

Linear Models in R (M1–MIDO)

Lab Session 3 – Student Sheet

Henri PANJO

Table of contents

Dataset Overview: <i>data_pokemon.csv</i>	3
Setup	4
Question 1. Loading dataset	5
Question 2: Exploring categorical variables	5
Question 3: Data management, variable creation	5
Question 4: Box plot	6
Question 5: Attack mean over group variables	6
Question 6: Dummy variables creation	7
Question 7: Using dummy variables in OLS regression	7
Question 8: Testing equality of coefficients	8
Question 9: Predicting mean attack for all category combinations	8
Question 10. Residual diagnostics	9

Dataset Overview: `data_pokemon.csv`

This dataset is adapted from a popular Kaggle Pokéémon dataset.

Even if you are not familiar with Pokéémon, 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 Pokéémon
- Battle statistics (health, attack, defense, special attack, special defense, speed)
- Categorical features (primary/secondary type, generation, legendary flag)

Fields (Codebook)

- `id`: Unique Pokéémon ID
- `name`: Pokéémon 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, \dots .
- 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(gtsummary)
library(kableExtra)
library(openxlsx)
library(janitor)
library(marginaleffects)
library(collapse)
library(tidyverse)

source("helper_functions3.R")
```

Question 1. Loading dataset

Import the dataset `data_pokemon.csv` with `read_csv()` and save it in an object called `pok`.

Using `select()`, keep only the variables `id`, `name`, `attack`, `speed`, `defense`, `hp`, `sp_attack`, and `sp_def`.

Display the first 10 rows of `pok` using `head()` or `slice()`.

Question 2: Exploring categorical variables

The Pokémon dataset contains 4 categorical variables

- `type_1` : the primary type (always present)
- `type_2` : the secondary type (may be missing)
- `legendary`: if the pokemon is legendary or not.
- `generation`: the pokemon generation.

1. Create frequency tables for `type_1`, `type_2`, `legendary` and `generation`. Display both the counts and the relative proportions.
2. Produce bar plots for the distributions of `type_1` and `type_2`.
 - Make one bar plot for `type_1` and one for `type_2`.
 - Ensure that categories on the x-axis are readable (e.g., rotate labels if necessary).

Question 3: Data management, variable creation

The variable (`type_1`), has 18 levels, which can be too many to include directly in an OLS regression. To simplify the analysis, we want to group these 18 types into 3 broader, meaningful groups:

- 1.Elemental/Environmental: Fire, Water, Grass, Electric, Ice, Flying, Poison
- 2.Physical/Material: Bug, Fighting, Ground, Rock, Steel, Normal
- 3.Mystical/Supernatural: Psychic, Ghost, Dragon, Fairy, Dark

1. In the `pok` dataframe, create a new factor variable called `type_group3` that assigns each Pokémon to one of the 3 groups above based on its primary type (`type_1`).
2. Create a new binary variable called `has_secondary_type` defined as follows:
 - “Yes” if the Pokémon has a secondary type (`type_2` is not “None”)
 - “No” if the Pokémon does not have a secondary type (`type_2` is “None”)
3. Transform the variables `legendary` and `generation` into factors.
4. Verify that the new variables are well created

Question 4: Box plot

We now want to explore how the Pokémon attack distribution varies across several categorical variables in the dataset. Boxplots are useful for comparing the distribution of a numeric variable across groups.

Using the Pokémon dataset, create four separate boxplots where the response variable is attack, and the grouping variables are `type_group3`, `has_secondary_type`, `legendary`, `generation`.

Produce the four boxplots either separately or arranged in a multi-panel layout (your choice).

Question 5: Attack mean over group variables

You have explored the distribution of the variable `attack` using boxplots. We now want to summarize these differences numerically by computing the mean Attack score for several grouping variables.

Using the updated Pokémon dataset, compute the mean value of `attack` for each level of the following categorical variables:

- `type_group3` (3-level grouped primary type)
- `has_secondary_type` (“No” / “Yes”)
- `legendary` (“No” / “Yes”)
- `generation` (“Gen1” to “Gen6”)

For each variable, produce a summary table showing at least:

- the group name
- the mean of `attack`
- the standard deviation of `attack`
- the number of observations in each group

Hint: Use `mean_by_group()` from `helper_functions3.R`

Question 6: Dummy variables creation

In the Pokémon dataset create these variables

$$\text{legend1} = \begin{cases} 1 & \text{if } \text{legendary} = \text{"Yes"} \\ 0 & \text{otherwise} \end{cases} \quad \text{legend0} = \begin{cases} 1 & \text{if } \text{legendary} = \text{"No"} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{typeg1} = \begin{cases} 1 & \text{if } \text{type_group3} = \text{"1.Elemental/Environmental"} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{typeg2} = \begin{cases} 1 & \text{if } \text{type_group3} = \text{"2.Physical/Material"} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{typeg3} = \begin{cases} 1 & \text{if } \text{type_group3} = \text{"3.Mystical/Supernatural"} \\ 0 & \text{otherwise} \end{cases}$$

Hint: use `ifelse()` or any other functions/methods

Question 7: Using dummy variables in OLS regression

You have created the dummy variables `legend1`, `legend0`, `typeg1`, `typeg2`, and `typeg3`, which encode information about whether a Pokémon is Legendary and which primary type-group it belongs to.

We now want to explore how these characteristics relate to the attack variable using simple and multiple linear regressions.

Using the Pokémon dataset and the dummy variables you created, estimate the following three OLS regression models, each with `attack` as the dependent variable:

$$\text{attack} = \beta_0 + \beta_1 \text{legend1} + \varepsilon \quad (\text{Model 1})$$

$$\text{attack} = \beta_0 + \beta_1 \text{typeg2} + \beta_2 \text{typeg3} + \varepsilon \quad (\text{Model 2})$$

$$\text{attack} = \beta_0 + \beta_1 \text{legend1} + \beta_2 \text{typeg2} + \beta_3 \text{typeg3} + \varepsilon \quad (\text{Model 3})$$

1. For each model, report the regression output and interpret the coefficients.
2. Compare the 3 models with `compare_performance()` from `{performance}`.
3. Refit the 3 models using the factor variables `legendary` and `type_group3`.

Question 8: Testing equality of coefficients

Consider OLS model (mod3):

$$\text{attack} = \beta_0 + \beta_1 \text{legend1} + \beta_2 \text{typeg2} + \beta_3 \text{typeg3} + \varepsilon$$

Parameter	Coefficient	SE	95% CI	t(796)	p
(Intercept)	72.435	1.676	[69.145, 75.725]	43.215	< .001
legend1	41.840	4.000	[33.988, 49.691]	10.460	< .001
typeg2	7.623	2.419	[2.876, 12.371]	3.152	0.002
typeg3	1.595	2.904	[-4.106, 7.296]	0.549	0.583

Perform the following test

$$H_0 : \beta_2 = \beta_3 \quad \text{versus} \quad H_1 : \beta_2 \neq \beta_3.$$

Question 9: Predicting mean attack for all category combinations

Consider the following regression model estimated using the Pokémon dataset (mod3bis):

$$\text{attack} = \beta_0 + \beta_1 \text{legendary}_{\text{Yes}} + \beta_2 \text{type_group3}_{2.\text{Physical/Material}} + \beta_3 \text{type_group3}_{3.\text{Mystical/Supernatural}} + \varepsilon$$

Parameter	Coefficient	SE	95% CI	t(796)	p
(Intercept)	72.4	1.7	[69.1, 75.7]	43.2	< .001
legendary [Yes]	41.8	4.0	[34.0, 49.7]	10.5	< .001
type group3 [2.Physical/Material]	7.6	2.4	[2.9, 12.4]	3.2	0.002
type group3 [3.Mystical/Supernatural]	1.6	2.9	[-4.1, 7.3]	0.5	0.583

We want to compute the predicted mean Attack for every combination of these two variables.

1. Create a prediction grid containing all $2 \times 3 = 6$ combinations of
 - $\text{legendary} \in \{\text{Yes}, \text{No}\}$
 - $\text{type_group3} \in \{1.\text{Elemental/Environmental}, 2.\text{Physical/Material}, 3.\text{Mystical/Supernatural}\}$
2. Compute the predicted mean of attack for each of the six combinations.
3. Present the results in a table showing:
 - the combination of categories
 - the predicted mean Attack with standard error or 95% confidence interval

Question 10. Residual diagnostics

Using `mod3bis` and functions from the file `helper_functions3.R`:

1. Plot residuals vs fitted values and vs each predictor.
2. Plot $\sqrt{|\text{Standardized residuals}|}$ vs fitted values and vs each predictor.
3. Plot studentized residuals vs fitted values and vs each predictor.
4. Plot residuals vs order of observation.
5. Plot a histogram and normal Q-Q plot of the standardized residuals.