Modelos Lineares Generalizados: avaliação de ferramentas de programação numa aplicação com bases de microdados reais

Ana Isabel Silva Marinho

UMinho | 2023

Universidade do Minho
Escola de Ciências

Ana Isabel Silva Marinho

Modelos Lineares Generalizados: avaliação de ferramentas de programação numa aplicação com bases de microdados reais

outubro 2023

Universidade do Minho
Escola de Ciências

Ana Isabel Silva Marinho

Modelos Lineares Generalizados: avaliação de ferramentas de programação numa aplicação com bases de microdados reais

Dissertação de Mestrado
Mestrado em Estatística para Ciência de Dados

Trabalho efetuado sob a orientação da

Prof. Doutora Susana Faria

Doutora Rita Sousa

outubro 2023

Despacho RT - 31 /2019 - Anexo 3

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho

Atribuição
CC BY

https://creativecommons.org/licenses/by/4.0/

Agradecimentos

Esta dissertação só se tornou uma realidade graças ao apoio fundamental de algumas pessoas que tornaram o meu percurso académico possível. Portanto, gostaria de agradecer às pessoas que estiveram ao meu lado e me apoiaram ao longo desta jornada.

Em particular, quero começar por agradecer às minhas orientadoras, a Professora Susana Faria e a Doutora Rita Sousa, pela orientação, disponibilidade, dedicação e apoio ao longo deste projeto.

Agradeço aos meus país e ao meu irmão Bruno que desempenharam um papel fundamental no meu percurso académico, obrigado por serem o meu maior exemplo de resiliência e determinação.

Por ultimo, mas não menos importante, quero agradecer aos meus amigos por serem uma parte tão importante da minha vida.

Despacho RT - 31 /2019 - Anexo 4

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Resumo

Os Modelos Lineares Generalizados (MLG) são frequentemente utilizados quando se pretende estudar a relação entre variáveis, especialmente quando se analisa o impacto que uma ou mais variáveis explicativas exercem sobre uma determinada variável de interesse (variável resposta).

A importância dos MLG não advém apenas de uma perspetiva aplicada, mas também do ponto de vista teórico. A relevância teórica desses modelos resulta do facto de estarem na base de muitos métodos estatísticos e de se utilizarem em diversas aplicações, destacando-se a centralidade da verosimilhança na teoria da inferência.

Esta dissertação tem como objetivo avaliar e comparar o desempenho de diferentes linguagens de programação na estimação dos MLG e aplicar estes modelos a uma base de microdados real.

Numa primeira fase, faz-se uma descrição detalhada da teoria dos MLG, com particular ênfase nos dois modelos de regressão que serviram como objeto de estudo: o Modelo de Regressão Logística e o Modelo de Regressão Poisson.

Dado que os MLG se podem estimar usando várias linguagens de programação, faz-se uma análise comparativa com aplicações em R, Stata e Python, com o propósito de avaliar o desempenho das mesmas na estimação. Nesse contexto, foram analisados diversos critérios, incluindo o desempenho computacional, o tempo e número de iterações necessário na estimação dos modelos. Esta análise teve como base um estudo de simulação cujo processo de criação dos dados é descrito de forma detalhada no decorrer da dissertação.

Os estudos de simulação são muito frequentes em Estatística, permitindo avaliar o desempenho e as propriedades de modelos estatísticos em cenários controlados e conhecidos. O estudo de simulação desenvolvido teve como principal objetivo avaliar e comparar a estimação dos coeficientes dos modelos e a capacidade de previsão dos mesmos usando funções de diferentes *packages* da linguagem de programação R. Na avaliação da previsão do modelo estudou-se a capacidade preditiva quando se estima o modelo com diferentes dimensões de subamostras.

Por fim, aplicou-se um MLG numa base de microdados real disponibilizada pelo Laboratório de Investigação em Microdados do Banco de Portugal (BPLIM). Neste caso de estudo pretendeu-se identificar quais as variáveis do Painel harmonizado da Central de Balanços que melhor explicam o facto de uma empresa ser ou não exportadora.

Com base nos estudos de simulação, concluiu-se no primeiro estudo que a função `bayesglm` apresenta estimativas menores nas medidas avaliadas, e no segundo estudo que a variabilidade das medidas de desempenho diminui à medida que a dimensão da amostra aumenta. No caso de estudo, obteve-se um modelo com uma Acurácia de aproximadamente 65%.

Palavras-chave: estimação, estudo de simulação, linguagens de programação, modelos lineares generalizados, predição.

Abstract

Generalized Linear Models (GLM) are often used when you want to study the relationship between variables, especially when analyzing the impact that one or more explanatory variables have on a given variable of interest (response variable). The importance of MLG does not only come from an applied perspective, but also from its deep theoretical meaning. The theoretical relevance of these models results from the fact that they are the basis of many statistical methods and are used in various applications, highlighting the centrality of likelihood in the theory of inference.

This dissertation aims to assess and compare the performance of different programming languages in the estimation of GLM and apply these models to a real microdata base.

Firstly, a detailed description of the MLG theory is made with particular emphasis on the two regression models that served as the object of study: the Logistic Regression Model and the Poisson Regression Model.

Given that MLG can be estimated in several programming languages, a comparative analysis is carried out with applications in R, Stata and Python, with the purpose of evaluating their performance. In this context, several criteria were analyzed, including computational performance, time and number of iterations required to estimate the models. This analysis was based on a simulation study whose data creation process is described in detail throughout the dissertation. Given the similarity of the results obtained in the various programming languages, the study was developed mainly in R.

Simulation studies are very common in Statistics, allowing the performance and properties of statistical models to be evaluated in controlled and known scenarios. The main objective of the simulation study developed was to estimate the model coefficients and their predictive capacity using functions from different packages of the R programming language, allowing their comparison. When evaluating the model's prediction, the predictive capacity was studied when estimating the model with different subsample dimensions.

Finally, an MLG was applied to a real microdata base provided by the Banco de Portugal Microdata Research Laboratory (BPLIM). In this case study, the aim was to identify which variables from the Harmonized Central Balance Sheet Panel best explain whether or not a company is an exporter.

Based on simulation studies, it was concluded, in the first study, that the `bayesglm` function provides lower estimates for the evaluated metrics, and in the second study that the variability of performance metrics decreases as the sample dimension increases. In the case of this study, a model was obtained with an accuracy of approximately 65%.

Keywords: estimation, simulation study, programming languages, generalized linear models, prediction.

Índice

Índice de Figuras

Índice de Tabelas

Lista de Códigos

Acrónimos

AIC Critério de informação de Akaike

BIC Critério de informação de Bayes

BPLIM Laboratório de Investigação em Microdados do Banco de Portugal

CV Coeficiente de Variação

fdp Função densidade de probabilidade

fgc Função geradora de cumulantes

fgm Função geradora de momentos

fmp Função massa de probabilidade

IDE Ambiente de desenvolvimento integrado

MAPE Mean absolute percentage error (em português: Erro Percentual Absoluto Médio)

MLG Modelos Lineares Generalizados

MSE Mean squared error (em português: Erro Quadrático Médio)

PRE Percent Relative Efficiency (em prtuguÊs: Eficiência Relativa em Percentagem)

SQR Soma dos Quadrados dos Resíduos

1 Introdução

O modelo linear normal foi, durante muitos anos, o modelo aplicado na modelação estatística. Este modelo pressupunha a normalidade dos erros e a variância constante. Quando o fenómeno sob estudo não apresentava uma resposta para a qual fosse possível assumir estes pressupostos, eram aplicadas transformações às variáveis. De entre as várias transformações, a mais conhecida é provavelmente a proposta por Box e Cox, em 1964, que transforma a variável resposta com o objetivo de se obter um melhor ajuste do modelo (Box & Cox, 1964).

Todavia, apesar das várias transformações, nem todas as situações sob estudo eram adequadamente explicadas pelo modelo linear normal e para contrariar essa adversidade, foram desenvolvidos modelos não lineares ou não normais, tais como o modelo complementar *log-log*, para ensaios de diluição, os modelos *probit* e *logit*, para proporções, os modelos *log-lineares*, para dados de contagens, e os modelos de regressão, para análise de dados de sobrevivência. (Turkman & Silva, 2000)

Os modelos lineares generalizados (MLG), introduzidos por Nelder e Wedderburn em 1972, vieram unificar os modelos enunciados acima (Nelder & Wedderburn, 1972). A ideia base destes modelos consiste em abrir o leque de opções para a distribuição da variável resposta, quando esta pertence à família exponencial de distribuições.

Estes modelos apresentam algumas limitações, como o facto de exigirem a independência entre as variáveis respostas, as distribuições da variável resposta se restringirem à família exponencial e por manterem a estrutura de linearidade. Contudo, apesar destas limitações, os MLG têm vindo a desempenhar um papel cada vez mais relevante na análise estatística.

As linguagens de programação, como o R , o Stata e o Python, são bastante utilizadas para estudar estes modelos. O intuito deste projeto passa por analisar estes modelos nessas diferentes linguagens.

No âmbito do Mestrado em Estatística para Ciência de Dados, da Universidade do Minho, foi realizado um estágio nas instalações do Departamento de Estudos Económicos do Banco de Portugal. Este estágio teve como destino específico o Laboratório de Investigação em Microdados (BPLIM).

O presente capítulo tem por objetivo fornecer uma visão detalhada do Banco de Portugal, com foco especial no Laboratório de Investigação em Microdados. Adicionalmente, são apresentados dados introdutórios sobre a instituição, incluindo uma breve contextualização da sua função e importância. Também são delineados os principais objetivos do estágio em questão e a estrutura global planeada para o relatório subsequente.

1.1 Laboratório de Investigação em Microdados do Banco de Portugal

O Banco de Portugal, tal como os demais bancos centrais, faz a ligação entre os produtores de microdados administrativos e a comunidade de investigadores. Estas entidades têm acesso a um vasto conjunto de dados

administrativos confidenciais e dispõem de uma equipa de investigadores que utilizam de forma adequada esses dados nas suas pesquisas. Após vários anos de pesquisa, o Banco de Portugal sentiu a necessidade de criar uma unidade de investigação centrada na análise de microdados. Surge, assim, o Laboratório de Investigação em Microdados do Banco de Portugal (BPLIM), com o intuito de aproveitar e melhorar o poder dos conjuntos de microdados administrativos portugueses.

O BPLIM está localizado na sucursal do Banco de Portugal no Porto e é uma unidade independente, que integra o Departamento de Estudos Económicos, tendo iniciado a sua atividade em 2016.

Através do BPLIM, vários investigadores têm ao seu dispor diferentes bases de microdados reais anonimizadas e bem documentadas, de modo a satisfazer as necessidades particulares de cada estudo. O BPLIM permite que os investigadores usem os recursos computacionais disponíveis no laboratório, ou que os mesmos acedam aos dados remotamente, sendo sempre o objetivo central estimular a investigação deste tipo de dados.

1.2 Objetivos

O tema proposto para a realização desta dissertação é "Modelos Lineares Generalizados: avaliação de ferramentas de programação numa aplicação com bases de microdados reais". Os objetivos propostos são:

- avaliar e comparar o desempenho de diferentes linguagens de programação na estimação dos modelos lineares generalizados;
- desenvolver estudos de simulação para estudar a capacidade preditiva dos modelos;
- aplicar os modelos lineares generalizados a uma base de microdados real.

1.3 Estrutura

A presente dissertação é constituída por 5 Capítulos. No primeiro capítulo introduz-se o tema central do estudo e são delimitados os objetivos que direcionam o trabalho. No segundo capítulo inicia-se um estudo aprofundando dos MLG onde são detalhados conceitos centrais incluindo a Família Exponencial, a função Geradora de Momentos e Cumulantes, bem como as três componentes essenciais dos MLG. Além disso são abordadas técnicas fundamentais de inferência, como estimação de parâmetros e testes de hipóteses. A discussão estende-se a métodos de seleção e validação de modelos, que incluem medidas de qualidade de ajustamento e a presença de observações atípicas. Para complementar essa abordagem, o capítulo conclui com dois exemplos de MLG: o Modelo de Regressão Logística e o Modelo de Regressão Poisson. O Capitulo 3 foca-se no estudo de simulação como parte integral da pesquisa. São exploradas diferentes linguagens de programação e

delineados os passos para a criação de dados simulados. Além disso, a comparação entre as linguagens de programação é abordada, bem como as etapas de estimação de parâmetros e previsão. O Capitulo 4 apresenta o caso de estudo. Inicialmente descreve-se as variáveis existentes na base de microdados em estudo (CBHP - Painel Harmonizado da Central de Balanços) e, em seguida, estima-se um MLG com as variáveis significativas para explicar a variável que identifica as empresas exportadoras. No Capitulo 5,apresentam-se as principais conclusões deste trabalho e o trabalho futuro.

2 Modelos Lineares Generalizados

Encontrar um modelo que se ajuste aos dados é uma das tarefas mais importantes de um estudo estatístico. O objetivo principal passa por encontrar o modelo mais simples possível que descreva corretamente os dados e que seja de fácil compreensão, para os seus utilizadores.

Os modelos de regressão são normalmente usados em problemas onde o objetivo principal é estudar a relação entre variáveis, ou seja, estudar a influência que uma ou mais variáveis explicativas têm sobre uma única variável resposta.

O modelo linear é o mais simples modelo de regressão e foi desenvolvido no início do século XIX. Contudo, este modelo apresenta algumas limitações, uma vez que não pode ser usado se a distribuição da variável resposta for diferente da distribuição Normal. Para solucionar este problema, Nelder e Wedderburn, em 1972, debruçaram-se sobre o assunto e demonstraram que diferentes técnicas utilizadas separadamente podem ser formuladas e unificadas em uma só classe de modelos de regressão, os Modelos Lineares Generalizados (MLG), que são uma extensão do modelo de regressão linear clássico (Nelder & Wedderburn, 1972).

2.1 Família Exponencial

Os MLG pressupõem que a variável resposta segue uma distribuição que pertence à Família Exponencial.

As distribuições da Família Exponencial surgem na Mecânica Estatística, no final do século XIX, sendo o conceito de Família Exponencial introduzido na Estatística por Fisher (Fisher, 1922). No entanto, a Família Exponencial começou a ter mais destaque depois de Nelder e Wedderbun definirem MLG.

A distribuição de uma variável aleatória, Y, pertence à Família Exponencial se a sua função densidade de probabilidade (fdp) ou função massa de probabilidade (fmp) se puder escrever na forma

$$f(y|\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}, \tag{2.1}$$

onde θ é o parâmetro canónico de localização, ϕ é o parâmetro de dispersão e $a(.)$, $b(.)$, $c(.)$ são funções reais específicas que determinam unicamente a distribuição (Turkman & Silva, 2000).

As distribuições Normal, Binomial, Binomial Negativa, Gama, Poisson, Normal Inversa, Multinomial, Beta, entre outras, pertencem à Família Exponencial.

O valor esperado e a variância da variável aleatória Y, que pertence à Família Exponencial, são dadas por

$$\mu = E[Y] = b'(\theta) \tag{2.2}$$

$$var(Y) = a(\phi)b''(\theta). \tag{2.3}$$

A variância de Y é o produto da função $a(\phi)$ e de $b''(\theta)$, onde $a(\phi)$ depende apenas do parâmetro de dispersão ϕ e $b''(\theta)$ depende apenas do parâmetro canónico θ, e portanto do valor médio, ao qual se dá o nome de Função de Variância e se representa por $V(\mu)$ (Davison, 2003),

$$V(\mu) = b''(\theta) = \frac{d\mu}{d\theta}. \tag{2.4}$$

A função $a(\phi)$ é da forma $a(\phi) = \frac{\phi}{w}$, onde w é um peso conhecido *à priori*. Assim, a função definida em (2.1) pode ser escrita na forma

$$f(y|\theta, \phi, w) = \exp\left\{\frac{w}{\phi}(y\theta - b(\theta)) + c(y, \phi)\right\}.$$

O processo, que consiste em averiguar se uma distribuição pertence ou não à Família Exponencial, é muito simples. Considerem-se os exemplos seguintes.

Exemplo 2.1.1. *Distribuição Normal:* $Y \sim N(\mu, \sigma^2)$

A fdp de Y é dada por:

$$\begin{aligned} f(y|\mu, \sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\} \\ &= exp\left\{\frac{1}{\sigma^2}\left(y\mu - \frac{\mu^2}{2}\right) - \frac{1}{2}\left(\frac{y^2}{\sigma^2} + ln(2\pi\sigma^2)\right)\right\}, \end{aligned}$$

A distribuição Normal pertence à família exponencial, considerando as seguintes igualdades

$$\begin{aligned} \theta = \mu, \quad b(\theta) = \tfrac{\mu^2}{2}, \quad \phi = \sigma^2, \quad w = 1, \\ a(\phi) = \sigma^2 \quad e \quad c(y, \phi) = -\tfrac{1}{2}\left(\tfrac{y^2}{\sigma^2} + ln(2\pi\sigma^2)\right). \end{aligned}$$

O valor médio e a variância são dados por

$$\begin{aligned} E[Y] &= b'(\theta) = \theta = \mu \\ var[Y] &= a(\phi)b''(\theta) = \sigma^2. \end{aligned}$$

A função de variância é, neste caso, $V(\mu)$*=1. O parâmetro de dispersão é* σ^2 *e o parâmetro canónico de localização é* μ*.*

Exemplo 2.1.2. *Distribuição Poisson:* $Y \sim P(\mu)$

A fmp de Y é dada por:

$$\begin{aligned} f(y|\mu) &= \frac{e^{-\mu}\mu^y}{y!} \\ &= exp\,(y\ ln(\mu) - \mu - ln(y!)). \end{aligned}$$

Considera-se que a distribuição Poisson pertence à família exponencial igualando

$$\theta = ln(\mu), \quad \phi = 1, \quad a(\phi) = 1, \quad w = 1,$$
$$b(\theta) = \mu = exp(\theta) \text{ e } c(y, \phi) = -ln(y!).$$

O valor médio e a variância são

$$E[Y] = b'(\theta) = e^{\theta} = \mu$$
$$var[Y] = b''(\theta)a(\phi) = e^{\theta} = \mu.$$

Neste caso, o parâmetro de dispersão é 1, o parâmetro canónico é ln(μ) e $V(\mu)$= μ.

Exemplo 2.1.3. *Distribuição Gama: $Y \sim G(\nu, \frac{\nu}{\mu})$*

A fdp de Y é dada por:

$$\begin{aligned} f(y|\nu, \mu) &= \frac{1}{\Gamma(\nu)} \left(\frac{\nu}{\mu}\right)^{\nu} y^{\nu-1} exp\left(-\frac{\nu}{\mu} y\right) \\ &= exp\left[\nu\left(-\frac{y}{\mu} - ln(\mu)\right) + (\nu - 1)\, ln(y) - ln(\Gamma(\nu)) + \nu\, ln(\nu)\right] \\ &= exp\left[\nu\,(\theta y + ln(-\theta)) + (\nu - 1)\, ln(y) - ln(\Gamma(\nu)) + \nu\, ln(\nu)\right] \end{aligned}$$

com $y > 0$ e $\theta = -\frac{1}{\mu}$.

A distribuição Gama pertence à família exponencial considerando-se

$$\theta = -\tfrac{1}{\mu}, \quad \phi = \tfrac{1}{\nu}, \quad a(\phi) = \phi, \quad w = 1,$$
$$b(\theta) = -ln(-\theta) \text{ e } c(y, \phi) = (\nu - 1)\, ln(y) - ln(\Gamma(\nu)) + \nu\, ln(\nu).$$

Obtendo-se o valor médio e variância

$$E[Y] = b'(\theta) = -\tfrac{1}{\theta} = \mu$$
$$var[Y] = b''(\theta)a(\phi) = \tfrac{1}{\theta^2}\tfrac{1}{\nu} = \tfrac{\mu^2}{\nu}.$$

A função de variância é, neste caso, V(μ)=μ^2 e o parâmetro de dispersão é $\frac{1}{\nu}$.

Existem, no entanto, outras distribuições que não pertencem à família exponencial mas que podem ser transformadas. Um exemplo disso é a distribuição Binomial.

Exemplo 2.1.4. *A distribuição Binomial não pertence à família exponencial, contudo, considerando a transformação*

$$Y = \frac{X}{m}, \text{ tal que } X \sim Binomial(m, \pi).$$

Pode-se mostrar que $Y \sim B(m, \pi/m)$ *pertence.*

A fmp de Y é dada por,

$$\begin{aligned} f(y|\pi) &= \binom{m}{my}\pi^{ym}(1-\pi)^{m-ym} \\ &= \exp\left\{ym\ ln(\pi) + m(1-y)\ ln(1-\pi) + ln\binom{m}{ym}\right\} \\ &= \exp\left\{m(y\theta - ln(1+e^{\theta})) + ln\binom{m}{ym}\right\}, \end{aligned}$$

com $y \in \left\{0, \frac{1}{m}, \frac{2}{m}, ..., 1\right\}$ *e* $\theta = ln\left(\frac{\pi}{1-\pi}\right)$.

A variável aleatória pertence à família exponencial com

$$\theta = ln\left(\frac{\pi}{1-\pi}\right), \quad b(\theta) = ln(1+e^{\theta}), \quad \phi = 1, \quad w = m,$$
$$a(\phi) = \frac{1}{m} \quad e \quad c(y,\phi) = ln\left(\binom{m}{ym}\right).$$

Consequentemente, o valor médio e a variância são

$$E[Y] = b'(\theta) = \frac{e^{\theta}}{1+e^{\theta}} = \pi$$
$$var[Y] = \frac{e^{\theta}}{m(1+e^{\theta})^2} = \frac{\pi(1-\pi)}{m}.$$

A função de variância é $V(\mu) = \pi(1-\pi)$ e o parâmetro de dispersão é 1.

2.2 Função Geradora de Momentos e Cumulantes

Seja Y uma variável aleatória, define-se função geradora de momentos (fgm) como, (Pestana & Velosa, 2010),

$$M_Y(t) = E[e^{tY}], \tag{2.5}$$

para todo o valor de t para o qual o valor esperado existe.

A fgm permite obter todos os momentos de uma variável aleatória. Tal como a fdp, permite identificar uma variável aleatória, isto é, uma fgm corresponde a uma única variável aleatória e por outro lado, uma variável aleatória possui uma única fgm .

Relativamente à Família Exponencial, pode-se definir a fgm como

$$M(t;\theta,\phi) = E[e^{tY}] = \exp\left\{\frac{b(a(\phi)t+\theta) - b(\theta)}{a(\phi)}\right\} \tag{2.6}$$

Demonstração

No caso de uma variável aleatória contínua tem-se

$$\begin{aligned}
&\int_A f(y;\theta,\phi)dy = 1 \\
\iff &\int_A \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\} dy = 1 \\
\iff &\frac{1}{\exp\left(\frac{b(\theta)}{a(\phi)}\right)} \int_A \exp\left\{\frac{y\theta}{a(\phi)} + c(y,\phi)\right\} dy = 1
\end{aligned}$$

$$\iff \int_A \exp\left\{\frac{y\theta}{a(\phi)} + c(y,\phi)\right\} dy = \exp\left(\frac{b(\theta)}{a(\phi)}\right) \tag{2.7}$$

Logo,

$$\begin{aligned}
M(t;\theta,\phi) &= E[e^{tY}] \\
&= \int_A \exp(ty) f(y) dy \\
&= \int_A \exp(ty)\, \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\} dy \\
&= \int_A \exp\left\{\frac{(a(\phi)t+\theta)y}{a(\phi)} - \frac{b(\theta)}{a(\phi)} + c(y,\phi)\right\} dy \\
&= \frac{1}{\exp\left(\frac{b(\theta)}{a(\phi)}\right)} \int_A \exp\left\{\frac{(a(\phi)t+\theta)y}{a(\phi)} + c(y,\phi)\right\} dy
\end{aligned}$$

Usando a equação (2.7) tem-se que

$$\begin{aligned}
M(t;\theta,\phi) &= \frac{1}{\exp\left(\frac{b(\theta)}{a(\phi)}\right)} \exp\left\{\frac{b(a(\phi)t+\theta)}{a(\phi)}\right\} \\
&= \exp\left\{\frac{b(a(\phi)t+\theta) - b(\theta)}{a(\phi)}\right\}
\end{aligned} \tag{2.8}$$

Quando se estudam variáveis aleatórias discretas, é possível substituir a integração pelo somatório, o que leva ao mesmo resultado.

A função geradora de cumulantes (fgc) correspondente define-se por

$$\varphi(t;\theta,\phi) = \ln[M(t;\theta,\phi)] = \frac{b(a(\phi)t+\theta) - b(\theta)}{a(\phi)} \tag{2.9}$$

A fgc está diretamente relacionada com a fgm, contudo, em Estatística, a fgc é mais relevante que a fgm , visto que uma grande parte da teoria assintótica depende das suas propriedades.

Derivando a expressão (2.9), verifica-se que existe uma relação de recorrência entre os cumulantes da família exponencial.

$$\begin{aligned}
\varphi'(t;\theta,\phi) &= b'(a(\phi)t+\theta) \\
\varphi''((t;\theta,\phi) &= b''(a(\phi)t+\theta)a(\phi) \\
\varphi'''((t;\theta,\phi) &= b'''(a(\phi)t+\theta)a(\phi)^2 \\
&\vdots \qquad\qquad \vdots \\
\varphi^{(r)}(t;\theta,\phi) &= b^{(r)}(a(\phi)t+\theta)(a(\phi))^{(r-1)}
\end{aligned}$$

Para $t=0$, obtém-se o r-ésimo cumulante da família exponencial.

$$\varphi^{(r)}(t;\theta,\phi) = b^{(r)}(\theta)a(\phi)^{(r-1)} \tag{2.10}$$

Obtém-se assim o valor esperado, $\varphi^{(1)}$, e a variância, $\varphi^{(2)}$, de uma variável aleatória cuja distribuição pertence à Família Exponencial através da equação (2.10).

Para $r=1$, $\varphi^{(1)}(t;\theta,\phi) = b'(\theta) = E[Y]$.

Para $r=2$, $\varphi^{(2)}(t;\theta,\phi) = b''(\theta)a(\phi) = Var[Y]$.

Através das equações (2.9) e (2.8) consegue-se obter facilmente a fgc e fgm de qualquer distribuição que pertença à Família Exponencial, tal como se pode confirmar nos exemplos seguintes.

