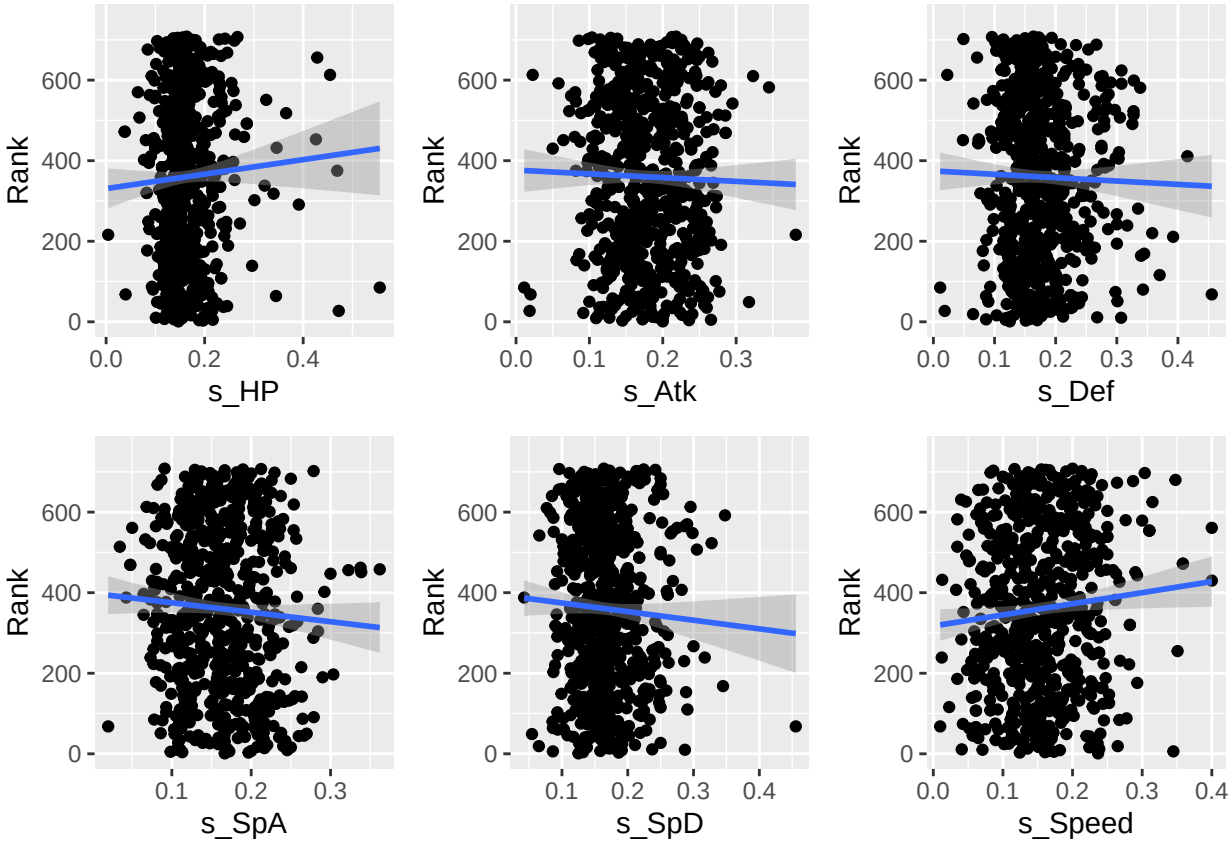
Indeed, we have a high R-squared and statistically significant coefficients; we can be confident that the relationship here is not simply due to chance. It's worth asking the question, though: does this relationship

show up for each of the stats individually, or just when they're combined? Below we see the relationship between each of the six stats and rank:



