Linear Models in R (M1–MIDO)
Lab Session 4 — Student Sheet

Henri PANJO

Table of contents

Dataset Overview: *data_pokemon.csv*

This dataset is adapted from a popular Kaggle Pokémon dataset.
Even if you are not familiar with Pokémon, the data is straightforward:
it combines numeric statistics with categorical attributes, making it well-suited for applying **Ordinary Least Squares (OLS)** in R.

What it contains

- Unique identifiers and names for each Pokémon
- Battle statistics (health, attack, defense, special attack, special defense, speed)
- Categorical features (primary/secondary type, generation, legendary flag)

Fields (Codebook)

- `id`: Unique Pokémon ID
- `name`: Pokémon name
- `type_1`: Primary type (e.g., Water, Fire)
- `type_2`: Secondary type (optional)
- `hp`: Hit points (overall health)
- `attack`: Physical attack strength (**we will use this as** y in most regressions)
- `defense`: Physical defense strength
- `sp_attack`: Special (non-physical) attack strength
- `sp_defense`: Special defense strength
- `speed`: Speed / turn order
- `generation`: Game generation label
- `legendary`: Indicator for legendary status (TRUE/FALSE)

Note on notation

- We treat `attack` as the outcome variable Y.
- Predictor variables (e.g., `defense`, `speed`) will be denoted as $x_1, x_2, \ldots$.
- Factors like `type_1` or `legendary` will be included as categorical predictors.

Setup

To keep numbers readable and reproducible, we set display options:

```
options(scipen = 999, digits = 5)
```

We also load the packages used during this session.

> ⚠ Warning
>
> Don't worry if you don't know them all — we'll introduce functions as we need them. Some provide regression tools, others are for data visualization or diagnostics.

```
library(broom)
library(performance)
library(parameters)
library(datawizard)
library(see)
library(effectsize)
library(insight)
library(correlation)
library(modelbased)
library(glue)
library(scales)
library(GGally)
library(ggpubr)
library(car)
library(lmtest)
library(multcomp)
library(rstatix)
library(matrixTests)
library(ggfortify)
library(qqplotr)
library(patchwork)
library(ggrepel)
library(gtsummary)
library(kableExtra)
library(openxlsx)
library(janitor)
library(marginaleffects)
library(collapse)
library(tidyverse)
```

```
source("helper_functions4.R")
```

Data management

We first load the dataset and create the same variables as lab session 3: `typeg`, `second_type`
We also transform the variables `legendary` and `generation` into `factors`.

- Load the data

```
pok <- read_csv("data_pokemon.csv", show_col_types = FALSE)
```

- Define the 3-level grouping map

```
type_map3 <- list(
  elemental_env = c("Fire", "Water", "Grass", "Electric", "Ice", "Flying", "Poison"),
  physical_material = c("Bug", "Fighting", "Ground", "Rock", "Steel", "Normal"),
  mystical_supernatural = c("Psychic", "Ghost", "Dragon", "Fairy", "Dark")
)
```

- Variable creation with `mutate()` from `{dplyr}` and labelling with `relabel()`

```
pok <- pok |>
  mutate(
    typeg = case_when(
      type_1 %in% type_map3$elemental_env ~ "Elemental", # "Elemental",
      type_1 %in% type_map3$physical_material ~ "Physical", # "Physical",
      type_1 %in% type_map3$mystical_supernatural ~ "Mystical", # "Mystical",
      .default = NA_character_
    )
  ) |>
  mutate(second_type = ifelse(type_2 == "None", 0, 1) |> factor(labels = c("No", "Yes"))) |>
  mutate(typeg = fct_infreq(typeg), legendary = as.factor(legendary)) |>
  mutate(generation = factor(generation, labels = paste0("Gen", 1:6))) |>
  relabel(
    typeg = "Primary Type", second_type = "Secondary type",
    legendary = "Legendary", generation = "Pokemon generation",
    attack = "Attack power", speed = "Speed power", defense = "Defense power",
    hp = "Hit points (health)", sp_attack = "Special attack power",
    sp_def = "Special defense power", id = "ID", name = "Pokemon name"
  )
```

