Linear Models in R (M1–MIDO)

Lab Session 2 — 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:

```r
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.

```r
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(rstatix)
library(matrixTests)
library(ggfortify)
library(qqplotr)
library(patchwork)
library(gtsummary)
library(kableExtra)
library(collapse)
library(tidyverse)
```

```r
source("helper_functions.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`.

```
pok <- select(pok, id, name, attack, speed, defense, hp, sp_attack, sp_def)
```

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

Question 2. Data management, label variable

Attach descriptive labels to each variable in the dataset `pok`.
This helps make outputs (e.g., summaries or regression tables) more readable.

Hint: use `relabel()` from the `{collapse}` package.

Question 3. Summary statistics

For the variables `attack`, `speed`, `defense`, `hp`, `sp_attack`, `sp_def`, compute summary statistics: mean, standard deviation, median, Q1, Q3, minimum, maximum.

Hint: `descr()`, `describe_distribution()`, `get_summary_stats()` `summarise()`, `tbl_summary()`

Question 4. Histogram, Scatter plots, Correlations

1. Plot histogram of `attack`, `speed`, `defense`, `hp`, `sp_attack`, `sp_def`.
2. Create scatter plots of `attack`, `speed`, `defense`, `hp`, `sp_attack`, `sp_def` against each other using `ggpairs()` from `{GGally}`.

Question 5. Multivariate linear gaussian regression model

We now fit a multivariate linear model of `speed`, `defense`, `hp`, `sp_attack`, `sp_def` on `attack`.

Scalar form

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i \\ y_i &= \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i \qquad p = 5,\ i = 1, \cdots, n, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2). \end{aligned}$$

with $y_i = \texttt{attack}_i$ and $(x_{i1}, \ldots, x_{i5}) = (\texttt{speed}_i,\ \texttt{defense}_i,\ \texttt{hp}_i,\ \texttt{sp\_attack}_i,\ \texttt{sp\_def}_i)$

$$\texttt{attack} = \beta_0 + \beta_1 \texttt{speed} + \beta_2 \texttt{defense} + \beta_3 \texttt{hp} + \beta_4 \texttt{sp\_attack} + \beta_5 \texttt{sp\_def} + \varepsilon$$

Matrix form

$$\mathbf{y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}_n, \sigma^2 \mathbf{I}_n\right), \mathrm{rank}(\mathbb{X}) = p + 1 = 6$$

1. Fit the model with `lm()` and save the result in `full_model`.
2. Interpret the output of:

```r
summary(full_model)
model_parameters(full_model, pretty_names = FALSE)
```

Question 6. Test of overall regression

We want to test

$$\begin{aligned} &H_0 : \beta_1 = \beta_2 = \cdots = \beta_5 = 0 \quad \text{versus} \quad H_1 : \text{At least one } \beta_j \neq 0 \\ \Longleftrightarrow &H_0 : (m_0)\ \mathbf{y} = \beta_0 \mathbf{1}_n + \boldsymbol{\varepsilon} \quad \text{versus} \quad H_1 : (m_1)\ \mathbf{y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \end{aligned}$$

Perform this test in 2 differents ways. **Hint:** Fisher test for nested models / General linear hypothesis tests

Method 1: Fisher test for nested models
The F-statistic is

$$F = \frac{\left\|P_{m_0}\mathbf{y} - P_{m_1}\mathbf{y}\right\|^2 / (p-q)}{\left\|\mathbf{y} - P_{m_1}\mathbf{y}\right\|^2 / (n-r)} = \frac{\left[\text{RSS}(m_0) - \text{RSS}(m_1)\right] / (p-q)}{\text{RSS}(m_1)/(n-r)} \sim F_{p-q,n-r} \quad (\text{under } H_0)$$

- Here $q = 0$ (reduced model has only an intercept),
- $r = p + 1 = 6$,
- $p - q = 5$.

