
Half Title

Introduction to Bayesian Inference:
A GUIded toolkit using R



Title Page

Introduction to Bayesian Inference:
A GUIded toolkit using R

by Andrés Ramírez-Hassan, PhD. Statistical Science.

LOC Page

To my parents, Nancy and Orlando.



Introduction



FIGURE 1
Supposedly portrait of Thomas Bayes.

Since late 90's Bayesian inference has gained a lot of popularity among researchers due to the computational revolution and availability of algorithms to solve complex integrals. However, many researchers, students and practitioners still lack understanding and application of this inferential approach. The main reason is the requirement of good programming skills.

Introduction to Bayesian inference: A GUIided toolkit using R mainly targets those who want to apply Bayesian inference having a good conceptual and formal understanding, but not necessarily having time to develop programming skills. Thus, this book provides a graphical user interface (GUI) to carry out Bayesian regression in a very friendly environment. The book also provides the basic theory, and its code implementation using **R** software [166], some applications to highlight the potential of Bayesian inference, and theory and computational exercises, for those who are interested in developing more complex models. In particular, this book contains the mathematical proofs step by step of the basic models, which are the base for obtaining the most relevant mathematical results of more complex models.

Our GUI is based on an interactive web application using shiny [35], and some packages in **R**. Users can estimate univariate, multivariate, time series, longitudinal/panel data, and Bayesian model average models using our GUI. In addition, it gives basic summaries and formal and graphical diagnostics of the

posterior chains. Our GUI can be run in any operating system, and is freely available at <https://github.com/besmarter/BSTApp>.

Users can get simulated and real datasets in the folders **DataSim**, and **DataApp**, respectively. The former folder also includes the files that were used to simulate different processes, so, the population parameters are available, and as a consequence, these files can be used as a pedagogical tool to show some statistical properties. The latter folder contains the datasets used in our applications. Users should use these datasets as templates to structuring their own datasets.

This book has three parts. The first part covers theory (conceptual and mathematical), programming, and simulation foundations (chapters 1 to 4). The second part focuses on applications of regression analysis with particular emphasis on the computational aspect of obtaining draws from the posterior distributions at three levels of programming skills: no skills at all using our GUI, intermediate level using specialized packages of **R** to perform Bayesian inference, and relatively advanced programming skills getting the posterior draws from scratch (chapters 5 to 10). The third part provides an introductory treatment of *advanced methods* in Bayesian inference (chapters 11 to 14). I show in some detail the mathematical deductions in the first part of the book, whereas I do not show most of the proofs in the second and third parts. However, same mathematical steps in the first part can be used to find the results of parts two and three of the book.

Chapter 1 begins with an introduction to formal concepts in Bayesian inference starting with the Bayes' rule, all its components with their formal definitions and basic examples. Then, it presents the basics of Bayesian inference based on decision theory under uncertainty. Chapter 2 presents conceptual differences between Bayesian and Frequentist statistical approaches, and a historical and philosophical perspective about Bayesian statistics and econometrics highlighting differences compared to the Frequentist approach. In Chapter 3 I introduce conjugate families in basic statistical models, solving them analytically and computationally. Simulation based methods are shown in Chapter 4, these algorithms are very important in modern Bayesian inference as most realistic models do not have standard forms or analytical solutions. I present our graphical user interface in Chapter 5, and univariate and multivariate regression models are presented in chapters 6 and 7, respectively. Chapter 8 presents univariate and multivariate time series models, and Chapter 9 presents Bayesian longitudinal/panel data models. Chapter 10 introduces Bayesian model averaging. In the third part, there are Chapter 11 exploring hierarchical models, Chapter 13 shows causal inference, Chapter 12 shows Bayesian methods in machine learning algorithms, and Chapter 14 describes some recent methodological developments such as approximate Bayesian computation (ABC), variational Bayes (VB), integrated nested Laplace approximations (INLA), and Bayesian exponential tilted empirical likelihood (BETEL).

About me

My name is Andrés Ramírez-Hassan, I am an applied and theory econometrician working as a Distinguished Professor in the School of Finance, Economics and Government at Universidad EAFIT (Medellín, Colombia). I got a PhD in Statistical Science, a masters degree in Finance, and another in Economics, and also a bachelor's degree in Economics. I was a research fellow at the Department of Econometrics and Business Statistics at Monash University, and a visiting Professor in the Department of Economics at the University of Melbourne and the University of Glasgow. Having completed my PhD degree, much of my research has been in the area of Bayesian Econometrics with applications in crime, finance, health, sports and utilities. My work has been published (or is forthcoming) in the *International Journal of Forecasting*, *Journal of Applied Econometrics*, *Econometric Reviews*, *Journal of Computational and Graphical Statistics*, *The R Journal*, *Economic Modelling*, *Spatial Economic Analysis*, *Economic Inquiry*, *World Development*, *Journal of Sport Economics*, *Empirical Economics*, *Australian and New Zealand Journal of Statistics*, *Brazilian Journal of Probability and Statistics*, and other highly regarded international research outlets.

I founded **BEsmarter** –Bayesian Econometrics: simulations, models and applications to research, teaching and encoding with responsibility–. This is a research group whose **mission** is to *lead and excel in the generation and dissemination of Bayesian Econometric knowledge through research, teaching and software*. We **envision** worldwide econometric research, teaching and applications based on the Bayesian framework that:

- Inspires new econometric ideas
- Creates a user friendly environment for applications of Bayesian econometrics
- Transforms classic econometric research, teaching and applications
- And where one of the main concerns of science is to solve social problems

mail: aramir21@gmail.com / aramir21@eafit.edu.co

website: <http://www.besmarter-team.org> <https://sites.google.com/view/arh-bayesian>



FIGURE 2

This book is licensed under the **Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License**.



Contents

Introduction	vii
Foreword	xi
Preface	xiii
Symbols	xv
I Foundations: Theory, simulation methods and programming	1
1 Basic formal concepts	3
1.1 The Bayes' rule	3
1.2 Bayesian framework: A brief summary of theory	8
1.2.1 Example: Health insurance	14
1.3 Bayesian reports: Decision theory under uncertainty	27
1.3.1 Example: Health insurance continues	30
1.4 Summary	32
1.5 Exercises	32
2 Conceptual differences of the Bayesian and Frequentist approaches	35
2.1 The concept of probability	35
2.2 Subjectivity is not the key	36
2.3 Estimation, hypothesis testing and prediction	37
2.4 The likelihood principle	41
2.5 Why is not the Bayesian approach that popular?	43
2.6 A simple working example	45
2.6.1 Example: Math test	46
2.7 Summary	47
2.8 Exercises	48
3 Cornerstone models: Conjugate families	51
3.1 Motivation of conjugate families	51
3.1.1 Examples of exponential family distributions	52
3.2 Conjugate prior to exponential family	56
3.2.1 Examples: Theorem 4.2.1	57

3.3	Linear regression: The conjugate normal-normal/inverse gamma model	72
3.4	Multivariate linear regression: The conjugate normal-normal/inverse Wishart model	86
3.5	Summary	90
3.6	Exercises	90
4	Simulation methods	95
4.1	Markov chain Monte Carlo methods	95
4.1.1	Gibbs sampler	96
4.1.2	Metropolis-Hastings	99
4.1.3	Hamiltonian Monte Carlo	103
4.2	Importance sampling	109
4.3	Particle filtering	114
4.4	Convergence diagnostics	125
4.4.1	Numerical standard error	125
4.4.2	Effective Number of Simulation Draws	126
4.4.3	Tests of convergence	127
4.4.4	Checking for errors in the posterior simulator	128
4.5	Summary	133
4.6	Exercises	133
II	Regression models: A GUIded toolkit	135
5	Graphical user interface	137
5.1	Introduction	137
5.2	Univariate models	139
5.3	Multivariate models	142
5.4	Time series model	144
5.5	Longitudinal/panel models	144
5.6	Bayesian model average	145
5.7	Warning	147
6	Univariate models	149
6.1	The Gaussian linear model	149
6.2	The logit model	154
6.3	The probit model	159
6.4	The multinomial probit model	162
6.5	The multinomial logit model	166
6.6	Ordered probit model	170
6.7	Negative binomial model	174
6.8	Tobit model	179
6.9	Quantile regression	183
6.10	Bayesian bootstrap regression	186
6.11	Summary	188
6.12	Exercises	188

7 Multivariate models	191
7.1 Multivariate regression	191
7.2 Seemingly unrelated regression	197
7.3 Instrumental variable	202
7.4 Multivariate probit model	206
7.5 Summary	211
7.6 Exercises	211
8 Time series models	215
8.1 State-space representation	216
8.2 ARMA processes	227
8.3 Stochastic volatility models	239
8.4 Vector Autoregressive models	246
8.5 Summary	253
8.6 Exercises	256
9 Longitudinal/Panel data models	259
9.1 Normal model	260
9.2 Logit model	269
9.3 Poisson model	273
9.4 Summary	278
9.5 Exercises	278
10 Bayesian model average	281
10.1 Foundation	281
10.2 The Gaussian linear model	285
10.3 Generalized linear models	298
10.4 Dynamic model averaging	304
10.5 Calculating the marginal likelihood	309
10.5.1 Savage-Dickey density ratio	311
10.5.2 Chib's methods	312
10.5.3 Gelfand-Dey method	312
10.6 Summary	318
10.7 Exercises	319
III Advanced methods: A brief introduction	323
11 Non-parametric and semi-parametric models	325
11.1 Additive non-parametric structure	325
11.1.1 Partial linear model	325
11.2 Hierarchical models	325
11.2.1 Finite mixtures	325
11.2.2 Dirichlet processes	325

