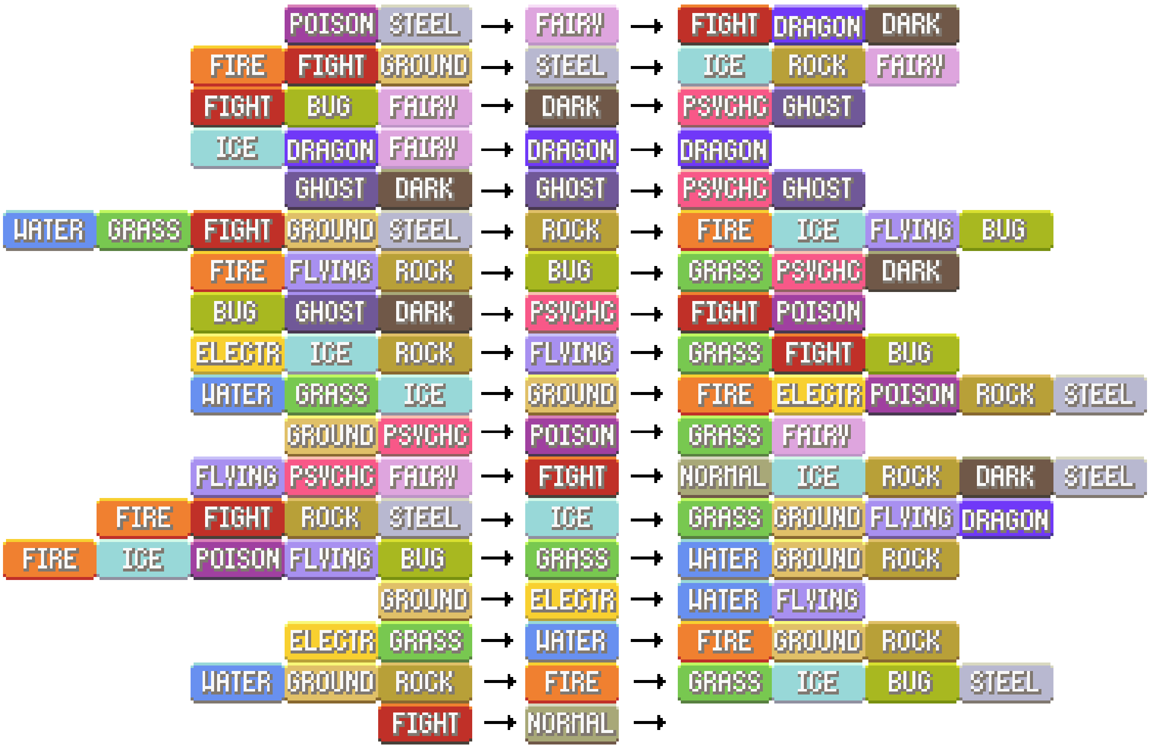
1. INTRODUCTION

Pokémon [1], is a digital game franchise that debuted in Japan in February 1996. It firstly introduced us on Nintendo’s bestselling handheld console Gameboy as Pokémon Green Version and Pokémon Red Version. The franchise later became wildly popular around the world. It becomes one of the main titles of their home consoles. Thus, it still holds its title up until now.

The original Pokémon is a role-playing game based around the role of Pokémon Trainers, capturing imaginary animals that has certain powers and train them to battle the others in a quest to become the very best like no one ever was.

Pokémon are divided into types, such as grass, water and fire. Each type has its weaknesses and its advantages. Indeed, it is a simple rock-paper-scissor game mechanic with more elements in it. For example, to gain an advantage over a Pokémon that cannot beat an opponent’s fire-type Pokémon, the player might substitute a water-type Pokémon that has at least equal experience with the opposing fire-type Pokémon. [2]


POISON STEEL → FAIRY → FIGHT DRAGON DARK
FIRE FIGHT GROUND → STEEL → ICE ROCK FAIRY
FIGHT BUG FAIRY → DARK → PSYCHC GHOST
ICE DRAGON FAIRY → DRAGON → DRAGON
GHOST DARK → GHOST → PSYCHC GHOST
WATER GRASS FIGHT GROUND STEEL → ROCK → FIRE ICE FLYING BUG
FIRE FLYING ROCK → BUG → GRASS PSYCHC DARK
BUG GHOST DARK → PSYCHC → FIGHT POISON
ELECTR ICE ROCK → FLYING → GRASS FIGHT BUG
WATER GRASS ICE → GROUND → FIRE ELECTR POISON ROCK STEEL
GROUND PSYCHC → POISON → GRASS FAIRY
FLYING PSYCHC FAIRY → FIGHT → NORMAL ICE ROCK DARK STEEL
FIRE FIGHT ROCK STEEL → ICE → GRASS GROUND FLYING DRAGON
FIRE ICE POISON FLYING BUG → GRASS → WATER GROUND ROCK
GROUND → ELECTR → WATER FLYING
ELECTR GRASS → WATER → FIRE GROUND ROCK
WATER GROUND ROCK → FIRE → GRASS ICE BUG STEEL
FIGHT → NORMAL →

Complete Pokémon-Type Chart

2. MATERIALS AND METHODS

This Pokémon dataset includes English and Japanese names, weights and heights, stats, abilities, types and type advantages and many more data of 802 Pokémon from all seven generations that came out until late 2016. (Even though the eighth generation came out in late 2019 and ninth in 2022, unfortunately the dataset does not include their data.)