Exemplo 2.2.1. *Distribuição Normal:* $N(\mu,\sigma^2)$

Considerando o Exemplo 2.1.1, tem-se que $a(\phi)=\sigma^2$, $\theta=\mu$ *e* $b(\theta)=\frac{\theta^2}{2}$. *A fgc é*

$$\begin{aligned}
\varphi(t) &= \frac{1}{\sigma^2}\left[\frac{(\sigma^2 t+\mu)^2}{2} - \frac{\mu^2}{2}\right] \\
&= \frac{1}{2}(\sigma^2 t^2 + 2t\mu) \\
&= t\mu + \frac{\sigma^2 t^2}{2}.
\end{aligned}$$

Logo, a fgm é

$$M(t) = exp\left\{t\mu + \frac{\sigma^2 t^2}{2}\right\}$$

Exemplo 2.2.2. *Distribuição Poisson:* $P(\mu)$

Considerando o Exemplo 2.1.2, tem-se $a(\phi) = 1$, $\theta = ln(\mu)$ *e* $b(\theta) = exp\,(\theta)$, *a fgc é*

$$\begin{aligned}\varphi(t) &= exp(t + ln\,(\mu)) - \mu \\ &= \mu(e^t - 1).\end{aligned}$$

Consequentemente, a fgm é

$$M(t) = exp(\mu(e^t - 1)).$$

Exemplo 2.2.3. *Distribuição Gama* $G(\nu, \frac{\nu}{\mu})$

Tendo em conta o Exemplo 2.1.3, note-se que $a(\phi) = \frac{1}{\nu}$, $\theta = -\frac{1}{\mu}$ *e* $b(\theta) = -ln(-\theta)$, *a fgc é*

$$\begin{aligned}\varphi(t) &= \nu\left(-ln\left(\frac{1}{\mu} - \frac{t}{\nu}\right) + ln\left(\frac{1}{\mu}\right)\right) \\ &= \nu ln\left(\frac{-t\mu^2 + \mu\nu}{\mu\nu}\right)^{-1} \\ &= ln\left(1 - \frac{t\mu}{\nu}\right)^{-\nu}\end{aligned}$$

Logo, a fgm é

$$M(t) = \left(1 - \frac{t\mu}{\nu}\right)^{-\nu}$$

As funções geradoras de momentos de algumas distribuições que pertencem à Família Exponencial encontram-se sintetizadas na Tabela 2.1.

Tabela 2.1: Funções geradoras de momentos de algumas distribuições

Distribuição	Função geradora de momentos
Normal: N(μ, σ^2)	exp $\left(t\mu + \frac{\sigma^2 t^2}{2}\right)$
Poisson: P(μ)	exp $[\mu(e^t - 1)]$
Binomial: B(m, π)	$\left(\frac{m-\mu}{m} + \frac{\mu}{m}e^t\right)^m$
Binomial Negativa: BN(μ, k)	$\left[1 + \frac{\mu}{k}(1 - e^t)\right]^{-k}$
Normal Inversa IG(μ, σ^2)	exp $\left[\frac{1}{\sigma^2}\left[\frac{1}{\mu} - \left(\frac{1}{\mu^2} - 2t\sigma^2\right)^{1/2}\right]\right]$, $t < \frac{1}{2\sigma^2\mu^2}$
Gama: G(μ, ν)	$\left(1 - \frac{t\mu}{\nu}\right)^{-\nu}$,$t < \frac{\nu}{\mu}$

2.3 Componentes

Os MLG apresentam três componentes. A componente aleatória identifica a distribuição condicionada de Y dadas as variáveis explicativas e pertence à família exponencial. A componente sistemática considera uma combinação linear das variáveis explicativas. Por fim, a função ligação, como o próprio nome sugere, faz a ligação entre as componentes aleatória e sistemática (Fox, 2016).

Componente Aleatória

Dado o vetor de covariáveis $\boldsymbol{x_i}$, as variáveis Y_i são condicionalmente independentes com distribuição pertencente à Família Exponencial

$$E[Y_i|\boldsymbol{x_i}] = b'(\theta_i) = \mu_i, \text{ para } i = 1, ..., n.$$

Componente Sistemática

Corresponde ao preditor linear, η_i,

$$\eta_i = \beta_0 + \sum_{j=1}^{p} X_j\beta_j = z_i^T\boldsymbol{\beta}, \tag{2.11}$$

onde $\beta_j, j = 1, ..., p$ são os coeficientes do modelo, $z_i = (1, x_i^T)$ e $\boldsymbol{\beta}$ é o vetor dos parâmetros de regressão.

Função de ligação

Esta função monótona e diferenciável, $g(.)$, relaciona o valor esperado, μ_i, com o preditor linear, η_i,

$$g(\mu_i) = \eta_i. \tag{2.12}$$

Quando o preditor linear coincide com o parâmetro canónico, ou seja,

$$g(\mu_i) = \eta_i = \theta_i,$$

a função de ligação correspondente diz-se então Função de Ligação Canónica.

As funções de ligação canónicas de algumas distribuições resumem-se na Tabela 2.2.

Tabela 2.2: Funções de ligação canónicas de algumas distribuições

Distribuição	Ligação
Normal	Identidade: $\eta = \mu$
Poisson	Logarítmica: $\eta = \ln\mu$
Binomial	Logística: $\eta = \ln(\mu/(1-\mu))$
Gamma	Recíproca: $\eta = \mu^{-1}$
Normal Inversa	Recíproca quadrática: $\eta = \mu^{-2}$

Uma das vantagens de serem usadas ligações canónicas é que estas garantem a concavidade da função de verosimilhança e, consequentemente, muitos resultados assintóticos são obtidos mais facilmente. Por exemplo, a concavidade da função de verosimilhança garante a unicidade da estimativa de máxima verosimilhança de $\boldsymbol{\beta}$, quando essa existe.

2.4 Estimação

Após a formulação do modelo, é necessário proceder à realização de inferências sobre o mesmo (Turkman & Silva, 2000). Nos Modelos Lineares Generalizados (MLG), a inferência é baseada na verosimilhança.

Verosimilhança e matriz de Fisher

Consideremos um MLG, definido como na secção 2.1

$$f(y_i|\theta_i,\phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right\}, \quad i = 1,...,n$$

com função de ligação $g(\mu_i) = \eta_i = \boldsymbol{z}_i^T\boldsymbol{\beta}$, sendo as variáveis aleatórias Y_i independentes.

A função de verosimilhança, como função de $\boldsymbol{\beta}$, é

$$\begin{aligned} L(\boldsymbol{\beta}) &= \prod_{i=1}^{n} f(y_i|\theta_i,\phi) \\ &= \prod_{i=1}^{n} \exp\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right] \\ &= \exp\left[\sum_{i=1}^{n}\frac{1}{a(\phi)}(y_i\theta_i - b(\theta_i)) + \sum_{i=1}^{n} c(y_i,\phi)\right] \end{aligned} \tag{2.13}$$

O logaritmo da função de verosimilhança, ou seja, a função log-verosimilhança, que se denota por ℓ, é dado por

$$\begin{aligned} \ln L(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) &= \sum_{i=1}^{n}\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi) \\ &= \sum_{i=1}^{n}\ell_i(\boldsymbol{\beta}), \end{aligned} \tag{2.14}$$

onde

$$\ell_i(\boldsymbol{\beta}) = \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi) \tag{2.15}$$

é a contribuição de cada observação y_i para a verosimilhança.

Os estimadores de máxima verosimilhança são obtidos através do seguinte sistema de equações (Turkman &

Silva, 2000)

$$\frac{\partial \ell(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial \ell_i(\boldsymbol{\beta})}{\partial \beta_j} = 0, \qquad j = 0, ..., p.$$

Assim, aplicando a regra da cadeia, obtém-se

$$\frac{\partial \ell_i(\boldsymbol{\beta})}{\partial \beta_j} = \frac{\partial \ell_i(\theta_i)}{\partial \theta_i} \frac{\partial \theta_i(\mu_i)}{\partial \mu_i} \frac{\partial \mu_i(\eta_i)}{\partial \eta_i} \frac{\partial \eta_i(\boldsymbol{\beta})}{\partial \beta_j}, \qquad j = 0, ..., p.$$

Usando a definição e as propriedades de MLG, tem-se

$$\frac{\partial \ell_i(\theta_i)}{\partial \theta_i} = \frac{y_i - b'(\theta_i)}{a(\phi)} = \frac{y_i - \mu_i}{a(\phi)}, \tag{2.16}$$

através da função de log-verosimilhança e da igualdade 2.2,

$$\frac{\partial \mu_i}{\partial \theta_i} = b''(\theta_i) = \frac{var(Y_i)}{a(\phi)}, \tag{2.17}$$

pela igualdade (2.3) e

$$\frac{\partial \eta_i(\boldsymbol{\beta})}{\partial \beta_j} = z_{ij}, \tag{2.18}$$

a partir de (2.11).

Assim, através de (2.16), (2.17), (2.18), concluí-se que

$$\begin{aligned} \frac{\partial \ell_i(\boldsymbol{\beta})}{\partial \beta_j} &= \frac{y_i - \mu_i}{a(\phi)} \frac{a(\phi)}{var(Y_i)} \frac{\partial \mu_i}{\partial \eta_i} z_{ij} \\ &= \frac{(y_i - \mu_i) z_{ij}}{var(Y_i)} \frac{\partial \mu_i}{\partial \eta_i}. \end{aligned} \tag{2.19}$$

e as equações de verosimilhança para $\boldsymbol{\beta}$ são

$$\sum_{i=1}^{n} \frac{(y_i - \mu_i) z_{ij}}{var(Y_i)} \frac{\partial \mu_i}{\partial \eta_i} = 0, \qquad j = 0, ..., p. \tag{2.20}$$

Em geral, estas equações são funções não lineares dos parâmetros β e são resolvidas numericamente por processos iterativos. No caso do MLG, usa-se o método de *scores* de Fisher que, em geral, é o mais simples. Para tal, recorre-se à função *score*.

Defina-se a função *score* como a derivada da função log-verosimilhança em ordem a $\boldsymbol{\beta}$

$$s(\boldsymbol{\beta}) = \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} s_i(\boldsymbol{\beta}), \tag{2.21}$$

onde $s_i(\boldsymbol{\beta})$ é o vetor de componentes $\frac{\partial \ell_i(\boldsymbol{\beta})}{\partial \beta_j}$ obtidas em (2.20).

Assim, o elemento j da função *score* é

$$\sum_{i=1}^{n} \frac{(y_i - \mu_i) z_{ij}}{var(Y_i)} \frac{\partial \mu_i}{\partial \eta_i}. \tag{2.22}$$

A matriz de informação de Fisher é a matriz de covariância da função *score*

$$cov(s(\boldsymbol{\beta})) = I(\boldsymbol{\beta}) = E\left[-\frac{\partial s(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right].$$

Logo, para obter a matriz de informação de Fisher, é necessário conhecer a segunda derivada de $\ell_i(\boldsymbol{\beta})$. Tem-se, para famílias regulares, que

$$\begin{aligned}
-E\left(\frac{\partial^2 \ell_i}{\partial \beta_j \partial \beta_k}\right) &= E\left(\frac{\partial \ell_i}{\partial \beta_j}\frac{\partial \ell_i}{\partial \beta_k}\right) \\
&= E\left[\left(\frac{(Y_i - \mu_i) z_{ij}}{var(Y_i)}\frac{\partial \mu_i}{\partial \eta_i}\right)\left(\frac{(Y_i - \mu_i) z_{ik}}{var(Y_i)}\frac{\partial \mu_i}{\partial \eta_i}\right)\right] \\
&= E\left[\frac{(Y_i - \mu_i)^2 z_{ij} z_{ik}}{(var(Y_i))^2}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2\right] \\
&= \frac{z_{ij} z_{ik}}{(var(Y_i))}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2.
\end{aligned}$$

Portanto, o elemento de ordem (j, k) da matriz de informação de Fisher é

$$-\sum_{i=1}^{n} E\left(\frac{\partial^2 \ell_i}{\partial \beta_j \partial \beta_k}\right) = \sum_{i=1}^{n} \frac{z_{ij} z_{ik}}{(var(Y_i))}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2. \tag{2.23}$$

Assim, a matriz de informação de Fisher pode ser escrita de forma matricial

$$I(\boldsymbol{\beta}) = Z^T W Z \tag{2.24}$$

onde W é a matriz diagonal de ordem n, cujo i-ésimo elemento é

$$w_i = \frac{\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2}{var(Y_i)} = \frac{w_i \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2}{\phi V(\mu_i)}. \tag{2.25}$$

Estimação de β

Partindo do pressuposto de que existe uma solução única para a equação (2.20), usa-se o método iterativo de mínimos quadrados ponderados para a obter. Este método iterativo é baseado na função *score* de Fisher e, contrariamente ao algoritmo de Newton-Raphson, que usa a matriz Hessiana, este usa a matriz de informação de Fisher, o que é vantajoso, uma vez que, em geral, é mais fácil calcular a matriz de informação de Fisher, para além

de que esta é uma matriz semi-definida positiva.

Considere-se $\hat{\beta}$ uma estimativa inicial para $\boldsymbol{\beta}$ e obtém-se sucessivas iterações através da relação

$$\hat{\boldsymbol{\beta}}^{(k+1)} = \hat{\boldsymbol{\beta}}^{(k)} + \left[I(\hat{\boldsymbol{\beta}}^{(k)})\right]^{-1} s(\hat{\boldsymbol{\beta}}^{(k)}), \tag{2.26}$$

onde $I(.)^{-1}$ é a inversa da matriz de informação de Fisher e $s(.)$ o vetor de *scores*.

Equivalentemente

$$\left[I(\hat{\boldsymbol{\beta}}^{(k)})\right] \hat{\boldsymbol{\beta}}^{(k+1)} = \left[I(\hat{\boldsymbol{\beta}}^{(k)})\right] \hat{\boldsymbol{\beta}}^{(k)} + s(\hat{\boldsymbol{\beta}}^{(k)}). \tag{2.27}$$

Atendendo a (2.22) e (2.23), pode-se escrever que (2.27) é um vetor com elemento genérico de ordem l dado por

$$\left[I(\hat{\boldsymbol{\beta}}^{(k)})\right] \hat{\boldsymbol{\beta}}^{(k+1)} = \sum_{j=1}^{p} \left[\sum_{i=1}^{n} \frac{z_{ij} z_{il}}{(Var(Y_i))} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2\right] {\beta_j}^{(k)} + \sum_{i=1}^{n} \frac{(y_i - \mu_i) z_{il}}{var(Y_i)} \frac{\partial \mu_i}{\partial \eta_i}. \tag{2.28}$$

Na forma matricial tem-se

$$\left[I(\hat{\boldsymbol{\beta}}^{(k)})\right] \hat{\boldsymbol{\beta}}^{(k+1)} = Z^T W^{(k)} \boldsymbol{u}^{(k)},$$

onde a matriz $W^{(k)}$ representa a matriz W definida em 2.25, calculada em $\hat{\boldsymbol{\mu}}^{(k)}$, e $\boldsymbol{u}^{(k)}$ é um vetor com elemento genérico

$$\begin{aligned} u_i^{(k)} &= \sum_{j=0}^{p} z_{ij} \beta_j^{(k)} + (y_i - \mu_i)^{(k)} \frac{\partial \eta_i^{(k)}}{\partial \mu_i^{(k)}} \\ &= \eta_i^{(k)} + (y_i - \mu_i)^{(k)} \frac{\partial \eta_i^{(k)}}{\partial \mu_i^{(k)}}. \end{aligned} \tag{2.29}$$

Assim, usando 2.24, tem-se a expressão final para a estimação de $\boldsymbol{\beta}$ na (k+1)-ésima iteração

$$\begin{aligned} &(Z^T W^{(k)} Z) \hat{\boldsymbol{\beta}}^{(k+1)} = Z^T W^{(k)} \boldsymbol{u}^{(k)} \\ &\iff (Z^T W^{(k)} Z)^{-1} (Z^T W^{(k)} Z) \hat{\boldsymbol{\beta}}^{(k+1)} = (Z^T W^{(k)} Z)^{-1} Z^T W^{(k)} \boldsymbol{u}^{(k)} \\ &\iff \hat{\boldsymbol{\beta}}^{(k+1)} = (Z^T W^{(k)} Z)^T Z^T W^{(k)} \boldsymbol{u}^{(k)}. \end{aligned} \tag{2.30}$$

Estimação de ϕ

O logaritmo de máxima verosimilhança também pode ser usado para a estimação do parâmetro de dispersão, mas existe um método mais simples, baseado na Estatística de Pearson Generalizada (Turkman & Silva, 2000). Para tal, suponha-se que se obteve uma estimativa $\hat{\boldsymbol{\beta}}$ para $\boldsymbol{\beta}$ pelo algoritmo dos mínimos quadrados ponderados.

As estimativas de máxima verosimilhança para os parâmetros μ_i são dadas por

$$\hat{\mu}_i = h(\boldsymbol{z}_i^T \hat{\boldsymbol{\beta}}),$$

onde $h(.)$ é a inversa da função de ligação.

Já foi visto que

$$var(Y_i) = b''(\theta_i)\frac{\phi}{w_i} = \frac{V(\mu_i)\phi}{w_i}, \ \ i = 1, ..., n,$$

então

$$E\left[\frac{w_i(Y_i - \mu_i)^2}{V(\mu_i)}\right] = \phi \ .$$

Pela Lei Fraca dos Grandes Números, se

$$\frac{1}{n^2}\sum_{i=1}^{n}\frac{w_i^2 E(Y_i - \mu_i)^4}{[V(\mu_i)]^2} \longrightarrow 0$$

quando n $\longrightarrow \infty$, então

$$\frac{1}{n}\sum_{i=1}^{n}\frac{w_i(Y_i - \mu_i)^2}{V(\mu_i)} \xrightarrow{P} \phi.$$

E se $V(.)$ é uma função contínua e $\hat{\mu}_i \xrightarrow{P} \mu_i$ para todo o i, então

$$\frac{1}{n-p}\sum_{i=1}^{n}\frac{w_i(Y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)} \xrightarrow{P} \phi.$$

Estima-se ϕ como

$$\hat{\phi} = \frac{1}{n-p}\sum_{i=1}^{n}\frac{w_i(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}. \tag{2.31}$$

Esta estimativa de ϕ, além de ser mais simples, produz, em geral, maior estabilidade numérica do que a estimativa de máxima verosimilhança (Turkman & Silva, 2000).

2.5 Testes de hipóteses

Os testes de hipóteses são usados em Estatística para verificar se existe evidência suficiente para suportar ou rejeitar uma hipótese sobre um determinado parâmetro. São frequentemente utilizados em MLG para testar a significância estatística dos coeficientes do modelo. Baseiam-se, essencialmente, na teoria de máxima verosimilhança. Segundo esta teoria, os testes de hipóteses são baseados em três estatísticas que são deduzidas através de distribuições assintóticas de funções adequadas das estimativas dos coeficientes do modelo.

Hipótese Simples

Sob a hipótese nula H_0 e supondo que o parâmetro de dispersão ϕ é conhecido, as três estatísticas que serão apresentadas convergem para uma variável aleatória com distribuição χ^2_p. Considere o teste de hipóteses simples, em que $\boldsymbol{\beta}^*$ é um vetor especificado para o vetor $\boldsymbol{\beta}$ de parâmetros desconhecidos.

As hipóteses são dadas por

$$H_0 : \boldsymbol{\beta} = \boldsymbol{\beta}^* \ \textit{versus} \ H_1 : \boldsymbol{\beta} \neq \boldsymbol{\beta}^*,$$

em que $\boldsymbol{\beta}^*$ é um vetor p-dimensional.

Teste de Wald

A estatística de Wald baseia-se na normalidade assintótica do estimador de máxima verosimilhança, ou seja, $\hat{\boldsymbol{\beta}} \sim N_{p+1}(\boldsymbol{\beta}, I^{-1}(\boldsymbol{\beta}))$, e nesse caso é obtida por

$$\begin{aligned} \xi_W &= (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)^T \hat{Var}^{-1}(\hat{\boldsymbol{\beta}})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \\ &= (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)^T I(\hat{\boldsymbol{\beta}})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \\ &= \frac{1}{\phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)^T (Z^T \hat{W} Z)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*). \end{aligned} \tag{2.32}$$

em que $I(\hat{\boldsymbol{\beta}}) = \hat{Var}^{-1}(\hat{\boldsymbol{\beta}})$ é a matriz de variância/covariância assintótica de $\hat{\boldsymbol{\beta}}$.

Assim, a hipótese nula é rejeitada, a um nível de significância α, se o valor observado de W for superior ao quantil de probabilidade $(1-\alpha)$ de uma χ^2_q. Esta estatística é, em geral, usada para testar hipóteses nulas sobre componentes individuais.

Particularmente, no caso em que $p = 1$, o teste de Wald é equivalente ao teste t^2 usual

$$\xi_W = \frac{(\hat{\beta} - \beta^*)^2}{\hat{Var}(\hat{\beta})}.$$

Este teste é dependente da parametrização utilizada, nomeadamente quando $\eta(\boldsymbol{\beta})$ é não linear em $\boldsymbol{\beta}$, ou seja, diferentes formas de $\eta(\boldsymbol{\beta})$ podem levar a diferentes valores de ξ_W (Gilberto, 2004).

Teste de Razão de Verosimilhança

A Estatística de Razão de Verosimilhança ou Estatística de Wilks, baseia-se na distribuição assintótica de razão do máximo das verosimilhanças sob as hipóteses H_0 e $H_0 \cup H_1$ e é definida por

$$\begin{aligned} \xi_{RV} &= -2ln\frac{max_{H_0}\ell(\boldsymbol{\beta})}{max_{H_0\cup H_1}\ell(\boldsymbol{\beta})} \\ &= -2\{\ell(\boldsymbol{\beta^*}) - \ell(\hat{\boldsymbol{\beta}})\}, \end{aligned} \tag{2.33}$$

onde $\ell(\boldsymbol{\beta^*})$ e $\ell(\hat{\boldsymbol{\beta}})$ são os valores do logaritmo da função de verosimilhança em $\hat{\beta}$ e β_0, respetivamente. Sob a hipótese nula, ξ_{RV} segue uma distribuição χ^2_p.

Segundo o teste de razão de verosimilhança, a hipótese nula é rejeitada, a um nível de significância α, se o valor observado em ξ_{RV} for superior ao quantil de probabilidade $1-\alpha$ de uma χ^2_q.

A estatística de razão de verosimilhança, em geral, é utilizada para comparar modelos encaixados, ou seja, dois modelos M_p e M_q tais que $M_p \in M_q$, com p e q parâmetros respetivamente, satisfazendo $p < q$.

Teste de Score

A Estatística de Rao, ou Estatística *Score*, baseia-se nas propriedades assintóticas da função *score* e é definida por

$$\begin{aligned} \xi_R &= U^T(\boldsymbol{\beta^*})\hat{Var}_0(\hat{\boldsymbol{\beta}})U(\boldsymbol{\beta^*}) \\ &= U^T(\boldsymbol{\beta^*})I_0^{-1}U(\boldsymbol{\beta^*}) \\ &= U^T(\boldsymbol{\beta^*})(X^T\hat{W}_0X)^{-1}U(\boldsymbol{\beta^*}), \end{aligned} \tag{2.34}$$

em que $\hat{W}_0$ é estimado sob H_0. Tal como para os outros testes, rejeita-se H_0, a um nível de significância α, se o valor observado em ξ_R for superior ao quantil de probabilidade $1-\alpha$ de uma χ^2_q.

A estatística de Rao, tal como a estatística de Wald, é usada em modelos encaixados, já que para o seu cálculo há necessidade de calcular os momentos de primeira e segunda ordem. Além disso, esta estatística apresenta uma vantagem em relação à estatística de razão de verosimilhança, esta não requer o cálculo do estimador não restrito de $\boldsymbol{\beta}$, é útil quando já se conhece o estimador restrito de $\boldsymbol{\beta}$.

2.6 Regiões de confiança

Qualquer uma das estatísticas anteriores permite construir um intervalo de confiança para $\boldsymbol{\beta}$ supondo ϕ conhecido. Uma região de confiança assintótica para $\boldsymbol{\beta}$ baseada na razão de verosimilhança é dada por

$$2\left[\ell(\hat{\boldsymbol{\beta}}; y) - \ell(\boldsymbol{\beta}; y)\right] \leq \chi^2_{q,1-\alpha}.$$

Enquanto que a região de confiança para β segundo a estatística de Wald inclui os seguintes valores

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T (X^T \hat{W} X)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq \phi \chi^2_{q,1-\alpha},$$

em que $\chi^2_{p,1-\alpha}$ é o percentil de ordem $(1-\alpha)$ de uma qui-quadrado com q graus de liberdade.

2.7 Avaliação do modelo

Nesta secção, explora-se a qualidade do modelo, uma parte essencial para a escolha de um modelo. Aborda-se o ajustamento do modelo, os critérios de seleção de modelos, técnicas de diagnóstico e a identificação de observações atípicas.

2.7.1 Qualidade de ajustamento

O objetivo central quando se seleciona uma modelo é encontrar um modelo parcimonioso, ou seja, um modelo com o menor número de covariáveis possíveis de forma a obter o máximo de informação em relação à variável resposta. Na prática encontrar esse modelo é desafiante, pois é necessário encontrar um equilíbrio entre um bom ajustamento dos dados e um modelo menos complexo.

Para esta secção é importante descrever alguns modelos que são usados neste processo de seleção.

- Modelo nulo: Modelo com um único parâmetro, assume-se que todas as variáveis Y_i têm o mesmo valor médio μ. É um modelo simples que dificilmente captura a estrutura inerente aos dados atribuindo toda a variação dos dados à componente aleatória.

- Modelo saturado: Modelo que possuiu um parâmetro para cada observação (p=n), contrariamente ao anterior este modelo atribui toda a variabilidade dos dados à componente sistemática. A hipótese deste modelo ser adequado é praticamente nula pois é um modelo não explicativo uma vez que reproduz os próprios dados.

- Modelo corrente: Modelo sob pesquisa.

Função Desvio

Quando se ajusta um modelo a um conjunto de observações, o que se pretende é substituir y por um conjunto de valores estimados $\hat{\boldsymbol{\mu}}$ para um modelo com menor número de parâmetros. Obviamente, os valores médios estimados não serão iguais aos valores de y, o cerne da questão é perceber o quanto eles diferem. O cálculo dessa discrepância é segundo (McCullagh & Nelder, 1989) feito através da Função de Desvio. É de notar que uma discrepância pequena pode ser tolerável enquanto que uma discrepância grande, não.

Define-se Função Desvio como

$$\begin{aligned} D^*(\boldsymbol{y};\hat{\boldsymbol{\mu}}) &= 2(\ell_S(\hat{\boldsymbol{\beta}}_S) - \ell_M(\hat{\boldsymbol{\beta}}_M)) \\ &= \frac{D(\boldsymbol{y};\hat{\boldsymbol{\mu}})}{\phi}, \end{aligned} \tag{2.35}$$

onde $\ell_S(\hat{\boldsymbol{\beta}}_S)$ e $\ell_M(\hat{\boldsymbol{\beta}}_M)$ são os máximos da função de verosimilhança para os modelos saturado e corrente, respetivamente.

A $D^*(\boldsymbol{y};\hat{\boldsymbol{\mu}})$ dá-se o nome de Desvio Reduzido, ao numerador $D(\boldsymbol{y};\hat{\boldsymbol{\mu}})$ dá-se o nome de Desvio.

O valor da Função Desvio é superior ou igual a zero. Quando o valor da Função Desvio é pequeno, sugere que, para um ajuste menor de parâmetros obtemos um ajuste tão bom quanto o ajuste com o modelo saturado. O desvio decresce para zero conforme se aumenta o número de parâmetros do modelo sob investigação, quando o desvio é zero significa que se está perante um modelo saturado.