12 Machine learning	327
12.1 Cross validation and Bayes factors	327
12.2 Regularization	327
12.2.1 Bayesian LASSO	327
12.2.2 Stochastic search variable selection	327
12.2.3 Non-local priors	327
12.3 Bayesian additive regression trees	327
12.4 Gaussian processes	327
13 Causal inference	329
13.1 Instrumental variables	329
13.1.1 Semi-parametric IV model	329
13.2 Regression discontinuity design	329
13.3 Regression kink design	329
13.4 Synthetic control	329
13.5 Difference in difference estimation	329
13.6 Event Analysis	329
13.7 Bayesian exponential tilted empirical likelihood	329
13.8 Double-Debiased machine learning causal effects	329
14 Approximation methods	331
14.1 Approximate Bayesian computation	331
14.2 Expectation propagation	331
14.3 Integrated nested Laplace approximations	331
14.4 Variational Bayes	331
Bibliography	333
Appendix	349

Foreword



Preface

The main goal of this book is to make more approachable the Bayesian inferential framework to students, researchers and practitioners who want to understand and apply this statistical/econometric approach, but who do not have time to develop programming skills. I tried to have a balance between applicability and theory. Then, this book comes with a very friendly graphical user interface (GUI) to implement the most common regression models, but also contains the basic mathematical developments, as well as their code implementation, for those who are interested in advancing in more complex models.

To instructors and students

This book is divided in three parts, foundations (chapters 1 to 4), regression analysis (chapters 5 to 10), and *Advanced* methods (chapters 11 to 14). Our graphical user interface (GUI) targets the second part. The source code can be found at <https://github.com/besmarter/BSTApp>. Instructors and students can have all codes, simulated and real data sets are also there. There are three ways to install our GUI: First, typing `shiny::runGitHub("besmarter/BSTApp", launch.browser=T)` in the **R** package console or any **R** code editor, and execute it. Second, getting into <https://posit.cloud/content/4328505>, log in or sing up in **Posit Cloud**, click on **BSTApp-master** folder in the **Files** tap of the right-bottom window, then click on **app.R** file, and click on **Run App** button. Third, using a **Docker** image typing

1. `docker pull magralo95/besmartergui:latest`
2. `docker run -rm -p 3838:3838 magralo95/besmartergui`

in the **Command Prompt**, then users can access our GUI going to <http://localhost:3838/>. See Chapter 5 for details.

Students should have basic knowledge of probability theory and statistics, with some basic background of econometrics and time series, particularly regression analysis. Familiarity with standard univariate and multivariate probability distributions is strongly recommended. In addition, students who want to master the material in this book should have programming skills in **R** software.¹

I included some formal and computational exercises at the end of each

¹This is an excellent starting point for **R** programming: <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>.

chapter. This would help students to have a better understanding of the material shown in each chapter. A manual with the solutions of exercises accompanies this book.

Instructors can use this book as a text in a course of introduction to Bayesian Econometrics/Statistics with a high emphasis on implementation and applications. This book is complementary, rather than substitute, of excellent books in the topic such as [76, 34, 189, 92, 85, 129] and [126].

Acknowledgments

I started our GUI in the 2016 after being diagnosed with cervical dystonia. I used to work in this side project on weekends, I named this time “nerd weekends”, and it was a kind of release from my health condition. Once I got better, I invited Mateo Graciano, my former student, business partner and friend, to be part of the project, he helped me a lot developing our GUI, and I am enormously thankful to Mateo. I would also like to thank members of the BEsmarter research group from Universidad EAFIT, and NUMBATS members from Monash University for your comments and recommendations to improve our GUI.

This book is an extension of the paper *A GUIded tour of Bayesian regression* [182], which is a brief user guide of our GUI. So, I decided to write this book to show the underlying theory and codes in our GUI, and use it as a text book in my course in Bayesian econometrics/statistics. I acknowledge and offer my gratitude to my students in this subject, their insight and thoughtful questions have helped me to get a better understanding of this material.

I also thank Chris Parmeter for your suggestions about how to present our user guide, Professor Raul Pericchi and Juan Carlos Correa who introduced me to Bayesian statistics, Liana Jacobi and Chun Fung Kwok (Jackson) from the University of Melbourne and David Frazier from Monash University for nice talks and amazing collaborations in Bayesian Econometrics/statistics, Professor Peter Diggle to support my career, and particularly, Professor Gael Martin, who gave me a chance to work with her, she is an inspiring intellectual figure. Finally, my colleagues and staff from Universidad EAFIT have always given me their support.

To my parents, Orlando and Nancy, who have given me their unconditional support. They have taught me that the primary aspect of the human being’s spiritual evolution is humility. I am in my way to learn this.

Symbols

Symbol Description

\neg	Negation symbol	$\mathbf{0}_l$	l -dimensional null vector
\propto	Proportional symbol	\max_r	Maximum over r
\perp	Independence symbol	\min_r	Minimum over r
\mathcal{R}	The Real set	∇	Gradiente operator
\emptyset	Empty set	$\stackrel{iid}{\sim}$	Independently and identically distributed
$\mathbb{1}$	Indicator function	$>$	Greater than
P	Probability measure	$<$	Less than
$:=$	Is defined as	\approx	Approximately equal to
argmax	Argument of the maximum	\gtrsim	Greater than or approximately equal to
argmin	Argument of the minimum	Δ	Difference operator
tr	Trace operator	\subseteq	Subset
vec	Vectorization operator	\subset	Proper subset
\lim	Limit	\xrightarrow{d}	Convergence in distribution (law)
\otimes	Kronecker product	\xrightarrow{p}	Convergence in probability
$\text{diag}\{\cdot\}$	Diagonal matrix	$\xrightarrow{a.s.}$	Almost surely convergence
$\dim\{\cdot\}$	Dimension of an object		
\mathbf{I}_l	l -dimensional identity matrix		



Part I

Foundations: Theory, simulation methods and programming



1

Basic formal concepts

We introduce formal concepts in Bayesian inference starting with the Bayes' rule, all its components with their formal definitions and basic examples. In addition, we present some nice features of Bayesian inference such as Bayesian updating, and asymptotic sampling properties, and the basics of Bayesian inference based on decision theory under uncertainty, presenting important concepts like loss function, risk function and optimal rules.

1.1 The Bayes' rule

As expected the point of departure to perform Bayesian inference is the Bayes' rule,¹ which is the Bayes' solution to the inverse probability of causes, this rule combines prior beliefs with objective probabilities based on repeatable experiments. In this way, we can move from observations to probable causes.

Formally, the conditional probability of A_i given B is equal to the conditional probability of B given A_i times the marginal probability of A_i over the marginal probability of B ,

$$\begin{aligned} P(A_i|B) &= \frac{P(A_i, B)}{P(B)} \\ &= \frac{P(B|A_i) \times P(A_i)}{P(B)}, \end{aligned} \tag{1.1}$$

where by the law of total probability $P(B) = \sum_i P(B|A_i)P(A_i) \neq 0$, $\{A_i, i = 1, 2, \dots\}$ is a finite or countably infinite partition of a sample space.

In the Bayesian framework, B is sample information that updates a probabilistic statement about an unknown object A_i following probability rules. This is done by means of the Bayes' rule using prior "beliefs" about A_i , that is, $P(A_i)$, sample information relating B to the particular state of the nature A_i through a probabilistic statement, $P(B|A_i)$, and the probability of observing that specific sample information $P(B)$.

¹Observe that I use the term "Bayes' rule" rather than "Bayes' theorem". It was Laplace [131] who actually generalized the Bayes' theorem [12]. His generalization is named the Bayes' rule.

Let's see a simple example, *the base rate fallacy*:

Assume that the sample information comes from a positive result from a test whose true positive rate (sensitivity) is 98%, $P(+|\text{disease}) = 0.98$. On the other hand, the prior information regarding being infected with this disease comes from a base incidence rate that is equal to 0.002, that is $P(\text{disease}) = 0.002$. Then, *what is the probability of being actually infected?*

This is an example of *the base rate fallacy*, where having a positive test result from a disease whose base incidence rate is tiny gives a low probability of actually having the disease.

