Pokemon

John Cruz

2023-03-01

R Markdown

National Pokedex

Pokemon Stats

Pokemon Types

A few kinds of analysis that could be done would be:

to look at which types of Pokémon tend to have the highest base stats to find the 6 Pokémon with the highest base speed to observe whether Pokémon with a single type have higher base stats than Pokémon with two types to find the 10 rarest Pokémon abilities, i.e. the abilities that the fewest number of Pokémon have access to

Required Libraries

```
library(tidyverse)
library(rvest)
library(xml2)
library(janitor)
```

```
url <- "https://www.serebii.net/pokemon/nationalpokedex.shtml"

web_table <- read_html(url)

# use XML to account for <br> with abilities and add '\n'
xml_find_all(web_table, ".//br") |>
  xml_add_sibling("p", "\n")
xml_find_all(web_table, ".//br") |>
  xml_remove()

web_table <-
  web_table |>
  html_element('.dextable') |>
  html_table()

pokemon_stats <- as.data.frame(web_table)
```

```r
# drop null values if Pokemon name is N/A
stats_df <-
  pokemon_stats |>
  drop_na(4)

# drop first row (duplicate header) and second column (pic)
stats_df <- stats_df[-1,-2]

# set column headers from first row and clean names
stats_df <-
  stats_df |>
  row_to_names(row_number = 1) |>
  clean_names()

# shift pokemon names, etc to left by 1 column
stats_df[c(2:10)] = stats_df[, c(3:11)]

# drop 'na' column
stats_df <-
  stats_df |>
  select(!c(na, type))

# split multiple abilities into long format based on created '\n'
stats_df <-
  stats_df |>
  separate_longer_delim(abilities, delim = "\n")

# change to pokemon number
stats_df$no <-
  parse_number(stats_df$no)
```

```r
url <- "https://bulbapedia.bulbagarden.net/wiki/List_of_Pok%C3%A9mon_by_National_Pok%C3%A9dex_number"

web_table <- read_html(url)

# use XML to account for <br> and replace with '\n'
xml_find_all(web_table, ".//br") |>
  xml_add_sibling("p", "\n")
xml_find_all(web_table, ".//br") |>
  xml_remove()

web_table <-
  web_table |>
  html_element('body') |>
  html_table()

pokemon_types <- as.data.frame(web_table)
```

```r
# drop null values if Pokemon name is N/A
types_df <-
  pokemon_types |>
  drop_na(2)
```

```r
# drop unnecessary columns
types_df <-
  types_df[, 1:5]

# set column headers from first row and clean names
types_df <-
  types_df |>
  row_to_names(row_number = 1) |>
  clean_names()

# change to pokemon number
types_df$ndex <-
  parse_number(types_df$ndex)

# drop N/A or zero (0) while keeping only distinct pokemon numbers
types_df <-
  types_df |>
  drop_na() |>
  filter(ndex != 0) |>
  distinct(ndex, .keep_all=TRUE)

# within same pokemon number, replace repeated types with N/A
types_df <-
  types_df |>
  mutate(type_2 = if_else(type_2 != type, type_2, NA)) |>
  select(-c(2)) |>
  rename(no = ndex)

# melt both type columns into one column
temp1 <-
  types_df |>
  select(1:3)

temp2 <-
  types_df |>
  select(1,2,4) |>
  rename(type = type_2)

types_df <-
  temp1 |>
  full_join(temp2) |>
  drop_na() |>
  select(!pokemon) |>
  arrange(no)
```