Do logaritmo da função de verosimilhança, obtém-se

$$\ell_S(\hat{\boldsymbol{\beta}}_S) = \phi^{-1}\sum_{i=1}^{n}\Big[y_i\tilde{\theta}_i - b(\tilde{\theta}_i)\Big] + \sum_{i=1}^{n} c(y_i,\phi)$$

e

$$\ell_M(\hat{\boldsymbol{\beta}}_M) = \phi^{-1}\sum_{i=1}^{n}\Big[y_i\hat{\theta}_i - b(\hat{\theta}_i)\Big] + \sum_{i=1}^{n} c(y_i,\phi),$$

onde $\hat{\theta}_i = \theta_i(\hat{\boldsymbol{\mu}}_{\boldsymbol{i}})$ e $\tilde{\theta}_i = \theta_i(\tilde{\boldsymbol{\mu}}_{\boldsymbol{i}})$ são as estimativas de máxima verosimilhança de θ para os modelos saturado e corrente, respetivamente.

A função desvio pode ser denotada de outra forma por

$$D(\boldsymbol{y};\hat{\boldsymbol{\mu}}) = 2w_i\sum_{i=1}^{n} y_i(\tilde{\theta}_i - \hat{\theta}_i) + b(\hat{\theta}_i) - b(\tilde{\theta}_i). \tag{2.36}$$

A distribuição do desvio não é conhecida, o que faz com que a análise do desvio seja usada apenas como um guia quando se estuda um modelo. Na prática, o usual é comparar os valores calculados da função desvio com o quantil de probabilidade $(1-\alpha)$ de uma distribuição $\chi^2_{(n-p)}$. Se o valor calculado for superior ao quantil, considera-se que o modelo não é adequado (Turkman & Silva, 2000).

Exemplo 2.7.1. *Distribuição Normal:* $N(\mu, \sigma^2)$

Para o caso do modelo normal a função log-verosimilhança é

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{1}{\sigma^2}(y_i\mu_i - \frac{\mu_i^2}{2}) + c(y_i, \phi).$$

Assim, para o modelo saturado

$$\ell_S(\hat{\boldsymbol{\beta}}_S) = \sum_{i=1}^{n} \frac{1}{\sigma^2}(y_i^2 - \frac{y_i^2}{2}) + c(y_i, \phi) = \sum_{i=1}^{n} \frac{y_i^2}{2\sigma^2} + c(y_i, \phi)$$

e para o modelo corrente

$$\ell_M(\hat{\boldsymbol{\beta}}_M) = \sum_{i=1}^{n} \frac{1}{\sigma^2}(y_i\hat{\mu}_i - \frac{\hat{\mu_i}^2}{2}) + c(y_i, \phi) = \sum_{i=1}^{n} \frac{1}{2\sigma^2}(2y_i\hat{\mu}_i - \hat{\mu_i}^2) + c(y_i, \phi)$$

Consequentemente, o desvio reduzido da distribuição Normal é definida por

$$D^*(\boldsymbol{y}; \hat{\boldsymbol{\mu}}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \hat{\mu_i})^2.$$

Exemplo 2.7.2. *Distribuição Poisson:* $P(\mu)$

Para o caso do modelo Poisson com função de ligação canónica, tem-se que a função log-verosimilhança é

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} y_i ln\mu_i - \mu_i - ln(y_i!),$$

sendo

$$\ell_S(\hat{\boldsymbol{\beta}}_S) = \sum_{i=1}^{n} y_i lny_i - y_i - ln(y_i!),$$

e

$$\ell_M(\hat{\boldsymbol{\beta}}_M) = \sum_{i=1}^{n} y_i ln\hat{\mu}_i - \hat{\mu}_i - ln(y_i!).$$

Consequentemente o desvio reduzido é

$$D^*(\boldsymbol{y}; \hat{\boldsymbol{\mu}}) = 2 \sum_{i=1}^{n} \left[y_i ln\left(\frac{y_i}{\hat{\mu}_i}\right) - (y_i - \hat{\mu}_i) \right].$$

Pode-se resumir outros exemplos de desvios na Tabela 2.3.

Tabela 2.3: Funções desvio de algumas distribuições

Distribuição	Desvio
Normal	$\sum_{i_1}^{n}(y_i - \hat{\mu_i})^2$
Poisson	$2\sum_{i=1}^{n}\left[y_i ln\left(\frac{y_i}{\hat{\mu_i}}\right) - (y_i - \hat{\mu_i})\right]$
Binomial	$2\left[\sum_{i=1}^{n} m_i y_i ln\frac{y_i}{\hat{\mu_i}} + \sum_{i=1}^{n} m_i(1-y_i) ln\frac{1-y_i}{1-\hat{\mu_i}}\right]$
Gama	$2\sum_{i=1}^{n}\left[-ln\frac{y_i}{\hat{\mu_i}} + \frac{y_i - \hat{\mu_i}}{\hat{\mu_i}}\right]$
Normal Inversa	$\sum_{i=1}^{n}\frac{(y_i-\hat{\mu_i})^2}{y_i\hat{\mu_i}^2}$

Estatística de Pearson Generalizada

Outra medida usada para verificar a qualidade de ajustamento de um modelo é a estatística de Pearson Generalizada, que é dada por,

$$X^2 = \sum_{i=1}^{n}\frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}, \tag{2.37}$$

onde $V(\hat{\mu}_i)$ é a função de variância estimada sob o modelo que está a ser estudado.

Para a distribuição Normal, a estatística X^2 coincide com a soma dos quadrados dos resíduos, e para os modelos Poisson e Binomial coincide com a estatística original de Pearson (Turkman & Silva, 2000).

Tal como a Função Desvio, na prática compara-se o resultado com o quantil de probabilidade $(1-\alpha)$ de uma distribuição $\chi^2_{(n-p)}$. Esta aproximação pelo χ^2 da distribuição de X^2 pode, para certos modelo, não ser a adequada.

Em suma, estes dois métodos são apropriados para verificar a qualidade de ajustamento de um modelo, contudo, a Função Desvio, devido à sua propriedade aditiva, é preferida em relação à estatística de Pearson Generalizada. No entanto, esta última tem a vantagem de ter uma interpretação mais direta.

2.7.2 Seleção de Modelos

Nos Modelos Lineares Generalizados (MLG) é recorrente trabalhar-se com um grande número de covariáveis que podem ser importantes para explicar a variabilidade do modelo. É ainda necessário investigar a influência de possíveis interações entre as covariáveis. Isto implica que é essencial trabalhar um grande número de modelos de forma a encontrar um que se ajuste melhor aos dados. Duas estatísticas essenciais para proceder a esta seleção serão abordados a seguir.

Critério de informação de Akaike (AIC)

O critério de informação de Akaike é baseado na função de verosimilhança (Akaike, 1974) e é dado por

$$AIC_k = D^*(y;\hat{\mu}) - 2\ell(\hat{\beta};y) + 2k, \tag{2.38}$$

em que k é o número de parâmetros. O objetivo passa por encontrar o modelo para qual o valor de AIC seja menor.

Critério de informação de Bayes (BIC)

A estatística de Schwartz consiste em maximizar $\ell(\beta)$ enquanto se minimiza o número de coeficientes de regressão (Schwarz, 1978). Essa estatística é conhecida por critério de informação de Bayes e é dado por

$$BIC_k = D^*(y;\hat{\mu}) - 2\ell(\hat{\mu}_i;y) + kln(n), \tag{2.39}$$

onde k é o número de parâmetros. Tal como o AIC, o menor valor de BIC pode indicar um melhor ajuste do modelo.

Ambos AIC e BIC fornecem abordagens bem fundamentadas e independentes para a comparação de modelos, embora com diferentes motivações e objetivos parcialmente diferentes.

O AIC é assintóticamente ótimo, baseado na ideia de minimizar o erro esperado de previsão. Já o BIC tem a propriedade de consistência, é baseado na ideia de minimizar a perda de informação da amostra em relação ao modelo verdadeiro. Portanto, a escolha entre os dois critérios depende do contexto da análise (Yang, 2005) .

Em geral, o AIC é mais adequado para amostras menores e para seleção de modelos não lineares, enquanto o BIC é mais adequado para amostras maiores e para seleção do modelo mais simples e parcimonioso (Bishop, 2006).

Contudo, é fundamental destacar que tanto o AIC quanto o BIC não fornecem testes de hipóteses para comparação de modelos e os seus valores não trazem qualquer informação sobre a qualidade do modelo por si só. Portanto, se todos os modelos considerados não se ajustam bem aos dados, essas medidas não fornecerão nenhuma informação útil nesse sentido.

Métodos sequências para a seleção de variáveis

Na seleção das variáveis utilizam-se procedimentos sequenciais, tais como: o método *backward*, o método *forward* e o método *stepwise*.

Método backward

O método *backward* inicia com o modelo completo que inclui todas as variáveis. Em seguida, remove

iterativamente a variável menos significativa. A decisão de eliminar uma variável é baseada na análise do desvio ou na medida AIC.

Método forward

Este método considera que o modelo inicial não inclui nenhuma variável (modelo nulo). A abordagem consiste em adicionar uma variável em cada etapa. A primeira a incluir é aquela que apresenta maior correlação com a variável resposta, com base na análise do desvio ou na medida AIC.

Método stepwise

O método *stepwise* é uma combinação dos métodos *forward* e *backward*. Do mesmo modo com base no desvio ou na medida AIC seleciona quais variáveis devem ser incluídas ou removidas do modelo. Começa com um modelo que inclui todas as variáveis independentes e, em seguida, adiciona ou remove as variáveis.

2.7.3 Técnicas de diagnóstico

A análise de resíduos permite avaliar a qualidade de ajustamento de um modelo relativamente à escolha da distribuição da variável resposta, da função de ligação e de termos do preditor linear. Permite também identificar observações mal ajustadas pelo modelo.

Os resíduos expressam a discrepância entre o valor observado ,y_i, e o valor ajustado pelo modelo, $\hat{\mu_i}$.

Na análise de diagnóstico é importante conhecer a matriz de projeção, é dada por

$$\begin{aligned} H &= W^{\frac{1}{2}} Z (Z^T W Z)^{-1} Z^T W^{\frac{1}{2}} \\ &= W^{\frac{1}{2}} Z I^{-1}(\boldsymbol{\beta}) Z^T W^{\frac{1}{2}}. \end{aligned}$$

A matriz H é simétrica e idempotente, tem traço(H)=característica (H)=$\sum_{i=1}^{n} h_{ii} = p + 1$ e os elementos da diagonal (h_{ii}) estão compreendidos entre 0 e 1. Esta matriz depende das variáveis independentes, da função de ligação e da função de variância.

Para uma análise adequada dos resíduos é conveniente que estes sejam padronizados e reduzidos, ou seja, que tenham variância constante e unitária. De seguida apresentam-se os tipos de resíduos mais comuns em Modelos Lineares Generalizados (MLG).

Resíduo de Pearson

O resíduo de Pearson é expresso por

$$R_i^P = \frac{y_i - \hat{\mu_i}}{\sqrt{\hat{var}(Y_i)}} = \frac{(y_i - \hat{\mu_i}) w_i}{\sqrt{\hat{\phi} V(\hat{\mu_i})}}.$$

Este resíduo corresponde à contribuição de cada observação para o cálculo da estatística de Pearson

Generalizada.

Resíduo de Pearson Padronizado

Atendendo a que, assintoticamente, $var(Y_i - \hat{\mu}_i) \approx var(Y_i)(1 - h_{ii})$, o resíduo Pearson padronizado é

$$R^{*P}_i = \frac{(y_i - \hat{\mu}_i)w_i}{\sqrt{\hat{\phi}V(\hat{\mu}_i)(1 - h_{ii})}}.$$

Este resíduo tem a desvantagem que a sua distribuição é, geralmente, bastante assimétrica para modelos não normais (Turkman & Silva, 2000).

Resíduo de Anscombe

Anscombe sugeriu fazer uma transformação adequada $A(y_i)$ da observação y_i, com o intuito de obter resíduos com uma distribuição o mais próxima possível da Normal, definindo resíduo como

$$R^A_i = \frac{A(y_i) - E[A(Y_i)]}{\sqrt{var[A(Y_i)]}}.$$

Através das aproximações $E[A(Y_i)] \approx A(\mu_i)$ e $var[A(y_i)] \approx [A'(\mu_i)]^2 var(Y_i)$ obtêm-se os resíduos de Anscombe

$$R^A_i = \frac{A(y_i) - A(\hat{\mu}_i)}{\sqrt{\hat{var(Y_i)}}A'(\hat{\mu}_i)}.$$

A transformação a considerar nos MLG foi descoberta por (Barndorff-Nielsen, 1978) e é a seguinte

$$A(x) = \int \frac{1}{V^{\frac{1}{3}}(x)} dx,$$

onde $V(x)$ é a função de variância.

Desvio Residual

Este resíduo baseia-se na função de desvio, usando a contribuição da i-ésima observação para a função de desvio

$$D(\boldsymbol{y}; \hat{\boldsymbol{\mu}}) = 2w_i \sum_{i=1}^{n} y_i(\tilde{\theta}_i - \hat{\theta}_i) + b(\hat{\theta}_i) - b(\tilde{\theta}_i) = \sum_{i=1}^{n} d_i,$$

para dar uma nova definição de resíduo.

Assim o desvio residual é dado por

$$R^D_i = \delta_i \sqrt{d_i},$$

onde δ_i corresponde ao sinal de $(y_i - \hat{\mu}_i)$.

Desvio residual padronizado

O desvio residual padronizado obtém-se através da divisão do desvio residual por $\sqrt{\hat{\phi}(1-h_{ii})}$, ou seja,

$$R^{*D}_i = \frac{R^D_i}{\sqrt{\hat{\phi}(1-h_{ii})}}$$

Resíduo quantílico

Os resíduos quantílicos apresentam distribuição Normal, independente da distribuição da variável resposta e da sua dispersão.

Seja $F(y;\hat{\mu},\phi)$ a função de distribuição cumulativa de uma variável aleatória Y. O resíduo quantílico é definido por

$$r_Q = \Phi^{-1}(u)$$

em que $\Phi^{-1}(.)$ representa a função quantílica da distribuição normal padrão e u é uma variável aleatória com distribuição uniforme no intervalo $(a,b]$, sendo $a = \lim_{\epsilon\uparrow 0} F(y+\epsilon;\hat{\mu},\phi)$ e $b = F(y;\hat{\mu},\phi)$ (a notação $\lim_{\epsilon\uparrow 0}$ significa o limite quando ϵ se aproxima de 0 de menor para maior, de modo que ϵ é sempre negativo) (Dunn & Smyth, 2017).

Exemplo 2.7.3. *Distribuição Normal:* $Y \sim N(\mu,\sigma^2)$

Na distribuição Normal, o resíduo de Pearson é dado por

$$R^P_i = y_i - \hat{\mu}_i.$$

Como $V(x) = 1$*, tem-se que* $A(x) = \int V^{\frac{-1}{3}}(x)dx = x$ *então*

$$R^A_i = y_i - \hat{\mu}_i.$$

Por outro lado, como $d_i = (y_i - \hat{\mu}_i)^2$*, tem-se que o desvio residual é*

$$R^D_i = y_i - \hat{\mu}_i.$$

Em suma, verifica-se que os três tipos de resíduos na distribuição Normal coincidem.

Exemplo 2.7.4. *Distribuição Poisson:* $Y \sim P(\mu)$

O resíduo de Pearson para a distribuição Poisson é dado por

$$R_i^P = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i}}.$$

Sabe-se que $V(x) = x$ *logo* $\int V^{\frac{-1}{3}}(x)dx = \frac{3}{2}x^{\frac{2}{3}}$ *e portanto os resíduos de Anscombe são dados por*

$$R_i^A = \frac{3(y_i^{\frac{2}{3}} - \hat{\mu_i}^{\frac{2}{3}})}{2\hat{\mu_i}^{\frac{1}{6}}}.$$

Consultando a tabela dos desvios obtém-se o desvio residual

$$R_i^D = \delta_i 2^{\frac{1}{2}}(y_i ln\frac{y_i}{\hat{\mu}_i} - y_i + \hat{\mu}_i)^{\frac{1}{2}}.$$

Resumem-se os três tipos de resíduos para as diferentes distribuições na Tabela 2.4.

Tabela 2.4: Resíduos de algumas distribuições

Distribuição	R_i^P	R_i^A	R_i^D
Normal	$y_i - \hat{\mu}_i$	$y_i - \hat{\mu}_i$	$y_i - \hat{\mu}_i$
Poisson	$\frac{y_i - \hat{\mu}_i}{\hat{\mu_i}^{\frac{1}{2}}}$	$\frac{3(y_i^{\frac{2}{3}} - \hat{\mu_i}^{\frac{2}{3}})}{2\hat{\mu_i}^{\frac{1}{6}}}$	$\delta_i 2^{\frac{1}{2}}(y_i ln\frac{y_i}{\hat{\mu_i}} - y_i + \hat{\mu}_i)^{\frac{1}{2}}$
Binomial	$\frac{m_i^{\frac{1}{2}}(y_i - \hat{\mu_i})}{[\hat{\mu_i}(1-\hat{\mu_i})]^{\frac{1}{2}}}$	$\frac{m_i^{\frac{1}{2}}[A(y_i) - A(\hat{\mu_i})]}{[\hat{\mu_i}(1-\hat{\mu_i})]^{\frac{1}{6}}}$	$\delta_i[2m_i(ln\frac{y_i}{\hat{\mu_i}} + (1-y_i)ln\frac{1-y_i}{1-\hat{\mu_i}}]^{\frac{1}{2}}$
Gama	$\frac{y_i - \hat{\mu_i}}{\hat{\mu_i}}$	$\frac{3(y_i^{\frac{1}{3}} - \hat{\mu_i}^{\frac{1}{3}})}{\hat{\mu_i}^{\frac{1}{3}}}$	$\delta_i\left[2ln\left(\frac{\hat{\mu_i}}{y_i} + \frac{y_i - \hat{\mu_i}}{\hat{\mu_i}}\right)\right]^{\frac{1}{2}}$
Normal Inversa	$\frac{y_i - \hat{\mu_i}}{\hat{\mu_i}^{\frac{3}{2}}}$	$\hat{\mu_i}^{\frac{-1}{2}} ln\frac{y_i}{\hat{\mu_i}}$	$\frac{y_i - \hat{\mu_i}}{y_i^{\frac{1}{2}}\hat{\mu_i}}$

$\delta_i = sinal(y_i - \hat{\mu}_i)$ e $A(x) = \int [x(1-x)]^{-\frac{1}{3}}\, dx$.

2.7.4 Observações atípicas

Quando um modelo é ajustado, é normal que certas observações não sigam o padrão das outras observações, estas são chamadas de observações atípicos.

Observações alavanca

São observações que estão afastadas, no espaço das variáveis explicativas, da maioria das outras observações. Têm como objetivo medir a influência de y_i sobre o próprio valor ajustado $\hat{y}_i$.

Alguns autores sugerem que essa medida é dada através da matriz de projeção H, pelo valor de h_{ii}, ou seja, o i-ésimo elemento da diagonal principal. Como foi visto que o *traço*$(H) = p + 1$ e considerando-se que, em

média, cada valor h_{ii} está próximo de $(p+1)/n$, então um ponto alavanca é dado por

$$h_{ii} > \frac{2p+1}{n}.$$

Uma forma informal de analisar observações alavancas é através da análise gráfica do h_{ii} contra os valores ajustados, considerando observações alavanca os valores ajustados que sejam superiores a $(2p+1)/n$.

Observações influentes

Observações que, individualmente ou coletivamente, influenciam o modelo estimado. A eliminação destas observações na estimação de modelos leva a grandes mudanças nas estimativas dos coeficientes.

A avaliação da existência de observações influentes pode ser feita através de diferentes métodos.

Distância de Cook

A distância de Cook mede a distância entre os valores ajustados obtidos através dos dados completos e os valores obtidos pela exclusão da i-ésima observação

$$C_i = \frac{\sum_{i=1}^{n}(\hat{y}_j - \hat{y}_{j(i)})^2}{\sigma^2(p+1)}, \quad com \;\; i = 1, 2, 3, ..., n$$

Equivalentemente,

$$C_i = \frac{R_i^2}{p+1}\frac{h_{ii}}{1-h_{ii}}, \quad com \;\; i = 1, 2, 3, ..., n \tag{2.40}$$

Alguns autores sugerem que observações podem ser declaradas influentes se $C_i > F(0.5, p+1, n-p-1)$, onde F é a distribuição F de Fisher-Snedecor. Na prática costuma-se considerar quando $C_i > 1$ (Chatterjee & Hadi, 2012).

Graficamente analisa-se a presença de pontos influentes através de um gráfico C_i *versus* números de observações.

Observações outliers

Consideram-se *outliers* as observações que têm resíduos de valor elevado quando comparados com as outras observações. Muitas vezes designam-se por valores atípicos, discordantes e aberrantes. Estas observações devem ser identificadas pois podem afetar significativamente as análises estatísticas e os resultados da estimação de modelos.

O gráfico *Q-Q plot* costuma ser usado para a análise deste tipo de observações. Em geral considera-se uma observação *outlier* quando $R_i > 3$. Depois de detetada a observação deve ser feito um estudo dos dados sem a

mesma e analisar o impacto que causa.

2.8 Exemplos de Modelos Lineares Generalizados

No contexto deste projeto, como já mencionado, serão explorados dois exemplos de modelos lineares generalizados: o modelo de regressão de Poisson e o modelo de regressão Logística. Nesta secção, apresenta-se cada um dos modelos, descreve-se como se estimam os coeficientes e introduzem-se medidas de ajustamento direcionadas para cada modelo.

2.8.1 Modelos de Regressão Poisson

Os modelos de regressão Poisson são usados quando se pretende entender a relação entre variáveis independentes e a ocorrências de eventos raros, especialmente quando se trabalha com dados de contagem. Por exemplo, podem aplicar-se estes modelos quando se quer analisar o número de acidentes de trânsito para uma determinada área e período de tempo.

A regressão de Poisson é considerada uma metodologia fundamental na modelação de dados de contagens devido à sua capacidade de lidar com a natureza discreta desses dados, uma vez que assume valores inteiros não negativos para o valor esperado da variável resposta. Em contraste, a distribuição Normal é contínua e permite valores não inteiros, o que a torna inadequada para dados de contagem. Tentar aplicar a distribuição Normal a esses dados poderia levar a interpretações erradas, visto que poderia gerar valores não inteiros e/ou não negativos (Winter & Bürkner, 2021). Contudo este tipo de modelo impõe que o valor médio e a variância sejam iguais, levando por vezes a problemas de sobredispersão. Para resolver esse problema em situações específicas, é possível recorrer ao modelo de regressão Binomial Negativa (Berk & MacDonald, 2008).

Seja $Y_1, ..., Y_n$ uma amostra aleatória que representa o número de ocorrências de um determinado evento. Dado um vetor de variáveis explicativas $X = (X_1, X_2, ..., X_p)$ e uma observação do indivíduo i, $x_i = (x_{i1}, ..., x_{ip})$,

$$Y|X = x_i \sim P(\mu(x_i))$$

Dado que o objetivo é modelar o valor médio de $Y|X = x_i$, a opção mais comum é utilizar uma transformação logarítmica para relacionar μ_i com a combinação linear das variáveis explicativas. A função de ligação canónica é a função logaritmo e a equação que define um modelo de regressão Poisson é a seguinte:

$$ln(\mu(x_i)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p \tag{2.41}$$

Equivalentemente

$$\mu(x_i) = e^{\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_p x_p} \tag{2.42}$$

Os coeficientes de regressão β_j,$j = 0, ..., p$ representam a variação esperada no logaritmo do valor médio, por unidade de variação na variável X_i.

Estimação dos coeficientes

A estimação dos coeficientes do modelo é feita através do método de Máxima Verosimilhança como explicado na Secção 2.4.

A função de massa de probabilidade de Poisson é

$$f(y_i|\mu_i) = \frac{\mathrm{e}^{-\mu_i}\mu_i^{y_i}}{y_i!},$$

logo a função de verosimilhança obtém-se como o produto destes termo, sendo

$$L = \prod_{i=1}^{n} \frac{\mathrm{e}^{-\mu_i}\mu_i^{y_i}}{y_i!}.$$

Aplicando-se o logaritmo, obtém-se a função de log-verosimilhança

$$\ell = \sum_{i=1}^{n} (y_i \ln(\mu_i) - \mu_i - \ln(y_i!)).$$

Substituindo por as equações (2.41) e (2.42)

$$\ell = \sum_{i=1}^{n} (y_i(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p) - e^{\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_p x_p} - \ln(y_i!)).$$

As estimativas dos parâmetros encontram-se maximizando esta função, recorrendo a métodos iterativos.

Medida de ajustamento

Como medida de ajustamento para o estudo de modelos de Regressão Poisson recorreu-se à estatística obtida por

$$SQR = \frac{\textit{soma dos quadrados dos resíduos de Pearson}}{\textit{n-p-1}}.$$

2.8.2 Modelo de regressão Logística

No dia à dia, muitas situações geram dados categóricos. Um exemplo comum é quando se quer prever se um indivíduo é propenso a ter uma condição específica, como por exemplo, cancro do pulmão. A análise destes dados faz-se com recursos a modelos de regressão Logística, que permitem prever a probabilidade de ocorrência de um evento binário, como "sim"ou "não", "positivo"ou "negativo", com base em variáveis explicativas (Agresti, 2013).

O que distingue um modelo de regressão Logística de um modelo de regressão linear é a variável resposta, que é binária ou dicotómica (Hosmer, Lemeshow, & Sturdivant, 2013).

Considera-se a variável resposta Y, seguindo uma distribuição de Bernoulli. Dada uma amostra $y_1, y_2, ..., y_n$ dessa distribuição, com valores que assumem apenas dois valores, atribuí-se $y_i = 1$ ao evento de interesse, denominado "sucesso", e $y_i = 0$ ao evento complementar, denominado "insucesso".

$$Y|X = x_i \sim B(1, p(x_i)),$$

onde $p(x_i) = p_i = P(Y = 1|X = x_i)$ é a probabilidade de sucesso para Y tendo em conta x_i.

No contexto da regressão Logística, ao tentarmos modelar a média de $Y|X = x_i$, não é adequado utilizar uma modelagem linear direta, pois $p(x_i)$ varia apenas entre 0 e 1, enquanto a combinação linear das variáveis explicativas pode variar de $-\infty$ a $+\infty$. Portanto, uma transformação é necessária, e a transformação mais comum é a transformação *logit*.

$$logit(p_i) = ln\left(\frac{p_i}{1-p_i}\right)$$

A função de ligação canónica é a função *logit* e a equação que define um modelo de regressão logística é

$$ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p, \tag{2.43}$$

onde p_i é a probabilidade da ocorrência do evento e β_p são os coeficientes de regressão a serem estimados.

Estimação dos coeficientes

Para estimar os coeficientes de regressão, emprega-se o método de Máxima Verosimilhança, descrito na Secção 2.4. A função de máxima verosimilhança para dados binários é obtida como o produto dos termos da função massa de probabilidade,

$$f(y_i|p_i) = p_i^{y_i}(1-p_i)^{1-y_i}, \quad y_i = 0, 1 \ \textit{e} \ i = 1, ..., n,$$

onde p_i é o parâmetro desconhecido.