The key to answer the question is based on understanding the difference between the probability of having the disease given a positive result, $P(\text{disease}|+)$, versus the probability of a positive result given the disease, $P(+|\text{disease})$. The former is the important result, and the Bayes' rule help us to get the answer. Using the Bayes' rule (equation 1.1):

$$\begin{aligned} P(\text{disease}|+) &= \frac{P(+|\text{disease}) \times P(\text{disease})}{P(+)} \\ &= \frac{0.98 \times 0.002}{0.98 \times 0.002 + (1 - 0.98) \times (1 - 0.002)} \\ &= 0.09, \end{aligned}$$

where $P(+) = P(+|\text{disease}) \times P(\text{disease}) + P(+|\neg\text{disease}) \times P(\neg\text{disease})$.²

R code. The base rate fallacy

```
1 PD <- 0.002 # Probability of disease
2 PPD <- 0.98 # True positive (Sensitivity)
3 PDP <- PD * PPD / (PD * PPD + (1 - PD)*(1 - PPD))
4 paste("Probability of disease given a positive test is", sep
      = " ", round(PDP, 2))
5 "Probability of disease given a positive test is 0.09"
```

We observe that despite of having a positive result, the probability of having the disease is low. This due to the base rate being tiny.

Another interesting example, which is at the heart of the origin of the Bayes' theorem [12], is related to the existence of God [207]. The Section X of David Hume's "An Inquiry concerning Human Understanding, 1748" is named *Of Miracles*. There, Hume argues that when someone claims to

² \neg is the negation symbol. In addition, we have that $P(B|A) = 1 - P(B|A^c)$ in this example, where A^c is the complement of A . However, it is not always the case that $P(B|A) \neq 1 - P(B|A^c)$.

have seen a miracle, this is poor evidence it actually happened, since it goes against what we see every day. Then, Richard Price, who actually finished and published “An essay towards solving a problem in the doctrine of chances” in 1763 after Bayes died in 1761, argues against Hume saying that there is a huge difference between *impossibility* as used commonly in conversation and *physical impossibility*. Price used an example of a dice with a million sides, where *impossibility* is getting a particular side when throwing this dice, and *physical impossibility* is getting a side that does not exist. In millions throws, the latter case never would occur, but the former eventually would.

Let's say that there are two cases of resurrection (Res), Jesus Christ and Elvis, and the total number of people who have ever lived is 108.5 billion,³ then the prior base rate is $2/(108.5 \times 10^9)$. On the other hand, let's say that the sample information comes from a very reliable witness whose true positive rate is 0.9999999. Then, *what is the probability of this miracle?*⁴

Using the Bayes' rule:

$$\begin{aligned} P(\text{Res}|\text{Witness}) &= \frac{P(\text{Witness}|\text{Res}) \times P(\text{Res})}{P(\text{Witness})} \\ &= \frac{2/(108.5 \times 10^9) \times 0.9999999}{2/(108.5 \times 10^9) \times 0.9999999 + (1 - 2/(108.5 \times 10^9)) \times (1 - 0.9999999)} \\ &= 0.000184297806959661 \end{aligned}$$

where $P(\text{Witness}) = P(\text{Witness}|\text{Res}) \times P(\text{Res}) + (1 - P(\text{Witness}|\text{Res})) \times (1 - P(\text{Res}))$.

Thus, 1.843×10^{-4} is the probability of a resurrection given a very reliable witness.

R code. Of Miracles

```

1 # Probability of resurrection
2 PR <- 2/(108.5 * 10^9)
3 PWR <- 0.999999 # True positive rate
4 PRW <- PR * PWR / (PR * PWR + (1 - PR)*(1 - PWR))
5 paste("Probability of resurrection given witness is", sep =
      " ", PRW)
6 "Probability of resurrection given witness is
      0.000184297806959661"

```

Observe that we can condition on many events in the Bayes' rule. Let's have two conditioning events B and C , then equation 1.1 becomes

³<https://www.wolframalpha.com/input/?i=number+of+people+who+have+ever+lived+on+Earth>

⁴<https://www.r-bloggers.com/2019/04/base-rate-fallacy-or-why-no-one-is-justified-to-believe-that-jesus-rose/>

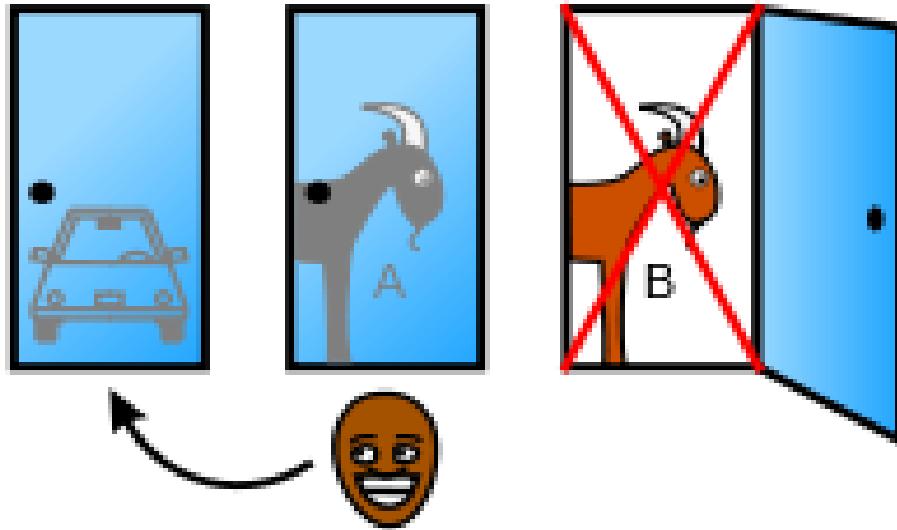


FIGURE 1.1
The Monty Hall problem.

$$\begin{aligned} P(A_i|B,C) &= \frac{P(A_i,B,C)}{P(B,C)} \\ &= \frac{P(B|A_i,C) \times P(A_i|C) \times P(C)}{P(B|C)P(C)}. \end{aligned} \quad (1.2)$$

Let's use this rule in one of the most intriguing statistical puzzles, *the Monty Hall problem*, to illustrate how to use equation 1.2 [196, 197]. This was the situation faced by a contestant in the American television game show *Let's Make a Deal*. There, the contestant was asked to choose a door where behind one door there is a car, and behind the others, goats. Let's say that the contestant picks door No. 1, and the host (Monty Hall), who knows what is behind each door, opens door No. 3, where there is a goat (see Figure 1.1). Then, the host asks the tricky question to the contestant, *do you want to pick door No. 2?*

Let's name P_i the event **contestant picks door No. i** , which stays close, H_i the event **host picks door No. i** , which is open, and there is a goat, and C_i the event **car is behind door No. i** . In this particular setting, the contestant is interested in the probability of the event $P(C_2|H_3, P_1)$. A naive answer would be that it is irrelevant as initially $P(C_i) = 1/3$, $i = 1, 2, 3$, and now $P(C_i|H_3) = 1/2$, $i = 1, 2$ as the host opened door No. 3. So, why bothering changing the initial guess if the odds are the same (1:1)? The important point here is that the host knows what is behind each door, and picks a door where there is a goat given contestant choice. In this particular

setting, $P(H_3|C_3, P_1) = 0$, $P(H_3|C_2, P_1) = 1$ and $P(H_3|C_1, P_1) = 1/2$. Then, using equation 1.2

$$\begin{aligned} P(C_2|H_3, P_1) &= \frac{P(C_2, H_3, P_1)}{P(H_3, P_1)} \\ &= \frac{P(H_3|C_2, P_1)P(C_2|P_1)P(P_1)}{P(H_3|P_1) \times P(P_1)} \\ &= \frac{P(H_3|C_2, P_1)P(C_2)}{P(H_3|P_1)} \\ &= \frac{1 \times 1/3}{1/2}, \end{aligned}$$

where the third equation uses the fact that C_i and P_i are independent events, and $P(H_3|P_1) = 1/2$ due to this depending just on P_1 (not on C_2).

Therefore, changing the initial decision increases the probability of getting the car from $1/3$ to $2/3$! Thus, it is always a good idea to change the door.

Let's see a simulation exercise to check this answer:

R code. The Monty Hall problem

```

1 set.seed(0101) # Set simulation seed
2 S <- 100000 # Simulations
3 Game <- function(switch = 0){
4   # switch = 0 is not change
5   # switch = 1 is to change
6   opts <- 1:3
7   car <- sample(opts, 1) # car location
8   guess1 <- sample(opts, 1) # Initial guess
9
10  if(car != guess1) {
11    host <- opts[-c(car, guess1)]
12  } else {
13    host <- sample(opts[-c(car, guess1)], 1)
14  }
15  win1 <- guess1 == car # Win no change
16  guess2 <- opts[-c(host, guess1)]
17  win2 <- guess2 == car # Win change
18  if(switch == 0){
19    win <- win1
20  } else {
21    win <- win2
22  }
23  return(win)
24 }
25
26 #Win probabilities not changing
27 Prob <- mean(replicate(S, Game(switch = 0)))
28 Prob
29 0.3334
30
31 #Win probabilities changing
32 Prob <- mean(replicate(S, Game(switch = 1)))
33 Prob
34 0.6654

```

1.2 Bayesian framework: A brief summary of theory

For two random objects θ and y , the Bayes' rule may be analogously used,⁵

⁵From a Bayesian perspective θ is fixed, but unknown. Then, it is treated as a random object despite the lack of variability (see Chapter 2).

