R Tutorial - Linear Regression and Log-Log Specifications

POL 501

2024-11-26

Table of Contents

## Introduction

This tutorial explores how nuclear energy production impacts CO2 emissions per capita, using simulated data and regression models. Specifically, we examine:

1. **Data Simulation**: Create a mock dataset for a hypothetical case study.
2. **Linear Regression**: Explore simple and multiple regression models.
3. **Elasticity**: Define and calculate elasticities for log-log and log-linear models.
4. **Control Variables**: Assess how GDP per capita affects the relationship.

### Case Study Context

We investigate how nuclear energy impacts carbon emissions. As nuclear energy replaces fossil fuels, it likely reduces CO2 emissions. However, other factors like total energy demand or economic activity (GDP per capita) may also influence emissions. This tutorial models these relationships, estimating elasticities (percent change in CO2 emissions for percent changes in nuclear energy) while controlling for other variables.

## 1. Defining Elasticity

**Elasticity** measures the responsiveness of one variable to another:

Elasticity can be derived from log-log and log-linear models:

### Case 1: Log-Log Model

Start with the log-log model:

Take the derivative w.r.t. :

Recognize the relationship:

Thus, directly represents elasticity.

### Case 2: Log-Linear Model

Start with:

Take the derivative w.r.t. :

Express in terms of percentage changes:

For small changes, a one-unit change in (measured in percentage points) leads to an approximate change in .

## 2. Simulating Data

Simulate data for years 2000–2015:

### R Code: Data Generation

# Load necessary library  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Set seed for reproducibility  
set.seed(123)  
  
# Define the years  
Years <- seq(2000, 2015)  
num\_years <- length(Years)  
  
# Simulate energy production and emissions  
NuclearEnergyProduced <- runif(num\_years, 0, 600) # Nuclear (billion kWh)  
OtherEnergyProduced <- runif(num\_years, 300, 900) # Other sources  
TotalEnergyProduced <- NuclearEnergyProduced + OtherEnergyProduced  
NuclearEnergyShare <- (NuclearEnergyProduced / TotalEnergyProduced) \* 100 # %  
CO2EmissionsPerCapita <- 10 - 0.005 \* NuclearEnergyProduced + rnorm(num\_years, 0, 0.5) # t/person  
GDPPerCapita <- runif(num\_years, 30000, 60000) # USD  
  
# Create data frame  
energy\_data <- data.frame(  
 Year = Years,  
 NuclearEnergyProduced,  
 OtherEnergyProduced,  
 TotalEnergyProduced,  
 NuclearEnergyShare,  
 CO2EmissionsPerCapita,  
 GDPPerCapita  
)  
  
# Add log-transformed variables  
energy\_data <- energy\_data %>%  
 mutate(  
 log\_CO2EmissionsPerCapita = log(CO2EmissionsPerCapita),  
 log\_NuclearEnergyProduced = log(NuclearEnergyProduced + 1) # Avoid log(0)  
 )  
  
# View the first few rows  
head(energy\_data)

## Year NuclearEnergyProduced OtherEnergyProduced TotalEnergyProduced  
## 1 2000 172.5465 447.6526 620.1992  
## 2 2001 472.9831 325.2357 798.2188  
## 3 2002 245.3862 496.7524 742.1386  
## 4 2003 529.8104 872.7022 1402.5126  
## 5 2004 564.2804 833.7236 1398.0040  
## 6 2005 27.3339 715.6820 743.0159  
## NuclearEnergyShare CO2EmissionsPerCapita GDPPerCapita  
## 1 27.821146 9.386193 54439.20  
## 2 59.254816 6.651776 43455.49  
## 3 33.064735 9.123747 54301.93  
## 4 37.775805 7.114552 54371.69  
## 5 40.363288 6.644686 53830.27  
## 6 3.678777 9.754343 43194.95  
## log\_CO2EmissionsPerCapita log\_NuclearEnergyProduced  
## 1 2.239240 5.156446  
## 2 1.894884 6.161172  
## 3 2.210881 5.506900  
## 4 1.962142 6.274405  
## 5 1.893817 6.337322  
## 6 2.277713 3.344059

## 3. Linear and Log-Log Models

### Simple Log-Log Regression

We fit a log-log model to estimate the elasticity of CO2 emissions with respect to nuclear energy.

### R Code: Log-Log Model

# Fit the log-log model  
log\_log\_model <- lm(log\_CO2EmissionsPerCapita ~ log\_NuclearEnergyProduced, data = energy\_data)  
  
# Model summary  
summary(log\_log\_model)

##   
## Call:  
## lm(formula = log\_CO2EmissionsPerCapita ~ log\_NuclearEnergyProduced,   
## data = energy\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.11711 -0.04013 -0.01226 0.04625 0.13304   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.82413 0.14037 20.120 9.92e-12 \*\*\*  
## log\_NuclearEnergyProduced -0.13182 0.02457 -5.366 9.96e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07963 on 14 degrees of freedom  
## Multiple R-squared: 0.6728, Adjusted R-squared: 0.6495   
## F-statistic: 28.79 on 1 and 14 DF, p-value: 9.958e-05

### Interpretation

If the coefficient of () is -0.13:

* A 1% increase in nuclear energy production reduces CO2 emissions per capita by 0.13%.
* Negative elasticity indicates an inverse relationship.