Obtém-se a função de máxima verosimilhança, como

$$L(\beta) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$$

Usando o logaritmo da função de verosimilhança, dado por

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{n}(y_i ln(p_i) + (1-y_i)ln(1-p_i)) \\ &= \sum_{i=1}^{n} y_i ln\left(\frac{p_i}{1-p_i}\right) + ln(1-p_i).\end{aligned}$$

Através da Equação 2.43 e de $p_i = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_p x_p}}$, tem-se

$$\ell(\beta) = \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p) - ln(1 + e^{\beta_0+\beta_1 x_1+\beta_2 x_2+....+\beta_p x_p})$$

A solução do valor de β que maximiza a função é obtido através da derivação de $\ell(\beta)$ em relação a β e, por vezes, para encontrar a solução é necessário recorrer a métodos iterativos.

Interpretação dos parâmetros do modelo

O *odds* descreve a relação entre a probabilidade de sucesso e a probabilidade de insucesso de um acontecimento, é uma forma de expressar as chances de que um evento ocorra em relação à sua não ocorrência. Calcula-se pelo quociente entre a probabilidade de sucesso e a probabilidade de insucesso de um certo acontecimento.

O *odds ratio*(OR) é uma medida que compara as *odds* de sucesso para dois indivíduos diferentes ou para dois grupos de indivíduos.

Um OR de 1 indica que não há diferença na probabilidade de sucesso entre os dois grupos. Um OR maior que 1 sugere que o primeiro grupo tem uma maior chance de sucesso em comparação com o segundo grupo, enquanto um OR menor que 1 sugere que o primeiro grupo tem uma menor chance de sucesso.

Sejam $X_1, ..., X_p$ variáveis explicativas do modelo sem interações.

Quando X_j, de $j = 1, ..., p$ toma valores contínuos, tem-se:

$$x = (x_1, ..., x_j, ..., x_p) \ \ e \ \ (x + c) = (x_1, ..., x_j + c, ..., x_p),$$

logo, recorrendo à definição de *odds ratio*:

$$\begin{aligned} log(OR(x+c,x)) &= ln\left(\frac{\textit{odds}\, p(x+c)}{\textit{odds}\, p(x)}\right) \\ &= ln\left(\frac{\frac{p(x+c)}{1-p(x+c)}}{\frac{p(x)}{1-p(x)}}\right) \\ &= logit\ p(x+c) - logit\ p(x) \\ &= (x_j+c)\beta_j - x_j\beta_j \\ &= c\beta_j. \end{aligned}$$

Assim, $c\beta_j$, representa a alteração provocada no *logit* da probabilidade por uma variação de c unidade na variável X_j, mantendo as restantes variáveis constantes.

$OR(x+c,x) = e^{c\beta_j}$, o que significa que, mantendo as restantes variáveis constantes e aumentando c unidades a variável X_j, o *odds* para o sucesso varia $e^{c\beta_j}$ vezes.

Quando X_j representa uma variável dicotómica, tem-se:

$$\begin{aligned} ln(OR(x=1,x=0)) &= ln\left(\frac{\textit{odds}\, p(1)}{\textit{odds}\, p(0)}\right) \\ &= ln\left(\frac{\frac{p(1)}{1-p(1)}}{\frac{p(0)}{1-p(0)}}\right) \\ &= logit\ p(1) - logit\ p(0) \\ &= \beta_j. \end{aligned}$$

Logo $OR(x=1,x=0) = e^{\beta_j}$ significa que é provável e^{β_j} vezes que o sucesso ocorra nos indivíduos $X_j = 1$ do que nos indivíduos com $X_j = 0$, mantendo as restantes variáveis explicativas constantes.

Quando X_j é uma variável categórica, contendo k categorias, a variável é representada por $k-1$ variáveis indicatrizes enquanto que uma das categorias é considerada de referência. As *odds ratio* são estimadas para cada uma das $k-1$ classes em relação à variável de referência.

Medida de ajustamento

Por vezes, para analisar o desempenho da previsão de um modelo, recorre-se a uma matriz de confusão. Esta fornece uma visão clara e eficiente do desempenho do modelo em classificar uma observação e não é mais do que uma tabela de dupla entrada que compara os valores reais dos dados com os valores previstos pelo modelo (Yun, 2021).

Para tal, considera-se valor positivo quando o evento ocorre e valor negativo quando o evento não ocorre.

Tabela 2.5: Matriz de Confusão

		Valor Real	
		Positivo	Negativo
Valor Previsto	Positivo	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	Negativo	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Quando um dado é classificado como:

- **Verdadeiro positivo (VP)** significa que os verdadeiros valores e os valores previstos são ambos valores positivo
- **Falso positivo (FP)** significa que o valor real é negativo, mas o valor previsto é positivo.
- **Falso negativo (FN)** significa que o modelo previu erradamente o valor real positivo.
- **Verdadeiro negativo (VN)** significa que o modelo previu corretamente o valor negativo.

A classificação dos erros através da matriz de confusão permite calcular algumas medidas de ajustamentos, tais como a Acurácia, a Sensibilidade a Precisão e a Especificidade (Powers, 2011).

A Acurácia representa a proporção de previsões corretas que o modelo consegue prever em relação ao total de previsões calculadas. É dada por:

$$\textit{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN}.$$

Note-se que, quando se trabalha com dados desiquilibrados em que o número de observações nas categorias da variável resposta é muito desproporcional, a Acurácia não é a medida mais adequada. De forma a controlar a situação descrita neste estudo, procurou-se sempre trabalhar com um conjunto de dados balanceado.

A Sensibilidade corresponde à proporção de verdadeiros positivos que foram corretamente identificados pelo modelo em relação ao total de casos positivos na amostra. Determina-se por:

$$\textit{Sensibilidade} = \frac{VP}{VP + FN}.$$

Quanto mais próximo de um 1 for o valor da sensibilidade melhor é a capacidade de o modelo detetar verdadeiros positivos.

A Precisão corresponde à proporção de verdadeiros positivos em relação a todos os casos positivos previstos pelo modelo. Calcula-se por:

$$\textit{Precisão} = \frac{VP}{VP + FP}.$$

Valores próximos de 1 significam que o modelo prevê corretamente casos positivos.

A Especificidade calcula a proporção de casos negativos que foram identificados corretamente pelo modelo em relação a todos os casos negativos detetados pelo modelo. É dada por:

$$\textit{Especificidade} = \frac{VN}{VN + FN}.$$

Quanto mais próximo for de 1 o valor da Especificidade significa que o modelo tem maior capacidade de identificar corretamente casos negativos.

3 Estudo de simulação

Um estudo de simulação assume um papel crucial como ferramenta estatística, viabilizando a geração de dados simulados por meios computacionais. Esses dados simulados são concebidos para replicar cenários reais, permitindo a avaliação de diversas hipóteses e a obtenção de resultados empíricos acerca do desempenho de métodos estatísticos em diferentes cenários. Este avalia e compara métodos para estimar um ou mais parâmetros populacionais, as designadas Estatísticas. Neste estudo, o foco é a aplicação da simulação como meio de examinar o desempenho dos Modelos Lineares Generalizados (MLG) em diferentes linguagens de programação.

Neste contexto, esta secção apresenta uma descrição detalhada das linguagens a serem usadas, bem como o processo de geração de bases de dados. Essas bases de dados serão cruciais para a comparação das diferentes linguagens em análise.

Além disso, com o propósito de complementar e aprimorar esta pesquisa, selecionou-se a linguagem de programação R para avaliar o desempenho dos MLG em amostras de diversas dimensões, ao mesmo tempo em que se examina sua capacidade de ajustamento. No ambiente R, os objetivos desta análise dividem-se em duas partes: a estimativa de parâmetros e a previsão de modelos. Esses objetivos serão alcançados através da aplicação de modelos de regressão Logística e modelos de regressão de Poisson.

3.1 Linguagens de programação

Para este estudo usam-se três linguagens de programação, R, Stata e Python, a escolha das mesmas deve-se ao facto de estas linguagens serem muitos usadas pelos investigadores que recorrem ao BPLIM.

R

R é uma linguagem de programação de análise de dados amplamente utilizada em Ciência de Dados e Estatística. Possui a grande vantagem de código aberto e de livre acesso, ou seja, os utilizadores podem aceder ao código R de forma gratuita, analisar o código e modificá-lo sempre que necessário de acordo com as suas necessidades. Além disso, existe uma grande comunidade de utilizadores de R que contribuem com novas funções e *packages*, tornando o R uma ferramenta poderosa e flexível para análise de dados.

Em geral, o R é uma ferramenta bastante eficaz a armazenar e manipular dados, realizar cálculos e testes estatísticos, fazer análises exploratórias e produzir gráficos.

No âmbito deste estudo será usado o *RStudio*. O *RStudio* é um ambiente de desenvolvimento integrado (IDE) para a linguagem de programação R, que oferece uma interface amigável e funcional. Tem como objetivo ajudar os utilizadores a escrever e executar código R de maneira mais eficiente.

O *RStudio* apresenta:

- Uma consola para executar código R;

- Um editor de código, que permite escrever e editar *scripts* R;
- Um painel de histórico para rever comandos realizados anteriormente;
- Um navegador de ambiente de trabalho para examinar os objetos criados na sessão atual do R;
- Ferramentas gráficas para visualizar dados;
- Um gerenciador de *packages* para instalar e atualizar *packages* R;
- Outras funcionalidades.

A Figura 1 apresenta o ambiente inicial do *RStudio*.

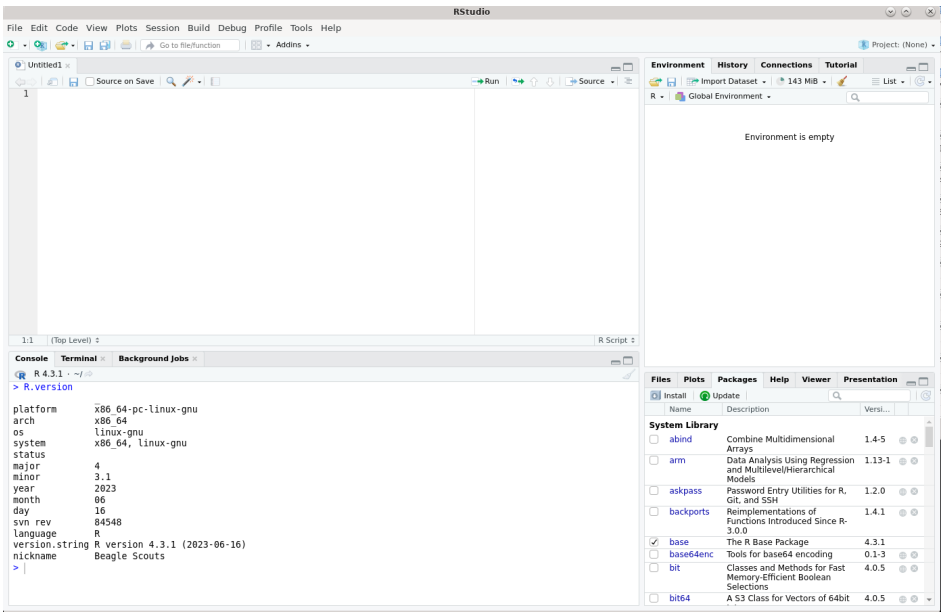

Figura 1: Ecrã inicial do RStudio

O *RStudio* está disponível como uma versão gratuita de código aberto e ainda como uma versão paga comercial. Este está disponível para diferentes sistemas operativos: *Windows, Mac e Linux*.

O R apresenta uma extensa lista de funções que podem ser usadas no estudo de MLG (Dunn & Smyth, 2017).

Python

Python é uma linguagem de programação muito utilizada em análise de dados com múltiplas bibliotecas disponíveis para vários sistemas operacionais. É uma linguagem de programação de uso geral, ou seja, pode ser usada para uma variedade de tarefas, além da análise de dados, como por exemplo desenvolvimento *web*, desenvolvimento de jogos e inteligência artificial. Estas características fazem com que as empresas recorram cada vez mais ao uso de Python.

Além disso, as bibliotecas do Python são de código aberto, ativamente desenvolvidas e mantidas por uma grande comunidade de investigadores, o que permite uma vasta gama de possibilidades de visualização e manipulação de dados.

As seguintes bibliotecas são fortemente analíticas e oferecem uma completa caixa de ferramentas de Ciência de Dados, constituída por funções altamente otimizadas para desempenho e uso de memória.

- *Pandas*: biblioteca voltada para a análise de dados que oferece estruturas de dados de alto desempenho e fáceis de usar.
- *Seaborn*: biblioteca que disponibiliza uma interface de alto nível para desenhar gráficos estatísticos e informativos.
- *Matplotlib*: biblioteca de gráficos 2D que permite criar visualizações de dados personalizadas e flexíveis.
- *Statsmodels*: biblioteca de modelação estatística, fornece diversos modelos estatísticos para análise de dados.

Contrariamente ao R, o Python não possui um ambiente de desenvolvimento integrado (IDE) oficial específico. No entanto, oferece várias opções populares de IDEs, como o *PyCharm, Anaconda, Visual Studio Code, Jupyter Notebook* e muitos outros, que permitem o desenvolvimento de código em Python. Neste trabalho recorreu-se ao *Jupyter*.

A Figura 2 apresenta o ecrã inicial do ambiente *Jupyter* usando a linguagem Python.

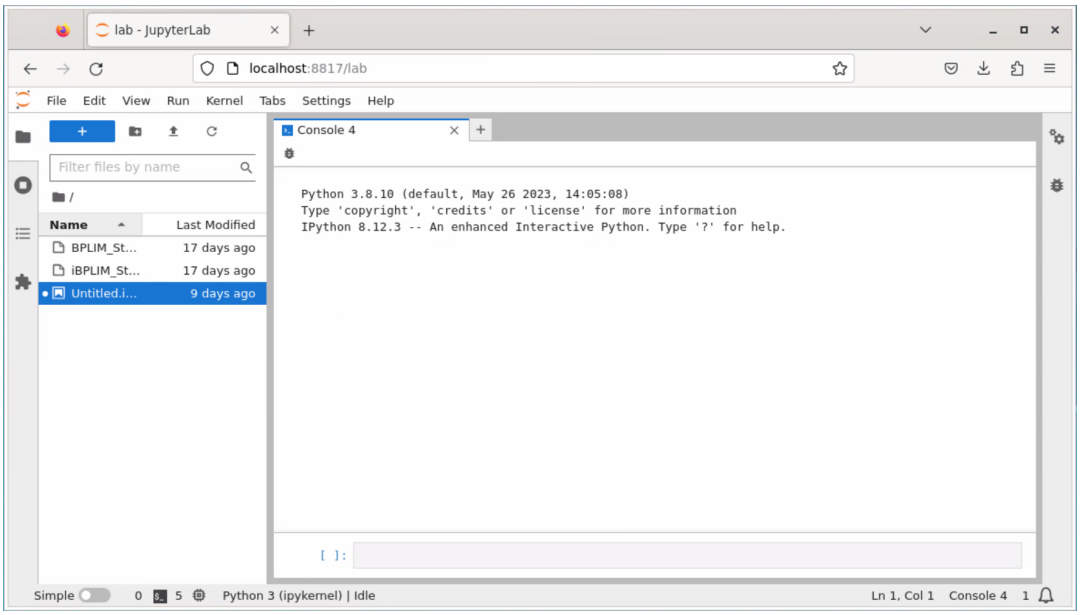

Figura 2: Ecrã inicial do *Jupyter*

Stata

O Stata é um programa estatístico que permite aos utilizadores trabalhar com bases de dados de grandes dimensões e disponibiliza uma grande diversidade de recursos para análises estatísticas: modelos lineares, análise de séries temporais, análise de sobrevivência, análise de dados categóricos, construção de gráficos e suporte a vários tipos de ficheiros de dados. Utiliza uma linguagem de programação para executar comandos, mas também oferece uma ampla variedade de comandos pré-programados acessíveis através da barra de menu.

O Stata quando iniciado, apresenta:

- Uma consola que é a principal área de trabalho do Stata e exibe informações sobre as operações realizadas, como comandos escritos, resultados de análises e mensagens de erro.
- Um painel de variáveis, onde exibe uma lista de todas as variáveis presentes na base de dados, incluindo os nomes, tipos e descrições.
- Uma janela de comandos que permite escrever comandos e executá-los ou editar *scripts* que já foram guardados, esta janela permite gravar o que é realizado, contrariamente à consola que apenas executa.
- Uma janela de resultados, onde mostra os resultados da análise realizada;
- Uma janela de gráficos, onde exibe gráficos criados a partir de dados, incluindo histogramas, gráficos de barras, gráficos de dispersão, entre outros.

A Figura 3 apresenta o ambiente inicial do Stata.

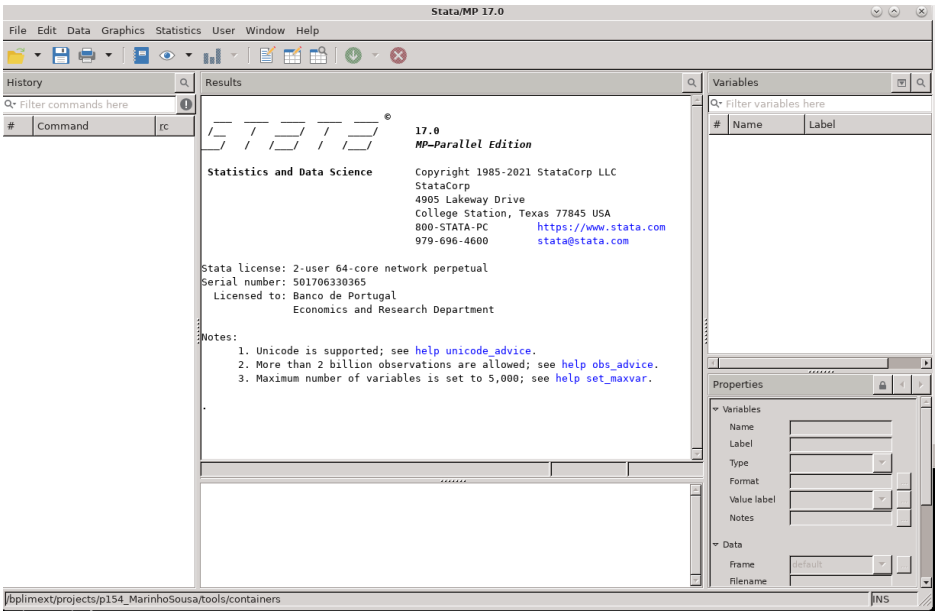

Figura 3: Ecrã inicial do Stata

O Stata sendo uma linguagem de programação muito utilizada em estatística apresenta uma grande variedade de funções e de bibliotecas direcionadas para MLG (Hardin & Hilbe, 2012). Apresenta a desvantagem de ser um *software* comercial e requerer uma licença para ser usado.

3.2 Criação de Dados

Neste estudo criam-se diferentes bases de dados que são aleatoriamente geradas. A linguagem de programação escolhida para a criação das bases de dados foi o R (R Core Team, 2023). As bases de dados criadas são de diferentes dimensões e as distribuições condicionadas da variável resposta seguem duas distribuições diferentes: a distribuição de Poisson e a distribuição de Bernoulli.

O processo de criação das bases de dados inicia-se com a geração de quatro variáveis, as variáveis explicativas, que desempenham um papel importante na criação da variável resposta do MLG. Dessas quatro variáveis, duas seguem uma distribuição Normal Padrão, $N(0,1)$, enquanto que as restantes seguem uma distribuição Uniforme no intervalo [0,1]. No Código 1 ilustra-se a criação dessas variáveis recorrendo à função `rnorm()` e `runif()`, onde n é a dimensão da amostra.

```
1  x1 <- rnorm(n); x2 <- rnorm(n); x3 <- runif(n); x4 <- runif(n)
```

Código 1: Criação das variáveis explicativas

O passo seguinte consiste na escolha dos coeficientes do modelo. Para esta escolha consideram-se quatros modelos distintos. No Modelo 1 a distribuição condicionada da variável resposta é a distribuição Poisson. O Modelo 2 pressupõe que a distribuição condicionada da variável resposta é a distribuição Bernoulli, em que o número casos de insucesso é muito maior que o número de casos de sucesso. No Modelo 3 também se assume uma distribuição condicionada da variável resposta, a distribuição Bernoulli, mas em que o número de casos de sucesso é praticamente igual ao número de casos de insucesso. Por fim, no Modelo 4, a distribuição condicionada é a distribuição Bernoulli, mas o número de casos de sucesso é maior que o número de casos de insucesso.

Assim sendo e tendo em consideração as características mencionadas anteriormente, a escolha dos parâmetros dos modelos é:

Modelo 1: $\beta_0 = 1.0$, $\beta_1 = 1.5$, $\beta_2 = 2$, $\beta_3 = -2.5$ e $\beta_4 = 3$.

Modelo 2: $\beta_0 = -1.5$, $\beta_1 = 3.5$, $\beta_2 = -5.2$, $\beta_3 = -6.8$ e $\beta_4 = 0.2$.

Modelo 3: $\beta_0 = -1.5$, $\beta_1 = 4$, $\beta_2 = -1.2$, $\beta_3 = -0.8$ e $\beta_4 = 3$.

Modelo 4: $\beta_0 = 3.5$, $\beta_1 = -2.5$, $\beta_2 = 7.2$, $\beta_3 = -3.8$ e $\beta_4 = 6.2$.

De seguida é necessário gerar a variável resposta. Para gerar a variável resposta com distribuição Bernoulli utiliza-se a função `rbinom()` do R. O Código 2 apresenta a criação da variável resposta para o Modelo 3. As

variáveis resposta do Modelo 2 e 4 são criadas de forma similar. No caso do Modelo 1, recorre-se à função `rpois()`. No Código 3 apresenta-se a criação da variável resposta deste modelo.

```
#Coeficientes do modelo
beta0 <- -1.5
beta1 <- 5.5
beta2 <- -6.5
beta3 <- -4.5
beta4 <- 3.5
#Valor médio
p <- exp(beta0 + beta1 * x1 + beta2 * x2 + beta3 * x3 + beta4 * x4) / (1 +
exp(beta0 + beta1 * x1 + beta2 * x2 + beta3 * x3 + beta4 * x4))
#Variável resposta
y <- rbinom(n, 1, p)
```

Código 2: Criação da variável resposta com distribuição Bernoulli(p)

```
#Coeficientes do modelo
beta0 <- 1
beta1 <- 1.5
beta2 <- 2.0
beta3 <- -2.5
beta4 <- 3.0
#Valor médio
lambda <- exp(beta0 + beta1 * x1 + beta2 * x2 + beta3 * x3 + beta4 * x4)
#Variável resposta
y <- rpois(n, lambda)
```

Código 3: Criação da variável resposta com distribuição Poisson, P(λ)

Para os Modelos 2, 3 e 4 apresenta-se a tabela de frequências da variável resposta gerada no exemplo de uma amostra de dimensão 10000.

Tabela 3.6: Frequências da variável resposta dos diferentes modelos (n=10000)

Resposta	
0	1
7596	2404
Modelo 2	

Resposta	
0	1
5342	4658
Modelo 3	

Resposta	
0	1
2743	7257
Modelo 4	

Para o Modelo 1, caso em que a variável resposta segue uma distribuição Poisson, apresenta-se um histograma da frequência da variável resposta gerada no exemplo de amostra de dimensão 10000 (Figura 4).

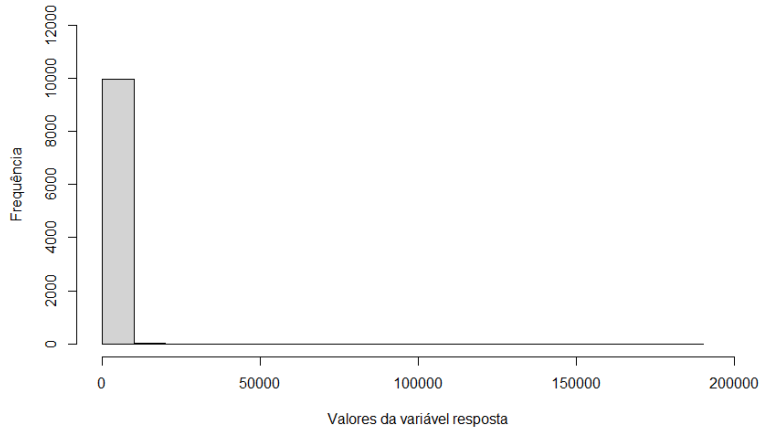

Figura 4: Histograma da variável resposta com distribuição Poisson (n=10000)

Para uma melhor interpretação do gráfico apresentam-se as estatísticas descritivas da variável resposta na Tabela 3.7.

Tabela 3.7: Estatísticas descritivas da variável resposta com distribuição Poisson (n=10000)

Mínimo	1° Quartil	Mediana	Media	3° Quartil	Máximo
0.0	0.0	4.0	152.8	23.0	189654.0

Da análise do histograma da Figura 4, identifica-se a presença de valores elevados (valor máximo 189654). Essa ocorrência deve-se aos valores elevados que o valor médio (λ) pode tomar. Para lidar com essa questão, decidiu-se restringir os valores médios (λ) excluindo os valores extremos da variável resposta cujo valor médio era superior ao terceiro quartil dos valores de λ. Para manter o tamanho da amostra geraram-se inicialmente amostras de dimensão $n_1 = n/0.75$, ou seja, $n_1 = 667$, $n_1 = 2667$, $n_1 = 13334$ e $n_1 = 26667$.

No Código 4 ilustra-se o processo descrito. Após calcular os valores médios (λ), selecionaram-se as observações com valor λ menor ou igual ao terceiro quartil dos valores de λ. Essa abordagem possibilitou a remoção dos valores elevados e melhorar a estabilidade do modelo.

```
#Valor médio
lambda <- exp(beta0 + beta1 * x1 + beta2 * x2 + beta3 * x3 + beta4 * x4)
dados<-data.frame(x1,x2,x3,x4,lambda)
terceiro_quartil <- quantile(lambda, 0.75)
#Restrinção dos valores de lambda
x1 <- dados$x1[lambda <= terceiro_quartil]
x2 <- dados$x2[lambda <= terceiro_quartil]
x3 <- dados$x3[lambda <= terceiro_quartil]
x4 <- dados$x4[lambda <= terceiro_quartil]
lambda <- lambda[lambda <= terceiro_quartil]
#Variável resposta
```

```r
y <- rpois(length(lambda), lambda)
```

Código 4: Criação da variável resposta ajustada com distribuição Poisson, P(λ)

A análise do histograma da Figura 5 evidencia uma melhor aproximação da variável resposta à distribuição Poisson, excluindo valores extremos da variável resposta.

