|  |  |  |
| --- | --- | --- |
| Checkpoint II | Checkpoint II: Data Cleaning & Processing | |
| Group: | G11 |
| Date: | 2022/10/10 |
|  |  |

# Initial Dataset

19 joint datasets, all regarding information related to the competitive Pokémon video game in 2022.

The total size of the datasets was (#Items x #Attributes): 20 \* 2 + 1099 \* 22 + 25 \* 5 + 822 \* 9 + 840 \* 7 + 268 \* 3 + 325 \* 4 + 1686 \* 2 + 3411 \* 3 + 153 \* 2 + 555 \* 2 + 772 \* 3 + 6819 \* 4 + 69105 \* 2 + 3271 \* 4 + 1892 \* 3 + 2176 \* 3 + 822 \* 2 + 33 \* 2 = 249546.

The total size of the current dataset is (derived measures included) (#Items x #Attributes): 983 \* 12 (df\_pokemon.csv) + 769 \* 9 (df\_moves) + 3144 \* 7 (df\_used\_with\_move) = 40716

Data sample:

(from “df\_pokemon.csv”)

Pokemon; Generation; Type1; Type2; Total; HP; […]; Monthly Usage (k)

Venusaur; 1; Grass; Poison; 525; 80; […]; 204

# Selected/Derived Data

The selected data is already described in the “Data Abstraction” section, along with its abstraction. The dataset is static, partly tabular partly multidimensional table (keys Move, Pokémon). The data is items (Pokémon, Moves, and their relation to each other).

Data derived :

* PP, Accuracy, and Power percentile of a Move (purpose: better compare the different ranges of these Move attributes) (computation: used Excel to compute percentile based on all values of the given attribute)
* Distribution of a given Stat (Total, HP, Speed…) per Pokémon Type(s) (purpose: compare Stats by Type(s)) and/or per Generation (purpose: compare values per generation) (computation: done by d3, group Pokémon by relevant attribute and compute average)
* Monthly Move Use Per Pokémon Type(s) per Damage Class (purpose: compare frequency of move Damage Class per Pokémon Type(s)) (computation: done by d3, groups Moves by
* (Pokémon) Monthly Usage per Pokémon Type(s) (purpose: compare frequency of use of each Type(s) in competition) (computation: done by d3, group Pokémon by Type(s), sum Monthly Usage)

|  |  |  |
| --- | --- | --- |
| Category | Data | Semantics |
| Ratio,  Sequential | df\_pokemon: Stats | Statistics of a Pokémon (Total, HP, Speed, …). Influence move power. |
| df\_moves: Power, Accuracy, PP | Attributes of a Pokémon Move (PP = Max. Uses Per Battle) |
| df\_moves: Percentiles | Percentile of a Pokémon Move’s attributes (Power, PP, Accuracy) |
| df\_moves: Use Percentage | Use percentage of a Pokémon Move for a given Pokémon |
| Average Stats | Average Stats of a Pokémon. Also grouped by Type(s) and/or Generation. |
| df\_used\_with\_move: Monthly Move Use | Monthly Usage of a Move per Pokémon. Also grouped by Pokémon Type(s) and Move Damage Class. |
| Nominal | df\_pokemon: Pokemon; df\_moves: Move; | Keys of a Pokémon (key: Pokemon) or Pokémon Move (key: Move) |
| df\_moves: Damage Class | Category of a Pokémon Move |
| df\_pokemon/df\_moves: Type | Category of a Pokémon or of a Pokémon Move |
| Ordinal | df\_pokemon: Generation | Era of Pokémon Games associated; can be used to build a timeline |

# Data Processing

We used Excel and pgAdmin to compute, select and sort through data. Datasets and attributes that were less interesting/relevant to the visualization were removed: Moves and Pokémon-related attributes were deemed most relevant. For missing Nominal values, we used a Sentinel value of “NULL”, if the attribute was non-applicable, or a researched Impute value otherwise. For missing Ratio values, we used Impute values (ex: “0” for the Monthly Usage(k) attribute) or Sentinel values (ex: “101” for the Accuracy attribute when infinite). We discarded items with Dynamax-related attributes (as their Ratio values were outliers), and Pokémon Restricted or Banned from competition (they mislead while comparing all Pokémon to the Pokémon used by competitive players). Other values for Nominal attributes were considered valid. No other Ratio outliers were identified with the Standard Deviation method. We used the attributes Pokemon, Move, and Type(1,2) as cross-reference keys to multiply the (Move) Use Percentage (df\_used\_with\_move) by the (Pokémon) Monthly Usage (df\_pokemon) to obtain the “Monthly Move Use” per Pokemon (df\_used\_with\_move), associated with a (Move) Damage Class and the Pokémon’s Type(1,2) (df\_pokemon).

# Mapping (Data sample/Questions)

|  |  |
| --- | --- |
| ID | Data |
| 1 | (from “df\_pokemon.csv”)  Pokemon; Generation; Type1; Type2; Total; HP; […]; Monthly Usage (k)  Venusaur; 1; Grass; Poison; 525; 80; […]; 204 |
| 2 | (from “df\_moves.csv”)  Move; Type; Power; Accuracy; PP; Damage Class; Power Percentile; Accuracy Percentile; PP Percentile  Absorb;Grass;20;100;25;Special;30;50;40;130 |
| 3 | (from “df\_used\_with\_move.csv”)  Move;Use Percentage,Pokemon; Monthly Move Use; Damage Class; Type1; Type2  Behemoth Blade; 99.996; Zacian Crowned Sword,1087.95648; Physical; Sword; Fairy; Steel |

|  |  |  |
| --- | --- | --- |
| Question | DataID | Usage |
| How has the HP Stat of Poison-Type Pokémon Evolved throughout generations? | 1 | d3 groups the HP stat of Pokémon by generation; its values per Generation may then be compared |
| How does a Pokémon’s Type (and number of Types) influence its Stats, and the Stats prioritized by competitive players? | 1 | d3 groups the values of each Stat by Type(1,2) and by Monthly Usage of its Pokémon: Stats of Pokémon with an M.O. greater than 0 (competitive Pokémon), and of an M.O. greater than or equal to 0 (all Pokémon); these values are compared |
| How does a Pokémon’s Type influence the Damage Class of Moves chosen by competitive players? | 3 | d3 can group the Moves by the Type(s) of the Pokémon who use them, then by Damage Class of the Move; the Monthly Move Use is compared |
| Do competitive players prioritize Move availability (PP) or Move Power, when choosing Moves for their Pokémon? | 2, 3 | d3 can only select moves with a Monthly Move Use over 0; their PP and Power percentiles are then compared, and Monthly Move User summed |
| Which Type combinations are preferred by competitive players? | 2 | d3 can sum Monthly Usage (k), grouped by Type combination; these sums may then be compared |