An introduction to regression

Lino Galiana
Version of September 29, 2026

Linear regression is the first statistical modeling to be discovered in a quantitative curriculum. It is a very intuitive and rich method. Machine Learning allows us to approach it in a different way to econometrics. With `scikit` and `statsmodels`, we have all the tools we need to satisfy both data scientists and economists.

Table of contents

The previous chapter aimed to propose a first model to understand the counties where the Republican Party wins. The variable of interest was bimodal (win or lose), placing us within the framework of a classification model.

Now, using the same data, we will propose a regression model to explain the Republican Party's score. The variable is thus continuous. We will ignore the fact that its bounds lie between 0 and 100, meaning that to be rigorous, we would need to transform the scale so that the data fits within this interval.

Machine learning materials in this course uses a unique dataset, presented in the introduction. All examples are based on US county level presidential election results combined with sociodemographic variables. Source code for data ingestion is available on `Github`.

```
!pip install geopandas openpyxl plotnine plotly
```

```
import requests

url     =     'https://raw.githubusercontent.com/linogaliana/python-datascientist/main/content/
modelisation/get_data.py'
r = requests.get(url, allow_redirects=True)
open('getdata.py', 'wb').write(r.content)

import getdata
votes = getdata.create_votes_dataframes()
```

This chapter will use several modeling *packages*, the main ones being `Scikit` and `Statsmodels`. Here is a suggested import for all these *packages*.

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import sklearn.metrics
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

1 General Principle

The general principle of regression consists of finding a law $h_\theta(X)$ such that

$$h_\theta(X) = \mathbb{E}_\theta(Y|X)$$

This formalization is extremely general and is not limited to linear regression.

In econometrics, regression offers an alternative to maximum likelihood methods and moment methods. Regression encompasses a very broad range of methods, depending on the family of models (parametric, non-parametric, etc.) and model structures.

1.1 Linear Regression

This is the simplest way to represent the law $h_\theta(X)$ as a linear combination of variables X and parameters θ. In this case,

$$\mathbb{E}_\theta(Y|X) = X\beta$$

This relationship is, under this formulation, theoretical. It must be estimated from the observed data y. The method of least squares aims to minimize the quadratic error between the prediction and the observed data (which explains why regression can be seen as a *Machine Learning* problem). In general, the method of least squares seeks to find the set of parameters θ such that

$$\theta = \arg\min_{\theta \in \Theta} \mathbb{E}\left[(y - h_\theta(X))^2 \right]$$

Which, in the context of linear regression, is expressed as follows:

$$\beta = \arg\min \mathbb{E}\left[(y - X\beta)^2 \right]$$

When the theoretical model ($\mathbb{E}_\theta(Y|X) = X\beta$) is applied to data, the model is formalized as follows:

$$Y = X\beta + \epsilon$$

With a certain distribution of the noise ϵ that depends on the assumptions made. For example, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ i.i.d., the estimator β obtained is equivalent to the Maximum Likelihood Estimator, whose asymptotic theory ensures unbiasedness and minimum variance (Cramer-Rao bound).

1.1.a Application

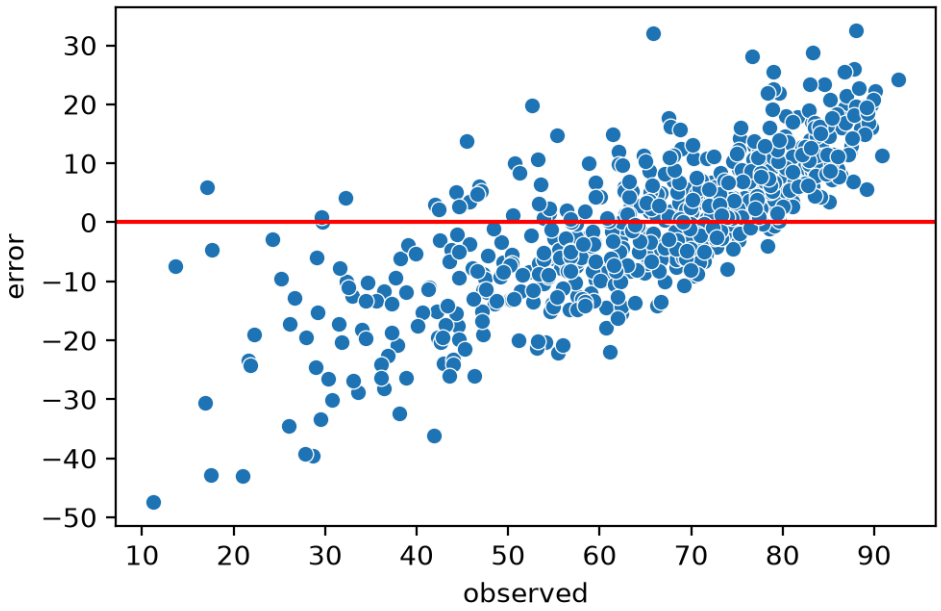
Under the guidance of the heirs of A. Siegfried [1], our objective in this chapter is to explain and predict the Republican score based on some socioeconomic variables. Unlike the previous chapter, where we focused on a binary outcome (Republican victory/defeat), this time we will model the Republican score directly.