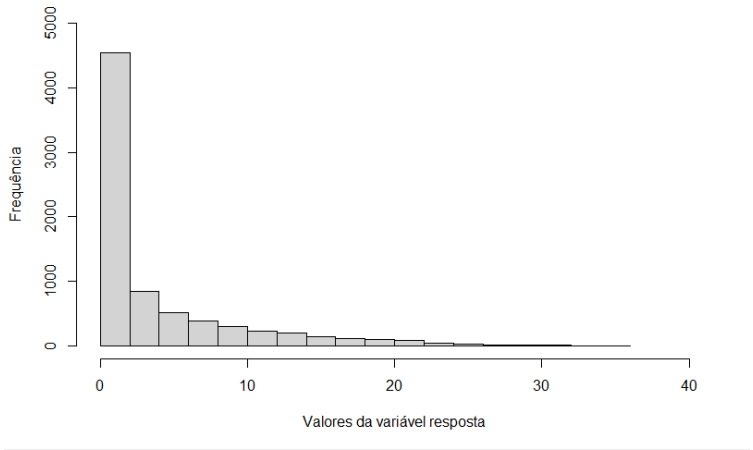

Figura 5: Histograma da variável resposta ajustada com distribuição Poisson (n=10000)

Do mesmo modo, apresenta-se a análise descritiva da variável resposta na Tabela 3.8.

Tabela 3.8: Estatísticas descritivas da variável resposta ajustada com distribuição Poisson (n=10000)

Mínimo	1° Quartil	Mediana	Media	3° Quartil	Máximo
0.0	0.0	1.0	3.844	5.0	35.0

3.3 Comparação das Linguagens de Programação

Na comparação das linguagens de programação, serão analisados os coeficientes do modelo, o tempo necessário na estimação do modelo e o número de iterações requeridas até à convergência do algoritmo.

È importante salientar que as comparações são executadas na mesma máquina, assegurando um ambiente equitativo e minimizando potenciais influências de variáveis externas nos resultados.

Para a realização deste trabalho, foi utilizado um servidor externo fornecido pelo BPLIM com um ambiente Linux, equipado com 56 CPUs, 816GB de memória RAM e 8TB de espaço de armazenamento.

Como referido os dados são gerados no ambiente R e posteriormente convertidos para formato *Excel*. Essa conversão foi realizada para possibilitar o uso das mesmas no Python e Stata, assegurando, assim, a consistência dos dados.

Com as bases de dados criadas, o próximo passo envolve a estimação dos modelos, um processo realizado em cada linguagem de programação.

Criação de MLG no RStudio

Usando a linguagem R, existem várias funções disponíveis para a estimação de MLG, no entanto, a mais utilizada pela maioria dos utilizadores é a função `glm` (Chambers, Hastie, & Pregibon, 1990).

A função `glm` encontra-se no *package stats* e permite ajustar MLG, especificado por uma descrição do preditor linear.

No Código 5 apresenta-se a estimação de um MLG usando a função *glm* da linguagem R, utilizando os seguintes argumentos:

- *formula* : descrição do modelo a ser ajustado;
- *family* : descrição da distribuição da variável resposta e função de ligação a ser usada no modelo;
- *data* : base de dados em estudo, pode-se introduzir uma lista ou ambiente que contenha as variáveis do modelo;
- *weights* : vetor opcional de pesos. Pode ser nulo ou um vetor numérico;
- *subset* : vetor opcional especificando um subconjunto de observações a serem ajustadas no processo de ajustamento;
- *na.action* : função que indica o que deve acontecer quando os dados contêm valores ausentes;
- *start* : valores iniciais para o preditor linear;
- *mustart* : valor médio inicial da variável resposta;
- *offset* : usado para especificar uma componente conhecida *à priori* a ser incluído no preditor linear durante o ajuste. Pode ser nulo ou um vetor numérico. Pode ser incluído mais que um termo, se tal acontecer, usa-se a soma;
- *control* : lista de parâmetros para controlar o processo de convergência;
- *model* : um valor lógico (TRUE OU FALSE) que indica se a estrutura do modelo deve ser incluída como uma componente do valor que retorna;
- *method* : método a ser usado no ajuste do modelo. O método padrão é o `glm.fit` que usa o método iterativo dos mínimos quadrados ponderados;
- *x* : valor lógico (TRUE ou FALSE) que indica se o vetor de resposta usado no ajustamento do modelo deve ser retornado. Por predefinição é TRUE;

- *Y*: valor lógico (TRUE ou FALSE) que indica se a matriz usada no ajustamento do modelo deve ser retornada. Por predefinição é TRUE;

- *singular.ok* : valor lógico (TRUE ou FALSE). Está predefinido TRUE, se FALSE significa que a função glm() irá interromper a execução e retornar uma mensagem de erro quando detetar uma matriz de desenho singular.

Existem, no entanto, outros argumentos que podem ser explorados usando o comando `help(glm)` do R.

```
glm(formula, family = gaussian, data, weights, subset, na.action, start = NULL,
etastart, mustart, offset, control = list(...), model = TRUE, method = "glm.fit",
x = FALSE, y = TRUE, singular.ok = TRUE, contrasts = NULL, ...)
```

Código 5: Estimação de um MLG em R

Um exemplo do uso desta função pode ser ilustrado através de uma base de dados criada seguindo os critérios da secção anterior. Para tal, recorreu-se a uma base de dados com dimensão n=500, em que a variável resposta segue uma distribuição Poisson e usando o Modelo 1. Uma maneira simples de analisar um modelo estimado é através da função `summary()`, que apresenta várias estatísticas relevantes do modelo. A Figura 6 apresenta o modelo estimado usando a linguagem R.

```
> # Ajustar o modelo GLM com distribuição Poisson
> modelo<-glm(y~x1+x2+x3+x4,data=dados,family = poisson(link = "log"))
> summary(modelo)

Call:
glm(formula = y ~ x1 + x2 + x3 + x4, family = poisson(link = "log"),
    data = dados)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5033  -0.7647  -0.2330   0.4131   3.1885

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.94423    0.06183   15.27   <2e-16 ***
x1           1.49166    0.03918   38.08   <2e-16 ***
x2           1.99211    0.04531   43.96   <2e-16 ***
x3          -2.39865    0.08661  -27.70   <2e-16 ***
x4           3.06543    0.09354   32.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 4155.25  on 499  degrees of freedom
Residual deviance:  453.51  on 495  degrees of freedom
AIC: 1571

Number of Fisher Scoring iterations: 5
```

Figura 6: Visualização do modelo estimado em R

Criação de um MLG no Python

Existem diversas formas de estimar um MLG no Python. Uma possibilidade é através da biblioteca *statsmodels* usando a função `sm.GLM`. Esta função apresenta 3 argumentos:

- *X*: variáveis explicativas do modelo;
- *Y*: variável resposta do modelo;
- *family*: descrição da distribuição de probabilidade da variável resposta e a função de ligação a serem usadas no modelo.

O Código 6 apresenta de forma genérica como criar um MLG usando a linguagem de programação Python.

```
import pandas as pd
import statsmodels.api as sm
modelo = sm.GLM(y, X, family=sm.families.Poisson())
```

Código 6: Estimação de um MLG em Python

A Figura 7 apresenta a aplicação da função `sm.GLM` à mesma base de dados com dimensão n=500 usada na estimação de um MLG no R, em que a variável resposta segue uma distribuição Poisson usando o Modelo 1. Os resultados são obtidos através das funções `modelo.fit()` e `summary()`.

```
resultado=modelo.fit()
resultado.summary()
```

[1]:

Generalized Linear Model Regression Results

Dep. Variable:	y	**No. Observations:**	500
Model:	GLM	**Df Residuals:**	495
Model Family:	Poisson	**Df Model:**	4
Link Function:	Log	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-780.50
Date:	Sat, 28 Oct 2023	**Deviance:**	453.51
Time:	23:48:53	**Pearson chi2:**	463.
No. Iterations:	6	**Pseudo R-squ. (CS):**	0.9994
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.9442	0.062	15.272	0.000	0.823	1.065
x1	1.4917	0.039	38.077	0.000	1.415	1.568
x2	1.9921	0.045	43.962	0.000	1.903	2.081
x3	-2.3986	0.087	-27.695	0.000	-2.568	-2.229
x4	3.0654	0.094	32.771	0.000	2.882	3.249

Figura 7: Visualização do modelo estimado em Python

Criação de um MLG em Stata

Para a criação de um MLG usando Stata recorre-se à função `glm()`.

No Código 7 apresenta-se a estimação de um MLG no Stata, utilizando os seguintes argumentos:

- *depvar* : variável resposta do modelo;
- *indepvars* : variáveis explicativas do modelo;
- *if* : expressão lógica que especifica quais observações devem ser incluídas no modelo;
- *in* : lista de números inteiros que especifica quais observações devem ser incluídas no modelo;
- *weight* : vetor de pesos para as observações;
- *family* : distribuição da variável resposta, por defeito o é a distribuição Normal;
- *link* : função de ligação da variável resposta, por defeito é a canónica;
- *vce* : matriz de covariância dos erros;
- *predict* : especifica se as previsões do modelo devem ser calculadas e armazenadas na memória.

Existem no entanto outra opções que podem ser incluídas (StataCorp, 2023).

```
glm depvar [indepvars] [if] [in] [weight] [, options]
```

Código 7: Estimação de um MLG em Stata

A Figura 8 apresenta os resultados do Modelo 1, estimado no Stata, usando a mesma base de dados com dimensão 500.

```
. glm y x1 x2 x3 x4, family(poisson) link(log) irls

Iteration 1:   deviance =  952.7697
Iteration 2:   deviance =  513.9186
Iteration 3:   deviance =   454.994
Iteration 4:   deviance =  453.5128
Iteration 5:   deviance =  453.5116
Iteration 6:   deviance =  453.5116

Generalized linear models                         Number of obs   =        500
Optimization     : MQL Fisher scoring             Residual df     =        495
                   (IRLS EIM)                     Scale parameter =          1
Deviance         =  453.5115978                   (1/df) Deviance =    .916185
Pearson          =  463.0852036                   (1/df) Pearson  =   .9355257

Variance function: V(u) = u                       [Poisson]
Link function    : g(u) = ln(u)                   [Log]

                                                  BIC             =  -2622.719
```

y	Coefficient	EIM std. err.	z	P>|z|	[95% conf.	interval]
x1	1.491658	.0391751	38.08	0.000	1.414876	1.568439
x2	1.992112	.0453142	43.96	0.000	1.903298	2.080926
x3	-2.398648	.0866106	-27.69	0.000	-2.568402	-2.228895
x4	3.065434	.0935422	32.77	0.000	2.882095	3.248774
_cons	.9442311	.0618278	15.27	0.000	.8230509	1.065411

Figura 8: Visualização do modelo estimado no Stata

Como se pode confirmar nas Figuras 6, 7 e 8 as estimativas obtidas dos coeficiente do modelo nas diferentes linguagens de programação foram as mesmas. Este resultado era esperado uma vez que as linguagens usam o mesmo método de estimação.

Tabela 3.9: Tempo e número de iterações na estimação de MLG em diferentes linguagens

Linguagem	**Tempo e Iterações**			
	n=500		n=20000	
	Tempo (s)	Iterações	Tempo (s)	Iterações
R	0.00336	**5**	0.05301	**5**
Python	**0.00029**	6	**0.00123**	6
Stata	0.04618	6	0.23578	6

A Tabela 3.9 apresenta uma análise comparativa do desempenho das linguagens de programação na estimação de modelos de MLG em dois cenários de tamanho de amostra, n=500 e n=20000, em ambos foi considerado um Modelo de Regressão Poisson estimada com a mesma base de dados.

Os resultados revelam que, em termos de eficiência computacional todas as linguagens requerem pouco tempo para estimar um modelo, no entanto a linguagem Python destaca-se, apresentando os tempos de execução mais baixos em ambas as amostras. O número de iterações é mais reduzido na linguagem R necessitando apenas de 5 iterações enquanto o Stata e o Python necessitam de 6. Note-se que os tempos de execução foram obtidos pela média de 500 repetições para uma maior consistência dos resultados.

Uma vez que os resultados são muito semelhantes nas três linguagens de programação, prossegue-se os estudos recorrendo à linguagem R.

3.4 Estimação do parâmetros

Nesta secção pretende-se comparar, diferentes funções do R na estimação de MLG.

O R apresenta diversos *packages* que permitem estimar MLG, tais como o *stats, glm2, arm, brglm, logistf, robustbase* e *VGAM*.

O *stats* é um dos *package* básicos da linguagem de programação R e faz parte da instalação padrão do R. Inclui um conjunto fundamental de funções e ferramentas estatísticas para análise de dados incluindo a função `glm()`.

- glm: Esta função é a mais conhecida para a criação de MLG e será usada como base neste estudo. Ajusta MLG através da especificação da variável resposta, das variáveis explicativas e indicação da família de distribuição. Faz o ajustamento a partir do Método Mínimos Quadrados Iterativos e permite criar modelos GLM com diferentes famílias de distribuição. A estimação deste modelo foi realizado para as duas famílias de distribuição em estudo: Poisson(`family=poisson(link="log")`) e Bernoulli(`family=binomial(link="logit")`). O controlo de convergência do modelo foi o definido por defeito, com se pode ver pelos argumentos da função `glm.control()`:

  $$\texttt{glm.control(epsilon = } 1 \times 10^{-8}\texttt{, maxit = 25, trace = FALSE)}.$$

  O Código 8 representa a estimação de MLG através da função `glm`.

```
1 glm(y ~ x1 + x2 + x3 + x4, data=dados, family, ...)
```

Código 8: Estimação de um MLG através da função `glm()`

O *package glm2*, utiliza a mesma especificação de modelo que o *glm*, mas com um método de ajustamento distinto, proporcionando maior estabilidade para modelos propensos a problemas de convergência (Marschner & Donoghoe, 2018). A função `glm2()` encontra-se neste *package*.

- `glm2`: Esta função usa a mesma especificação do modelo que a *glm*, mas adota um método de ajuste diferente. Nesse método, denominado método Mínimos Quadrados Reponderados com redução pela metade modificada, o desvio é forçado a diminuir a cada iteração. Esta função foi aplicada a duas distribuições: Poisson(`family=poisson(link="log")`) e Bernoulli(`family=binomial(link="logit")`). O controlo de convergência do modelo foi o definido por defeito, que é igual ao da função `glm()`,

  $$\texttt{glm.control(epsilon = } 1 \times 10^{-8}\texttt{, maxit = 25, trace = FALSE)}.$$

  A estimação deste modelo foi executada conforme ilustrado no Código 9.

```
glm2(y ~ x1 + x2 + x3 + x4, data=dados, family, ... )
```

Código 9: Estimação de um modelo através da função `glm2()`

O *package brglm* é direcionado para MLG onde a variável resposta segue uma distribuição Binomial, usando uma abordagem de pontuação ajustada para redução de viés (Kosmidis, 2021). Uma das principais funções deste *package* é a função `brglm()` .

- `brglm`: Esta função é usada para ajustar modelos de regressão logística e outros modelos de resposta binomial com viés reduzido. A família de distribuição tem como argumento `binomial(link="log")`. O controlo de convergência usado foi o predefinido, que é igual ao da função `glm()`,

$$\texttt{glm.control(epsilon = } 1 \times 10^{-8}\texttt{, maxit = 25, trace = FALSE)}.$$

  O Código 10 apresenta como se estima o modelo através desta função.

```
brglm(y ~ x1 + x2 + x3 + x4, data=dados, family, ... )
```

Código 10: Estimação de um modelo através da função `brglm()`

O *package logistf*, por sua vez, é dedicado ao ajuste de modelos de regressão logística e usa o método de redução de viés de Firth, equivalente à penalização da probabilidade logarítmica pelo método de Jeffreys anterior (Heinze, Ploner, Dunkler, Southworth, & Jiricka, 2022). A função `logistf()` encontra-se neste *package*.

- `logistf` : Esta função encontra-se no *package logistf* e permite ajustar modelos de regressão logística utilizando correção de Firth aplicada à verossimilhança (Firth, 1993). O ajuste deste modelo foi realizado conforme o Código 13, aplicado à família Bernoulli (`family=binomial(link="logit")`) e com o controlo de convergência predefinido pelo modelo, com os argumentos da função `logistf.control()`:

$$\texttt{logistf.control(maxit = 25, maxhs = 0, maxstep = 5, lconv = } 1 \times 10^{-5}\texttt{,}$$
$$\texttt{gconv = } 1 \times 10^{-5}\texttt{, xconv = } 1 \times 10^{-5}\texttt{, collapse = TRUE, fit = "NR")}.$$

  O Código 11 apresenta a estimação de um modelo recorrendo a esta função.

```
logistf(y ~ x1 + x2 + x3 + x4, data=dados, family, ... )
```

Código 11: Estimação de um modelo através da função `logistf()`

O *package robustbase* fornece métodos robustos para a análise de dados, incluindo estimação de parâmetros robustos em MLG. É útil quando se lida com dados que podem conter valores atípicos ou desvios substanciais da normalidade (Maechler et al., 2023). Dentro deste *package* encontra-se a função `glmrob()`.

- `glmrob`: Esta usa estimadores robustos do tipo Mallows ou Huber. Aplica-se para famílias Binomial, Poisson, Gamma e Normal. No contexto deste estudo aplicou-se às famílias: Bernoulli(`family=binomial(link="logit")`) e Poisson(`family=poisson(link="log")`. O método de controlo de convergência usado foi o predefinido pelo modelo, com os argumentos da função `glmrobMqle.control()`:

  `glmrobMqle.control(acc =` 1×10^{-4}`, test.acc = "coef", maxit = 50, tcc = 1.345)`.

  O Código 12 demonstra como se usa esta função.

```
glmrob(y ~ x1 + x2 + x3 + x4, data=dados, family, ... )
```

Código 12: Estimação de um modelo através da função `glmrob()`

O *package VGAM* é uma ferramenta versátil para ajustar Modelos Lineares Generalizados Aditivos (MLGA) e Modelos Lineares Generalizados Aditivos Multivariados (MLGAM), permite a inclusão de componentes não lineares e interações complexas em análises de regressão (Yee, 2023). Entre várias funções encontra-se a função `vglm()`.

- `vglm`: Esta função ajusta modelos lineares generalizados vetoriais (VGLM) incluindo MLG como um caso especial. O ajuste do modelo é realizado utilizando o Método dos Mínimos Quadrados. Neste estudo aplicou-se à família: Bernoulli(`family=binomialff`) e Poisson(`family=poissonff`). O controlo de convergência foi o predefinido, com os argumentos da função `vglm.control()`:

  `vglm.control(checkwz = TRUE, Check.rank = TRUE, Check.cm.rank = TRUE, criterion = names(.min.criterion.VGAM), epsilon =` 1×10^{-7}`, half.stepsizing = TRUE, maxit = 30, noWarning = FALSE, stepsize = 1, save.weights = FALSE,trace = FALSE, wzepsilon = .Machine$double.eps0.75, xij = NULL, ...)`

  O Código 13 ilustra como se estima um modelo através desta função.

```
vglm(y ~ x1 + x2 + x3 + x4, data=dados, family, ... )
```

Código 13: Estimação de um modelo através da função `vglm()`

O *package arm* disponibiliza uma estrutura bayesiana para análise de dados e estimação de parâmetros e inclui a função `bayesglm()` (Gelman et al., 2022).

- `bayesglm`: Esta função é uma versão modificada do glm no R que permite incorporar informações prévias durante a construção do modelo. Usa um algoritmo especial chamado EM aproximado para atualizar os

coeficientes do modelo, tendo em conta informações anteriores por meio de uma regressão aumentada. Está disponível para a estimação de modelos a diversas funções de distribuição. No âmbito deste estudo usaram-se as famílias: Bernoulli(`family=binomial(link="log)`) e Poisson(`family=poisson(link="log")`, o controlo de convergência foi o predefinido, que é igual ao da função `glm()`, com argumento:

$$\texttt{glm.control(epsilon =}1 \times 10^{-8}\texttt{, maxit = 25, trace = FALSE)}.$$

O Código 14 apresenta como usar a função.

```
bayesglm(y ~ x1 + x2 + x3 + x4, data=dados, family, ... )
```

Código 14: Estimação de um modelo através da função `bayesglm()`

Além destas funções, existem diversas funções alternativas para estimar MLG, a decisão sobre qual utilizar depende dos objetivos específicos que se pretende alcançar.

O estudo iniciou-se com a análise das sete funções diferentes dentro destes *packages* mencionados.

Para analisar o desempenho das diversas funções na estimação dos parâmetros de um MLG utilizam-se o Erro Quadrático Médio (MSE), a Eficiência Relativa em Percentagem (PRE) e o Erro Percentual Absoluto Médio (MAPE).

O Erro Quadrático Médio das estimativas dos parâmetros β_j pode ser calculada por

$$MSE_{\beta_j} = \sum_{s=1}^{S} \frac{(\hat{\beta}_{j,s} - \beta_j)^2}{S},$$

para $j = 0, .., 4$, onde β_j é o valor verdadeiro do j-ésimo coeficiente do modelo e $\hat{\beta}_{j,s}$ é o valor estimado do j-ésimo coeficiente do modelo na s-ésima simulação.

O MSE é uma medida estatística utilizada para quantificar a discrepância entre os valores verdadeiros dos parâmetros do modelo e os valores estimados com base nos dados. Valores menores de MSE indicam um ajuste mais preciso e coerente.

Como complemento ao MSE, calculou-se a Eficiência Relativa em Percentagem das diferentes funções (PRE). O PRE permite comparar as estimativas do MSE quando se aplicam diversas funções na estimação. Neste estudo compara-se a estimativa dos valores do MSE aplicando diferentes funções em relação à estimativa do MSE quando se usa função *glm*, que se designa por f_1, este pode ser calculado como

$$PRE_{f_k}(\%) = \frac{MSE(f_k)}{MSE(f_1)} \times 100,$$

onde f_k são as diversas funções usadas no estudo para estimar MLG.

O Erro Percentual Absoluto Médio (MAPE) pode ser calculado por

$$MAPE_{\beta_j}(\%) = \frac{1}{S}\left|\frac{\sum_{s=1}^{S}(\hat{\beta}_{j,s} - \beta_j)}{\beta_j}\right| \times 100,$$

com $j = 0, .., 4$, onde β_j é o valor verdadeiro do j-ésimo coeficiente do modelo e $\hat{\beta}_{j,s}$ é o valor estimado do j-ésimo coeficiente do modelo na s-ésima simulação.

Este indica como a média das proporções dos erros de estimação relativamente ao verdadeiro valor do parâmetro diferem, em termos percentuais (Eker, Poudel, & Özçelik, 2017).

Além do estudo do MSE, do PRE e do MAPE, foram também realizadas análises do tempo de execução e do número de iterações necessárias para a convergência do algoritmo quando se aplicam as funções na estimação dos modelos.

Esta última análise pretende estudar o desempenho computacional das diversas funções na estimação dos modelos. Por vezes, quando se trabalha com grandes volumes de dados, o tempo de execução é um aspeto crucial, pois modelos mais complexos podem exigir um tempo significativo na sua estimação, sendo assim relevante esta componente ser estudada. Por outro lado, a avaliação do número de iterações é também um critério importante para compreender a convergência do modelo. Funções que requerem um número elevado de iterações necessário para a convergência do algoritmo podem indicar uma dificuldade maior em se atingir a estabilidade do modelo.

Inicialmente o estudo foi conduzido com as sete funções referidas, no entanto, observa-se que as estimativas dos parâmetros são iguais quando se aplica diferentes funções. Isso deve-se ao facto de algumas das funções aplicadas usarem métodos de estimação similares, o que leva a resultados semelhantes. Perante desta constatação, optou-se por prosseguir o estudo sem considerar as funções que geravam resultados redundantes.

Assim:

- Entre as funções `glm`, `glm2` e `vglm` decidiu-se utilizar apenas a função `glm`, uma vez que todas são estimadas pelo método dos Mínimos Quadrados Reponderados;
- Entre as funções `logistf` e `brglm`, optou-se por selecionar apenas a função `brglm` uma vez que ambas fazem uso da verosimilhança penalizada máxima na estimação do modelo.

Assim sendo, a análise de resultados foi realizada considerando somente quatro funções distintas: `glm`, `brglm`, `glmrob` e `bayesglm`.

O planeamento do estudo de simulação é o seguinte:

- Dimensão da amostra (n): 500, 2000, 10000 e 20000
- Número de simulações: 500

- Variáveis explicativas: $X_1 \sim N(0,1)$, $X_2 \sim N(0,1)$, $X_3 \sim U(0,1)$ e $X_4 \sim U(0,1)$
- Variável resposta:Y $\sim$Bernoulli (μ) (Modelo de Regressão Logística)

  Y $\sim$Poisson (λ) (Modelo de Regressão Poisson)
- Funções usadas: `glm, glmrob, brglm` e `bayesglm` (Modelo de Regressão Logística)

  `glm, glmrob` e `bayesglm` (Modelo de Regressão Poisson)
- Parâmetros do modelo:

  Modelo 1: $\beta_0 = 1.\beta_1 = 1.5, \beta_2 = 2, \beta_3 = -2.5$ e $\beta_4 = 3.$ (Modelo de Regressão Poisson)

  Modelo 2: $\beta_0 = -1.5$, $\beta_1 = 3.5$, $\beta_2 = -5.2$, $\beta_3 = -6.8$ e $\beta_4 = 0.2.$ (Modelo de Regressão Logística)

  Modelo 3: $\beta_0 = -1.5.\beta_1 = 4, \beta_2 = -1.2, \beta_3 = -0.8$ e $\beta_4 = 3.$ (Modelo de Regressão Logística)

  Modelo 4: $\beta_0 = 3.5.\beta_1 = -2.5, \beta_2 = 7.2, \beta_3 = -3.8$ e $\beta_4 = 6.2.$ (Modelo de Regressão Logística)
- Estatísticas calculadas: MSE, MAPE, PRE, tempo e o número de iterações necessárias para a convergência.

Modelo de Regressão Poisson

O estudo de simulação foi realizado com recurso ao R e os resultados registados nas Tabelas 3.10, 3.11 e 3.12. Das quatro funções mencionadas foram aplicadas apenas a função `glm(), glmrob()` e `glmbayes()` uma vez que a função `brglm()` apenas se encontra definida para modelos em que a variável resposta pertence à família de distribuição Binomial.

Tabela 3.10: MSE dos coeficientes de regressão do modelo de regressão Poisson (Modelo 1)

		MSE			**PRE(%)**	
n	β_i	f_1=`glm`	f_2=`glmrob`	f_3=`bayesglm`	f_2/f_1	f_3/f_1
500	β_0	0.004651	0.004804	**0.004632**	103.29	99.59
	β_1	0.001883	0.002059	**0.001867**	109.35	99.15
	β_2	0.003229	0.003501	**0.003171**	108.24	98.20
	β_3	0.010045	0.010912	**0.009926**	108.63	98.81
	β_4	0.011539	0.012224	**0.011382**	105.94	98.64
2000	β_0	**0.001168**	0.001256	0.001169	107.53	100.09
	β_1	**0.000453**	0.000473	0.000455	104.42	100.44
	β_2	**0.000713**	0.000758	0.000717	106.31	105.72
	β_3	0.002407	0.002546	**0.002405**	105.77	99.92
	β_4	**0.002776**	0.002948	0.002779	106.20	100.12
10000	β_0	**0.000203**	0.000221	**0.000203**	108.87	100.00
	β_1	**0.000093**	0.000099	**0.000093**	106.45	100.00
	β_2	**0.000139**	0.000151	**0.000139**	108.63	100.00
	β_3	**0.000486**	0.000523	**0.000486**	107.61	100.00
	β_4	**0.000531**	0.000567	0.000532	106.78	100.19
20000	β_0	**0.000109**	0.000117	**0.000109**	107.34	100.00
	β_1	**0.000043**	0.000048	**0.000043**	111.63	100.00
	β_2	0.000067	0.000074	**0.000066**	110.45	98.51
	β_3	**0.000207**	0.000231	**0.000207**	111.59	100.00
	β_4	**0.000275**	0.000298	**0.000275**	108.36	100.00

Os resultados da Tabela 3.10 indicam diferenças na precisão entre as estimativas dos coeficientes quando estas são obtidas usando diferentes funções, realçando a negrito o valor com menor MSE em cada coeficiente.