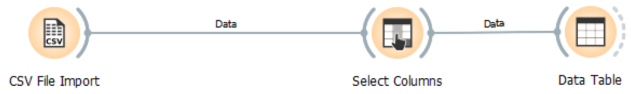
Since this dataset includes bunch of different data type, I cannot use every one of them in this project. So, I picked a goal which is building the ultimate Pokémon team including six different Pokémon, which is the maximum size of a Pokémon team. In this purpose, I am going to need stats, type and type advantages.

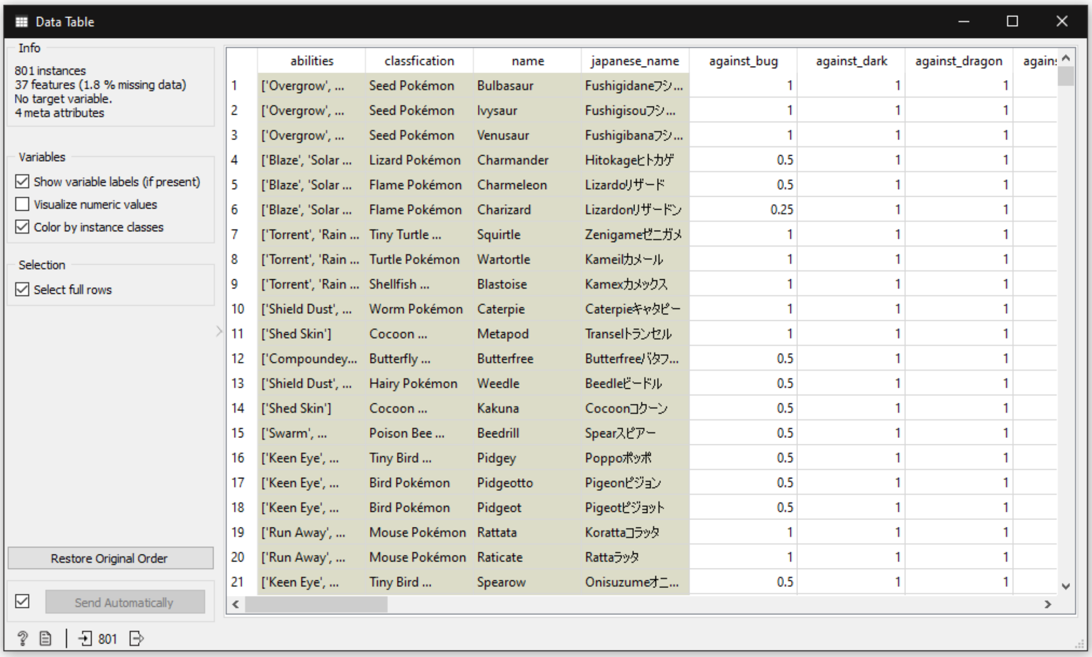
Pokémon battles are hugely focused on those three data types. Knowing your's and your opponent's Pokémon's type, and their relationship with each other, is the key to the victory. Yet, those Pokémon's stats are also having a very big role in the battle.

Let me give an example: You have a Fire-Type Pokémon and your opponent has a Grass-Type Pokémon, which is weak to Fire-Type, so it is going to take double damage and give half of its damage. However, assume your Pokémon have a 6 point in Attack Stat, while your opponent's Pokémon has a 26. (In this scenario, I am going to simplify the mechanic just to cover the basics of it.) So, when your Pokémon attacks, it simply gives 12 points of damage, while your opponent's Pokémon gives 13 points of damage.

3. TOOL

I choose Orange [3] as my tool for preparing the data and display. Because I think it has a simple and useful interface to work on. I started with opening "csv" file and get all the data it includes with Data Table feature.