The next exercise aims to demonstrate how to perform linear regression using `scikit`. In this area, `statsmodels` is significantly more comprehensive, as the following exercise will demonstrate. The

main advantage of performing regressions with `scikit` is the ability to compare the results of linear regression with other regression models in the context of selecting the best predictive model.

> Exercise 1a: Linear Regression with scikit
>
> 1. Using a few variables, for example, *'Unemployment_rate_2019', 'Median_Household_Income_2021', 'Percent of adults with less than a high school diploma, 2018-22', "Percent of adults with a bachelor's degree or higher, 2018-22"*, explain the variable `per_gop` using a training sample `X_train` prepared beforehand.
>
> ⚠️ Use the variable `Median_Household_Income_2021` in `log` form; otherwise, its scale might dominate and obscure other effects.
>
> 2. Display the values of the coefficients, including the constant.
> 3. Evaluate the relevance of the model using R^2 and assess the fit quality with the MSE.
> 4. Plot a scatter plot of observed values and prediction errors. Do you observe any specification issues?

In question 4, it can be observed that the distribution of errors is clearly not random with respect to X.



The model therefore suffers from a specification issue, so work will need to be done on the selected variables later. Before that, we can redo this exercise using the `statsmodels` package.

> Exercise 1b: Linear Regression with statsmodels
>
> This exercise aims to demonstrate how to perform linear regression using `statsmodels`, which offers features more similar to those of `R` and less oriented toward *Machine Learning*.
>
> The goal is still to explain the Republican score based on a few variables.
>
> 1. Using a few variables, for example, *'Unemployment_rate_2019', 'Median_Household_Income_2021', 'Percent of adults with less than a high school diploma, 2015-19', "Percent of adults with a bachelor's degree or higher, 2015-19"*, explain the variable `per_gop`.
>
>   ⚠️ Use the variable `Median_Household_Income_2021` in `log` form; otherwise, its scale might dominate and obscure other effects.

2. Display a regression table.
3. Evaluate the model's relevance using the R^2.
4. Use the `formula` API to regress the Republican score as a function of the variable `Unemployment_rate_2019`, `Unemployment_rate_2019` squared, and the log of `Median_Household_Income_2021`.

```
R2:  0.43016652968844415
```

Tip

To generate a well-formatted table for a report in $\LaTeX$, you can use the method `Summary.as_latex`. For an HTML report, you can use `Summary.as_html`.

Note

Users of `R` will find many familiar features in `statsmodels`, particularly the ability to use a formula to define a regression. The philosophy of `statsmodels` is similar to that which influenced the construction of `R`'s `stats` and `MASS` packages: providing a general-purpose library with a wide range of models.

However, `statsmodels` benefits from being more modern compared to `R`'s packages. Since the 1990s, `R` packages aiming to provide missing features in `stats` and `MASS` have proliferated, while `statsmodels`, born in the 2010s, only had to propose a general framework (the *generalized estimating equations*) to encompass these models.

1.2 Logistic regression

We applied our linear regression to a continuous *outcome* variable. How do we handle a binary distribution?
In this case, $\mathbb{E}_\theta(Y|X) = \mathbb{P}_\theta(Y = 1|X)$.
Logistic regression can be seen as a linear probability model:

$$\text{logit}\Big(\mathbb{E}_\theta(Y|X)\Big) = \text{logit}\Big(\mathbb{P}_\theta(Y = 1|X)\Big) = X\beta$$

The logit function is $]0, 1[\rightarrow \mathbb{R} : p \mapsto \log(\frac{p}{1-p})$.

It allows a probability to be transformed into $\mathbb{R}$. Its reciprocal function is the sigmoid ($\frac{1}{1+e^{-x}}$), a central concept in *Deep Learning*.

It should be noted that probabilities are not observed; what is observed is the binary *outcome* (0/1). This leads to two different perspectives on logistic regression:

- In econometrics, interest lies in the latent model that determines the choice of the outcome. For example, if observing the choice to participate in the labor market, the goal is to model the factors determining this choice;
- In *Machine Learning*, the latent model is only necessary to classify observations into the correct category.