Em geral, observa-se que a função `glmrob` produz valores de MSE ligeiramente maiores em comparação com a função `glm`. Isso é evidenciado pelo valor PRE entre as duas funções, cujos valores são superiores a 100% na maioria dos casos, sugerindo assim que a estimação pela função `glmrob` é menos precisa do que com a função `glm`.

Por outro lado, a função `bayesglm` destaca-se por apresentar valores de MSE menores em comparação com as funções `glmrob` e `glm` quando n =500. Os valores de PRE são em alguns casos menores que 100%, indicando maior precisão. Quando n=2000 nota-se um menor valor de MSE nas estimativas obtidas com a função `glm`.

Além disso, observa-se uma tendência geral relacionada com o tamanho da amostra. À medida que o tamanho da amostra aumenta, a precisão das estimativas dos parâmetros melhora, conforme evidenciado pela diminuição dos valores de MSE. De realçar também que conforme a dimensão da amostra aumenta os valores de MSE produzidos pelas funções `glm` e `glmbayes` são muito idênticos.

Portanto, a escolha do método de estimação dependerá dos objetivos analíticos e das características dos dados. Se a precisão for fundamental, as funções `bayesglm` e `glm` podem ser uma escolha sólida quando comparadas com a função `glmrob`.

Tabela 3.11: MAPE das estimativas dos coeficientes de regressão no modelo de regressão Poisson (Modelo 1)

		MAPE (%)		
n	β_i	`glm`	`glmrob`	`bayesglm`
500	β_0	0.1868	0.3563	**0.0535**
	β_1	0.1704	0.2104	**0.0553**
	β_2	0.3547	0.4140	**0.1375**
	β_3	0.3658	0.3887	**0.1186**
	β_4	0.3400	0.4417	**0.0976**
2000	β_0	**0.1036**	0.1069	0.1630
	β_1	**0.0920**	0.1150	0.1479
	β_2	**0.0711**	0.0873	0.1249
	β_3	0.0337	0.0093	**0.0280**
	β_4	**0.0228**	0.0293	0.0827
10000	β_0	0.0927	**0.0924**	0.1047
	β_1	0.0056	0.0095	**0.0055**
	β_2	**0.0204**	0.0245	0.0311
	β_3	0.0162	0.0098	**0.0040**
	β_4	**0.0346**	0.0366	0.0465
20000	β_0	**0.0227**	0.0295	0.0287
	β_1	0.0109	0.0088	**0.0054**
	β_2	0.0106	0.0067	**0.0053**
	β_3	0.0381	0.0365	**0.0320**
	β_4	0.0105	0.0103	**0.0046**

Os resultados da Tabela 3.11 demonstram que a função `bayesglm` apresenta em geral, menores valores de MAPE.

Além disso, observa-se como seria de esperar que, em geral, à medida que o tamanho da amostra aumenta, os valores de MAPE tendem a diminuir, indicando uma melhoria progressiva na precisão das estimativas.

Em resumo, os resultados indicam que a função `bayesglm` produz estimativas mais precisas em termos de MAPE e MSE em comparação com as funções `glm` e `glmrob` em diferentes tamanhos de amostra.

Tabela 3.12: Tempo e iterações necessárias na estimação do modelo de regressão Poisson (Modelo 1)

Tempo e Iterações				
n		glm	glmrob	bayesglm
500	Tempo (s)	**0.0027**	0.0062	0.0052
	Iterações	5.000	**3.186**	5.998
2000	Tempo (s)	**0.0068**	0.0180	0.0177
	Iterações	5.000	**2.638**	5.016
10000	Tempo (s)	**0.0406**	0.0899	0.0920
	Iterações	5.000	**2.008**	5.000
20000	Tempo (s)	**0.0893**	0.1828	0.1781
	Iterações	5.000	**2.000**	5.000

Pela análise da Tabela 3.12 nota-se que o tempo aumenta conforme se aumenta a dimensão da amostra o que era de esperar uma vez que se está a exigir um maior esforço computacional. No entanto, realizando uma análise global do tempo em relação às três funções avaliadas, observa-se que a função `glm` apresenta os menores tempos de execução.

No que diz respeito ao número de iterações necessárias para a convergência do algoritmo na estimação do modelo, observa-se que a função `glm` requer em média 5 iterações, independentemente do tamanho da amostra. A função `glmbayes` produz resultados muito semelhantes, inicialmente exigindo cerca de 6 iterações, mas estabilizando em 5 iterações à medida que a dimensão da amostra aumenta. Por outro lado, a função `glmrob` destaca-se por necessitar de um menor número de iterações para atingir a convergência, em comparação com as restantes funções.

Estas conclusões ressaltam a eficiência da função `glm` em relação ao tempo de processamento, enquanto que a função `glmrob` se destaca ao requerer um menor número de iterações para alcançar a convergência. No entanto deve-se realçar que quando se aplica esta função as estimativas dos coeficientes do modelo são as menos precisas

Para análises de regressão com distribuição Poisson, a função `bayesglm` pode ser a escolha preferível em termos de precisão das estimativas. No entanto, a seleção do método ainda deve considerar outros fatores, como a convergência do modelo e a adequação aos pressupostos do problema.

Modelo de Regressão Logística

O mesmo processo descrito para o modelo de regressão Poisson foi realizado para o modelo de regressão Logística e aplicado aos Modelos 2, 3 e 4, usando as funções `glm`, `brglm`, `glmrob` e `bayesglm`. Nas Tabelas 3.13 e 3.14 apresentam-se os resultados obtidos para o Modelo 3.

Tabela 3.13: MSE das estimativas dos coeficientes de regressão do modelo de regressão Logística (Modelo 3)

		MSE				**PRE(%)**		
n	β_i	f_1=glm	f_2=brglm	f_3=glmrob	f_4=bayesglm	f_2/f_1	f_3/f_1	f_4/f_1
500	β_0	0.215829	0.201521	0.258877	**0.194389**	93.37	119.95	90.07
	β_1	0.188031	0.156123	0.283612	**0.151679**	83.03	150.83	80.67
	β_2	0.042467	0.038542	0.051858	**0.037757**	90.76	122.11	88.91
	β_3	0.351090	0.333307	0.395740	**0.315569**	94.93	112.72	89.88
	β_4	0.359644	0.334286	0.445476	**0.326331**	92.95	123.87	90.74
2000	β_0	0.053352	0.052131	0.060539	**0.051470**	97.71	113.47	96.47
	β_1	0.038110	0.036706	0.050826	**0.036452**	96.32	133.37	95.65
	β_2	0.008359	0.008171	0.010393	**0.008131**	97.75	124.33	97.27
	β_3	0.075618	0.074772	0.081794	**0.073879**	98.88	108.17	97.70
	β_4	0.092626	0.090189	0.112010	**0.088981**	97.37	120.93	96.06
10000	β_0	0.009750	0.009680	0.010610	**0.009643**	99.28	108.82	98.90
	β_1	0.006093	0.006034	0.008083	**0.006025**	99.03	132.66	98.88
	β_2	0.001691	0.001682	0.001976	**0.001680**	99.47	116.85	99.35
	β_3	0.015590	0.015570	0.016587	**0.015550**	99.87	106.40	99.74
	β_4	0.016985	0.016867	0.019451	**0.016802**	99.31	114.52	98.92
20000	β_0	0.004816	0.004803	0.005254	**0.004797**	99.73	109.09	99.61
	β_1	0.003774	0.003739	0.004717	**0.003734**	99.07	124.99	98.94
	β_2	0.000810	0.000808	0.000958	**0.000807**	99.75	118.27	99.63
	β_3	0.008212	0.008203	0.009234	**0.008193**	99.89	112.45	99.77
	β_4	0.009091	0.009062	0.010188	**0.009046**	99.68	112.07	99.51

Analisando a Tabela 3.13 que apresenta os valores do MSE das estimativas dos coeficientes do Modelo 3 destaca-se que a função `bayesglm` apresenta sempre menores valores de MSE em comparação com as outras funções aplicadas.

Além disso, ao analisar os valores do PRE, observa-se que tanto a função `brglm` como a função `bayesglm` apresentam valores de MSE inferiores relativamente à função `glm`, enquanto que a função `glmrob` apresenta valores superiores.

Numa análise geral, à medida que a dimensão da amostra aumenta, observa-se, em geral, uma redução nos valores do MSE. Este comportamento é o esperado, isto é, com o aumento da dimensão da amostra a precisão das estimativas dos coeficientes do modelo são melhores.

Tabela 3.14: MAPE das estimativas dos coeficientes de regressão no modelo de regressão Logística (Modelo 3)

		Erro Percentual Absoluto Médio (%)			
n	β_i	f_1=glm	f_2=brglm	f_3=glmrob	f_4=bayesglm
500	β_0	3.2860	**0.5695**	4.1767	0.7775
	β_1	3.1647	0.3664	4.6698	**0.0702**
	β_2	2.7687	**0.0243**	4.1392	1.6046
	β_3	**0.9742**	3.6061	1.2056	6.4068
	β_4	1.9763	**0.7198**	2.8156	2.7671
2000	β_0	1.9360	1.2797	2.3319	**0.9447**
	β_1	0.5885	**0.0776**	0.7662	0.1548
	β_2	0.5890	**0.0775**	0.7536	0.4747
	β_3	**1.3246**	1.9599	1.4622	2.6778
	β_4	1.1728	0.5192	1.4087	**0.0046**
10000	β_0	0.8085	0.6793	0.8614	**0.6117**
	β_1	0.1394	**0.0073**	0.2317	0.0083
	β_2	0.1467	**0.0145**	0.2335	0.0655
	β_3	1.1604	1.2874	**0.9056**	1.4350
	β_4	0.3571	0.2280	0.4192	**0.1236**
20000	β_0	0.2303	0.1662	0.2808	**0.1326**
	β_1	0.1616	0.0955	0.1921	**0.0877**
	β_2	0.0707	**0.0046**	0.1056	0.0353
	β_3	0.1820	0.2462	**0.0524**	0.3206
	β_4	0.1613	0.0969	0.2129	**0.0448**

Pela análise da Tabela 3.14 não se observa uma função que apresente os menores valores de MAPE para todos os coeficientes. Contudo, uma análise mais detalhada, permite verificar que as funções `brglm` e `bayesglm`, em geral, apresentam valores mais reduzidos que as funções `glm` e `glmrob`.

Numa análise direcionada para a alteração dos valores do MAPE com a dimensão da amostra observa-se que na maioria dos casos há uma redução do valor com o aumento da dimensão da amostra.

Tabela 3.15: Tempo e iterações necessárias para estimar o modelo de regressão Logística (Modelo 3)

Tempo e Iterações					
n		glm	brglm	glmrob	bayesglm
500	Tempo (s)	0.0029	0.0148	0.0069	0.0061
	Iterações	6.730	1	4.992	9.276
2000	Tempo (s)	0.0084	0.0390	0.0185	0.0201
	Iterações	6.874	1	3.704	7.992
10000	Tempo (s)	0.0437	0.1807	0.0903	0.1008
	Iterações	6.992	1	2.994	7
20000	Tempo (s)	0.0835	0.3544	0.1708	0.1982
	Iterações	6.998	1	2.902	7

Analisando a Tabela 3.15 observa-se que em todos as funções analisadas o tempo necessário para o modelo convergir aumenta com o aumento da dimensão da amostra. Por outro lado, a função `glm` apresenta menor tempo nas diferentes dimensões comparativamente às outras funções.

Em relação ao número de iterações que o modelo necessita para convergir, destaca-se a função `brglm` que necessita sempre de apenas 1 iteração independentemente da dimensão da amostra. A função `glm` apresenta pequenas alterações para cada dimensão tendo valores a variar entre 6.730 e o 6.998. Já as funções `glmrob` e `glmbayes` apresentam um comportamento idêntico: o número de iterações necessário para o modelo convergir diminui com o aumento da dimensão da amostra. No caso da função `glmrob` observa-se uma redução de 4.992

para 2.902 enquanto que na função `bayesglm` os valores reduzem de 9.276 para 7 iterações.

Os resultados do Modelo 2 e do Modelo 4 encontram-se no Anexo 1. (Tabela 1.1 até Tabela 1.4)

Em relação ao Modelo 2 observa-se que, tal como no Modelo 3, a função `bayesglm` é a que apresenta menor valores na estimativa do MSE. No entanto ao analisar o PRE, observa-se que as diferenças entre a função `glm` e as outras funções em estudo são muito maiores, obtendo-se valores de PRE de 224 %. Além disso, o MAPE no Modelo 2 comporta-se de forma semelhante ao MAPE no Modelo 3.

Os resultados do Modelo 4 estão em concordância com os Modelos 2 e 3 em relação ao valor das estimativas do MSE uma vez que também se destaca a função `bayesglm` como a que apresenta menores valores.

Numa análise geral do MAPE observa-se que, ao contrário dos outros modelos, o Modelo 4 apresenta menor MAPE quando se usa a função `brglm` nas amostras com dimensões 500, 2000 e 10000. Quando se analisa a amostra com dimensão 20000 a função que se destaca é a função `bayesglm`.

3.5 Previsão

Nesta secção pretende-se avaliar a capacidade preditiva do modelo em dados com diferentes dimensões e estudar diferentes subamostras, ou seja, estudar a capacidade preditiva do modelo em diferentes dimensões de subamostras da dimensão dos dados iniciais. Para este efeito definiram-se diferentes subamostras de dimensões a variar entre 10% e 100% em intervalos de 10%, ou seja, no total estudou-se para cada dimensão de dados 10 subamostras de dimensões diferentes.

Em cada simulação, foi necessário dividir os dados em dois conjuntos, conjunto de treino e conjunto de teste na proporção de 80% e 20%, respetivamente. Esta divisão nos dois conjuntos realiza-se através da função `createDataPartition()` do R que se encontra no *package caret* como descrito no Código 15.

```
1 library(caret)
2 divisão_dados <- createDataPartition(dados$y, p = 0.8, list = FALSE)
3 treino <- dados[divisão_dados, ]
4 teste <- dados[-divisão_dados, ]
```

Código 15: Divisão dos dados em dados de treino e dados de teste

Os dados de treino são utilizados para ajustar o modelo, enquanto que os dados de teste são utilizados para avaliar a capacidade preditiva do modelo.

Como referido, o objetivo é avaliar a capacidade preditiva do modelo em relação à dimensão inicial tendo em conta as diversas subamostras de diferentes dimensão dos dados totais e para tal usa-se a estatística SQR descrita na Secção 2.8.1. Como complemento ao estudo, calcula-se também o Coeficiente de Variação (CV), uma medida que mostra o quão variável uma medida é em relação à sua média. Este coeficiente, representa-se em percentagem

e é calculado dividindo o desvio padrão pela média e multiplicando por 100. Valores de CV mais baixos indicam menor variabilidade em relação à média, enquanto valores mais altos sugerem maior variabilidade.

$$CV_m(\%) = \frac{\textit{desvio padrão(m)}}{\textit{média(m)}} \times 100,$$

em que m é a medida em estudo.

O estudo simulação é aplicado em 500 repetições para que os dados sejam mais consistentes e usa-se a função `bayesglm()` por ser a que se revelou mais precisa na secção anterior.

O planeamento do estudo de simulação é o seguinte:

- Dimensão da amostra (n): 500, 2000, 10000 e 20000
- Número de simulações: 500
- Partição de dados de treino: 80%
- Partição de dados de teste: 20%
- Variáveis explicativas: $X_1 \sim N(0,1)$, $X_2 \sim N(0,1)$, $X_3 \sim U(0,1)$ e $X_4 \sim U(0,1)$
- Variável resposta:Y $\sim$Bernoulli (μ) (Modelo de Regressão Logística)

  Y $\sim$Poisson (λ) (Modelo de Regressão Poisson)
- Parâmetros do modelo:

  Modelo 1: $\beta_0 = 1.\beta_1 = 1.5, \beta_2 = 2, \beta_3 = -2.5$ e $\beta_4 = 3.$ (Modelo de Regressão Poisson)

  Modelo 2: $\beta_0 = -1.5$, $\beta_1 = 3.5$, $\beta_2 = -5.2$, $\beta_3 = -6.8$ e $\beta_4 = 0.2.$ (Modelo de Regressão Logística)

  Modelo 3: $\beta_0 = -1.5.\beta_1 = 4, \beta_2 = -1.2, \beta_3 = -0.8$ e $\beta_4 = 3.$ (Modelo de Regressão Logística)

  Modelo 4: $\beta_0 = 3.5.\beta_1 = -2.5, \beta_2 = 7.2, \beta_3 = -3.8$ e $\beta_4 = 6.2.$ (Modelo de Regressão Logística)
- Dimensão das subamostras: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% e 100% em relação à dimensão dos dados iniciais.
- Estatísticas calculadas:

  Acurácia, Sensibilidade, Especificidade, Precisão e CV (Modelo de Regressão Logística)

  SQR e CV (Modelo de Regressão Poisson)

Modelo de regressão Poisson

No Código 16 descreve-se o estudo de simulação para o modelo de regressão Poisson que consiste:

1. Definir as subamostras que se pretende estudar recorrendo à função `seq()`;

2. Definir o número de simulações a serem usadas;

3. Iniciar um ciclo *for* onde o número de iterações do ciclo varia entre 1 e número de subamostras, ou seja, 10;

4. Criar um ciclo interno *for*, onde o número de iterações do ciclo varia entre 1 até ao número de simulações. Dentro desse ciclo:

    - Criar amostras aleatórias sem reposição de dados de treino para a respetiva dimensão da subamostra.
    - Ajustar um modelo de regressão Poisson às subamostras de dados de treino das subamostras.
    - Determinar as previsões dos dados de teste com com base no modelo ajustado para os dados de treino usando a função `predict()`.
    - Calcular os valores da estatística SQR para cada simulação.

5. Após todas as repetições para uma determinada subamostra, armazenar os valores de SQR numa matriz *m_sqr_valores* na coluna correspondente e armazenar noutra matriz o valor do desvio padrão,que é calculado com o recurso à função `apply()`.

6. Após a conclusão do ciclo externo para todas as subamostras, os valores das estatísticas SQR e os desvios padrão por subamostra são armazenados em listas *sqr_values_n* e *desvios_padrao_n*, respetivamente.

```r
#Definição das partições e do número de repetições
subamostras<-seq(0.1, 1, by = 0.1)
rep<-500
sqr_valores<-NULL
m_sqr_valores<-matrix(nrow = rep, ncol = length(subamostras))
#Ciclo para calcular a estatística SQR nas diferentes subamostras
for (i in 1:length(subamostras)) {
  for (j in 1:rep) {
  #Criar amostras com  diferentes dimensões
  indices_aleatorios<-sample(nrow(treino), nrow(treino) * subamostras[i])
  dados_aleatorios<-treino[indices_aleatorios, ]
  #Estimar o modelo
  modelo<-glm(y~x1+x2+x3+x4, data=dados_aleatorios, family=poisson(link="log"))
  #Calcular as predições e a estatística
  predicao<-predict(modelo, newdata = teste, type = "response")
  sqr_valores[j]<-sum((((teste$y-predicao)/sqrt(predicao))^2),na.rm = T)/
  (nrow(teste)-5)
  }
```

```
  #Guardar a informação referente a cada repetição
  m_sqr_valores[,i]<-sqr_valores
  desvio_padrao_por_subamostras<-apply(m_sqr_values, 2, sd)/apply
  (m_sqr_values,2, mean)
 }
#Armazenar os resultados em uma lista separada por dimesões
sqr_valores_n[[as.character(n)]]<-m_pearson_valores
desvios_padrao_n[[as.character(n)]]<-desvio_padrao_por_subamostras
```

Código 16: Estimação do modelo de regressão Poisson (Modelo 1) e cálculo da medida de desempenho

Na Figura 9, apresentam-se os resultados da estatística SQR para diferentes dimensões dos dados. Cada gráfico apresenta uma dimensão e é composto por dez caixas com bigodes, correspondentes às 10 subamostras criadas de dimensão a variar entre 10% e 100% da amostra inicial. No eixo horizontal estão dispostas as diferentes subdivisões das amostras, enquanto que no eixo vertical são apresentados os valores correspondentes à estatística SQR.

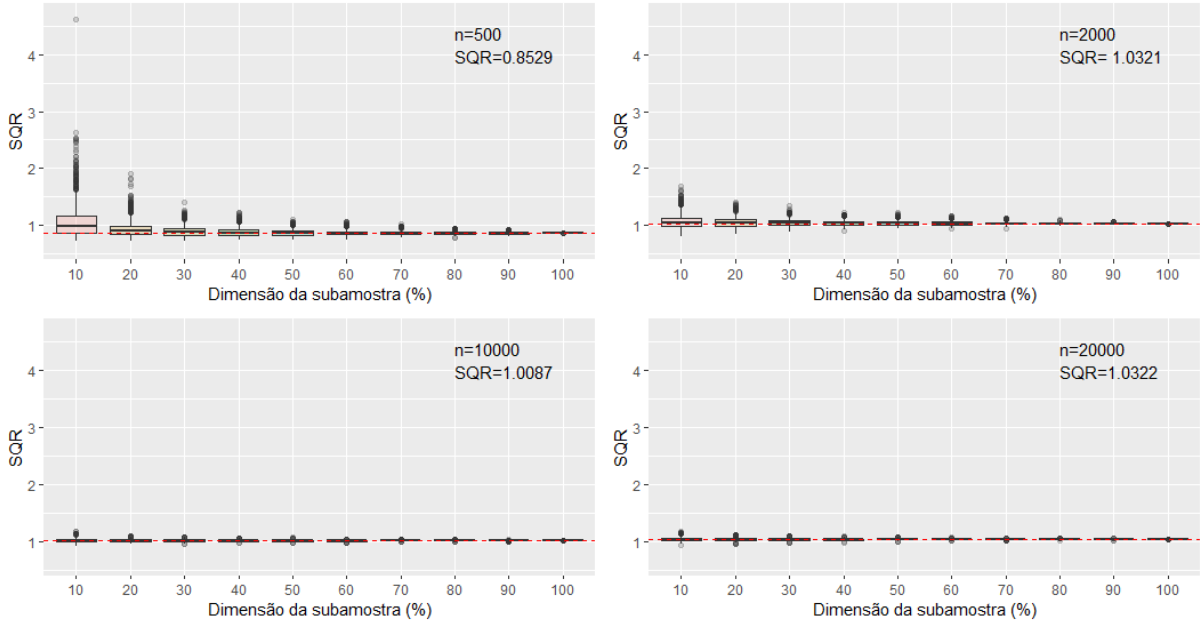

Figura 9: Caixas com bigodes da estatística SQR em diferentes dimensões da amostra

Na Figura 9, a linha tracejada a vermelho serve como referência para o valor da mediana de SQR quando é utilizada a totalidade dos dados da amostra. Esse valor é o indicado no canto superior direito de cada gráfico juntamente com a dimensão da amostra.

Analisando os quatro gráficos, é evidente uma tendência de redução na variabilidade à medida que a dimensão da amostra aumenta: realça-se uma expressiva diminuição na variabilidade para a amostra n=20000, comparativamente ao gráfico de n=500. Os gráficos ressaltam que, independentemente do tamanho inicial da amostra, a dispersão das estimativas tende a diminuir à medida que a análise incorpora um maior volume de

dados.

Adicionalmente, ao analisar individualmente cada gráfico, torna-se evidente que, à medida que a dimensão da subamostra aumenta, a variabilidade dos dados diminui. Isso enfatiza ainda mais a influência positiva do aumento da dimensão da amostra na estabilidade das estimativas. A Figura 9 é a Figura 10 quando se muda a escala do eixo y, de forma a ser percetível as diferenças que existem entre cada dimensão das subamostras.

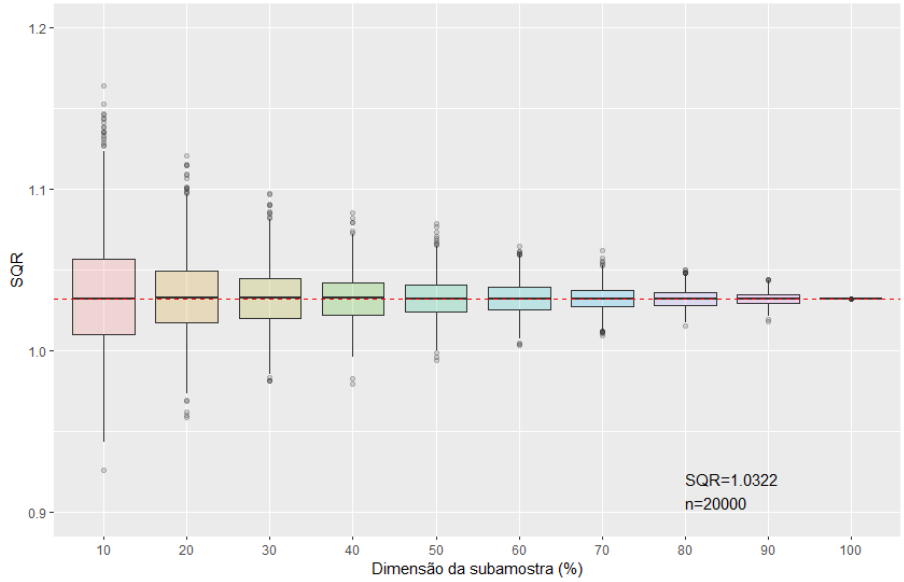

Figura 10: Caixa com bigodes da estatística SQR (n=20000)

Tabela 3.16: CV (%) da estatística SQR

Dimensão da subamostra	n=500	n=2000	n=10000	n=20000
10%	27.4509	11.6089	3.7038	3.4716
20%	15.0166	7.4769	2.4562	2.3240
30%	9.9297	5.7519	1.9500	1.8080
40%	8.1319	4.6004	1.5071	1.4211
50%	6.2739	3.7907	1.2356	1.1829
60%	5.1925	3.1294	1.0159	0.9702
70%	3.9917	2.5470	0.8024	0.7597
80%	3.1059	1.8341	0.6128	0.5673
90%	2.0881	1.2516	0.4091	0.3989
100%	0.0000	0.0000	0.0000	0.0000

A Tabela 3.16 sintetiza os valores do CV da estatística SQR para cada combinação da amostra e dimensão da subamostra. Ao observar os valores presentes na tabela, observa-se uma tendência decrescente dos valores de CV da estatística SQR à medida que as dimensões da amostra e da subamostra aumentam. Este padrão é coerente com as tendências observadas nos gráficos anteriores.

