Data Analysis and Visualization Exercise 3, Data Wrangling

Vicente Yépez, Žiga Avsec

5 November 2019

Setup

```r
library(data.table)
library(magrittr)   # Needed for %>% operator
library(tidyr)
library(readxl)
library(dplyr)
DATADIR <- "../extdata"
```

Questions

Section I. Reading and cleaning up data

1. Read the pokemon.xlsx file. Assign the contents of each of the four sheets (tabs) into different data.tables: poke_dt, moves_dt, evolution_dt, typeChart_dt. Open the data.tables and understand the data they contain. *Hint:* read_excel from the readxl package.
2. Check the columns' names and classes of poke_dt. Rename the column "#" as "Number". Replace the spaces in the columns' names with underscores. *Hint:* colnames(), setnames(), gsub()
3. Note that the "Number" column is a character vector that sometimes begins with a space. To remove this space use the following:

```r
# poke_dt[, Number := gsub(intToUtf8(160),'', Number)]
```

Then, change the class of "Number" from character to integer and "Type" from character to factor.

Hint: as.integer(), as.factor()

4. Remove all 'Mega' pokemons. Then, subset to include pokemon from the first generation only (from Bulbasaur #1 until Mewtwo #150). *Hint:* grep()

Section II. Data exploration

1. Which are all the types of pokemons? How many pokemon are there per type?
2. Which are all the Ice pokemon?
3. Create a character vector called stats that contains the column names of the pokemons stats (HP, attack, etc.). Which is the maximum value of each stat? First, think how would it be just for one, eg. HP. *Hint:* sapply (or .SD)

```
stats <- c("HP", "Attack", "Defense", "Special_Attack", "Special_Defense", "Speed", "Total")
```

4. Remember that R has the functions `colMeans` and `colSums`. Create the function `colMax(DT)` that takes as input a data.table and returns a named vector with the maximum value of all columns. Use it to obtain the same results as in the previous exercise.
5. Go to the help page of `which.min`. You will see that this function returns the index of the **first** minimum of a vector. Create the functions `which_min` and `which_max` whose input is a numeric vector and output is a vector with **all** the indices where the minimum and maximum values are located. *Hint:* which()
6. Of each Type, which are the: first, strongest and weakest pokemon? We define 'strongest' as the one with the highest Total value. *Hint*: .SD, which_max()
7. On average, which is the strongest Type? Specifically, give a sorted data.table with all pokemon types and their 'Total' means. *Hint*: order()

Section III. Merging and manipulating

1. In evolution_dt, rename the columns 'Evolving from' -> 'Name' and 'Evolving to' -> 'Evolution'. Subset it in order to include only the pokemons from the first generation (As subsetted in question section I question 4).
2. Which pokemons neither evolve nor are an evolution from others? *Hint*: There's no need to merge poke_dt with evo_dt, we are simply interested in the pokemon that are not in `evo_dt$Name` or `evo_dt$Evolution`.
3. Add a column to the poke_dt with the 1st generation evolutions of each pokemon (only the first one) and the level it requires to evolve. *Hint*: merge()
4. Which pokemon requires the highest level to evolve? By type?
5. Now let's examine `moves_dt`. It contains all moves (or attacks) that pokemons can do. Check the columns' classes. Try to explain the first row.
6. Expand the names (use the full names) of the columns Cat., Acc. and Prob. Change Accuracy class to numeric.
7. Which are the different categories of the moves?
8. Which are all the Grass attacks?
9. Assuming that a pokemon can only perform attacks based on its type(s), which are all the attacks that eg. Bulbasaur can do? There is no need to merge poke_dt with moves_dt.

10. Add a column `Real_Power` with the real power of an attack which we define as Power * Accuracy %. Which is the most powerful attack of all? Which are the most powerful attacks per type and category? *Hint:* by = .(Category, Type)

11. Open typeChart_dt. Check classes. Fairy attacks are effective (i.e. Multiplier > 1) against which defense types?

12. We use the sum of multiplier to evaluate the effectiveness of one type. Which is the most effective type, when attacking, against others? Which is the most effective, when defending, against others? Specifically, provide a sorted data.table with all the types and the sum of effectiveness for attack and another data table for defense.

Section IV. Joining all knowledge

1. Make a function called `best_attack()` that receives the name of 2 pokemons (attacker and defender) and returns the most powerful attack that the first pokemon should do (Power * Accuracy * Effectiveness Multiplier). For simplicity, consider that the defending pokemon only has its first type. Ignore 'Status' moves as they don't inflict any damage. Try some examples like: best_attack("Gastly", "Tangela"), best_attack("Pikachu", "Bellsprout")