Parameter estimation for β can be performed using maximum likelihood or regression, both of which are equivalent under certain assumptions.

Note

By default, `scikit` applies regularization to penalize non-parsimonious models (a behavior different from that of `statsmodels`). This default behavior should be kept in mind if the objective is not prediction.

Exercise 2a: Logistic Regression with scikit

Using `scikit` with training and test samples:

1. Evaluate the effect of the already-used variables on the probability of Republicans winning. Display the values of the coefficients.
2. Derive a confusion matrix and a measure of model quality.
3. Remove regularization using the `penalty` parameter. What effect does this have on the estimated parameters?

Exercise 2b: Logistic Regression with statsmodels

Using training and test samples:

1. Evaluate the effect of the already-used variables on the probability of Republicans winning.
2. Perform a likelihood ratio test regarding the inclusion of the (log)-income variable.

The p-value of the likelihood ratio test being close to 1 means that the log-income variable almost certainly adds information to the model.

Tip

The test statistic is:

$$LR = -2\log\left(\frac{\mathcal{L}_\theta}{\mathcal{L}_{\theta_0}}\right) = -2(\ell_\theta - \ell_{\theta_0})$$

2 Going Further

This chapter only introduces the concepts of regression in a very introductory way. To expand on this, it is recommended to explore further based on your interests and needs.

In the field of *machine learning*, the main areas for deeper exploration are:

- Alternative regression models like random forests.
- *Boosting* and *bagging* methods to learn how multiple models can be trained jointly and their predictions combined democratically to converge on better decisions than a single model.
- Issues related to model explainability, a very active research area, to better understand the decision criteria of models.

In the field of econometrics, the main areas for deeper exploration are:

- Generalized linear models to explore regression with more general assumptions than those we have made so far;
- Hypothesis testing to delve deeper into these questions beyond our likelihood ratio test.

References

Informations additionnelles

i Python environment

This site was built automatically through a `Github` action using the `Quarto` reproducible publishing software (version 1.10.18).

The environment used to obtain the results is reproducible via uv. The `pyproject.toml` file used to build this environment is available on the `linogaliana/python-datascientist` repository

```
[project]
name = "python-datascientist"
version = "0.1.0"
description = "Source code for Lino Galiana's Python for data science course"
readme = "README.md"
requires-python = "≥3.13,<3.14"
dependencies = [
    "altair≥6.0.0",
    "cartiflette",
    "contextily==1.6.2",
    "duckdb≥0.10.1",
    "folium≥0.19.6",
    "gdal==3.11.4",
    "graphviz==0.20.3",
    "great-tables≥0.12.0",
    "gt-extras≥0.0.8",
    "ipykernel≥6.29.5",
    "jupyter≥1.1.1",
    "jupyter-cache≥1.0.0",
    "kaleido≥0.2.1",
    "langchain-community≥0.3.27",
    "loguru==0.7.3",
    "markdown≥3.8",
    "nbclient≥0.10.0",
    "nbformat≥5.10.4",
    "nltk≥3.9.1",
    "pandas≥3.0",
    "pip≥25.1.1",
    "plotly≥6.1.2",
    "plotnine≥0.15",
    "polars≥1.8.2",
    "pyarrow≥17.0.0",
    "pynsee≥0.1.8",
    "python-dotenv≥1.0.1",
    "python-frontmatter≥1.1.0",
    "pywaffle≥1.1.1",
    "requests≥2.32.3",
    "scikit-image≥0.24.0",
    "scikit-learn≥1.8.0",
    "scipy≥1.13.0",
    "seaborn≥0.13.2",
    "selenium<4.39.0",
    "spacy≥3.8.4",
    "webdriver-manager≥4.0.2",
```

```
    "wordcloud==1.9.3",
]

[tool.uv.sources]
cartiflette = { git = "https://github.com/inseefrlab/cartiflette" }
gdal = [
  { index = "gdal-wheels", marker = "sys_platform == 'linux'" },
  { index = "geospatial_wheels", marker = "sys_platform == 'win32'" },
]

[[tool.uv.index]]
name = "geospatial_wheels"
url = "https://nathanjmcdougall.github.io/geospatial-wheels-index/"
explicit = true

[[tool.uv.index]]
name = "gdal-wheels"
url = "https://gitlab.com/api/v4/projects/61637378/packages/pypi/simple"
explicit = true

[dependency-groups]
dev = [
    "nb-clean>=4.0.1",
]
```

To use exactly the same environment (version of `Python` and *packages*), please refer to the documentation for `uv`.

Bibliography

[1] A. Siegfried, *Tableau politique de la France de l'ouest sous la troisième république: 102 cartes et croquis, 1 carte hors texte*. A. Colin, 1913.