Os dados sugerem diferenças pequenas com valores CV menores de 1% para n=10000 e para n=20000 a partir de 70 % e 60% da subamostra, respetivamente. Este resultado mostra que, em situações com uma grande dimensão de dados, a estatística SQR é estável e a variabilidade é mínima.

No entanto, com amostras menores, como n=500 e n=2000, observamos uma variabilidade inicialmente

mais alta. Por exemplo, em n=500, o valor inicial do CV é 27% para apenas 10% da subamostra, mas diminui para menos de 5% quando mais de 60% da subamostra é considerada. No caso de n=2000, o valor inicial do CV é cerca de 12% e reduz para menos de 5% a partir dos 40% da subamostra.

Modelo de Regressão Logística

O estudo de simulação para a Regressão Logística é realizado de forma semelhante ao estudo de simulação para Regressão Poisson. No entanto, neste contexto a análise incide nas quatro medidas distintas: Acurácia, Sensibilidade, Precisão e Especificidade, conforme detalhadas na Secção 2.8.2. Essas medidas requerem a computação dos valores VP, FP, FN e VF, que são calculados através da previsão. De salientar que, considera-se como casos positivos os que apresentam um valor predito superior a 0.5.

No Código 17 descreve-se o estudo de simulação para o modelo de regressão Logística.

```
#Matrizes para guardar os resultados das medidas de desempenho
acuracia_valores<-matrix(nrow=rep, ncol=length(subamostras))
sensibilidade_valores<-matrix(nrow=rep, ncol=length(subamostras))
especificidade_valores<-matrix(nrow=rep, ncol=length(subamostras))
precisao_valores<-matrix(nrow=rep, ncol=length(subamostras))
variacao_acuracia_valores<-matrix(nrow=rep, ncol=length(subamostras))
for (i in 1:length(subamostras)) {
  for (j in 1:rep) {
   indices_aleatorios<-sample(nrow(treino), nrow(treino) * subamostras[i])
   dados_aleatorios<-treino[indices_aleatorios, ]
#Estimação do modelo
    modelo<-glm(y~x1+x2+x3+x4, data=dados_aleatorios, family=binomial(link="logit"))
    #previsão do modelo em 0 e 1
    predicao<-predict(modelo, newdata=teste, type="response")
    y_pred_binary<-ifelse(predicao>=0.5, 1, 0)
    #cálculo das observações VP, VN, FN, FP
    VP<-sum(teste$y==1 & y_pred_binary==1)
    VN<-sum(teste$y==0 & y_pred_binary==0)
    FN<-sum(teste$y==1 & y_pred_binary==0)
    FP<-sum(teste$y==0 & y_pred_binary==1)
#Cálculo das medidas de desempenho
          sensibilidade[j]<-TP / (TP + FN)
          especificidade[j]<-TN / (TN + FP)
          precisao[j]<-TP / (TP + FP)
          acuracia[j]<-(TP + TN) / (TP + TN + FP + FN)
        }
```

```r
#Guardar a informação de cada repetição
    acuracia_valores[, i]<-acuracia
    sensibilidade_valores[, i]<-sensibilidade
    especificidade_valores[, i]<-especificidade
    precisao_valores[, i]<-precisao
    variacao_acuracia_valores<-apply(acuracia_valores, 2, sd)/apply(acuracia_valores,
    2, mean)
  }
  #Armazenar os resultados em lista separada por dimensões
  acuracia_lista[[as.character(n)]]<-acuracia_valores
  sensibilidade_lista[[as.character(n)]]<-sensibilidade_valores
  especificidade_lista[[as.character(n)]]<-especificidade_valores
  precisao_lista[[as.character(n)]]<-precisao_valores
  variacao_lista[[as.character(n)]]<-variacao_acuracia_valores
```

Código 17: Estimação do modelo regressão Logística e cálculo das medidas de desempenho

A análise de resultados apresentada neste relatório é apenas referente à Acurácia, contudo o Código apresentado permite calcular todas as medidas de desempenho mencionadas.

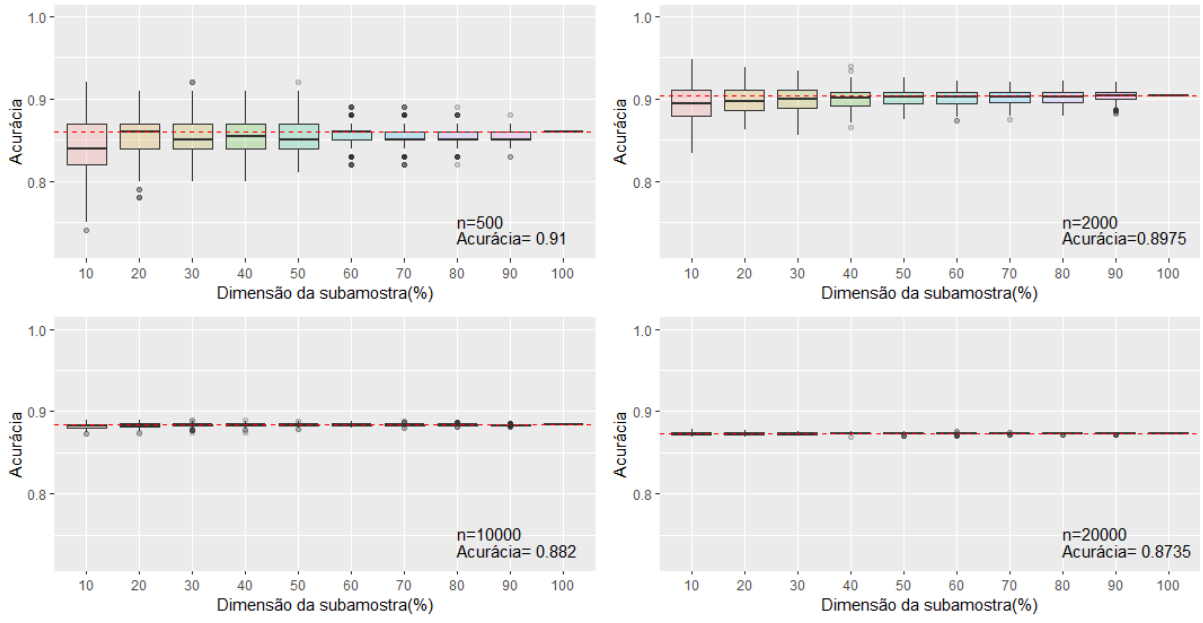

Figura 11: Caixas com bigodes da Acurácia em diferentes dimensões

Na Figura 11 a linha vermelha tracejada representa o valor mediano da Acurácia quando se utiliza a totalidade dos dados da amostra. Esse valor é indicado no canto inferior direito do gráfico em conjunto com a dimensão da amostra.

Ao analisar cada gráfico individualmente, observa-se que a variabilidade nas estimativas da Acurácia tende a diminuir à medida que se aumenta a dimensão das subamostras, ou seja, quantos mais dados são utilizados,

menor é a dispersão nas previsões, sugerindo uma maior estabilidade e consistência nas classificações obtidas pelo modelo. Além disso, quando se comparam os quatros gráficos em conjunto, constata-se, conforme esperado, que a variabilidade entre as diferentes caixas com bigodes também tende a diminuir à medida que a dimensão inicial da amostra aumenta.

Esse padrão reflete a ideia de que quanto maior a dimensão da amostra inicial menor variabilidade é observada nos resultados, indicando uma melhoria nas estimativas da Acurácia. A Figura 12 é o caso apresentado na Figura 11 quando se muda a escala no eixo do y ilustrando de forma clara esta conclusão quando n=20000.

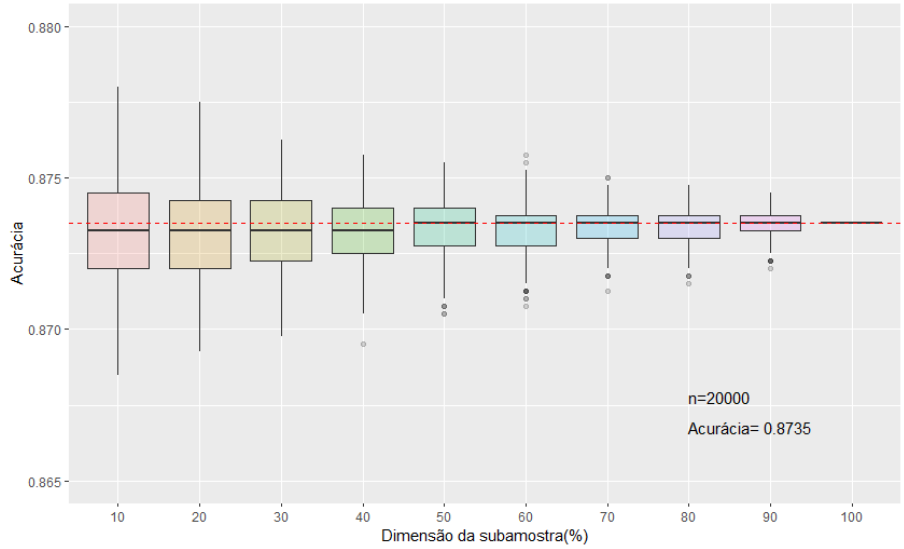

Figura 12: Caixa com bigodes da Acurácia (n=20000)

Tabela 3.17: CV (%) da Acurácia para diferentes tamanhos da amostra

Dimensão da subamostra	n=500	n=2000	n=10000	n=20000
10%	3.6960	1.2560	0.3662	0.1970
20%	2.8459	0.8177	0.3120	0.1681
30%	2.5308	0.7083	0.2459	0.1384
40%	2.2045	0.6173	0.2144	0.1197
50%	1.9416	0.5739	0.1833	0.1050
60%	1.6168	0.5261	0.1546	0.0942
70%	1.4648	0.4622	0.1309	0.0691
80%	1.1566	0.4402	0.1153	0.0622
90%	0.8318	0.4069	0.0809	0.0457
100%	0.0000	0.0000	0.0000	0.0000

A Tabela 3.17 resume os valores do CV para cada combinação de dimensão de amostra e subamostra, fornecendo uma visão quantitativa da variabilidade associada a cada configuração.

Ao observar os valores do CV, é possível notar uma concordância notável entre as tendências evidenciadas nos gráficos e os resultados da medida. Em geral, observa-se que à medida que se aumenta o tamanho da amostra e a subamostra, os valores do CV diminuem, indicando uma redução na dispersão das estimativas de Acurácia.

Isso está em consonância com a conclusão anterior, que sugere que uma maior quantidade de dados resulta em previsões mais consistentes e confiáveis.

Examinando os valores de CV com mais detalhes, fica evidente que, para qualquer subamostra considerada em qualquer dimensão, os resultados são consistentemente inferiores a 4%, o que sugere que a Acurácia apresenta pouca variabilidade em relação à sua média.

Notavelmente, para as dimensões n=10000 e n=20000, os CV são, em geral, menores de 1%. No caso de n=2000, os valores do CV são inferiores a 1% a partir de 20% da subamostra, enquanto que, para n=500, é necessário considerar mais de 80% da amostra para obter resultados com CV menores do que 1%.

Na Tabela 3.18 apresenta-se o resumo das quatro medidas (Acurácia, Sensibilidade, Previsão e Especificidade) para os diferentes modelos, em diferentes dimensões da amostra.

Tabela 3.18: Medidas de desempenho para os modelos em diferentes dimensões

	n	Acurácia	Sensibilidade	Especificidade	Precisão
Modelo 2	500	0.9500	0.9200	0.9600	0.8846
	2000	0.9375	0.8511	0.9641	0.8791
	10000	0.9275	0.8468	0.9551	0.8655
	20000	0.93275	0.8446	0.9596	0.8640
Modelo 3	500	0.8600	0.8889	0.8364	0.8163
	2000	0.8925	0.8812	0.9040	0.9036
	10000	0.8835	0.8733	0.8930	0.8834
	20000	0.8735	0.8530	0.8914	0.8727
Modelo 4	500	0.9700	0.9867	0.9200	0.9737
	2000	0.9650	0.9799	0.9208	0.9734
	10000	0.9445	0.9631	0.8970	0.9598
	20000	0.94675	0.9665	0.8966	0.9595

No que diz respeito ao Modelo 2, observa-se que, de uma maneira geral, à medida que a dimensão da amostra aumenta, a Acurácia e a Sensibilidade tendem a diminuir, embora mantenham valores elevados. A Especificidade do Modelo 2 tende a permanecer alta em todos os cenários, indicando que o modelo é consistente na identificação de verdadeiros negativos, o que é coerente, dado que o número de observações da variável resposta do modelo contém mais casos de insucesso que casos de sucesso.

No Modelo 3 à medida que a dimensão da amostra aumenta, a Acurácia, a Especificidade e a Precisão tendem a aumentar. Realça-se que ao comparar o Modelo 3 com os Modelos 2 e 4, é o Modelo 3 que apresenta valores mais baixos nas quatro medidas em estudo.

O Modelo 4 exibe um desempenho sólido em todos os cenários, com alta Acurácia, Sensibilidade e Precisão. À medida que a dimensão da amostra aumenta, as quatro medidas tendem a diminuir. Como esperado, a Sensibilidade permanece alta em todos os cenários, indicando que o modelo é consistente na identificação de verdadeiros positivos. A precisão varia, mas geralmente permanece em valores elevados, sugerindo que o Modelo 3 faz previsões precisas.

4 Caso de Estudo

Este capítulo tem como objetivo a aplicação de Modelos Lineares Generalizados (MLG) a uma base de microdados reais. A base de microdados em análise foi submetida a um processo de anonimização, no qual identificadores diretos foram removidos e os números de identificação das entidades foram substituídos por combinações aleatórias de dígitos, garantindo a privacidade e a confidencialidade dos dados.

4.1 Descrição da Base de Microdados

A base de microdados Painel Harmonizado da Central de Balanços (CBHP) é construída com base nas informações da Central de Balanços (CB) e disponibiliza informação económica e financeira sobre as sociedades não financeiras portuguesas (Banco de Portugal Microdata Research Laboratory (BPLIM), 2023). Esta base de dados está organizada em cinco módulos: Rosto, Contas, Pessoal, AMarc (acontecimentos marcantes) e mg (mercados geográficos). Para cada módulo é disponibilizado um ficheiro em que cada linha corresponde a uma empresa num determinado ano. Os dados correspondem ao período compreendido entre 2006 e 2021, no caso em estudo iremos usar apenas o ano 2021.

No âmbito deste estudo serão usadas as variáveis presentes no módulo de rosto que contém a caracterização das empresas e informação sobre a declaração fiscal, as variáveis B001, B061, B080, B025 e E001 do módulo Contas e a variável D001 do módulo Pessoal.

De seguida, descrevem-se as variáveis:

- *tina*: número de identificação fiscal anonimizado da entidade;
- *ano*: ano de referência dos dados;
- *dataintrib*: data de início do ano fiscal;
- *datafimtrib*: data de encerramento do ano fiscal;
- *numdias*: número total de dias do ano fiscal;
- *planocont*: sistema de contabilidade segundo o qual a empresa reporta;

Tabela 4.19: Categorias da variável *planocont*

Categoria	Significado
0	POC - Plano Oficial de Contabilidade
1	SNC - Sistema de Normalização Contabilística

- *regime*: com a introdução do SNC, as empresa devem selecionar os seus relatórios padrão. Esta variável só está disponível a partir de 2009 uma vez que foi o ano em que o SNC foi introduzido;

Tabela 4.20: Categorias da variável *regime*

Categoria	Significado
-1	Não especificado (N/E)
1	Normas Internacionais de Contabilidade (NIC's)
2	Normas Contabilísticas e de Relato Financeiro (NCRF's)
3	Norma Contabilística e de Relato Financeiro para Pequenas Entidades (NCRF-PE)
4	Norma Contabilística para Microentidades (NC-ME)

Com base nesta variável criou-se a variável *normasint*.

- *normasint*: variável categórica que assume valor 1 se a empresa segue as normas internacionais de Contabilidade e 0 caso contrário. Os casos não especificados são considerados como valores em falta;
- *motivodec*: indica o motivo do envio da declaração;

Tabela 4.21: Categorias da variável *motivodec*

Categoria	Significado
0	Normal
1	Consolidação
2	Período de cessação
3	Período especial de tributação – antes alteração
4	Período especial de tributação – após alteração
5	Exercício do início da tributação

- *numestabnac*: número de estabelecimentos da empresa em território nacional;
- *numestabest*: número de estabelecimentos da empresa fora do território nacional;
- *numestab*: número de estabelecimentos da empresa;
- *sitempresa*: indica a situação da empresa no final do período fiscal;

Tabela 4.22: Categorias da variável *sitempresa*

Categoria	Significado
1	Em atividade
2	Fim de atividade
3	Dissolvida
4	Liquidada

- *datasitempresa*: data de referência da situação da empresa, esta informação é necessária caso a empresa reporte os códigos 2, 3 e 4 na variável *sitempresa*. Com base nesta variável criou-se a variável *liquidacao*.
- *liquidacao*: variável categórica que assume valor 1 quando a empresa se encontra em liquidação e 0 caso contrário;

- *pervvn*: indica a proporção do volume de negócios que a atividade económica principal representa entre todas as atividades desenvolvidas pela empresa;
- *distrito*: distrito onde a empresa está localizada. Com esta variável criou-se a variável *continente*.
- *continente* : variável categórica que assume valor 1 quando o distrito pertence a Portugal continental ou valor 0 caso contrário;
- *sucursal*: variável categórica que assume valor 1 para sucursais de empresas estrangeiras localizadas em Portugal e o valor 0 caso contrário;
- *exporta*: variável categórica que assume valor 1 se a empresa exporta ou valor 0 caso contrário;

Tabela 4.23: Categorias da variável *exporta*

Categoria	Significado
0	Não exporta
1	Exporta para o Mercado Comunitário
	Exporta para o Mercado Extra-Comunitário
	Exporta para os Mercados Comunitário e Extra-Comunitário

- *unipessoal*: variável categórica que assume valor 1 para sociedades sob a forma jurídica "Sociedade Unipessoal por Quotas" e 0 caso contrário;
- *sectorinstfinal*: informa o setor institucional ao qual a empresa pertence;
- *indactiecon*: reporta a situação da empresa revista pelo Departamento de Estatística do Banco de Portugal.

Tabela 4.24: Categorias da variável *indactiecon*

Categoria	Significado
0	Ignorado / Desconhecido
10	Aguarda início de atividade
20	Em atividade
30	Atividade suspensa
40	Cessão de atividade
97	Inválido
98	Não especificado

- *dimcomissao*: classificação das empresas de acordo com quatro categorias seguindo a Recomendação da Comissão 2003/361/CE12:

Tabela 4.25: Categorias da variável *dimcomissao*

Categoria	Significado
1	Microempresas
2	Pequenas empresas
3	Médias empresas
4	Grandes empresas

- *ancon*: corresponde ao ano de início da empresa de acordo com o Indicador da Atividade Económica. Com base nesta variável criou-se a variável *antiguidade*.
- *antiguidade*: indica o número de anos da empresa;
- *natju* : variável que reporta a forma jurídica da empresa;
- *cae3* e *cae21*: reportam o principal setor de atividade da empresa pela Rev. 3 e Rev. 2.1, respetivamente. A lista completa de código por ser consultada em (Instituto Nacional de Estatística., 2007).
- *caekotu* : variável categórica que assume valor 1 se a atividade económica principal da empresa estiver inserida numa das seguintes secções: K, O, T ou U.
- *E001* : número de pessoas ao serviço da empresa (remuneradas e não remuneradas);
- *B001* : total do ativo;
- *B061* : capital próprio;
- *B080* : total do passivo;
- *B025* : investimentos financeiros;
- *D001* : vendas e serviços prestados.

Neste estudo, pretende-se estimar um modelo de Regressão Logística para investigar a relação entre as variáveis apresentadas e a variável resposta *exporta*. O objetivo principal é identificar as variáveis que influenciam a probabilidade de uma empresa ser exportadora.

Para tal, optou-se por restringir a base de microdados original a uma amostra de menor dimensão. Foram mantidos os dados do ano mais recente (2021) e selecionaram-se apenas as pequenas e médias empresas (categorias 2 e 3 da variável *dimcomissao*). Dessa forma, a base de microdados em estudo é composta por 49900 observações e 26 variáveis.

4.2 Modelo de Regressão Logística

A estimação do modelo de regressão logística inicia-se com o ajuste de uma regressão logística simples considerando cada uma das variáveis explicativas que podem ter impacto no modelo. O objetivo desta análise é verificar a influência que cada variável explicativa tem na variável resposta.

Dada a presença de disparidades nas escalas de valores das variáveis numéricas, optou-se por realizar a padronização das mesmas usando a função `scale()` do R. Esta função ajusta os valores de cada variável, subtraindo a média da variável e dividindo pelo desvio padrão da mesma.

Na Tabela 4.26 apresenta-se os resultados do ajuste de cada regressão logística simples.

Tabela 4.26: Estimativas do modelo de regressão logística simples para cada uma das variáveis explicativas

Variáveis explicativas	Estimativa dos coeficientes	Erro padrão	Teste Wald	Valor-p
Constante	-0.3956	0.0092	-43.00	$<2\times10^{-16}$
antiguidade	0.2548	0.0093	27.45	$\mathbf{<2\times10^{-16}}$
Constante	-0.3914	0.0091	-42.88	$<2\times10^{-16}$
pervvn	-0.0052	0.0090	-4.45	$\mathbf{<9\times10^{-6}}$
Constante	-0.3913	0.0091	-42.88	$<2\times10^{-16}$
numestab	0.0117	0.0090	1.29	0.1960
Constante	-0.3892	0.0091	-42.51	$<2\times10^{-16}$
numestabest	0.1657	0.0261	6.3500	$\mathbf{<3\times10^{-10}}$
Constante	-0.3913	0.0091	-42.88	$<2\times10^{-16}$
numestabnac	0.0093	0.0090	1.03	0.3000
Constante	-0.3910	0.0091	-42.85	$<2\times10^{-16}$
caekootu1	-11.1750	69.6386	-0.16	0.8730
Constante	-0.5424	0.0100	-54.11	$<2\times10^{-16}$
dimcomissao3	1.0297	0.0264	38.95	$\mathbf{<2\times10^{-16}}$
Constante	-0.3122	0.0101	-30.84	$<2\times10^{-16}$
unipessoal1	-0.4140	0.0237	-17.47	$\mathbf{<2\times10^{-16}}$
Constante	-0.3905	0.0093	-42.00	$<2\times10^{-16}$
E001	0.4076	0.0106	38.29	$\mathbf{<2\times10^{-16}}$
Constante	-0.3675	0.0094	-38.90	$<2\times10^{-16}$
D001	0.6655	0.0167	39.90	$\mathbf{<2\times10^{-16}}$
Constante	-0.3908	0.0091	-42.81	$<2\times10^{-16}$
B061	0.0912	0.0213	4.29	$\mathbf{<2\times10^{-5}}$
Constante	-0.3908	0.0091	-42.80	$<2\times10^{-16}$
B001	0.0926	0.0202	4.58	$\mathbf{<5\times10^{-6}}$
Constante	-0.3911	0.0091	-42.86	$<2\times10^{-16}$
B080	0.0500	0.0159	3.14	**0.0017**
Constante	-0.3913	0.0091	-42.88	$<2\times10^{-16}$
B025	0.0183	0.0111	1.65	0.1000
Constante	-0.3905	0.0092	-42.52	$<2\times10^{-16}$
normasint1	-0.0589	0.0811	-0.73	0.4680
Constante	-1.7271	0.0605	-28.56	$<2\times10^{-16}$
continente1	1.3824	0.0612	22.59	$\mathbf{<2\times10^{-16}}$

Pela análise da Tabela 4.26 observa-se que as variáveis *numestab*, *numestabnac*, *caekotu1*, *B025* e *normasint1* não são estatisticamente significativas (valor-p>0.05).

A construção do modelo inicia-se com as variáveis consideradas significadas como as variáveis explicativas. Desta forma como a variável resposta do modelo segue uma distribuição Bernoulli de parâmetro p, tem-se:

$$exporta \sim Bernoulli(p)$$

$$log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \times antiguidade + \beta_2 \times pervvn + \beta_3 \times numestaest + \beta_4 \times dimcomissao + \beta_5 \times unipessoal + \beta_6 \times E001 + \beta_7 \times D001 + \beta_8 \times B061 + \beta_9 \times B001 + \beta_{10} \times B080 + \beta_{11} \times continente.$$

Aplicaram-se os métodos de seleção sequenciais *forward, backward* e *stepwise* recorrendo à função `step()` do R. Baseados no critério AIC, verifica-se que o modelo final é o mesmo para os três métodos de seleção apresentados. Assim, o modelo escolhido para o estudo tem como equação:

$$exporta \sim Bernoulli(p)$$

$$log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \times antiguidade + \beta_2 \times pervvn + \beta_3 \times numestaest + \beta_4 \times unipessoal + \beta_5 \times E001 + \beta_6 \times D001 + \beta_7 \times B061 + \beta_8 \times B001 + \beta_9 \times continente.$$

Depois do ajustamento do modelo, procede-se à avaliação da qualidade de ajustamento. O teste de Wald e o teste de razão de Verosimilhança são os métodos utilizados para avaliar a qualidade de ajustamento do modelo.

Tabela 4.27: Estimativas do modelo final de regressão logística

Variáveis explicativas	Estimativa dos coeficientes	*Odds ratio*	Intervalo de confiança	Erro padrão	Teste Wald	Valor-p
Constante	-1.7609	0.1719	(0.1515, 0.1943)	0.0635	-27.74	$<2\times10^{-16}$
antiguidade	0.1622	1.1761	(1.1531,1.1996)	0.0101	16.08	$<2\times10^{-16}$
numestabest	0.1440	1.1549	(1.0994,1.2271)	0.0276	5.22	$<2\times10^{-7}$
unipessoal1	-0.1779	0.8370	(0.7961, 0.8798)	0.0255	-6.98	$<3\times10^{-12}$
E001	0.2317	1.2607	(1.2320, 1.2904)	0.0118	19.60	$<2\times10^{-16}$
D001	0.4735	1.6056	(1.5512,1.6632)	0.0178	26.67	$<2\times10^{-16}$
B061	0.1003	1.1055	(1.0442, 1.1809)	0.0313	3.20	0.0014
B001	-0.1365	0.8724	(0.8063, 0.9347)	0.0377	-3.62	0.0003
continente1	1.4685	4.3427	(3.8368, 4.9320)	0.0640	22.93	$<2\times10^{-16}$

De acordo com os valor-p do Teste de Wald apresentados na Tabela 4.27, existe evidência estatística de que os coeficientes estimados são todos estatisticamente significativos ao nível de significância de 1%.

O teste de Verosimilhança indica que o modelo de ajusta bem aos dados ($\xi_{RV} = 4069.8$, valor-p < 0.01).

A análise de resíduos permite verificar a qualidade de ajustamento do modelo. Na Figura 13 apresenta-se o gráfico dos resíduos do desvio do modelo *vs* valores ajustados do modelo.

