

Linear Models in R (M1–MIDO)

Lab Session 1 – Student Sheet

Henri PANJO

Table of contents

Dataset Overview: <i>data_pokemon.csv</i>	3
Setup	4
Question 1. Loading dataset	5
Question 2. Summary statistics	5
Question 3. Histogram and Scatter plots	5
Question 4. Correlation matrix	5
Question 5. Simple Linear Regression (SLR)	6
Question 6. Fitted values, residuals	6
Question 7. Residual diagnostics	7
Note: Standardized vs Studentized residuals	7
Question 8. Prediction & Intervals	9
Session Info	10

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.

i Note

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.

- If you do not have the packages installed in your computer

```
pk <- c(
  "broom", "performance", "parameters", "datawizard", "see",
  "effectsize", "insight", "correlation", "modelbased", "glue",
  "scales", "GGally", "ggpubr", "car", "lmtest", "rstatix",
  "matrixTests", "ggfortify", "qqplotr", "collapse", "tidyverse"
)

install.packages(pk)
```

- Loading packages

```
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(collapse)
library(tidyverse)
```

Question 1. Loading dataset

Import the `data_pokemon.csv` file with `read_csv()`. Save the data in an object called `pok`.

- Quickly examine the data using `glimpse()` from `{dplyr}`
- Display the first 10 rows of `pok` using `head()` or `slice()`.

Question 2. Summary statistics

For the variables attack, speed, defense, hp, compute summary statistics: number of missing values, number of distinct values, mean, median, and standard deviation.

Hint: `summary()`, `descr()`, `describe_distribution()`, `get_summary_stats()`, `summarise()`, `mean()`, `sd()`, `median()`, `n_distinct()`, `is.na()`

Question 3. Histogram and Scatter plots

1. Plot histogram of attack and speed. Hint: `geom_histogram()`
2. Create scatter plots of attack against each numeric predictor speed, defense, hp. Hint: `geom_point()`, with `geom_smooth(method = "lm")`

Question 4. Correlation matrix

Compute and interpret the correlation matrix of the predictors speed, defense, hp.

Hint: `cor()`, `correlation()`

Question 5. Simple Linear Regression (SLR)

1. For each predictors speed, defense, hp, fit a SLR to explain the variable attack:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

$$\text{attack} = \beta_0 + \beta_1 \text{speed} + \varepsilon$$

$$\text{attack} = \beta_0 + \beta_1 \text{defense} + \varepsilon$$

$$\text{attack} = \beta_0 + \beta_1 \text{hp} + \varepsilon$$

Save the models in 3 objects: `slr_speed`, `slr_defense`, `slr_hp`. Interpret intercept and slope.

Hint: `lm()`

2. Interpret the results of the three hypothesis tests $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$?

Hint: `summary()` or `coeftest()` on `slr_speed`, `slr_defense`, `slr_hp`

3. Compute 95% CIs for coefficients.

Hint: `confint()`, `tidy()`, `model_parameters()`

Question 6. Fitted values, residuals

1. In the `pok` database, create the following variables

- `yhat_speed`, which represents the fitted values of the `slr_speed` model: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
- `res_speed`, which represents the residuals of the `slr_speed` model: $\hat{\varepsilon}_i = y_i - \hat{y}_i$

Hint: retrieve $\hat{\beta}_0$ and $\hat{\beta}_1$ with `coef()`

Display the first 10 rows of `pok` with the fitted values and residuals added.

2. Compute the fitted values and residuals using the functions `predict()` (or `fitted`) and `residuals()` (or `resid()`) on `slr_speed`. Also try the handy function `augment()` on `slr_speed`

Question 7. Residual diagnostics

Note: Standardized vs Studentized residuals

$$\hat{\varepsilon} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbb{X}\hat{\beta} = \mathbf{y} - P_{\mathbb{X}}\mathbf{y} = (\mathbf{I}_n - P_{\mathbb{X}})\mathbf{y} = (\mathbf{I}_n - P_{\mathbb{X}})\boldsymbol{\varepsilon}$$

- 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*
- We have $\text{Cov}(\boldsymbol{\varepsilon}) = \sigma^2 \mathbf{I}_n$ but $\text{Cov}(\hat{\varepsilon}) = \sigma^2 (\mathbf{I}_n - H_{\mathbb{X}})$
- The residuals are not independant, however, in many cases, especially if n is large, the h_{ii} 's tend to be small.
The impact of this is usually small and diagnostics can reasonably be applied to the residuals in order to check the assumptions on the error but we can also modify the residuals to adjust for this effect.
- 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 .

In practice:

- Use standardized residuals for quick checks.
- Use studentized residuals when formally testing for outliers or influential observations.

1. For the `slr_speed` model, with the help of `autoplot()`, plot residuals vs fitted values and $\sqrt{|\text{Standardized residuals}|}$ vs fitted values
2. Plot *studentized residuals* vs fitted: Hint: `fitted()`, `rstudent()`
3. Plot residuals vs speed and vs defense
4. Plot residuals in the order of observation (to detect time dependence). Hint: `augment()`
5. Plot an histogram of the standardized residuals.
6. Plot a normal Q-Q plot of the standardized residuals. Hint: `qqnorm()`, `qqline()`, `augment()`, `stat_qq()`, `stat_qq_line()`
7. Performs the Breusch–Pagan test for heteroskedasticity: Hint: `bptest()`, `ncvTest()`
8. Perform the Durbin–Watson test on the residuals: Hint: `dwttest()`, `durbinWatsonTest()`
9. Perform test for normality on standardized residuals. Hint: `shapiro.test()`, `shapiro_test()`, `col_jarquebera()`

Question 8. Prediction & Intervals

Consider the model `s1r_speed`

1. Compute 95% Confidence Interval for the mean $\mathbb{E}(\text{attack})$ at speed = 30, 70, 110, 150.

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

2. Compute 95% Prediction Interval for new responses of attack at speed = 20, 60, 100, 140.

Hint: `predict(..., interval = "prediction")`, `estimate_prediction()`

3. On the graph representing the scatter plot between `attack` and `speed`, overlay the regression line and its 95% confidence band.

Session Info

- R version 4.5.1 (2025-06-13 ucrt)
- Rstudio version 2025.9.1.401 (Cucumberleaf Sunflower)

Package	Version
broom	1.0.10
car	3.1-3
collapse	2.1.3
correlation	0.8.8
datawizard	1.3.0
effectsize	1.0.1
GGally	2.4.0
ggfortify	0.4.19
ggpubr	0.6.1
glue	1.8.0
insight	1.4.2
lmtest	0.9-40
matrixTests	0.2.3
modelbased	0.13.0
parameters	0.28.2
performance	0.15.2
qqplotr	0.0.7
rstatix	0.7.2
scales	1.4.0
see	0.12.0
tidyverse	2.0.0