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta})}{p(\mathbf{y})}, \quad (1.3)$$

where $\pi(\boldsymbol{\theta}|\mathbf{y})$ is the posterior density function, $\pi(\boldsymbol{\theta})$ is the prior density, $p(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood (statistical model), and

$$p(\mathbf{y}) = \int_{\Theta} p(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta} = \mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})] \quad (1.4)$$

is the marginal likelihood or prior predictive. Observe that for this expected value to be meaningful the prior should be a proper density, that is, integrates to one, otherwise, it does not make sense.

Observe that $p(\mathbf{y}|\boldsymbol{\theta})$ is not a density in $\boldsymbol{\theta}$. In addition, $\pi(\boldsymbol{\theta})$ does not have to integrate to 1, that is, $\pi(\boldsymbol{\theta})$ can be an improper density function, $\int_{\Theta} \pi(\boldsymbol{\theta})d\boldsymbol{\theta} = \infty$. However, $\pi(\boldsymbol{\theta}|\mathbf{y})$ is a proper density function, that is, $\int_{\Theta} \pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} = 1$. For instance, set $\pi(\boldsymbol{\theta}) = c$, where c is a constant, then $\int_{\Theta} cd\boldsymbol{\theta} = \infty$. However, $\int_{\Theta} \pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} = \int_{\Theta} \frac{p(\mathbf{y}|\boldsymbol{\theta}) \times c}{\int_{\Theta} p(\mathbf{y}|\boldsymbol{\theta}) \times cd\boldsymbol{\theta}} d\boldsymbol{\theta} = 1$ where c cancels out.

$\pi(\boldsymbol{\theta}|\mathbf{y})$ is a sample updated “probabilistic belief” version of $\pi(\boldsymbol{\theta})$, where $\pi(\boldsymbol{\theta})$ is a prior probabilistic belief which can be constructed from previous empirical work, theory foundations, expert knowledge and/or mathematical convenience. This prior usually depends on parameters, which are named *hyperparameters*. In addition, the Bayesian approach implies using a probabilistic model about \mathbf{y} given $\boldsymbol{\theta}$, that is, $p(\mathbf{y}|\boldsymbol{\theta})$, where its integral over Θ , $p(\mathbf{y})$ is named *the model evidence* due to being a measure of model fit to the data.

Observe that the Bayesian inferential approach is conditional, that is, what can we learn about an unknown object $\boldsymbol{\theta}$ given that we already observed \mathbf{y} ? The answer is also conditional on the probabilistic model, that is $p(\mathbf{y}|\boldsymbol{\theta})$. So, what if we want to compare different models, let’s say \mathcal{M}_m , $m = \{1, 2, \dots, M\}$. Then, we should make explicit this in the Bayes’ rule formulation,

$$\pi(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M}_m) = \frac{p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}_m) \times \pi(\boldsymbol{\theta}|\mathcal{M}_m)}{p(\mathbf{y}|\mathcal{M}_m)}. \quad (1.5)$$

The posterior model probability is

$$\pi(\mathcal{M}_m|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{M}_m) \times \pi(\mathcal{M}_m)}{p(\mathbf{y})}, \quad (1.6)$$

where $p(\mathbf{y}|\mathcal{M}_m) = \int_{\Theta} p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}_m) \times \pi(\boldsymbol{\theta}|\mathcal{M}_m)d\boldsymbol{\theta}$ due to equation 1.5, and $\pi(\mathcal{M}_m)$ is the prior model probability.

Calculating $p(\mathbf{y})$ in equations 1.3 and 1.6 is very demanding most of the realistic cases. Fortunately, it is not required when performing inference about $\boldsymbol{\theta}$ as this is integrated out from it. Then, all what you need to know about the

shape of $\boldsymbol{\theta}$ is in $p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}_m) \times \pi(\boldsymbol{\theta}|\mathcal{M}_m)$ or without explicitly conditioning on \mathcal{M}_m ,

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}). \quad (1.7)$$

Equation 1.7 is a very good shortcut to perform Bayesian inference about $\boldsymbol{\theta}$.

We also can avoid calculating $p(\mathbf{y})$ when performing model selection (hypothesis testing) using posterior odds ratio, that is, comparing models \mathcal{M}_1 and \mathcal{M}_2 ,

$$\begin{aligned} PO_{12} &= \frac{\pi(\mathcal{M}_1|\mathbf{y})}{\pi(\mathcal{M}_2|\mathbf{y})} \\ &= \frac{p(\mathbf{y}|\mathcal{M}_1)}{p(\mathbf{y}|\mathcal{M}_2)} \times \frac{\pi(\mathcal{M}_1)}{\pi(\mathcal{M}_2)}, \end{aligned} \quad (1.8)$$

where the first term in equation 1.8 is named the Bayes factor, and the second term is the prior odds. Observe that the Bayes factor is a ratio of ordinates for \mathbf{y} under different models. Then, the Bayes factor is a measure of relative sample evidence in favor of model 1 compared to model 2.

However, we still need to calculate $p(\mathbf{y}|\mathcal{M}_m) = \int_{\boldsymbol{\Theta}} p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}_m) \pi(\boldsymbol{\theta}|\mathcal{M}_m) d\boldsymbol{\theta} = \mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}_m)]$. For this integral to be meaningful, the prior must be proper. Using improper prior has unintended consequences when comparing models, for instance, parsimonious models are favored by posterior odds or Bayes factors depend on units of measure (see Chapter 3).

A nice feature of comparing models using posterior odds is that if we have an exhaustive set of competing models such that $\sum_{m=1}^M \pi(\mathcal{M}_m|\mathbf{y}) = 1$, then we can recover $\pi(\mathcal{M}_m|\mathbf{y})$ without calculating $p(\mathbf{y})$. In particular, given two models \mathcal{M}_1 and \mathcal{M}_2 such that $\pi(\mathcal{M}_1|\mathbf{y}) + \pi(\mathcal{M}_2|\mathbf{y}) = 1$. Then, $\pi(\mathcal{M}_1|\mathbf{y}) = \frac{PO_{12}}{1+PO_{12}}$ and $\pi(\mathcal{M}_2|\mathbf{y}) = 1 - \pi(\mathcal{M}_1|\mathbf{y})$. In general, $\pi(\mathcal{M}_m|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{M}_m) \times \pi(\mathcal{M}_m)}{\sum_{l=1}^M p(\mathbf{y}|\mathcal{M}_l) \times \pi(\mathcal{M}_l)}$. These posterior model probabilities can be used to perform Bayesian model averaging.

Table 1.1 shows guidelines for the interpretation of $2 \log(PO_{12})$ [120]. This transformation is done to replicate the structure of the likelihood ratio test statistic. However, posterior odds do not require nested models as the likelihood ratio test does. Observe that the posterior odds ratio is a relative criterion, that is, we specify an exhaustive set of competing models, and compare them. However, we may want to check the performance of a model in its own or use a non-informative prior. In this case, we can use the *posterior predictive p-value* [73, 74].⁶

⁶[9] show potential issues due to using data twice in the construction of the predictive p values. They also present alternative proposals, for instance, *the partial posterior predictive p value*.

TABLE 1.1
Kass and Raftery guidelines.

$2 \times \log(PO_{12})$	PO_{12}	Evidence against \mathcal{M}_2
0 to 2	1 to 3	Not worth more than a bare mention
2 to 6	3 to 20	Positive
6 to 10	20 to 150	Strong
> 10	> 150	Very strong

Notes: [120] proposed these guidelines for model selection using posterior odds in a Bayesian framework.

The intuition behind the predictive p-value is simple: analyze discrepancy between model's assumptions and data by checking a potential extreme tail-area probability. Observe that this approach does not check if a model is true, its focus is on potential discrepancies between a model and the data at hand.

This is done simulating pseudo-data from our sampling model $(\mathbf{y}^{(s)}, s = 1, 2, \dots, S)$ using draws from the posterior distribution, and then calculating a discrepancy measure, $D(\mathbf{y}^{(s)}, \boldsymbol{\theta})$, to estimate the posterior predictive p-value, $p_D(\mathbf{y}) = P[D(\mathbf{y}^{(s)}, \boldsymbol{\theta}) \geq D(\mathbf{y}, \boldsymbol{\theta})]$ using the proportion of the S draws for which $D(\mathbf{y}^{(s)}, \boldsymbol{\theta}^{(s)}) \geq D(\mathbf{y}, \boldsymbol{\theta}^{(s)})$. Extreme tail probabilities ($p(D_y) \leq 0.05$ or $p(D_y) \geq 0.95$) suggest potential discrepancy between the data and the model. [74] also suggest the posterior predictive p-value based on the *minimum discrepancy*, $D_{min}(\mathbf{y}) = \min_{\boldsymbol{\theta}} D(\mathbf{y}, \boldsymbol{\theta})$, and the *average discrepancy* statistic $D(\mathbf{y}) = \mathbb{E}[D(\mathbf{y}, \boldsymbol{\theta})] = \int_{\Theta} D(\mathbf{y}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}$. These alternatives can be more computational demanding.