Info

801 instances
37 features (1.8 % missing data)
No target variable.
4 meta attributes

Variables

☑ Show variable labels (if present)
☐ Visualize numeric values
☑ Color by instance classes

Selection

☑ Select full rows

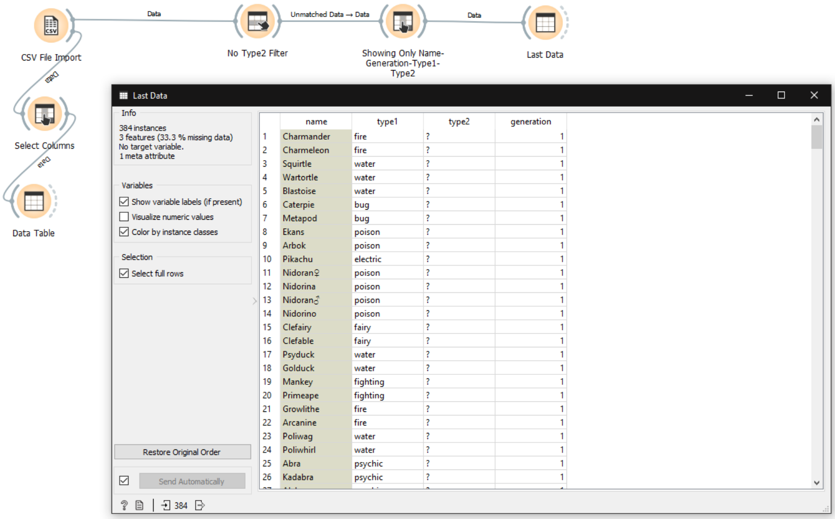
Restore Original Order

☑ Send Automatically

801

	abilities	classfication	name	japanese_name	against_bug	against_dark	against_dragon	again:
1	['Overgrow', ...	Seed Pokémon	Bulbasaur	Fushigidaneフシ...	1	1	1	
2	['Overgrow', ...	Seed Pokémon	Ivysaur	Fushigisouフシ...	1	1	1	
3	['Overgrow', ...	Seed Pokémon	Venusaur	Fushigibanaフシ...	1	1	1	
4	['Blaze', 'Solar ...	Lizard Pokémon	Charmander	Hitokageヒトカゲ	0.5	1	1	
5	['Blaze', 'Solar ...	Flame Pokémon	Charmeleon	Lizardoリザード	0.5	1	1	
6	['Blaze', 'Solar ...	Flame Pokémon	Charizard	Lizardonリザードン	0.25	1	1	
7	['Torrent', 'Rain ...	Tiny Turtle ...	Squirtle	Zenigameゼニガメ	1	1	1	
8	['Torrent', 'Rain ...	Turtle Pokémon	Wartortle	Kameilカメール	1	1	1	
9	['Torrent', 'Rain ...	Shellfish ...	Blastoise	Kamexカメックス	1	1	1	
10	['Shield Dust', ...	Worm Pokémon	Caterpie	Caterpieキャタピー	1	1	1	
11	['Shed Skin']	Cocoon ...	Metapod	Transelトランセル	1	1	1	
12	['Compoundey...	Butterfly ...	Butterfree	Butterfreeバタフ...	0.5	1	1	
13	['Shield Dust', ...	Hairy Pokémon	Weedle	Beedleビードル	0.5	1	1	
14	['Shed Skin']	Cocoon ...	Kakuna	Cocoonコクーン	0.5	1	1	
15	['Swarm', ...	Poison Bee ...	Beedrill	Spearスピアー	0.5	1	1	
16	['Keen Eye', ...	Tiny Bird ...	Pidgey	Poppoポッポ	0.5	1	1	
17	['Keen Eye', ...	Bird Pokémon	Pidgeotto	Pigeonピジョン	0.5	1	1	
18	['Keen Eye', ...	Bird Pokémon	Pidgeot	Pigeotピジョット	0.5	1	1	
19	['Run Away', ...	Mouse Pokémon	Rattata	Korattaコラッタ	1	1	1	
20	['Run Away', ...	Mouse Pokémon	Raticate	Rattaラッタ	1	1	1	
21	['Keen Eye', ...	Tiny Bird ...	Spearow	Onisuzumeオニ...	0.5	1	1	

Then, I tried to reach empty values by simply selecting all values and display the unmatched data.



Info

384 instances
3 features (33.3 % missing data)
No target variable.
1 meta attribute

Variables

☑ Show variable labels (if present)
☐ Visualize numeric values
☑ Color by instance classes