## 4. Log(Y) - X Model

When the independent variable is in percentage points (e.g., NuclearEnergyShare), we use a log-linear model.

### R Code: Log(Y)-X Model

# Fit the log-linear model  
logY\_X\_model <- lm(log\_CO2EmissionsPerCapita ~ NuclearEnergyShare, data = energy\_data)  
  
# Model summary  
summary(logY\_X\_model)

##   
## Call:  
## lm(formula = log\_CO2EmissionsPerCapita ~ NuclearEnergyShare,   
## data = energy\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.141940 -0.073807 -0.009119 0.041073 0.196785   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.314592 0.063839 36.257 3.04e-15 \*\*\*  
## NuclearEnergyShare -0.006908 0.001734 -3.985 0.00136 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09529 on 14 degrees of freedom  
## Multiple R-squared: 0.5314, Adjusted R-squared: 0.498   
## F-statistic: 15.88 on 1 and 14 DF, p-value: 0.001356

### Interpretation

If the coefficient of NuclearEnergyShare () is -0.01:

* A one-percentage-point increase in Nuclear Energy Share reduces CO2 emissions per capita by 0.01% (approx.).

## 5. Adding Control Variables

### Including GDP Per Capita

To isolate the impact of nuclear energy on CO2 emissions, we control for GDP per capita. To keep the interpretation of the slopes as elasticities, we should log transform GDP per capita.

### R Code: Model with Control Variable

# Log transform GDP per capita  
energy\_data <- energy\_data %>%  
 mutate(  
 log\_GDPPerCapita = log(GDPPerCapita)  
 )  
  
# Fit the model with a control variable  
logY\_X\_control\_model <- lm(log\_CO2EmissionsPerCapita ~ NuclearEnergyShare + log\_GDPPerCapita, data = energy\_data)  
  
# Model summary  
summary(logY\_X\_control\_model)

##   
## Call:  
## lm(formula = log\_CO2EmissionsPerCapita ~ NuclearEnergyShare +   
## log\_GDPPerCapita, data = energy\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.155689 -0.044393 -0.008213 0.047555 0.192660   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.445759 1.469122 0.984 0.34303   
## NuclearEnergyShare -0.007049 0.001791 -3.935 0.00171 \*\*  
## log\_GDPPerCapita 0.081539 0.137738 0.592 0.56402   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09758 on 13 degrees of freedom  
## Multiple R-squared: 0.5437, Adjusted R-squared: 0.4736   
## F-statistic: 7.746 on 2 and 13 DF, p-value: 0.006093

### Interpretation

If coefficients are:

* **NuclearEnergyShare ()** = -0.01: A 1pp increase in Nuclear Energy Share reduces CO2 emissions by 0.01% (rounded).
* **GDPPerCapita ()** = 0.08: A 1% increase in GDP per capita increases CO2 emissions by 0.08% (rounded).

Ceteris paribus, these effects hold independent of each other.

**Ceteris paribus**, meaning “all other things being equal,” implies that the effect of one variable (e.g., NuclearEnergyShare or GDPPerCapita) on the dependent variable (CO2 emissions) is measured while holding the other variables in the model constant. In statistical terms, this is achieved by isolating the unique contribution of each independent variable through the regression model, which controls for the influence of other included variables. For example, the coefficient for NuclearEnergyShare reflects its direct impact on CO2 emissions, independent of GDPPerCapita, and vice versa. This ensures the estimated relationship between each predictor and the outcome is not confounded by correlations with other variables in the model.

## Conclusion

In this tutorial, we:

1. Simulated a dataset for analyzing nuclear energy’s impact on CO2 emissions.
2. Defined elasticity for log-log and log-linear models, deriving interpretations mathematically.
3. Implemented regression models, including control variables, in R.
4. Interpreted results to explain the economic and environmental impacts.

### Appendix: Proof that

#### Objective

Demonstrate that the differential of the natural logarithm of is equal to the differential of divided by :

#### Proof

The differential represents the infinitesimal change in corresponding to an infinitesimal change in .

**Step 1: Recall the Derivative of**

The derivative of the natural logarithm function with respect to is:

**Step 2: Express the Differential**

By definition of the differential:

**Step 3: Conclude the Relationship**

Thus, we have:

#### Interpretation

* **Infinitesimal Changes**: The equation shows that the infinitesimal change in is equal to the proportional (or percentage) infinitesimal change in .
* **Application to Elasticities**: In economics, this property underlies the calculation of elasticities in regression models. It implies that small proportional changes in correspond directly to small absolute changes in .

#### Extension to Finite Changes

For small but finite changes , we can approximate:

**Justification:**

* **Taylor Series Approximation**: The first-order Taylor series expansion of around is:
* **Compute :**
* **Validity for Small :** The approximation holds when is small relative to , ensuring higher-order terms are negligible.