Method 2: General linear hypothesis test
We can also write

$$\begin{aligned} &H_0 : \beta_1 = \beta_2 = \cdots = \beta_5 = 0 \quad \text{versus} \quad H_1 : \text{At least one } \beta_j \neq= 0 \\ \Longleftrightarrow &H_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{0}_5 \quad \text{versus} \quad H_1 : \mathbf{C}\boldsymbol{\beta} \neq \mathbf{0}_5 \end{aligned}$$

It is clear that

$$\mathbf{C}\boldsymbol{\beta} = \mathbf{0}_5 \Longleftrightarrow \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix} = \mathbf{0}_5$$

The corresponding test statistic is

$$F = \frac{(\mathbf{C}\hat{\boldsymbol{\beta}})^\top \left[\mathbf{C}(\mathbb{X}^\top\mathbb{X})^{-1}\mathbf{C}^\top\right]^{-1} (\mathbf{C}\hat{\boldsymbol{\beta}})/q}{\hat{\sigma}^2} \sim F_{q,n-r}, \qquad q = 5.$$

Question 7. Indices of model performance for regression

Compute indices of performance for the `full_model`. **Hint:** `glance()`, `model_performance()`

Question 8. Joint hypothesis test

Consider the full regression model `full_model`

$$\texttt{attack} = \beta_0 + \beta_1 \texttt{speed} + \beta_2 \texttt{defense} + \beta_3 \texttt{hp} + \beta_4 \texttt{sp\_attack} + \beta_5 \texttt{sp\_def} + \varepsilon$$

We want to test jointly whether the coefficients on `sp_attack` and `sp_def` are equal to zero:

$$H_0 : \beta_4 = \beta_5 = 0 \quad \text{versus} \quad H_1 : \text{at least one of } \beta_4, \beta_5 \text{ is nonzero.}$$

Hints:

- Compare the **full model** with a **restricted model** (without `sp_attack` and `sp_def`) using an F-test (`anova()`).
- Use a **joint Wald test** (`linearHypothesis()`, `waldtest()`).

Question 9. Prediction & Intervals

Consider `full_model`

1. Compute 95% Confidence Interval for the mean $\mathbb{E}(\texttt{attack})$ given $\texttt{speed} = \texttt{30}, \texttt{70}, \texttt{110}, \texttt{150}$ and fixing the other predictors at their mean

Hint: `predict(..., interval = "confidence")`, `estimate_expectation()`

2. Suppose a new Pokémon is created with the following characteristics :

```
   speed   defense        hp sp_attack     sp_def
      50        42       100       135         60
```

Predict the `attack` for Pokémon and the appropriate 95%CI.

Question 10. Residual diagnostics

Note: Standardized vs Studentized residuals

- Let denote by h_{ij} the element of the projector $P_{\mathbb{X}} = H_{\mathbb{X}}$ such that $P_{\mathbb{X}} = H_{\mathbb{X}} = [h_{ij}]$
- The diagonal elements $h_{ii} \in [0, 1]$ are called the *leverages*
- If $h_{ii} > 2p/n$ (sometimes $h_{ii} > 3p/n$), then the observation i is consider an *outlier*

- **Standardized residuals** (from `rstandard()`)
  Raw residuals are rescaled by their estimated standard deviation, taking into account leverage.

$$\hat{r}_i = \frac{\hat{\varepsilon}_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

where $\hat{\varepsilon}_i$ is the raw residual and h_{ii} is the leverage of observation i.
These make residuals roughly comparable across observations.

- **Studentized residuals** (from `rstudent()`)
  Go one step further: each residual is scaled using a variance estimate that **excludes the** i-th observation.
  This gives more accurate standard errors and makes large outliers easier to detect.

$$t_i^* = \frac{\hat{\varepsilon}_i}{\hat{\sigma}_{(-i)}\sqrt{1 - h_{ii}}}$$

where $\hat{\sigma}_{(-i)}$ is the error standard deviation estimated without observation i.

Using `full_model` and functions from the file `helper_functions.R`:

1. Plot residuals vs fitted values and vs each predictor speed, defense, hp, sp_attack, sp_def.
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 in the order of observation (to detect dependence).
5. Plot a histogram of the standardized residuals.
6. Perform a normality test on standardized residuals.
7. Plot a normal Q-Q plot of standardized residuals.
8. Perform the Breusch–Pagan test for heteroskedasticity.
9. Perform the Durbin–Watson test on the residuals.

Session Info

Package	Version
broom	1.0.10
car	3.1-3
collapse	2.1.4
correlation	0.8.8
datawizard	1.3.0
effectsize	1.0.1
GGally	2.4.0
ggfortify	0.4.19
ggpubr	0.6.2
glue	1.8.0
gtsummary	2.4.0
insight	1.4.2
kableExtra	1.4.0
lmtest	0.9-40
matrixTests	0.2.3.1
modelbased	0.13.0
parameters	0.28.2
patchwork	1.3.2
performance	0.15.2
qqplotr	0.0.7
rstatix	0.7.3
scales	1.4.0
see	0.12.0
tidyverse	2.0.0