Selection

☑ Select full rows

Restore Original Order

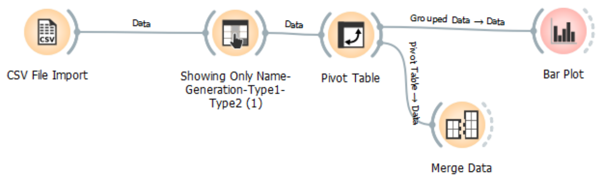
☑ Send Automatically

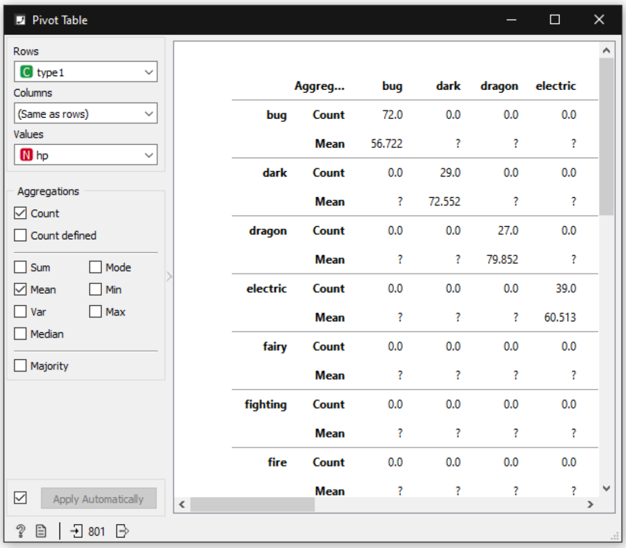
384

	name	type1	type2	generation
1	Charmander	fire	?	1
2	Charmeleon	fire	?	1
3	Squirtle	water	?	1
4	Wartortle	water	?	1
5	Blastoise	water	?	1
6	Caterpie	bug	?	1
7	Metapod	bug	?	1
8	Ekans	poison	?	1
9	Arbok	poison	?	1
10	Pikachu	electric	?	1
11	Nidoran♀	poison	?	1
12	Nidorina	poison	?	1
13	Nidoran♂	poison	?	1
14	Nidorino	poison	?	1
15	Clefairy	fairy	?	1
16	Clefable	fairy	?	1
17	Psyduck	water	?	1
18	Golduck	water	?	1
19	Mankey	fighting	?	1
20	Primeape	fighting	?	1
21	Growlithe	fire	?	1
22	Arcanine	fire	?	1
23	Poliwag	water	?	1
24	Poliwhirl	water	?	1
25	Abra	psychic	?	1
26	Kadabra	psychic	?	1

Unfortunately, I had a struggle with preparing the tables that I should display. Because even though Orange has a simple and a useful interface, it cannot perform any complicated process.

I aimed to get the average value of each stats Pokémon had but I cannot find a proper source that tells me how to do it other than using Python, R or any other scripting. Finally, I found a feature in the official website of Orange. Pivot Table calculates sum, mode, mean, min, max, var and median of any value that I pick.





Rows: type1

Columns: (Same as rows)

