

# **Quarto Template**

**for Master's and Bachelor's theses**

**Valentin Klotzbücher**

**Master of Science in Economics**

**Invalid Date**

**Department of Economics**

**University of Freiburg**

## Table of contents

<b>Preface</b>	<b>3</b>
<b>1 Introduction</b>	<b>4</b>
<b>2 The Importance of Modern Tools in Economic Research</b>	<b>5</b>
<b>3 Probit Estimation in Economics</b>	<b>7</b>
3.1 The Probit Model . . . . .	7
3.2 Estimating the Model . . . . .	7
3.2.1 Code Example: Estimating a Probit Model . . . . .	8
<b>4 First chapter title</b>	<b>12</b>
<b>References</b>	<b>13</b>

## Preface

This is a Quarto book.

To learn more about Quarto books visit <https://quarto.org/docs/books>.

```
1 + 1
```

```
[1] 2
```

# 1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

What follows is generated with GPT4o as blindtext to illustrate formatting options.

## 2 The Importance of Modern Tools in Economic Research

Embarking on a thesis in economics is an intellectually stimulating yet demanding journey, one that requires not just analytical rigor but also clear communication and effective organization of complex ideas. As you delve into this process, the tools you choose can greatly influence both your efficiency and the quality of your work. This is where Quarto and literate programming come into play, offering a robust framework that integrates writing, coding, and data visualization into a cohesive whole. By adopting these tools, you are not only streamlining your workflow but also ensuring that your research is transparent, reproducible, and professionally presented.

In the realm of economic research, clarity is paramount. Your thesis must convey intricate methodologies and detailed results in a manner that is accessible and understandable, not just to your thesis committee but to the broader academic community. Quarto allows you to write your thesis in a format that interweaves narrative and analysis, making it easier for your readers to follow the logical flow of your argument. The integration of literate programming takes this a step further by embedding your code within the document itself. This approach ensures that every piece of analysis is directly linked to the code that generated it, providing a level of transparency that is essential for the credibility of your work. Readers can see exactly how you arrived at your conclusions, which not only enhances the trustworthiness of your findings but also facilitates the replication of your results by other researchers.

Reproducibility is a cornerstone of good research, particularly in economics, where the ability to replicate findings is critical to the validation of results. With Quarto, you can create dynamic documents that automatically update analyses and results whenever data or parameters are changed. This means that as your research evolves, your thesis remains current without the need for extensive manual updates. This feature is particularly useful in the context of collaborative research or when working with large datasets, where maintaining consistency and accuracy across multiple revisions can be challenging. By using literate programming within Quarto, you are not just writing a static document; you are creating a living, breathing piece of research that evolves with your work.

Efficiency is another significant benefit of using Quarto and literate programming. The process of writing a thesis is inherently time-consuming, and any tool that can reduce the time spent on formatting, revising, or switching between different software platforms is invaluable. Quarto brings together writing, coding, and data visualization in a single environment, allowing you to focus on the intellectual aspects of your research rather than the mechanics of document preparation. This integrated approach minimizes the risk of errors and inconsistencies that can arise from using multiple tools, and it accelerates the overall workflow, giving you more time to refine your analysis and polish your arguments.

The presentation of your results is equally important, particularly in a field like economics, where the clarity and professionalism of your graphs, tables, and equations can significantly impact how your work is received. Quarto supports advanced typesetting features, enabling you to produce documents that are not only clear and well-organized but also aesthetically pleasing. Whether you are presenting complex econometric models, detailed statistical tables, or intricate graphs, Quarto ensures that your work looks professional and meets the high standards expected in academic publications.

Furthermore, by adopting Quarto and literate programming, you are equipping yourself with modern skills that extend beyond the completion of your thesis. In today's data-driven world, the ability to produce reproducible research and to document the research process in a clear and

transparent way is highly valued, both in academia and in the private sector. Mastering these tools gives you a competitive edge, whether you are pursuing further academic research, entering the job market, or working in a professional setting where data analysis and communication are key components of the role.

In conclusion, the use of Quarto and literate programming in your economics thesis is more than just a choice of tools; it represents a commitment to producing high-quality, transparent, and reproducible research. By integrating your writing, coding, and data analysis into a single, dynamic document, you are not only enhancing the quality of your thesis but also streamlining the process and developing skills that will serve you well in your future career. Embracing these modern approaches to research documentation will not only make your thesis journey smoother and more efficient but also ensure that your work stands out in the academic community for its clarity, rigor, and professionalism.

### 3 Probit Estimation in Economics

In many economic research contexts, we are interested in modeling binary outcome variables, such as whether an individual chooses to participate in the labor force or whether a firm decides to enter a particular market. Probit models are commonly used for this purpose, as they allow us to estimate the probability that a particular outcome occurs given a set of explanatory variables.

#### 3.1 The Probit Model

The Probit model assumes that there is an underlying latent variable  $Y^*$  that is a linear function of the explanatory variables  $X$  and a normally distributed error term  $\epsilon$ :

$$Y^* = X\beta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 1)$$

However, we do not observe  $Y^*$  directly. Instead, we observe a binary outcome  $Y$  defined as:

$$Y = \begin{cases} 1 & \text{if } Y^* > 0 \\ 0 & \text{if } Y^* \leq 0 \end{cases}$$

The probability that  $Y = 1$  (i.e., the event of interest occurs) is given by:

$$P(Y = 1 | X) = \Phi(X\beta)$$

where  $\Phi(\cdot)$  is the cumulative distribution function (CDF) of the standard normal distribution.