- We use `tab_freq1()` from `helper_functions4.R` to get `factor` distributions

```
tab_freq1(pok, c("typeg", "second_type", "legendary", "generation"), digits = 1) |>
  kable(align = "l", padding = 2) |>
  row_spec(c(1, 5, 8, 11), bold = TRUE)
```

Variable	Count (n)	Percent (%)
Primary Type		
Elemental	334	41.8%
Physical	297	37.1%
Mystical	169	21.1%
Secondary type		
No	386	48.2%
Yes	414	51.7%
Legendary		
No	735	91.9%
Yes	65	8.1%
Pokemon generation		
Gen1	166	20.8%
Gen2	106	13.2%
Gen3	160	20.0%
Gen4	121	15.1%
Gen5	165	20.6%
Gen6	82	10.2%

Question 1: Attack mean with respect to `second_type` and `typeg`

Plot the mean value of `attack` for each combination of `second_type` and `typeg`.

Question 2: Effect of speed on attack for each primary type

Create separate scatter plots of `attack` versus `speed` for each Pokémon type (`typeg`).

Hint: You can use `scatter_plot()` (`helper_functions4.R`)

Question 3: Interaction between 2 categorical variables (1)

1. Fit the two models `mod_main1` and `mod_interact1` below, and interpret estimated coefficients.

$$\texttt{attack} = \beta_0 + \beta_1 \texttt{second\_type}_{\texttt{Yes}} + \beta_2 \texttt{typeg}_{\texttt{Physical}} + \beta_3 \texttt{typeg}_{\texttt{Mystical}} + \varepsilon$$

$$\begin{aligned}\texttt{attack} = \beta_0 + \beta_1 \texttt{second\_type}_{\texttt{Yes}} + \beta_2 \texttt{typeg}_{\texttt{Physical}} + \beta_3 \texttt{typeg}_{\texttt{Mystical}} + \\ \beta_4 \texttt{second\_type}_{\texttt{Yes}} \times \texttt{typeg}_{\texttt{Physical}} + \beta_5 \texttt{second\_type}_{\texttt{Yes}} \times \texttt{typeg}_{\texttt{Mystical}} + \varepsilon\end{aligned}$$

2. Determine whether `mod_interact1` provides a significantly better fit than `mod_main1`. (Equivalently: test whether the interaction terms are jointly equal to zero.)

Question 4: Interaction between 2 categorical variables (2)

Run the code below and interpret the coefficients

```r
mod_interact1bis1 <- lm(attack ~ typeg + second_type:typeg, data = pok)
mod_interact1bis2 <- lm(attack ~ second_type + typeg:second_type, data = pok)
```

Question 5: Interaction between a categorical and a continuous

We now fit the 3 following models. Run the code below and interpret the estimated coeffcients.

$$\texttt{attack} = \beta_0 + \beta_1 \texttt{speed} + \beta_2 \texttt{typeg}_{\texttt{Physical}} + \beta_3 \texttt{typeg}_{\texttt{Mystical}} + \varepsilon$$

$$\texttt{attack} = \beta_0 + \beta_1 \texttt{speed} + \beta_2 \texttt{typeg}_{\texttt{Physical}} + \beta_3 \texttt{typeg}_{\texttt{Mystical}} + \beta_4 \texttt{speed} \times \texttt{typeg}_{\texttt{Physical}} + \beta_5 \texttt{speed} \times \texttt{typeg}_{\texttt{Mystical}} + \varepsilon$$

$$\begin{aligned}\texttt{attack} = \beta_0 + \beta_1 \texttt{typeg}_{\texttt{Physical}} + \beta_2 \texttt{typeg}_{\texttt{Mystical}} + \beta_3 \texttt{typeg}_{\texttt{Elemental}} \times \texttt{speed} + \beta_4 \texttt{typeg}_{\texttt{Physical}} \times \texttt{speed} + \\ \beta_5 \texttt{typeg}_{\texttt{Mystical}} \times \texttt{speed} + \varepsilon\end{aligned}$$

```
mod_main2 <- lm(attack ~ speed + typeg, data = pok)
mod_interact2 <- lm(attack ~ speed * typeg, data = pok)
mod_interact2bis <- lm(attack ~ typeg + speed:typeg, data = pok)
```