Values: hp

Aggregations

☑ Count
☐ Count defined

☐ Sum ☐ Mode
☑ Mean ☐ Min
☐ Var ☐ Max
☐ Median

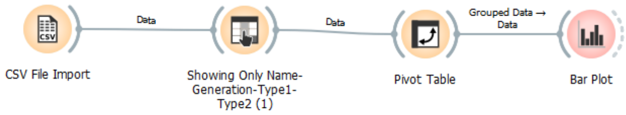
☐ Majority

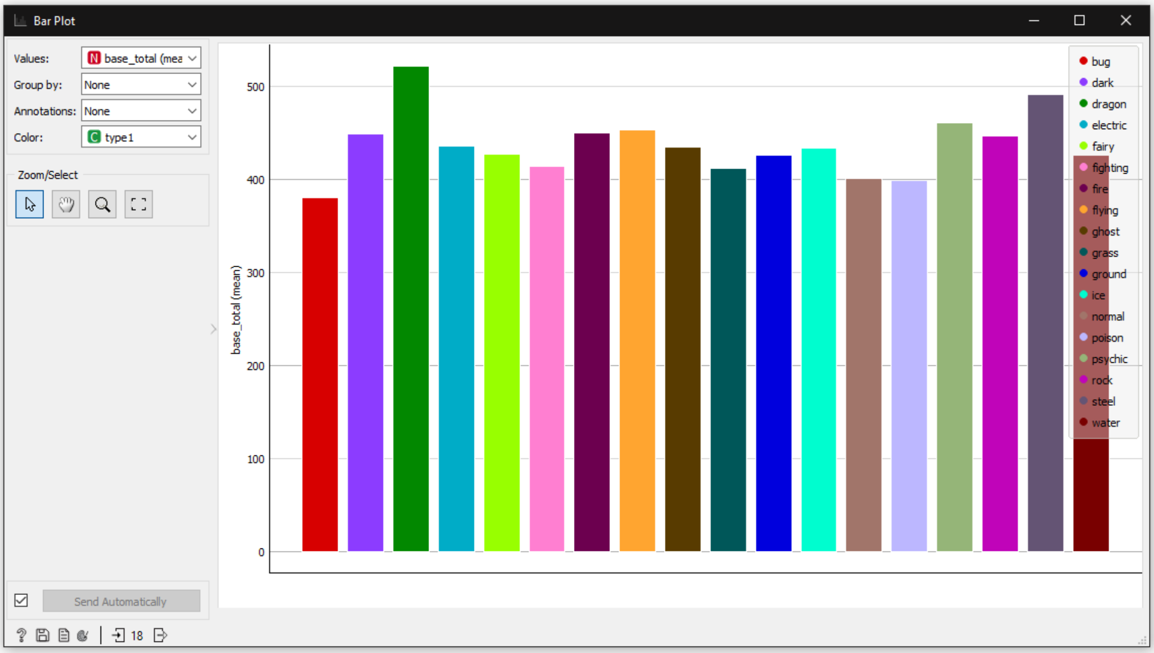
☑ Apply Automatically

801

	Aggreg...	bug	dark	dragon	electric
bug	Count	72.0	0.0	0.0	0.0
	Mean	56.722	?	?	?
dark	Count	0.0	29.0	0.0	0.0
	Mean	?	72.552	?	?
dragon	Count	0.0	0.0	27.0	0.0
	Mean	?	?	79.852	?
electric	Count	0.0	0.0	0.0	39.0
	Mean	?	?	?	60.513
fairy	Count	0.0	0.0	0.0	0.0
	Mean	?	?	?	?
fighting	Count	0.0	0.0	0.0	0.0
	Mean	?	?	?	?
fire	Count	0.0	0.0	0.0	0.0
	Mean	?	?	?	?

So, I picked the mean value of each stats that Pokémon have and prepared some box plots with Type1 attribute to see which type has the highest stats.




Bar Plot

Values: base_total (mea

Group by: None

Annotations: None

Color: type1

☑ Send Automatically

This table shows only the mean of total stats. I prepared a table for every stat. But unfortunately, I could not merge them into one table like I did in the coding. So, showing only the table of total stats would be enough. Since it shows the Pokémon-Type that has the highest stats. Thus, the details about the strongest Pokémon-Type are further below in “Codes” part.

4. CODES

```
library(tidyverse)
pkmn <- read_csv("pokemon.csv")
names(pkmn)
```

Now, we should start with getting the data from the “csv” file in the code.

```
##  [1] "abilities"          "against_bug"        "against_dark"
##  [4] "against_dragon"     "against_electric"   "against_fairy"
##  [7] "against_fight"     "against_fire"      "against_flying"
## [10] "against_ghost"     "against_grass"     "against_ground"
## [13] "against_ice"       "against_normal"    "against_poison"
## [16] "against_psychic"   "against_rock"      "against_steel"
## [19] "against_water"     "attack"            "base_egg_steps"
## [22] "base_happiness"    "base_total"        "capture_rate"
## [25] "classfication"     "defense"           "experience_growth"
## [28] "height_m"          "hp"                "japanese_name"
## [31] "name"              "percentage_male"   "pokedex_number"
## [34] "sp_attack"         "sp_defense"        "speed"
## [37] "type1"             "type2"             "weight_kg"
## [40] "generation"        "is_legendary"
```

Here we can see the dataset of all 802 Pokémon including 41 columns of attributes for each Pokémon.

```
sapply(pkmn, function(x) sum(is.na(x)))
```

Indeed, there is some missing data in some of the columns. I can check the missing columns by searching “NA” part of them. Which means “Not Available”.

```
##         abilities       against_bug      against_dark    against_dragon
##                 0                 0                 0                 0
##  against_electric     against_fairy     against_fight      against_fire
##                 0                 0                 0                 0
##    against_flying     against_ghost     against_grass    against_ground
##                 0                 0                 0                 0
##       against_ice    against_normal    against_poison   against_psychic
##                 0                 0                 0                 0
##      against_rock     against_steel     against_water            attack
##                 0                 0                 0                 0
##    base_egg_steps    base_happiness        base_total      capture_rate
##                 0                 0                 0                 0
##     classfication           defense experience_growth          height_m
##                 0                 0                 0                20
##                hp     japanese_name              name   percentage_male
##                 0                 0                 0                98
##    pokedex_number         sp_attack        sp_defense             speed
##                 0                 0                 0                 0
##             type1             type2         weight_kg        generation
##                 0               384                20                 0
##      is_legendary
##                 0
```

There are some missing values in Weight, Height, Percentage of Being Male and Type2 attributes. Especially in Type2 attribute, almost half of the Pokémon has no values. It is simply because almost half of the Pokémon have 1 primary type, therefore their Type2 attribute have NA as its value. So, the missing values basically means that Pokémon does not have that attribute at all. Thus, excluding Type2 attribute, I can assume that I am not going to need these attributes while looking for the ultimate team.