The Bayesian approach is also suitable to get probabilistic predictions, that is, we can obtain a posterior predictive density

$$\begin{aligned}\pi(\mathbf{Y}_0|\mathbf{y}, \mathcal{M}_m) &= \int_{\Theta} \pi(\mathbf{Y}_0, \boldsymbol{\theta}|\mathbf{y}, \mathcal{M}_m) d\boldsymbol{\theta} \\ &= \int_{\Theta} \pi(\mathbf{Y}_0|\boldsymbol{\theta}, \mathbf{y}, \mathcal{M}_m) \pi(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M}_m) d\boldsymbol{\theta}. \end{aligned} \quad (1.9)$$

Observe that equation 1.9 is again an expectation $\mathbb{E}[\pi(\mathbf{Y}_0|\boldsymbol{\theta}, \mathbf{y}, \mathcal{M}_m)]$, this time using the posterior distribution. Therefore, the Bayesian approach takes estimation error into account when performing prediction.

As we have shown many times, expectation (integration) is a common feature in Bayesian inference. That is why the remarkable relevance of computation based on *Monte Carlo integration* in the Bayesian framework (see Chapter 4).

Bayesian model average (BMA) allows considering model uncertainty in prediction or any unknown probabilistic object. In the prediction case,

$$\pi(\mathbf{Y}_0|\mathbf{y}) = \sum_{m=1}^M \pi(\mathcal{M}_m|\mathbf{y}) \pi(\mathbf{Y}_0|\mathbf{y}, \mathcal{M}_m), \quad (1.10)$$

and parameters case,

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \sum_{m=1}^M \pi(\mathcal{M}_m|\mathbf{y})\pi(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M}_m), \quad (1.11)$$

where

$$\mathbb{E}(\boldsymbol{\theta}|\mathbf{y}) = \sum_{m=1}^M \hat{\boldsymbol{\theta}}_m \pi(\mathcal{M}_m|\mathbf{y}), \quad (1.12)$$

and

$$Var(\theta_k|\mathbf{y}) = \sum_{m=1}^M \pi(\mathcal{M}_m|\mathbf{y}) \widehat{Var}(\theta_{km}|\mathbf{y}, \mathcal{M}_m) + \sum_{m=1}^M \pi(\mathcal{M}_m|\mathbf{y}) (\hat{\theta}_{km} - \mathbb{E}[\theta_{km}|\mathbf{y}])^2, \quad (1.13)$$

$\hat{\boldsymbol{\theta}}_m$ is the posterior mean and $\widehat{Var}(\theta_{km}|\mathbf{y}, \mathcal{M}_m)$ is the posterior variance of the k -th element of $\boldsymbol{\theta}$ under model \mathcal{M}_m .

Observe how the variance in equation 1.13 encloses extra variability due to potential differences between mean posterior estimates associated with each model, and the posterior mean involving model uncertainty in equation 1.12.

A nice advantage of the Bayesian approach, which is very useful in *state space representations* (see Chapter 8), is the way that the posterior distribution updates with new sample information. Given $\mathbf{y} = \mathbf{y}_{1:t+1}$ a sequence of observations from 1 to $t+1$, then

$$\begin{aligned} \pi(\boldsymbol{\theta}|\mathbf{y}_{1:t+1}) &\propto p(\mathbf{y}_{1:t+1}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}) \\ &= p(y_{t+1}|\mathbf{y}_{1:t}, \boldsymbol{\theta}) \times p(\mathbf{y}_{1:t}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}) \\ &\propto p(y_{t+1}|\mathbf{y}_{1:t}, \boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}|\mathbf{y}_{1:t}). \end{aligned} \quad (1.14)$$

We observe in Equation 1.14 that the new prior is just the posterior distribution using the previous observations. This is particular useful under the assumption of *conditional independence*, that is, $y_{t+1} \perp \mathbf{y}_{1:t} | \boldsymbol{\theta}$, then $p(y_{t+1}|\mathbf{y}_{1:t}, \boldsymbol{\theta}) = p(y_{t+1}|\boldsymbol{\theta})$ such that the posterior can be recovered recursively [163]. This facilities online updating due to all information up to t being in $\boldsymbol{\theta}$. Then, $\pi(\boldsymbol{\theta}|\mathbf{y}_{1:t+1}) \propto p(y_{t+1}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}|\mathbf{y}_{1:t}) \propto \prod_{h=1}^{t+1} p(y_h|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta})$. This recursive expression can be calculated faster at some specific point in time t compared to a batch mode algorithm, which requires processing simultaneously all information up to t .

It is also important to wonder about the sampling properties of “Bayesian estimators”. This topic has attracted attention of statisticians and econometricians long time ago. For instance, asymptotic posterior concentration at the population parameter vector is discussed by [20]. Convergence of posterior distributions is stated by the Bernstein-von Mises theorem [132, 213], which creates a link between *credible intervals (sets)* and confidence intervals (sets),

where a credible interval is an interval in the domain of the posterior distribution within which an unknown parameter falls with a particular probability. Credible intervals treat bounds as fixed and parameters as random, whereas confidence intervals reverse this. There are many settings in parametric models where Bayesian credible intervals with α level converge asymptotically to confidence intervals at α level. This suggests that Bayesian inference is asymptotically correct from a sampling perspective in these settings.

A heuristic approach to show this in the simplest case where we assume random sampling and $\theta \in \mathcal{R}$ is the following: $p(\mathbf{y}|\theta) = \prod_{i=1}^N p(y_i|\theta)$ such that the log likelihood is $l(\mathbf{y}|\theta) = \log p(\mathbf{y}|\theta) = \sum_{i=1}^N \log p(y_i|\theta) = N \times \bar{l}(\mathbf{y}|\theta)$ where $\bar{l} \equiv \frac{1}{N} \sum_{i=1}^N \log p(y_i|\theta)$ is the mean likelihood.⁷ Then, the posterior distribution is proportional to

$$\begin{aligned}\pi(\theta|\mathbf{y}) &\propto p(\mathbf{y}|\theta) \times \pi(\theta) \\ &= \exp\left\{N \times \bar{l}(\mathbf{y}|\theta)\right\} \times \pi(\theta).\end{aligned}\quad (1.15)$$

Observe that as the sample size gets large, that is, $N \rightarrow \infty$, the exponential term should dominate the prior distribution as long as this does not depend on N such that the likelihood determines the posterior distribution asymptotically.

Maximum likelihood theory shows that $\lim_{N \rightarrow \infty} \bar{l}(\mathbf{y}|\theta) \rightarrow \bar{l}(\mathbf{y}|\theta_0)$ where θ_0 is the population parameter of the data generating process. In addition, doing a second order Taylor expansion of the log likelihood at the Maximum likelihood estimator,

$$\begin{aligned}l(\mathbf{y}|\theta) &\approx l(\mathbf{y}|\hat{\theta}) + \frac{dl(\mathbf{y}|\theta)}{d\theta}\Big|_{\hat{\theta}} (\theta - \hat{\theta}) + \frac{1}{2} \frac{d^2l(\mathbf{y}|\theta)}{d\theta^2}\Big|_{\hat{\theta}} (\theta - \hat{\theta})^2 \\ &= l(\mathbf{y}|\hat{\theta}) + \frac{1}{2} \sum_{i=1}^N \frac{d^2l(y_i|\theta)}{d\theta^2}\Big|_{\hat{\theta}} (\theta - \hat{\theta})^2 \\ &= l(\mathbf{y}|\hat{\theta}) - \frac{1}{2} N [-\bar{l}''\Big|_{\hat{\theta}}] (\theta - \hat{\theta})^2 \\ &= l(\mathbf{y}|\hat{\theta}) - \frac{N}{2\sigma^2} (\theta - \hat{\theta})^2\end{aligned}$$

where $\frac{dl(\mathbf{y}|\theta)}{d\theta}\Big|_{\hat{\theta}} = 0$, $\bar{l}'' \equiv \frac{1}{N} \sum_{i=1}^N \frac{d^2l(y_i|\theta)}{d\theta^2}\Big|_{\hat{\theta}}$ and $\sigma^2 := [-\bar{l}''\Big|_{\hat{\theta}}]^{-1}$.⁸ Then,

⁷Take into account that in the likelihood function the argument is θ . However, we keep the notation for facility in exposition.

⁸The last definition follows from standard theory in maximum likelihood estimation (see [33, Chap. 10] and [221, Chap. 13]).

$$\begin{aligned}
\pi(\theta|\mathbf{y}) &\propto \exp\{l(\mathbf{y}|\theta)\} \times \pi(\theta) \\
&\approx \exp\left\{l(\mathbf{y}|\hat{\theta}) - \frac{N}{2\sigma^2}(\theta - \hat{\theta})^2\right\} \times \pi(\theta) \\
&\propto \exp\left\{-\frac{N}{2\sigma^2}(\theta - \hat{\theta})^2\right\} \times \pi(\theta)
\end{aligned}$$

Observe that we have that the posterior density is proportional to the kernel of a normal density with mean $\hat{\theta}$ and variance σ^2/N as long as $\pi(\hat{\theta}) \neq 0$. This kernel dominates as the sample size gets large due to N in the exponential term. Observe that the prior should not exclude values of θ that are logically possible, such as $\hat{\theta}$.