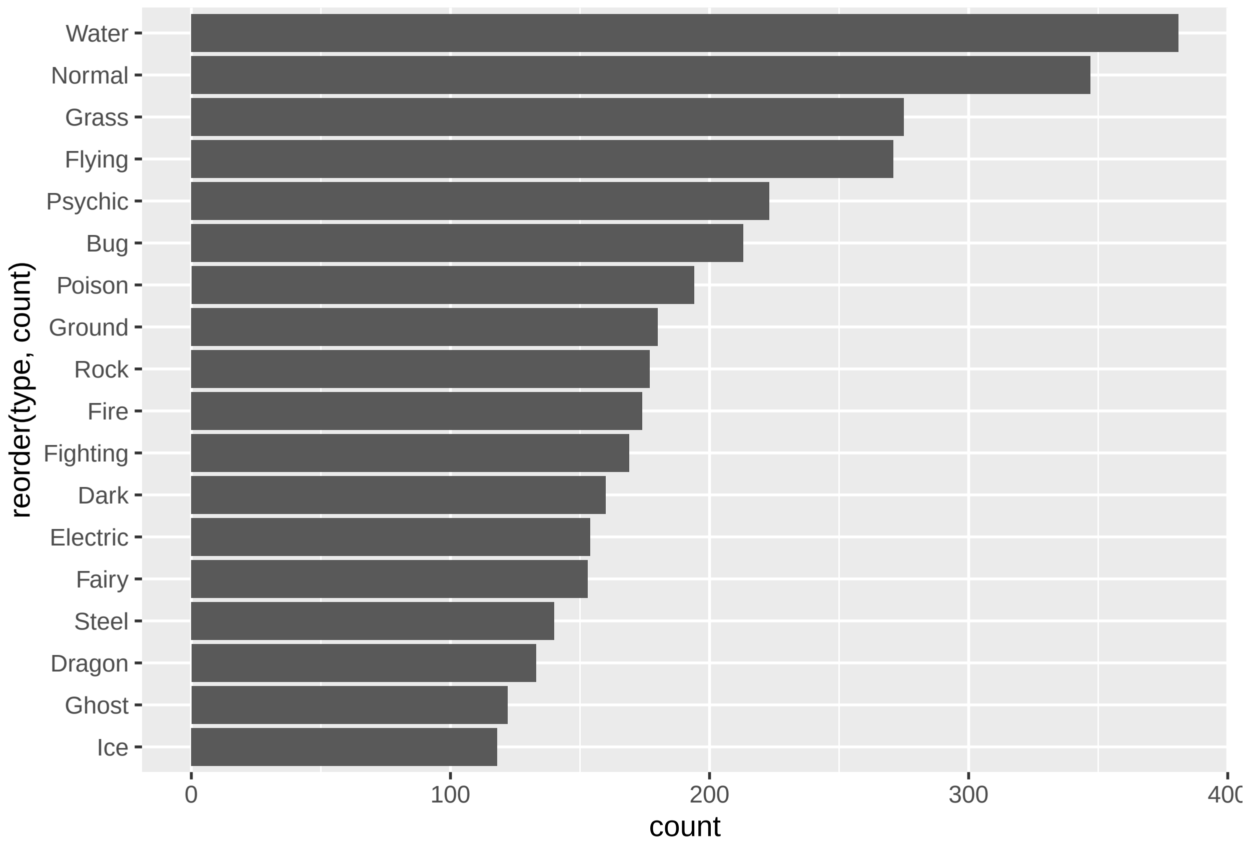
```
## Joining with ‘by = join_by(no, pokemon, type)‘
```

```r
stats_types_df <-
  stats_df |>
  inner_join(types_df) |>
  relocate(type, .after = name)
```

```
## Joining with ‘by = join_by(no)‘
## Warning in inner_join(stats_df, types_df): Each row in ‘x‘ is expected to match at most 1 row in ‘y‘
## i Row 1 of ‘x‘ matches multiple rows.
## i If multiple matches are expected, set ‘multiple = "all"‘ to silence this
##   warning.
```

to observe the frequency of Pokémon by type

```
stats_types_df |>
  group_by(type) |>
  summarise(count = n()) |>
  ggplot(aes(x = count, y = reorder(type, count))) +
  geom_bar(stat = 'identity')
```



```
stats_types_df |>
  group_by(type) |>
  summarise(count = n()) |>
    arrange(desc(count))
```

```
## # A tibble: 18 x 2
##    type     count
##    <chr>    <int>
##  1 Water      381
##  2 Normal     347
##  3 Grass      275
##  4 Flying     271
##  5 Psychic    223
##  6 Bug        213
##  7 Poison     194
##  8 Ground     180
##  9 Rock       177
## 10 Fire       174
## 11 Fighting   169
## 12 Dark       160
## 13 Electric   154
## 14 Fairy      153
## 15 Steel      140
## 16 Dragon     133
## 17 Ghost      122
## 18 Ice        118
```