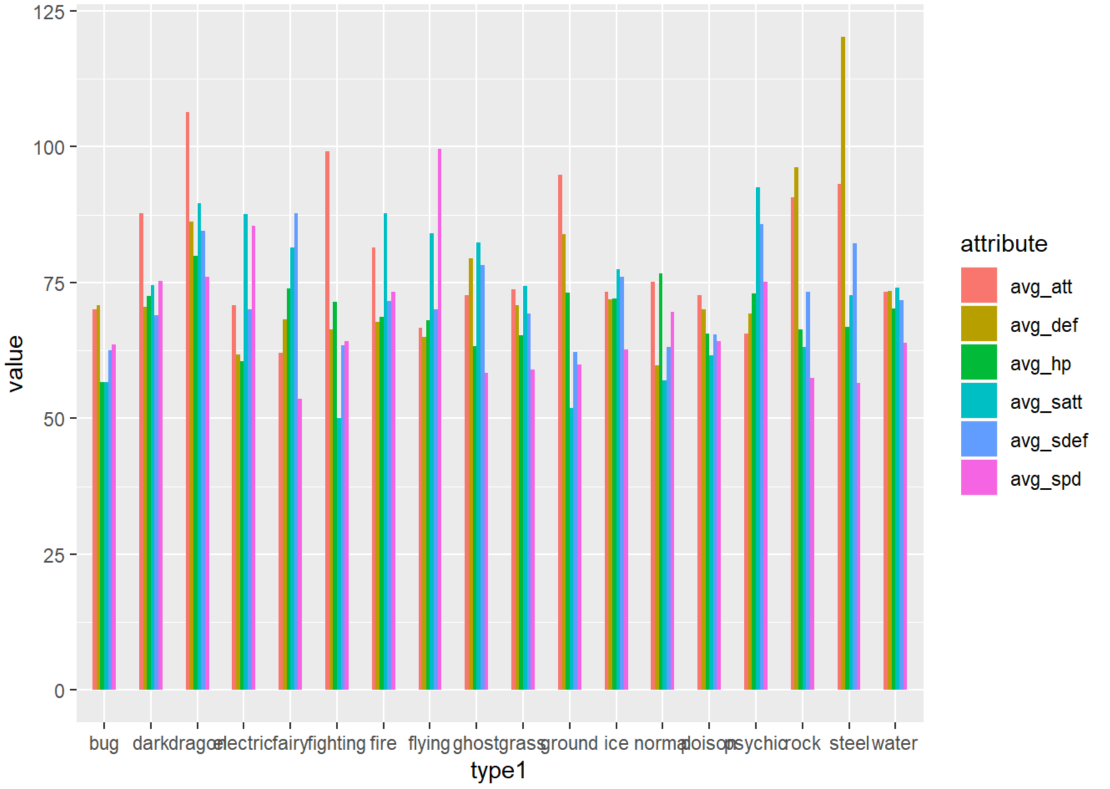
"Which attribute should I use to get the strongest team?"

This question is essential to reach the solution. In the battle mechanic of the Pokémon games, there are several attributes that should be considered as core of the mechanic:

- Stats: Higher stats equals higher attack and defense.
- Type Relations: Having less weaknesses and more resistances in the Type Chart makes Pokémon resist on more attacks and give more damage.
- Capture Rate: Every Pokémon has a capture rate, which relates to their power. The stronger Pokémon gets, harder it is to be caught.

```
pkmn %>%
  group_by(type1) %>%
  summarise(
    avg_hp = mean(hp),
    avg_att = mean(attack),
    avg_def = mean(defense),

    avg_satt = mean(sp_attack),

    avg_sdef = mean(sp_defense),

    avg_spd = mean(speed)

  ) %>%

  gather(`avg_hp`,`avg_att`,`avg_def`,`avg_satt`,`avg_sdef`,`avg_spd`, key=
"attribute", value="value") %>%

  ggplot() +

      geom_col(aes(x=type1, fill=attribute, y=value), position='dodge', wid
th=0.5)
```