#### 3.2 Estimating the Model

In practice, we estimate the vector of coefficients  $\beta$  using maximum likelihood estimation (MLE). The likelihood function for a sample of observations  $i = 1, \dots, n$  is given by:

$$L(\beta) = \prod_{i=1}^n \Phi(X_i\beta)^{Y_i} [1 - \Phi(X_i\beta)]^{1-Y_i}$$

The log-likelihood, which is typically maximized numerically to find the MLE of  $\beta$ , is:

$$\log L(\beta) = \sum_{i=1}^n [Y_i \log \Phi(X_i\beta) + (1 - Y_i) \log (1 - \Phi(X_i\beta))]$$

### 3.2.1 Code Example: Estimating a Probit Model

In practice, you might use a statistical software package to estimate a Probit model. For example, in R, you could use the following code:

```
# Load necessary libraries
library(tidyverse) # For data manipulation and visualization

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

library(margins) # For calculating marginal effects
library(modelsummary)
```

`modelsummary` 2.0.0 now uses `tinytable` as its default table-drawing backend. Learn more at: <https://vincentarelbundock.github.io/tin>

Revert to `kableExtra` for one session:

```
options(modelsummary_factory_default = 'kableExtra')
options(modelsummary_factory_latex = 'kableExtra')
options(modelsummary_factory_html = 'kableExtra')
```

Silence this message forever:

```
config_modelsummary(startup_message = FALSE)
```

```
library(flextable)
```

Attache Paket: 'flextable'

Das folgende Objekt ist maskiert 'package:purrr':

```
compose
```



```
# Load the built-in dataset
data("mtcars")

# Inspect the data
glimpse(mtcars)
```

```
Rows: 32
Columns: 11
$ mpg <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8, 1
$ cyl <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 4, 4, 4,
$ disp <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 146.7
$ hp <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 180,
$ drat <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92, 3
$ wt <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.190,
$ qsec <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20.00
$ vs <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1,
$ am <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
$ gear <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4, 4, 4
$ carb <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4, 1, 2, 1
```

```
# Create a binary outcome variable: 1 if mpg > 20, 0 otherwise
mtcars <- mtcars %>%
  mutate(high_mpg = ifelse(mpg > 20, 1, 0))

# Fit the Probit model
model <- glm(high_mpg ~ hp + wt + cyl,
             family = binomial(link = "probit"),
             data = mtcars)
```

Warning: glm.fit: Algorithmus konvergierte nicht

Warning: glm.fit: Angepasste Wahrscheinlichkeiten mit numerischem aufgetreten

```
# Summarize the model results
summary(model)
```

Call:

```
glm(formula = high_mpg ~ hp + wt + cyl, family = binomial(link = "pr
      data = mtcars)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	227.9647	59356.0515	0.004	0.997
hp	-0.5989	173.1890	-0.003	0.997
wt	-60.3149	16456.6441	-0.004	0.997
cyl	6.2871	4190.6791	0.002	0.999

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 4.386e+01 on 31 degrees of freedom  
 Residual deviance: 8.744e-09 on 28 degrees of freedom  
 AIC: 8

Number of Fisher Scoring iterations: 25

```
# Calculate marginal effects
margins_model <- margins(model)

# Summarize the marginal effects
#msummary(margins_model)

# Visualize marginal effects
plot(margins_model)

# Interpretation:
# The summary(model) output provides the estimated coefficients
# The coefficients indicate the direction of the effect, but the
# calculated via the margins() function tell us how a unit change
# affects the probability of a car having high MPG.
```

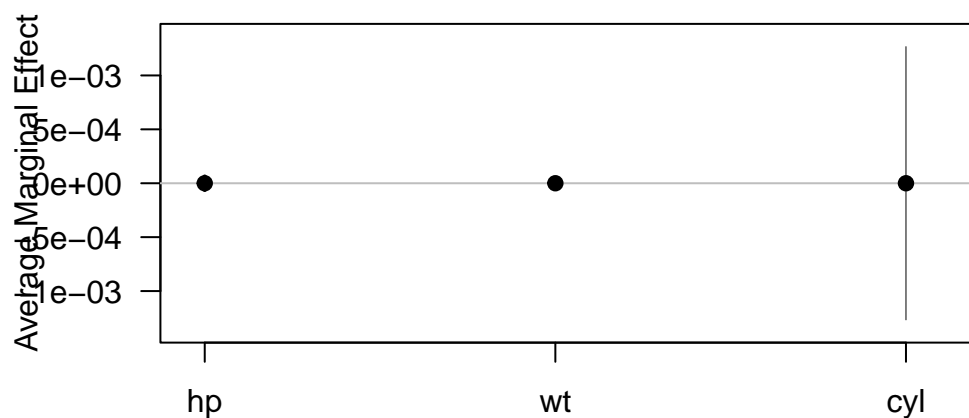


Figure 3.1: Marginal effects

Figure 3.1 shows the marginal effects.

Table 3.1: Marginal effects

```

summarytable = msummary(list(margins_model),
  output = "flextable",
  fmt = fmt_statistic(estimate = 3,
    std.error = 3,
    conf.low = 3,
    conf.high = 3,
    nobs = 0),
  stars = c('*' = 0.1, '**' = 0.05, '***' = 0.01),
  estimate = "{estimate}{stars}",
  statistic = "({std.error})",
  conf_level = .95,
  align="lc",
  gof_omit = 'AIC|BIC|R2 Adj.|RMSE|Std.Errors|R2 Within A
  coef_rename = c("lag_aid_gdp" = "Aid/GDP, t-1"
    )
)

```

## 4 First chapter title

In summary, this book has no content whatsoever.

1 + 1

[1] 2

## References

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.