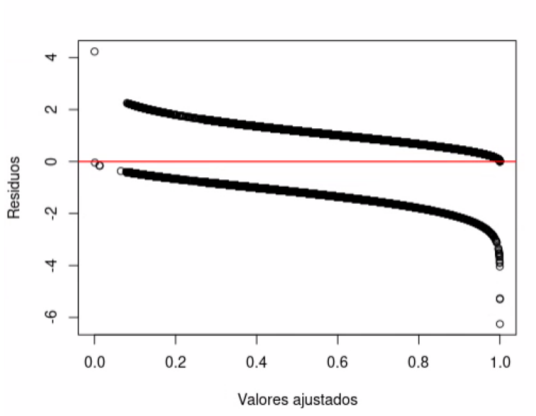

Figura 13: Residuos *vs* valores ajustados do modelo de regressão Logística ajustado

Observa-se pela análise da Figura 13 presença de várias observações *outliers*, que podem ter um impacto significativa no modelo. Posto isto, prossegue-se com um estudo detalhado para identificar as observações consideradas *outliers*, influentes ou alavanca. A Figura 14 apresenta os seguintes gráficos: resíduos do modelo *vs* índice das observações, o gráfico dos pontos alavanca *vs* o índice das observações e distância de Cook *vs* índice das observações.

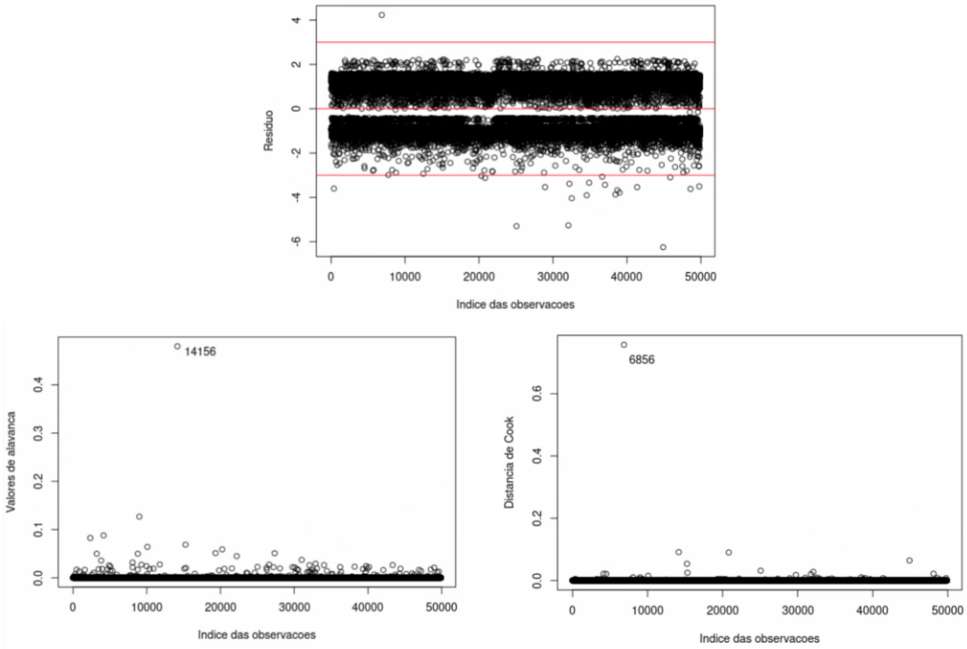

Figura 14: Análise de diagnóstico para o modelo de regressão Logística ajustado

No que diz respeito ao gráfico de resíduos do modelo, observou-se a presença de 21 observações com valores de resíduos fora do intervalo [-3, 3]. Portanto, classificam-se essas 21 observações como *outliers*. Quanto ao gráfico dos valores de alavanca, uma análise visual revela uma observação que se destaca, a observação de número 14156. No gráfico da Distância de Cook, destaca-se a observação 6856 com uma distância superior a 0.6.

De forma a perceber o impacto que estas observações tem na qualidade do modelo, ajusta-se o modelo sem

as mesmas. Como resultado, não se observaram diferenças nas medidas da qualidade de ajustamento do modelo, o que leva a prosseguir o estudo com o mesmo modelo.

Interpretação do modelo

Analisando os coeficientes das diferentes variáveis explicativas é possível detetar de forma mais pormenorizada a influência que estas têm no modelo.

Um coeficiente positivo, como o valor de 0.1588 na variável *antiguidade*, sugere que, à medida que a antiguidade da empresa aumenta, a probabilidade de a empresa exportar também aumenta. Para um aumento de uma unidade no desvio padrão da variável *antiguidade* as chances da empresa exportar aumenta cerca de 1.17 vezes (OR=1.1761 e $IC_{95\%}$=(1.1531,1.1996)).

A variável *numestaest* apresenta um coeficiente positivo o que sugere que o aumento de uma unidade no desvio padrão nessa variável implica um aumento de 1.15 na chance de uma empresa ser exportadora (OR=1.1549 e $IC_{95\%}$=(1.0994,1.2271)).

O *odds* para o sucesso num empresa *unipessoal* é aproximadamente 16.3% menor do que o *odds* para o sucesso em empresas não unipessoais (OR=0.8370 e $IC_{95\%}$=(0.7961,0.8798)).

As variáveis *E001*, *D001* e *B061* apresentam ambas um coeficiente positivo o que sugere que o aumento de cada uma delas aumenta a *odds* da empresa exportar. Para um aumento de uma unidade no desvio padrão das variáveis a chance da empresa exportar aumenta 1.26, 1.61 e 1.11 vezes, respetivamente e os intervalos de confiança para OR são (1.2320, 1.2904), (1.5512,1.6632) e (1.0442, 1.1809), respetivamente . Pelo contrário a variável *B001* apresenta um coeficiente negativo, o que significa que o aumento dessa variável diminui a *odds* da empresa exportar, ou seja, com o aumento de uma unidade no desvio padrão do total do ativo a chance da empresa exportar reduz 12,76% (OR=0.8724 e $IC_{95\%}$=(0.8062,0.9347)).

Para empresas portuguesas sediadas em Portugal continental, o *odds* para o sucesso é 4.34 vezes o *odds* para o sucesso em empresas não sediadas em Portugal continental (OR=4.3427 e $IC_{95\%}$=(3.8368, 4.9320)) .

Avaliação preditiva do modelo

Procede-se à avaliação da capacidade preditiva do modelo com o objetivo de avaliar a sua eficácia na classificação de empresas como exportadoras ou não exportadoras, tendo-se em consideração as variáveis explicativas.

Dado que a variável de interesse em análise é binária, a avaliação é realizada recorrendo a uma matriz de confusão. Essa matriz resume as previsões do modelo com base nos dados de teste, possibilitando a avaliação do desempenho do modelo na classificação desses dados.

Tabela 4.28: Matriz de Confusão do modelo final de regressão Logística

		Valor Real	
		Positivo	Negativo
Valor Previsto	Positivo	1021	467
	Negativo	3005	5487

Através da Tabela 4.28 é possível estimar os valores das medidas de desempenho. Verifica-se que a Acurácia, ou seja, a proporção de observações corretamente classificados foi de 65.21% (1021+5482/1021+467+3005+5487). A Sensibilidade e a Especificidade apresentam valores de 25.36% e 92.16% respetivamente, o que sugere que o modelo tem dificuldade em avaliar verdadeiros positivos, ou seja, empresas que exportam. Por outro lado, raramente classifica mal os casos negativos, ou seja, empresas que não exportam. Esta análise era esperada uma vez que nos dados há um maior número de empresas que não exportam relativamente ao número de casos de empresas que exportam. Em relação à Precisão, obtém-se um valor de 68.62%, o que corresponde à percentagem de previsões positivas feitas pelo modelo que são realmente positivas.

Como complemento ao estudo decidiu-se estudar como se comportam as medidas de desempenhos em diferentes subamostras dos dados totais. Para tal construí-se a Tabela 4.29 que apresenta de forma detalha o valor das diferentes medidas: Acurácia, Sensibilidade, Especificidade e Previsão nas diferentes dimensões da subamostra.

Tabela 4.29: Medidas de desempenho do modelo final de regressão Logística

Dimensão	10%	20%	30 %	40 %	50%	60 %	70 %	80 %	90%	100 %
Acurácia	0.6540	0.6544	0.6547	0.6549	0.6551	0.6552	0.6554	0.6555	0.6556	0.6557
Sensibilidade	0.2640	0.2607	0.2605	0.2597	0.2601	0.2595	0.2598	0.2598	0.2600	0.2598
Especificidade	0.9177	0.9206	0.9212	0.9221	0.9222	0.9227	0.9228	0.92299	0.9230	0.9234
Precisão	0.6853	0.6899	0.6911	0.6928	0.6933	0.6942	0.6948	0.6953	0.6963	0.6964

Pela análise da Tabela 4.29 observa-se que, a Acurácia do modelo varia de 0.6540 a 0.6557 à medida que se aumenta a dimensão das subamostras.

A Sensibilidade, que mede a capacidade do modelo de identificar corretamente os casos positivos reais, varia de 0.2640 a 0.2598, o que sugere que o modelo diminui ligeiramente a capacidade de detetar casos negativos com o aumento da dimensão das subamostras.

A Especificidade e a Precisão aumentam à medida que se aumenta a dimensão das subamostras, variando de 0.9177 para 0.9234 e de 0.6853 para 0.6964, respetivamente.

5 Conclusão e Trabalho Futuro

Os Modelos Lineares Generalizados (MLG) desempenham um papel de destaque em estudos estatísticos devido à sua capacidade de se adaptarem a diversos tipos de dados, mesmo quando estes não seguem uma distribuição normal. Uma representação eficaz dos dados envolve a seleção criteriosa das variáveis explicativas e a introdução de uma função de ligação adequada.

Neste trabalho começou-se por estudar a metodologia dos MLG. Um critério fulcral para o estudo destes modelos é a variável resposta pertencer à família exponencial, critério este que foi abordado na apresentação teórica destes modelos. Descreveram-se também as diferentes componentes presentes no modelo, o processo de estimação e algumas medidas para avaliar a qualidade do ajustamento.

Para estudar e compreender melhor estes modelos analisaram-se de forma mais detalha dois exemplos muito usados de MLG: Modelo de Regressão Poisson que é muito aplicado em dados de contagem e o Modelo de Regressão Logística que é aplicado em dados categóricos.

A grande maioria dos investigadores do BPLIM tem interesse na utilização de processos de estimação e predição, com recurso a MLG. As bases de microdados disponibilizadas pelo BPLIM são, em geral, de grande dimensão e o uso de ferramentas adequadas na sua análise é fulcral para os resultados de investigação.

Nesse sentido, o estudo realizado nesta Dissertação de Mestrado pretendeu aplicar e comparar MLG em diferentes linguagens de programação: R, Stata e Python.

Numa primeira abordagem, foram estudadas as diferentes metodologias para estimar MLG usando diferentes linguagens de programação. Através do acesso remoto ao servidor externo do BPLIM foi possível estimar em cada linguagem um MLG com as mesmas características e comparar diferentes parâmetros tais como, os coeficientes do modelo, as previsões geradas pelo mesmo, o tempo necessário para estimar o modelo e o número de iterações requeridas até convergir.

Nesta análise inicial concluiu-se que não existem diferenças nas estimativas dos coeficientes do modelo obtidas com diferentes linguagens de programação. Ainda em relação à estimação dos modelos, constatou-se que o Python apresentou menores tempos de execução.

Tendo em conta a homogeneidade de resultados nas diferentes linguagens decidiu-se aprofundar o estudo de MLG na linguagem R. Neste sentido recorreu-se a dois estudos de simulação. O primeiro estudo teve como objetivo avaliar e comparar a estimação de parâmetros do modelo usando diferentes funções do R, enquanto que, o segundo estudo focou-se nos resultados de previsão do modelo.

Para estes estudos de simulação foi necessário criar bases de dados com diferentes dimensões e distribuições: Poisson e Bernoulli balanceada e não balanceada.

No primeiro exploram-se diferentes *packages* do R, tendo sido as funções `glm`, `brglm`, `glmrob` e `bayesglm` as utilizadas. Para cada função foram analisadas diversas medidas, tais como MSE, MAPE e PRE. A função *bayesglm* destacou-se por apresentar menor valor nas diversas medidas comparativamente com as outras

funções nos 4 modelos em estudo.

No segundo estudo, observou-se, como seria de esperar, que à medida que a quantidade de dados iniciais aumenta, a variabilidade nas medidas de desempenho diminui. Além disso, observou-se que a variabilidade das medidas também diminui à medida que a dimensão da subamostra aumenta dentro de cada dimensão.

Por fim, usou-se uma base de microdados fornecida pelo BPLIM, a base Painel Harmonizado da Central de Balanços e estimou-se um MLG com o objetivo de perceber que variáveis conseguiam explicar se uma empresa exporta ou não. Restringiu-se a base de microdados para o ano 2021 e consideraram-se as pequenas e médias empresas. Estimou-se um MLG com variável resposta binária, que assume 1 quando a empresa exporta e 0 caso contrário, com família de distribuição Bernoulli e função de ligação *logit*. Depois de aplicar diferentes metodologias definiu-se um modelo onde se considerou as variáveis explicativas mais relevantes, obtendo-se um modelo com uma Acurácia de aproximadamente 65%. Na análise das variáveis deste modelo foram duas as variáveis que se destacaram positivamente, a relativa ao número de estabelecimentos e a que identifica as empresas sediadas em território continental, que revelaram aumentar as chances de exportação de uma empresa. Por outro lado, as variáveis *unipessoal* e o total do ativo (*B001*) apresentaram um coeficiente negativo o que sugere uma diminuição chance da empresa exportar. No final foram também analisadas as medidas de desempenho do modelo.

Neste estudo abordaram-se três linguagens de programação, no entanto o estudo acabou por se focar na aplicação das metodologias com a linguagem de programação R. Assim, num trabalho futuro seria interessante desenvolver uma análise comparativa e mais aprofundada nas linguagens de programação Stata e Python, que tal como o R, também dispõem de uma diversidade de funções para estimar MLG.

Além das linguagens abordadas, uma direção promissora seria a extensão deste estudo para incluir o *software* Julia. Julia é uma linguagem de programação que tem ganho destaque na análise de dados e estatística. Explorar como o Julia estima MLG e compará-lo com os resultados obtidos, poderia enriquecer esta pesquisa.

No que diz respeito ao caso em estudo e ao modelo que pretende explicar os fatores preponderantes para o mercado da exportação, poderia ser interessante cruzar o Painel Harmonizado da Central de Balanços com outras bases de microdados para encontrar novas variáveis explicativas e melhorar a qualidade preditiva do modelo.

Bibliografia

Agresti, A. (2013). *Categorial data analysis* (3° ed.). John Wiley & Sons, Inc.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716-723. doi: 10.1109/TAC.1974.1100705

Banco de Portugal Microdata Research Laboratory (BPLIM). (2023). *Central balance sheet - harmonized panel (cbhp)*. ("Dataset". Em: BANCO DE PORTUGAL) doi: https://doi.org/10.17900/SI.Apr2021.V1

Barndorff-Nielsen, O. E. (1978). *Information and exponential families in statistical theory*. Wiley, New York.

Berk, R., & MacDonald, J. M. (2008). Overdispersion and poisson regression. *J Quant Criminol*, *24*, 269–284.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, *26*(2), 211–252.

Chambers, J. M., Hastie, T. J., & Pregibon, D. (1990). *Statistical models in s*. Compstat. Physica-Verlag HD. (ed. Momirović, K., Mildner, V.) doi: https://doi.org/10.1007/978-3-642-50096-1_48

Chatterjee, S., & Hadi, A. S. (2012). *Regression analysis by example* (5° ed.). 5th Edition. Wiley, New York.

Davison, A. C. (2003). *Statistical models*. Cambridge University Press. doi: 10.1017/CBO9780511815850

Dunn, P. K., & Smyth, G. K. (2017). *Generalized linear models with examples in r*. Springer Texts in Statistics.

Eker, M., Poudel, K., & Özçelik, R. (2017, Dec). Aboveground biomass equations for small trees of brutian pine in turkey to facilitate harvesting and management. *Forests*, *8*(12), 477. (`http://dx.doi.org/10.3390/f8120477`) doi: 10.3390/f8120477

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, *80*, 27-38. doi: https://doi.org/10.1093/biomet/80.1.27

Fisher, R. A. (1922). On the interpretation of x2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, *85*, 87-94. doi: https://doi.org/10.2307/2340521

Fox, J. (2016). *Applied regression analysis and generalized linear models*. Sage Publications, Inc.

Gelman, A., Su, Y.-S., Yajima, M., Hill, J., Pittau, M. G., Kerman, J., ... Dorie, V. (2022). *Firth's bias-reduced logistic*

regression. (https://cran.r-project.org/package=arm)

Gilberto, P. A. (2004). *Modelos de regressão: com apoio computacional.* São Paulo: IME-USP.

Hardin, J. W., & Hilbe, J. M. (2012). *Generalized linear models and extensions* (3° ed.). Stata Press.

Heinze, G., Ploner, M., Dunkler, D., Southworth, H., & Jiricka, L. (2022). *Firth's bias-reduced logistic regression.* (https://cran.r-project.org/package=logistf)

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression.* John Wiley & Sons, Inc.

Instituto Nacional de Estatística. (2007). *Classificação das atividades económicas (cae) - revisão 3.* (https://www.ine.pt/ine_novidades/semin/cae/CAE_REV_3.pdf)

Kosmidis, I. (2021). *Bias reduction in binomial-response generalized linear models.* (https://CRAN.R-project.org/package=brglm)

Maechler, M., Rousseeuw, P., Croux, C., Todorov, V., Ruckstuhl, A., Salibian-Barrera, M., ... di Palma, M. A. (2023). *Basic robust statistics.* (https://cran.r-project.org/package=robustbase)

Marschner, I., & Donoghoe, M. W. (2018). *glm2: Fitting generalized linear models.* (https://CRAN.R-project.org/package=glm2)

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models* (2° ed.). Chapman and hall, London. doi: http://dx.doi.org/10.1007/978-1-4899-3242-6

Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 370-384.

Pestana, D. D., & Velosa, S. (2010). *"introdução à probabilidade e à estatística", 4ª ed. revista*. Calouste Gulbenkian Edition.

Powers, D. M. W. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies, no. 1, 2*, 37-63.

R Core Team. (2023). *R: A language and environment for statistical computing.* Vienna, Austria. (https://www.R-project.org/)

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, *6*(2), 461 – 464. doi: 10.1214/aos/1176344136

StataCorp. (2023). *Rglm manual.* (https://www.stata.com/manuals/rglm.pdf)

Turkman, M. A. A., & Silva, G. L. (2000). *Modelos lineares generalizados - da teoria à prática*. Sociedade Portuguesa de Estatística. (Mini-curso no VIII Congresso da Soc. Port. Estatística)

Winter, B., & Bürkner, P.-C. (2021). Poisson regression for linguists: A tutorial introduction to modelling count data with brms. *Language and Linguistics Compass*.

Yang, Y. (2005). Can the strengths of aic and bic be shared? a conflict between model indentification and regression estimation. *Biometrika*, *92*(4), 937–950. Retrieved from http://www.jstor.org/stable/20441246

Yee, T. (2023). *Vector generalized linear and additive models.* (https://cran.r-project.org/package=VGAM)

Yun, H. (2021). Prediction model of algal blooms using logistic regression and confusion matrix. *International Journal of Electrical and Computer Engineering (IJECE)*, *11*, 24072413.

Anexo I - Resultados da qualidade de ajustamento

Tabela 1.1: MSE dos coeficientes de regressão do modelo com Regressão Logística (Modelo 2).

		Erro Quadrático Médio				**PRE(%)**		
n	β_i	f_1=glm	f_2=brglm	f_3=glmrob	f_4=bayesglm	f_2/f_1	f_3/f_1	f_4/f_1
500	β_0	0.413212	0.356819	0.554910	**0.317135**	86.35	134.29	76.75
	β_1	0.321765	0.225303	0.656946	**0.197610**	70.02	204.17	61.41
	β_2	0.652943	0.461735	1.463445	**0.406621**	70.72	224.13	62.28
	β_3	1.356739	1.078963	2.820343	**1.054363**	79.53	207.87	77.71
	β_4	0.677520	0.608282	0.854221	**0.511170**	89.78	126.08	75.45
2000	β_0	0.082302	0.080335	0.088849	**0.078223**	97.61	107.95	95.04
	β_1	0.060563	0.055789	0.083890	**0.054242**	92.12	138.63	89.56
	β_2	0.122614	0.113745	0.175126	**0.111065**	92.77	142.83	90.58
	β_3	0.349530	0.326791	0.456678	**0.317780**	93.49	130.65	90.92
	β_4	0.146235	0.142710	0.165095	**0.136923**	97.59	112.90	93.63
10000	β_0	0.017524	0.017500	0.019660	**0.017406**	99.86	112.19	99.33
	β_1	0.010673	0.010566	0.015087	**0.010558**	99.00	141.36	98.92
	β_2	0.020445	0.020240	0.029662	**0.020223**	99.00	145.08	98.91
	β_3	0.057874	0.056928	0.075967	**0.056509**	98.37	131.26	97.64
	β_4	0.030024	0.029887	0.033673	**0.029643**	99.54	121.15	98.73
2000	β_0	0.007008	0.007009	0.008141	**0.006990**	100.01	116.17	99.74
	β_1	0.005347	0.005314	0.007293	**0.005308**	99.38	136.39	99.27
	β_2	0.010078	0.010029	0.013433	**0.010026**	99.51	133.29	99.48
	β_3	0.028693	0.028291	0.036632	**0.028030**	98.60	127.67	97.69
	β_4	0.012828	0.012798	0.014495	**0.012743**	99.77	113.00	99.34

Tabela 1.2: MAPE das estimativas dos coeficientes de regressão no modelo de Regressão Logística (Modelo 2).

		Erro Percentual Absoluto Médio (%)			
n	β_i	f_1=glm	f_2=brglm	f_3=glmrob	f_4=bayesglm
500	β_0	7.2102	2.0763	10.2686	**1.9943**
	β_1	6.2634	**0.9357**	10.0955	2.6163
	β_2	5.5373	**0.2458**	9.2314	2.9519
	β_3	4.1961	**0.9964**	8.0459	5.4677
	β_4	14.4207	9.0275	11.3965	**0.4690**
2000	β_0	**0.2294**	0.9125	1.1478	0.9082
	β_1	1.3646	**0.1521**	2.1007	0.7166
	β_2	1.1469	**0.0636**	1.8303	0.8459
	β_3	1.5085	**0.2991**	2.1419	0.8114
	β_4	2.9049	1.6782	3.8605	**0.3732**
10000	β_0	0.5154	0.7399	**0.2701**	0.7380
	β_1	0.1540	**0.0835**	0.2044	0.2552
	β_2	0.1415	**0.0959**	0.2160	0.2513
	β_3	0.3642	0.1275	0.4221	**0.0930**
	β_4	1.1549	1.3895	**0.5182**	1.7987
20000	β_0	**0.3197**	0.4321	0.3208	0.4314
	β_1	0.0945	**0.0240**	0.0889	0.1099
	β_2	0.0648	**0.0537**	0.0705	0.1314
	β_3	0.3330	0.2149	0.3569	**0.1045**
	β_4	0.8261	0.7073	1.1210	**0.5004**

Tabela 1.3: MSE dos coeficientes de regressão do modelo com distribuição Binomial (Modelo 4).

		Erro Quadrático Médio				**PRE (%)**		
n	β_i	f_1=glm	f_2=brglm	f_3=glmrob	f_4=bayesglm	f_2/f_1	f_3/f_1	f_4/f_1
500	β_0	0.645933	0.549077	1.076114	**0.508791**	85.01	166.60	78.77
	β_1	0.199476	0.151557	0.375424	**0.143396**	75.98	188.21	71.89
	β_2	1.277393	0.933097	2.541137	**0.837412**	73.05	198.93	65.56
	β_3	1.012184	0.860911	1.657017	**0.825975**	85.05	163.71	81.60
	β_4	1.652470	1.300768	2.880454	**1.219268**	78.72	174.31	73.78
2000	β_0	0.139805	0.131895	0.186449	**0.127182**	94.34	133.36	90.97
	β_1	0.037919	0.035656	0.054876	**0.035348**	94.03	144.72	93.22
	β_2	0.236075	0.217233	0.344632	**0.211198**	92.02	145.98	89.46
	β_3	0.214197	0.203330	0.269977	**0.197247**	94.93	126.04	92.09
	β_4	0.304640	0.288905	0.388585	**0.286566**	94.83	127.56	94.07
10000	β_0	0.025059	**0.024995**	0.029931	0.025025	99.74	119.44	99.86
	β_1	0.006846	0.006749	0.009126	**0.006727**	98.58	133.30	98.26
	β_2	0.040362	0.039927	0.057332	**0.039917**	98.92	142.04	98.90
	β_3	0.036065	**0.036051**	0.043908	0.036409	99.96	121.75	100.95
	β_4	0.060304	**0.059764**	0.078700	0.059767	99.10	130.51	99.11
10000	β_0	0.013379	0.013263	0.016770	**0.013183**	99.13	125.35	98.54
	β_1	0.003485	**0.003469**	0.004813	0.003472	99.54	138.11	99.63
	β_2	0.020692	0.020388	0.031177	**0.020227**	98.53	150.67	97.75
	β_3	0.020815	0.020654	0.025715	**0.020517**	99.23	123.54	98.57
	β_4	0.026021	0.025761	0.036262	**0.025616**	99.00	139.36	98.44

Tabela 1.4: MAPE das estimativas dos coeficientes de regressão no modelo de Regressão Logística (Modelo 4).

		Erro Percentual Absoluto Médio (%)			
n	β_i	f_1=glm	f_2=brglm	f_3=glmrob	f_4=bayesglm
500	β_0	2.9633	**2.5633**	7.0447	6.4743
	β_1	4.7982	**0.9192**	8.9877	5.9748
	β_2	4.6186	**1.0884**	8.7925	5.0836
	β_3	3.6793	**1.9514**	7.9842	8.9295
	β_4	5.1708	**0.5457**	9.3859	6.3413
2000	β_0	1.6293	**0.3180**	1.8974	0.7347
	β_1	1.0666	**0.2603**	1.7052	1.5525
	β_2	1.2191	**0.1091**	1.9472	1.1430
	β_3	1.7012	**0.3749**	2.0382	1.4737
	β_4	1.0608	**0.2584**	2.0432	1.7504
10000	β_0	**0.0454**	0.3010	0.0537	0.5120
	β_1	0.2493	**0.0113**	0.3108	0.2706
	β_2	0.1426	**0.1178**	0.2373	0.3243
	β_3	0.1646	0.4225	**0.0957**	0.7930
	β_4	0.1742	**0.0847**	0.2083	0.3829
20000	β_0	0.2808	0.1525	0.4591	**0.0455**
	β_1	0.0715	**0.0585**	0.2008	0.1888
	β_2	0.2146	0.0843	0.3471	**0.0198**
	β_3	0.3093	0.1796	0.5461	**0.0072**
	β_4	0.2247	0.0951	0.3620	**0.0547**