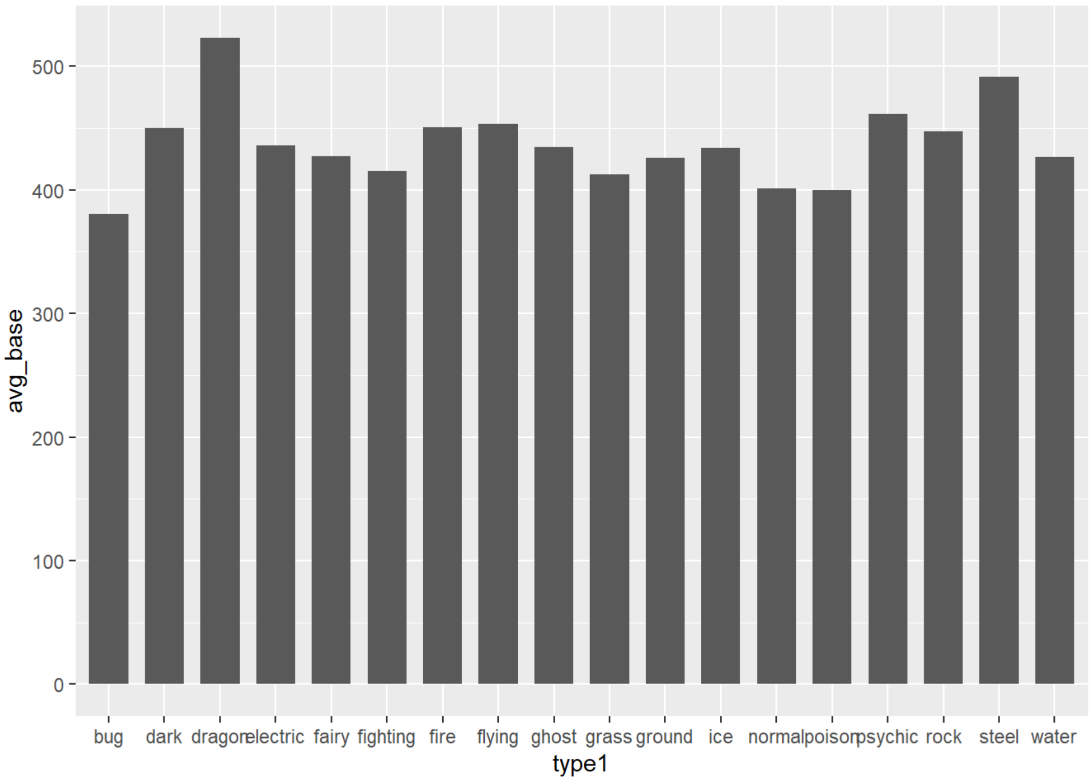
Here I compared each Type with 6 different stats in the games which is Attack, Defense, Special Attack, Special Defense, Speed and Hit Points.



With this data, I see that Dragon-Type Pokémon have the highest Attack while Steel-Type have even higher Defense stats. Yet these are not enough to define a Pokémon as strong. Even though it has the highest attack. If it has a low defense and got hit by a faster Pokémon to be knocked out, it is not going to matter for it to have a high Attack stat.

```
pkmn %>%
  group_by(type1) %>%
  summarise(
    avg_base = mean(base_total),
  ) %>%
    ggplot() +
      geom_col(aes(x=type1, y=avg_base), width=0.7)
```

Here I made a plot that shows Pokémon typing with total stats. “base_total” in the code means the total sum of all stats.



So basically, if a Pokémon has a high Attack but Low Defense, while another one has a decent Attack and a decent Defense. Second one is going to be considered as stronger than the first one. Yet Dragon-Type and Steel-Type can be considered as the strongest type of all again. With this I can be sure that

Dragon-Type is the strongest, while Steel-Type is coming after it. However, those types are not invincible.

I cannot build a team includes only Dragon-Type and/or Steel-Type Pokémon. If I would do that way, then my team would have a huge weakness against certain type(s). To prevent having such weakness, I should check the resistance rate of each Pokémon against all types. Since there are 18 unique types, Pokémon's resistance against it is going to have more resistance points. So, the lower resistance points mean more defensive the Pokémon is.

```
pkmn = pkmn %>%

  mutate(tot_resist = against_bug+against_dark+against_dragon+against_elect
ric+against_fairy+against_fight+against_fire+against_flying+against_ghost+a
gainst_grass+against_ground+against_ice+against_normal+against_poison+again
st_psychic+against_rock+against_steel+against_water)
```

This is not going to give me an exact solution since some Pokémon having two types brings out different resistance. However, it is going to give me enough information to use.

5. RESULTS

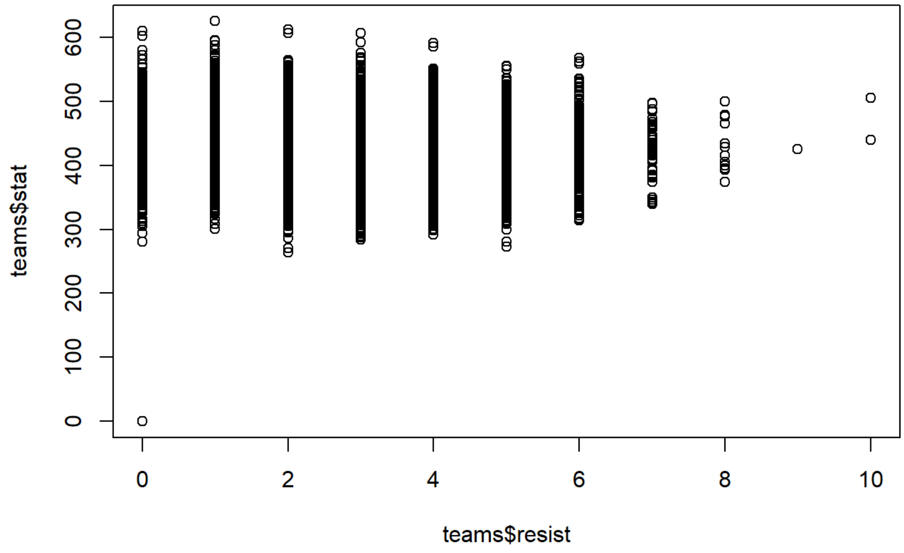
Building the ultimate team is not an easy task. Yet I gathered all the data I am going to need. (PS: I am going to ignore moves, abilities and synergy between Pokémon since those are alters drastically of a Pokémon's effectiveness.)