1.2.1 Example: Health insurance

Suppose that you are analyzing to buy a health insurance next year. To make a better decision you want to know *what is the probability that you visit your Doctor at least once next year?* To answer this question you have records of the number of times that you have visited your Doctor the last 5 years, $\mathbf{y} = \{0, 3, 2, 1, 0\}$. How to proceed?

Assuming that this is a random sample⁹ from a data generating process (statistical model) that is Poisson, that is, $Y_i \sim P(\lambda)$, and your probabilistic prior beliefs about λ are well described by a Gamma distribution with shape and scale parameters α_0 and β_0 , $\lambda \sim G(\alpha_0, \beta_0)$, then, you are interested in calculating the probability $P(Y_0 > 0|\mathbf{y})$. You need to calculate the posterior predictive density $\pi(Y_0|\mathbf{y})$ to answer this question in a Bayesian way.

In this example, $p(\mathbf{y}|\lambda)$ is Poisson, and $\pi(\lambda)$ is Gamma. Then, using 1.9

$$\pi(Y_0|\mathbf{y}) = \int_0^\infty \frac{\lambda^{y_0} \exp\{-\lambda\}}{y_0!} \times \pi(\lambda|\mathbf{y}) d\lambda,$$

where the posterior distribution is $\pi(\lambda|\mathbf{y}) \propto \lambda^{\sum_{i=1}^N y_i + \alpha_0 - 1} \exp\left\{-\lambda \left(\frac{\beta_0 N + 1}{\beta_0}\right)\right\}$ by equation 1.3.

Observe that the last expression is the kernel of a Gamma distribution with parameters $\alpha_n = \sum_{i=1}^N y_i + \alpha_0$ and $\beta_n = \frac{\beta_0}{\beta_0 N + 1}$. Given that $\int_0^\infty \pi(\lambda|\mathbf{y}) d\lambda = 1$, then the constant of proportionality in the last expression is $\Gamma(\alpha_n) \beta_n^{\alpha_n}$, where $\Gamma(\cdot)$ is the gamma function. Thus, the posterior density function $\pi(\lambda|\mathbf{y})$ is $G(\alpha_n, \beta_n)$.

Observe that

⁹Independent and identically distributed draws.

$$\begin{aligned}
\mathbb{E}[\lambda|\mathbf{y}] &= \alpha_n \beta_n \\
&= \left(\sum_{i=1}^N y_i + \alpha_0 \right) \left(\frac{\beta_0}{\beta_0 N + 1} \right) \\
&= \bar{y} \left(\frac{N\beta_0}{N\beta_0 + 1} \right) + \alpha_0 \beta_0 \left(\frac{1}{N\beta_0 + 1} \right) \\
&= w\bar{y} + (1-w)\mathbb{E}[\lambda],
\end{aligned}$$

where \bar{y} is the sample mean, which is the maximum likelihood estimator of λ in this example, $w = \left(\frac{N\beta_0}{N\beta_0 + 1} \right)$ and $\mathbb{E}[\lambda] = \alpha_0 \beta_0$ is the prior mean. The posterior mean is a weighted average of the maximum likelihood estimator (sample information) and the prior mean. Observe that $\lim_{N \rightarrow \infty} w = 1$, that is, the sample information asymptotically dominates.

The predictive distribution is

$$\begin{aligned}
\pi(Y_0|\mathbf{y}) &= \int_0^\infty \frac{\lambda^{y_0} \exp\{-\lambda\}}{y_0!} \times \frac{1}{\Gamma(\alpha_n)\beta_n^{\alpha_n}} \lambda^{\alpha_n-1} \exp\{-\lambda/\beta_n\} d\lambda \\
&= \frac{1}{y_0! \Gamma(\alpha_n) \beta_n^{\alpha_n}} \int_0^\infty \lambda^{y_0+\alpha_n-1} \exp\left\{-\lambda \left(\frac{1+\beta_n}{\beta_n} \right)\right\} d\lambda \\
&= \frac{\Gamma(y_0 + \alpha_n) \left(\frac{\beta_n}{\beta_n+1} \right)^{y_0+\alpha_n}}{y_0! \Gamma(\alpha_n) \beta_n^{\alpha_n}} \\
&= \binom{y_0 + \alpha_n - 1}{y_0} \left(\frac{\beta_n}{\beta_n + 1} \right)^{y_0} \left(\frac{1}{\beta_n + 1} \right)^{\alpha_n}.
\end{aligned}$$

The third equality follows from the kernel of a Gamma density, and the fourth from $\binom{y_0 + \alpha_n - 1}{y_0} = \frac{(y_0 + \alpha_n - 1)(y_0 + \alpha_n - 2)\dots\alpha_n}{y_0!} = \frac{\Gamma(y_0 + \alpha_n)}{\Gamma(\alpha_n)y_0!}$ using a property of the Gamma function.

Observe that this is a Negative Binomial density, that is $Y_0|\mathbf{y} \sim NB(\alpha_n, p_n)$ where $p_n = \frac{\beta_n}{\beta_n + 1}$.

Up to this point, we have said nothing about the hyperparameters, which are required to give a concrete response to this exercise. Thus, we show two approaches to set them. First, we set $\alpha_0 = 0.001$ and $\beta_0 = 1/0.001$ which imply vague prior information about λ due to having a large degree of variability compared to the mean information.¹⁰ In particular, $\mathbb{E}[\lambda] = 1$ and $\text{Var}[\lambda] = 1000$.

In this setting, $P(Y_0 > 0|\mathbf{y}) = 1 - P(Y_0 = 0|\mathbf{y}) \approx 0.67$. That is, the probability of visiting the Doctor at least once next year is approximately 0.67.

¹⁰We should be aware that there may be technical problems using this kind of hyperparameters in this setting [77].

Another approach is using *Empirical Bayes*, where we set the hyperparameters maximizing the logarithm of the marginal likelihood,¹¹ that is, $\left[\hat{\alpha}_0 \hat{\beta}_0 \right]^\top = \underset{\alpha_0, \beta_0}{\operatorname{argmax}} \ln p(\mathbf{y})$ where

$$\begin{aligned} p(\mathbf{y}) &= \int_0^\infty \left\{ \frac{1}{\Gamma(\alpha_0)\beta_0^{\alpha_0}} \lambda^{\alpha_0-1} \exp\{-\lambda/\beta_0\} \prod_{i=1}^N \frac{\lambda^{y_i} \exp\{-\lambda\}}{y_i!} \right\} d\lambda \\ &= \frac{\int_0^\infty \lambda^{\sum_{i=1}^N y_i + \alpha_0 - 1} \exp\left\{-\lambda \left(\frac{\beta_0 N + 1}{\beta_0}\right)\right\} d\lambda}{\Gamma(\alpha_0)\beta_0^{\alpha_0} \prod_{i=1}^N y_i!} \\ &= \frac{\Gamma(\sum_{i=1}^N y_i + \alpha_0) \left(\frac{\beta_0}{N\beta_0 + 1}\right)^{\sum_{i=1}^N y_i} \left(\frac{1}{N\beta_0 + 1}\right)^{\alpha_0}}{\Gamma(\alpha_0) \prod_{i=1}^N y_i} \end{aligned}$$

Using the empirical Bayes approach, we get $\hat{\alpha}_0 = 51.8$ and $\hat{\beta}_0 = 0.023$, then $P(Y_0 > 0|\mathbf{y}) = 1 - P(Y_0 = 0|\mathbf{y}) \approx 0.70$.

Observe that we can calculate the posterior odds comparing the model using an Empirical Bayes prior (model 1) versus the vague prior (model 2). We assume that $\pi(\mathcal{M}_1) = \pi(\mathcal{M}_2) = 0.5$, then

$$\begin{aligned} PO_{12} &= \frac{p(\mathbf{y}|\text{Empirical Bayes})}{p(\mathbf{y}|\text{Vague prior})} \\ &= \frac{\frac{\Gamma(\sum_{i=1}^N y_i + 51.807) \left(\frac{0.023}{N \times 0.023 + 1}\right)^{\sum_{i=1}^N y_i} \left(\frac{1}{N \times 0.023 + 1}\right)^{51.807}}{\Gamma(51.807)}}{\frac{\Gamma(\sum_{i=1}^N y_i + 0.001) \left(\frac{1/0.001}{N/0.001 + 1}\right)^{\sum_{i=1}^N y_i} \left(\frac{1}{N/0.001 + 1}\right)^{0.001}}{\Gamma(0.001)}} \\ &\approx 919. \end{aligned}$$

Then, $2 \times \log(PO_{12}) = 13.64$, there is very strong evidence against the vague prior model (see Table 1.1). In particular, $\pi(\text{Empirical Bayes}|\mathbf{y}) = \frac{919}{1+919} = 0.999$ and $\pi(\text{Vague prior}|\mathbf{y}) = 1 - 0.999 = 0.001$. These probabilities can be used to perform Bayesian model average (BMA). In particular,

$$\begin{aligned} \mathbb{E}(\lambda|\mathbf{y}) &= 1.2 \times 0.999 + 1.2 \times 0.001 = 1.2 \\ Var(\lambda|\mathbf{y}) &= 0.025 \times 0.999 + 0.24 \times 0.001 \\ &\quad + (1.2 - 1.2)^2 \times 0.999 + (1.2 - 1.2)^2 \times 0.001 = 0.025. \end{aligned}$$

The BMA predictive distribution is a mix of negative binomial distributions, that is, $y_0|\mathbf{y} \sim 0.999 \times NB(57.8, 0.02) + 0.001 \times NB(6.001, 0.17)$.