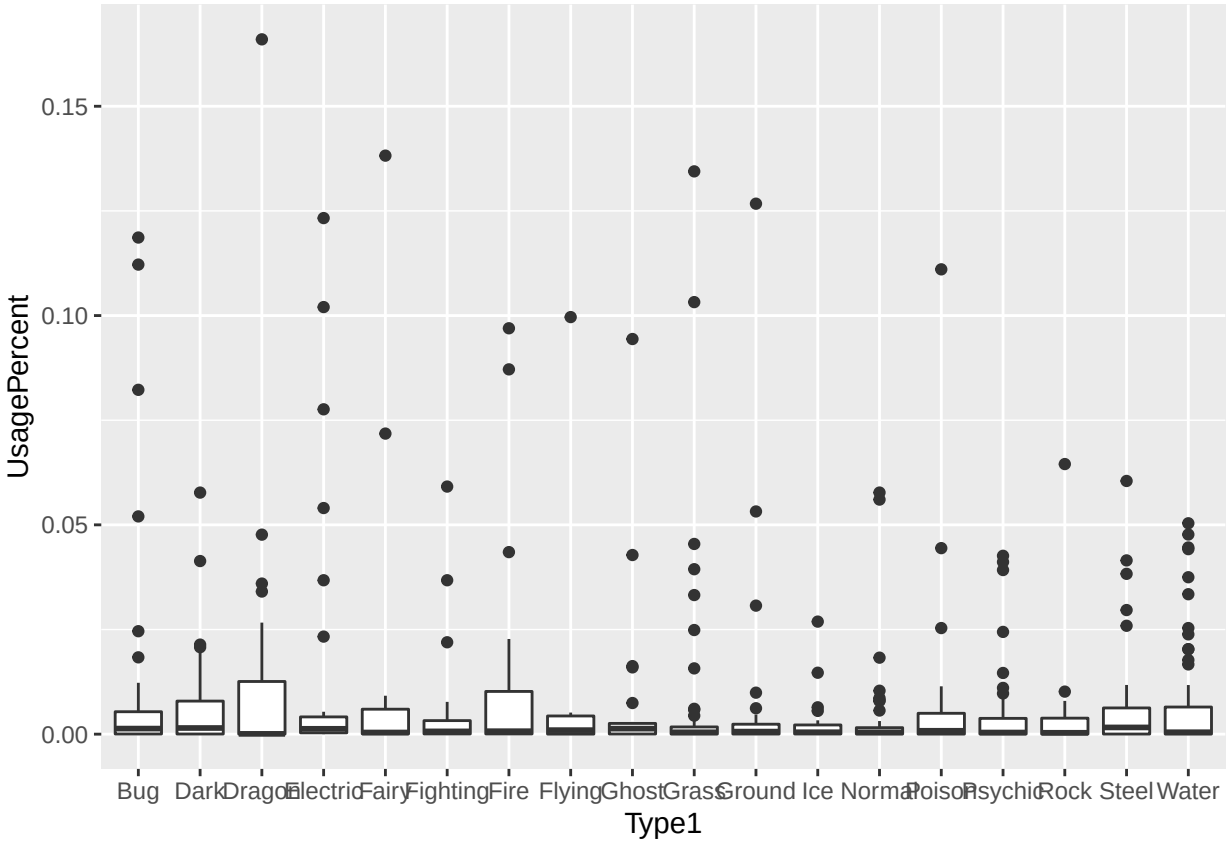
It's clear that the correlations become considerably weaker as we separate out each stat, but a negative correlation is still visible for each.

But what will happen if we standardize the stats? Let's take each stat as a percentage of the given Pokemon's stat total (essentially setting all stat totals to 1), and look at how those correlate with rank:



Here we can see that those negative correlations from each stat all but evaporate when standardized, suggesting that the strength of those negative correlations were due largely to the magnitude of the stat totals, rather than inherent characteristics of each stat.

Finally, we ought to take a look at how usage corresponds to Pokemon type. Observe the below boxplot of primary types compared to percent usage (rather than rank; this will give us better information on how much different Pokemon are actually being used):



Unfortunately, most Pokemon usage percentages seem to be clustered around 0, making a boxplot difficult to read. Let's filter our results to only look at Pokemon with a usage percentage of 0.5% or better. First, let's see how many Pokemon of each primary type made it to that threshold:

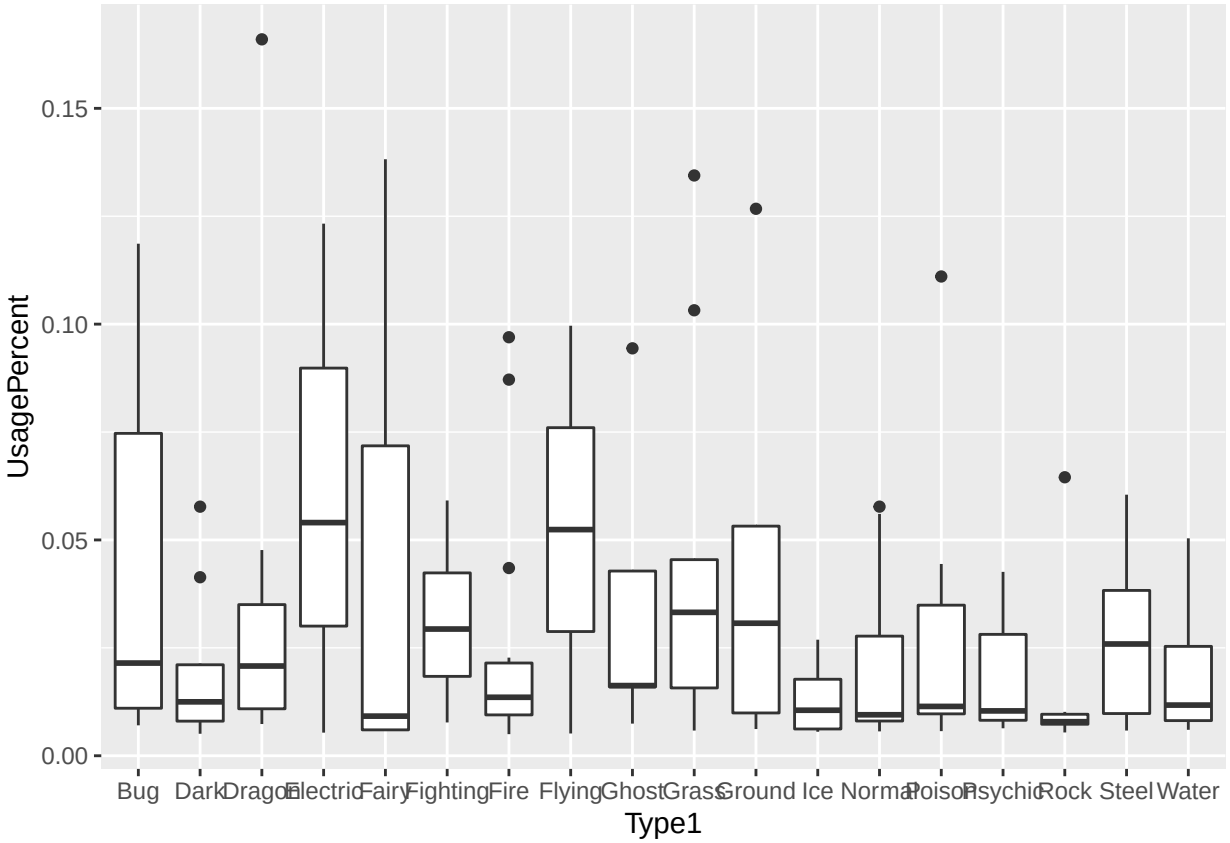
```
## # A tibble: 18 x 2
##    Type1    Count
##    <chr>    <int>
##  1 Water       25
##  2 Fire        14
##  3 Psychic     12
##  4 Dark        11
##  5 Dragon      11
##  6 Bug         10
##  7 Grass        9
##  8 Steel        9
##  9 Normal       8
## 10 Electric     7
## 11 Poison       7
## 12 Rock         6
## 13 Fairy        5
## 14 Ghost        5
## 15 Ground       5
## 16 Fighting     4
## 17 Ice          4
## 18 Flying       2
```

Water seems like an obvious winner here, but Water is also the most common primary type, as we saw

earlier; similarly, Flying being underrepresented is exactly what we would expect given the prior data. If we standardize these values as a percentage of their total representation, we get the following:

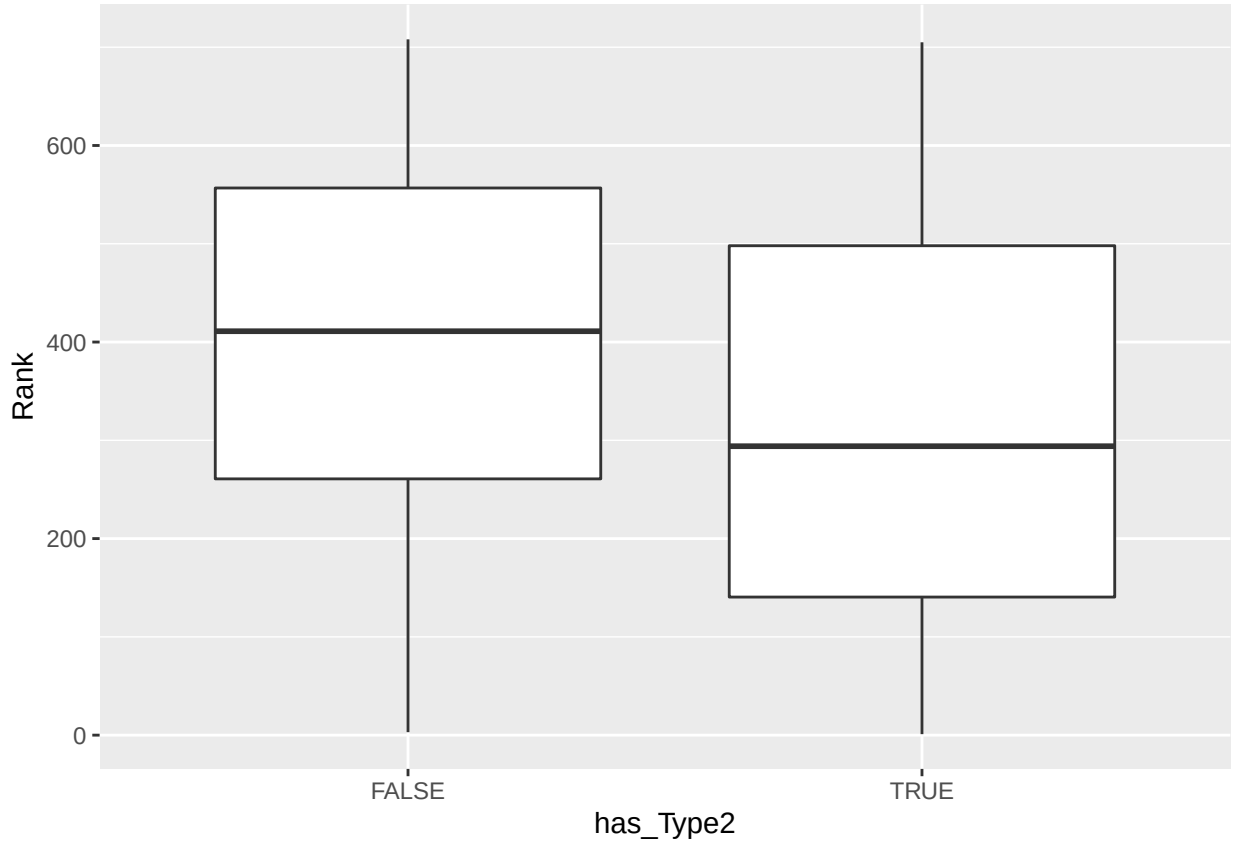
```
## # A tibble: 18 x 3
##    Type1    Count CountPercentage
##    <chr>    <int>           <dbl>
##  1 Flying       2           0.286
##  2 Water       25           0.194
##  3 Normal       8           0.186
##  4 Ice          4           0.182
##  5 Electric     7           0.175
##  6 Poison       7           0.175
##  7 Steel        9           0.170
##  8 Bug         10           0.164
##  9 Ground       5           0.161
## 10 Grass        9           0.161
## 11 Dragon      11           0.159
## 12 Rock         6           0.154
## 13 Ghost        5           0.147
## 14 Dark        11           0.139
## 15 Psychic     12           0.136
## 16 Fairy        5           0.135
## 17 Fighting     4           0.129
## 18 Fire        14           0.126
```

Conveniently, we find that (though Flying is still an odd outlier), the representation of the various primary types falls within a reasonably narrow band. Now let's take a look at the boxplot of these Pokemon:



Though the presence of outliers complicates our analysis somewhat, we can clearly see certain types clustered at lower percentages (most notably Rock), while other types are spread out more and contain a larger percentage of high-usage Pokemon, such as Electric and Fairy.

Finally, let's wrap up here by looking at the difference in rank between Pokemon with and without a second type (again, by using a boxplot).



Perhaps unsurprisingly, Pokemon without a second type are quite a bit lower-ranked (with a larger rank number) than those with a second type.