With all this information I gathered, it is time to write an algorithm that consists of:

- Calculates the team's total stat and selects the highest stat. (base_total)
- Checks for having at least one Pokémon for each weakness. (against_type)

- Gets a point for each weakness that could not be covered and keep the sum at minimum. (against_type)
- Store 6 Pokémon to have these conditions in a data. (data.frame)
- Repeat this algorithm until it finds the ultimate team.

```
teams = data.frame(
  p1 = 0,
  p2 = 0,
  p3 = 0,
  p4 = 0,
  p5 = 0,
  p6 = 0,
  resist = 0,
  stat = 0)

set.seed(565)
for (i in 1:10000) {
  team = pkmn %>%
    dplyr::select(33, 2:19, 23) %>%
    sample_n(6)

  team_resist = team %>%
    dplyr::select(2:19) %>%
    t() %>%
    as.data.frame() %>%
    mutate(V7 = ifelse(
      V1 < 1, 0, ifelse(
        V2 < 1, 0, ifelse(
          V3 < 1, 0, ifelse(
            V4 < 1, 0, ifelse(
              V5 < 1, 0, ifelse(
                V6 < 1, 0, ifelse(
                  V1 == 1 & V2 == 1 & V3 == 1 & V4 == 1 & V5 == 1 & V6 ==
1, 0, 1
                )
              )
            )
          )
        )
      )
    )
    )
  team_resist = sum(team_resist$V7)
  team_stat = mean(team$base_total)

  new = c(team$pokedex_number[1],
    team$pokedex_number[2],
    team$pokedex_number[3],
    team$pokedex_number[4],
    team$pokedex_number[5],
    team$pokedex_number[6],
    team_resist,
    team_stat
  )

  teams = rbind(teams, new)
}

plot(teams$resist, teams$stat)
```



Now, I should get the top teams with the highest stats.

```
teams %>%
  filter(resist == 0, stat > 0) %>%
  arrange(desc(stat)) %>%
  top_n(10)
```

With this, I should be able to get the top 10 team. And the result is:

p1 <dbl>	p2 <dbl>	p3 <dbl>	p4 <dbl>	p5 <dbl>	p6 <dbl>	resist <dbl>
612	635	681	254	485	382	0
491	792	306	342	445	465	0
91	149	493	462	699	377	0

p1	p2	p3	p4	p5	p6	resist
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
658	497	306	308	768	460	0
677	146	598	80	487	248	0
658	344	125	715	212	245	0
244	483	237	474	103	466	0
631	384	295	760	467	476	0
82	733	381	716	398	455	0
364	34	282	523	248	169	0

Great! I have a list of 10 teams that has the highest total stat while having the least amount of resistance. Finally, I am going to choose the first team and get their data's shown.

```
pkmn %>%
  slice(612, 635, 681, 254, 485, 382) %>%
  dplyr::select(name, type1, type2, base_total, tot_resist, is_legendary)
```

name	type1	type2	base_total	tot_resist
<chr>	<chr>	<chr>	<dbl>	<dbl>
Haxorus	dragon	*NA*	540	19.00
Hydreigon	dark	dragon	600	21.00
Aegislash	steel	ghost	520	14.25

name	type1	type2	base_total	tot_resist
<chr>	<chr>	<chr>	<dbl>	<dbl>
Sceptile	grass	*NA*	630	21.00
Heatran	fire	steel	600	16.25
Kyogre	water	*NA*	770	18.00

6. DISCUSSION AND CONCLUSIONS

Trying to build the ultimate team was a difficult task while building a competitive team in games is already hard enough. Yet I tried my best to examine this dataset and evaluate the data that I need to use to reach my goal. In this project, I picked the attributes that is essential to use by having their graphics of certain situations to make sure that they are affecting to build a strong team. Then I used an algorithm to use these data and get the strongest team just to pick one of them.

The team is consisting of having multiple type while keeping their stats at maximum. Thus, covering each other weakness by having a counter-typing. However, in the games this cannot be the exact solution. Because there are many more attributes that affects the team's competitiveness such as items, abilities, synergy etc. Yet getting the ultimate team with including these facts would be near impossible. So, I have tried my best to reach my goal using the dataset as much as I can.

REFERENCES

[1] https://www.pokemon.com/us/

[3] https://www.britannica.com/topic/Pokemon-electronic-game

[3] https://orangedatamining.com/