¹¹Empirical Bayes methods are criticized due to double-using the data. First to set the hyperparameters, and second, to perform Bayesian inference.

R code. Health insurance, predictive distribution using vague hyperparameters

```
1 set.seed(010101)
2 y <- c(0, 3, 2, 1, 0) # Data
3 N <- length(y)
4 ProbBo <- function(y, a0, b0){
5   N <- length(y)
6   #sample size
7   an <- a0 + sum(y)
8   # Posterior shape parameter
9   bn <- b0 / ((b0 * N) + 1)
10  # Posterior scale parameter
11  p <- bn / (bn + 1)
12  # Probability negative binomial density
13  Pr <- 1 - pnbinom(0, size=an,prob=(1 - p))
14  # Probability of visiting the Doctor at least once next
15  # year
16  # Observe that in R there is a slightly different
17  # parametrization.
18  return(Pr)
19 }
20 # Using a vague prior:
21 a0 <- 0.001 # Prior shape parameter
22 b0 <- 1 / 0.001 # Prior scale parameter
23 PriMeanV <- a0 * b0 # Prior mean
24 PriVarV <- a0 * b0^2 # Prior variance
25 Pp <- ProbBo(y, a0 = 0.001, b0 = 1 / 0.001)
26 # This setting is vague prior information.
27 Pp
28 0.67
```

R code. Health insurance, predictive distribution using empirical Bayes

```

1 # Using Empirical Bayes
2 LogMgLik <- function(theta, y){
3   N <- length(y)
4   #sample size
5   a0 <- theta[1]
6   # prior shape hyperparameter
7   b0 <- theta[2]
8   # prior scale hyperparameter
9   an <- sum(y) + a0
10  # posterior shape parameter
11  if(a0 <= 0 || b0 <= 0){
12    #Avoiding negative values
13    lnp <- -Inf
14  }else{
15    lnp <- lgamma(an) + sum(y)*log(b0/(N*b0+1)) - a0*log(N*b0
16      +1) - lgamma(a0)
17  }
18  # log marginal likelihood
19  return(-lnp)
20 }
21 theta0 <- c(0.01, 1/0.1)
22 # Initial values
23 control <- list(maxit = 1000)
24 # Number of iterations in optimization
25 EmpBay <- optim(theta0, LogMgLik, method = "BFGS", control =
26   control, hessian = TRUE, y = y)
27 # Optimization
28 EmpBay$convergence
29 0
30 a0EB <- EmpBay$par[1]
31 # Prior shape using empirical Bayes
32 a0EB
33 51.81
34 b0EB <- EmpBay$par[2]
35 # Prior scale using empirical Bayes
36 b0EB
37 0.023
38 PriMeanEB <- a0EB * b0EB
39 # Prior mean
40 PriVarEB <- a0EB * b0EB^2
41 # Prior variance
42 PpEB <- ProbBo(y, a0 = a0EB, b0 = b0EB)
43 # This setting is using empirical Bayes.
44 PpEB
45 0.70

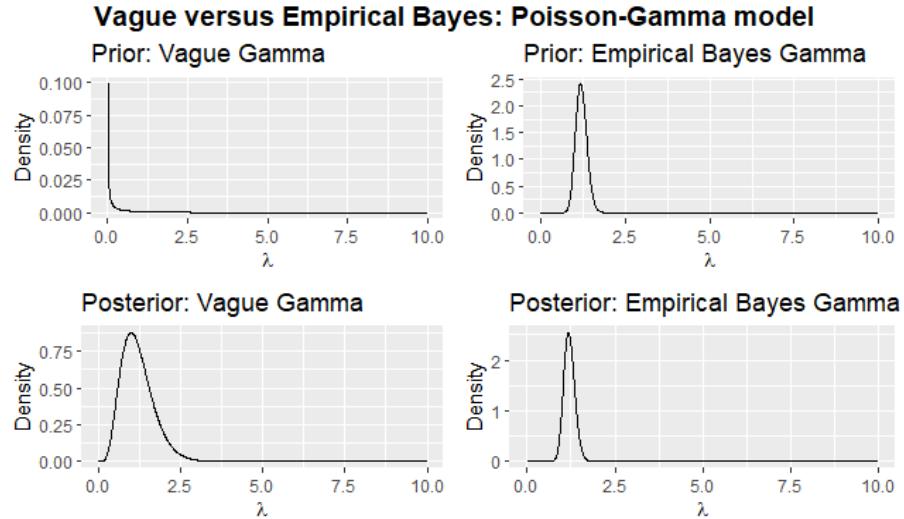
```

R code. Health insurance, density plots

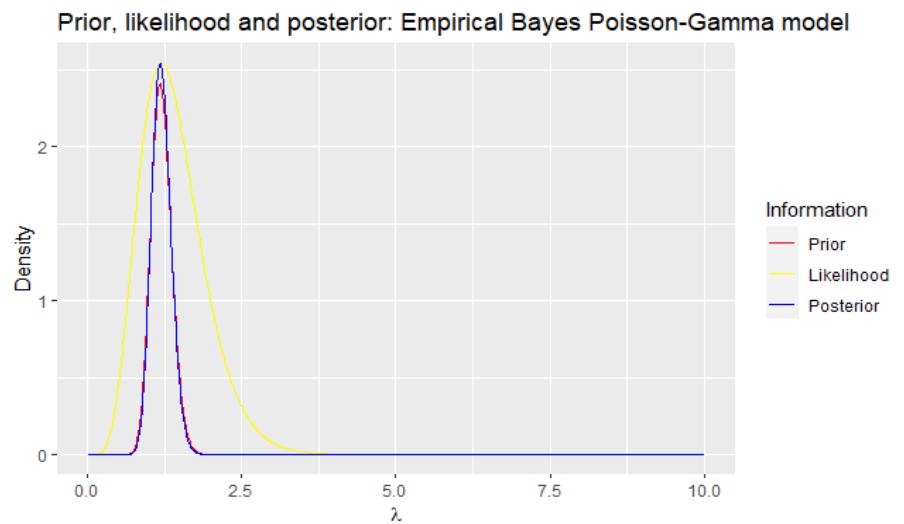
```

1 # Density figures:
2 # This code helps plotting densities
3 lambda <- seq(0.01, 10, 0.01)
4 # Values of lambda
5 VaguePrior <- dgamma(lambda, shape=a0, scale = b0)
6 EBPrior <- dgamma(lambda, shape=a0EB, scale = b0EB)
7 PosteriorV <- dgamma(lambda, shape = a0 + sum(y), scale = b0
8 / ((b0 * N) + 1))
9 PosteriorEB <- dgamma(lambda, shape = a0EB+sum(y), scale =
b0EB / ((b0EB * N) + 1))
10 # Likelihood function
11 Likelihood <- function(theta, y){
12 LogL <- dpois(y, theta, log = TRUE)
13 Lik <- prod(exp(LogL))
14 return(Lik)
15 }
16 Liks <- sapply(lambda, function(par) {Likelihood(par, y = y)
})
17 Sc <- max(PosteriorEB)/max(Liks)
18 #Scale for displaying in figure
19 LiksScale <- Liks * Sc
20 data <- data.frame(cbind(lambda, VaguePrior, EBPrior,
PosteriorV, PosteriorEB, LiksScale)) #Data frame
21 require(ggplot2) # Cool figures
22 require(latex2exp) # LaTeX equations in figures
23 require(ggpubr) # Multiple figures in one page
24 fig1 <- ggplot(data = data, aes(lambda, VaguePrior)) + geom_
line() + xlab(TeX("\lambda")) + ylab("Density") +
ggtitle("Prior: Vague Gamma")
25 fig2 <- ggplot(data = data, aes(lambda, EBPrior)) + geom_
line() + xlab(TeX("\lambda")) + ylab("Density") +
ggtitle("Prior: Empirical Bayes Gamma")
26 fig3 <- ggplot(data = data, aes(lambda, PosteriorV)) + geom_
line() + xlab(TeX("\lambda")) + ylab("Density") +
ggtitle("Posterior: Vague Gamma")
27 fig4 <- ggplot(data = data, aes(lambda, PosteriorEB)) + geom_
line() + xlab(TeX("\lambda")) + ylab("Density") +
ggtitle("Posterior: Empirical Bayes Gamma")
28 FIG <- ggarrange(fig1, fig2, fig3, fig4, ncol = 2, nrow = 2)
29 annotate_figure(FIG, top = text_grob("Vague versus Empirical
Bayes: Poisson-Gamma model", color = "black", face =
"bold", size = 14))
30 dataNew <- data.frame(cbind(rep(lambda, 3), c(EBPrior,
PosteriorEB, LiksScale), rep(1:3, each = 1000))) #Data
frame
31 colnames(dataNew) <- c("Lambda", "Density", "Factor")
32 dataNew$Factor <- factor(dataNew$Factor, levels=c("1", "3",
"2"),
33 labels=c("Prior", "Likelihood", "Posterior"))
34 ggplot(data = dataNew, aes_string(x = "Lambda", y = "Density
", group = "Factor")) + geom_line(aes(color = Factor)) +
xlab(TeX("\lambda")) + ylab("Density") + ggtitle("Prior,
likelihood and posterior: Empirical Bayes Poisson
-Gamma model") + guides(color=guide_legend(title=
"Information")) + scale_color_manual(values = c("red",
"yellow", "blue"))

```

**FIGURE 1.2**

Vague versus Empirical Bayes: Poisson-Gamma model.

**FIGURE 1.3**

Prior, likelihood and posterior: Empirical Bayes Poisson-Gamma model.

Figure 1.2 displays prior and posterior densities based on vague and Empirical Bayes hyperparameters. We see that prior and posterior densities using the latter are more informative as expected.

Figure 1.3 shows the prior, scaled likelihood and posterior densities of λ based on the hyperparameters of the Empirical Bayes approach. The posterior density is a compromise between prior and sample information.

R code. Health insurance, Predictive density

```

1 # Predictive distributions
2 PredDen <- function(y, y0, a0, b0){
3   N <- length(y)
4   #sample size
5   an <- a0 + sum(y)
6   # Posterior shape parameter
7   bn <- b0 / ((b0 * N) + 1)
8   # Posterior scale parameter
9   p <- bn / (bn + 1)
10  # Probability negative binomial density
11  Pr <- dnbinom(y0, size=an, prob=(1 - p))
12  # Predictive density
13  # Observe that in R there is a slightly different
     parametrization.
14  return(Pr)
15 }
16 y0 <- 0:10
17 PredVague <- PredDen(y=y, y0=y0, a0=a0, b0=b0)
18 PredEB <- PredDen(y=y, y0=y0, a0=aOEB, b0=bOEB)
19 dataPred <- as.data.frame(cbind(y0, PredVague, PredEB))
20 colnames(dataPred) <- c("y0", "PredictiveVague", "
     PredictiveEB")
21 ggplot(data = dataPred) + geom_point(aes(y0, PredictiveVague
     , color = "red")) +
22 xlab(TeX("\$y_0\$")) + ylab("Density") + ggtitle("Predictive
     density: Vague and Empirical Bayes priors") + geom_point
     (aes(y0, PredictiveEB, color = "yellow")) +
23 guides(color = guide_legend(title="Prior")) + scale_color_
     manual(labels = c("Vague", "Empirical Bayes"), values =
     c("red", "yellow")) + scale_x_continuous(breaks=seq
     (0,10,by=1))

```

Figure 1.4 displays the predictive probability mass of not having any visits to a physician the next year, having one, two, and so on using Empirical Bayes and vague hyperparameters. The predictive probability of not having any visits are approximately equal to 30% and 33% based on the Empirical Bayes and vague hyperparameters.

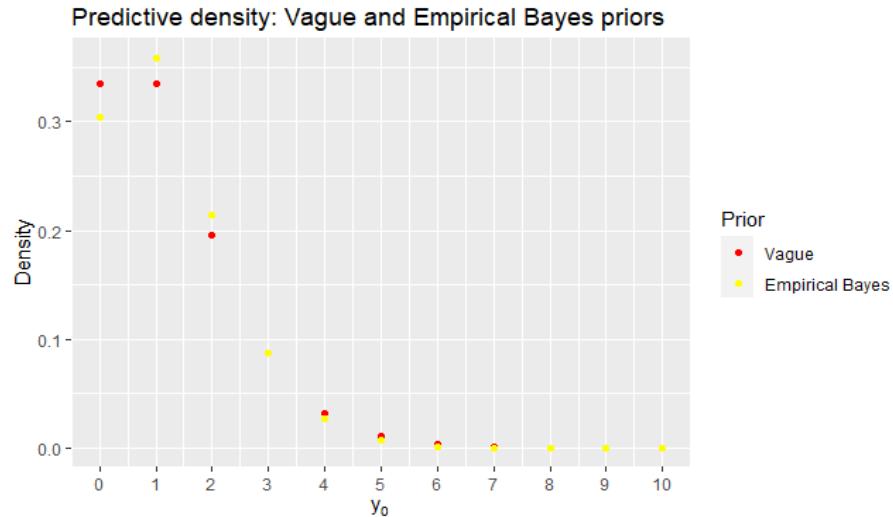


FIGURE 1.4
Predictive density: Vague and Empirical Bayes.

R code. Health insurance, Bayesian model average

```

1 # Posterior odds: Vague vs Empirical Bayes
2 P012 <- exp(-LogMgLik(c(a0EB, b0EB), y = y))/exp(-LogMgLik(c
  (a0, b0), y = y))
3 P012
4 919
5 PostProMEM <- P012/(1 + P012)
6 PostProMEM
7 0.998
8 # Posterior model probability Empirical Bayes
9 PostProbMV <- 1 - PostProMEM
10 PostProbMV
11 0.002
12 # Posterior model probability vague prior
13 # Bayesian model average (BMA)
14 PostMeanEB <- (a0EB + sum(y)) * (b0EB / (b0EB * N + 1))
15 # Posterior mean Empirical Bayes
16 PostMeanV <- (a0 + sum(y)) * (b0 / (b0 * N + 1))
17 # Posterior mean vague priors
18 BMAMean <- PostProMEM * PostMeanEB + PostProbMV * PostMeanV
19 BMAMean
20 1.2
21 # BMA posterior mean
22 PostVarEB <- (a0EB + sum(y)) * (b0EB/(b0EB * N + 1))^2
23 # Posterior variance Empirical Bayes
24 PostVarV <- (a0 + sum(y)) * (b0 / (b0 * N + 1))^2
25 # Posterior variance vague prior
26 BMAMVar <- PostProMEM * PostVarEB + PostProbMV*PostVarV +
  PostProMEM * (PostMeanEB - BMAMean)^2 + PostProbMV * (
    PostMeanV - BMAMean)^2
27 # BMA posterior variance
28 BMAMVar
29 0.025

```

R code. Health insurance, Bayesian model average

```

1 # BMA: Predictive
2 BMAPred <- PostProMEM * PredEB+PostProbMV * PredVague
3 dataPredBMA <- as.data.frame(cbind(y0, BMAPred))
4 colnames(dataPredBMA) <- c("y0", "PredictiveBMA")
5 ggplot(data = dataPredBMA) + geom_point(aes(y0,
  PredictiveBMA, color = "red")) + xlab(TeX("\$y_0\$")) +
  ylab("Density") + ggtitle("Predictive density: BMA") +
  guides(color = guide_legend(title="BMA")) + scale_color_
  manual(labels = c("Probability"), values = c("red")) +
  scale_x_continuous(breaks=seq(0,10,by=1))
6

```

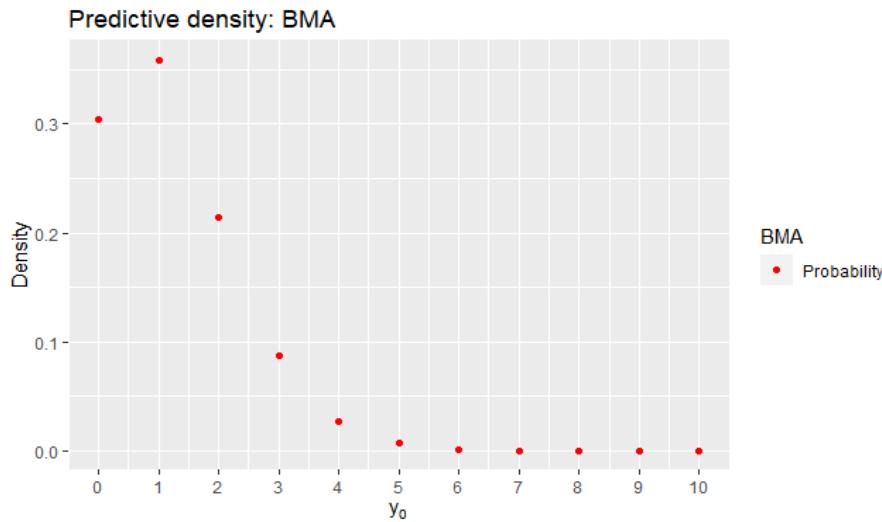


FIGURE 1.5
Bayesian model average: Predictive density.

Figure 1.5 displays the predictive density using Bayesian model average based on the vague and Empirical Bayes hyperparameters. This figure essentially resembles the predictive probability mass function based on the Empirical Bayes framework, as the posterior model probability for that setting is nearly one.

Figure 1.6 displays how the posterior distribution updates given new sample information based on an initial non-informative prior (iteration 1). We

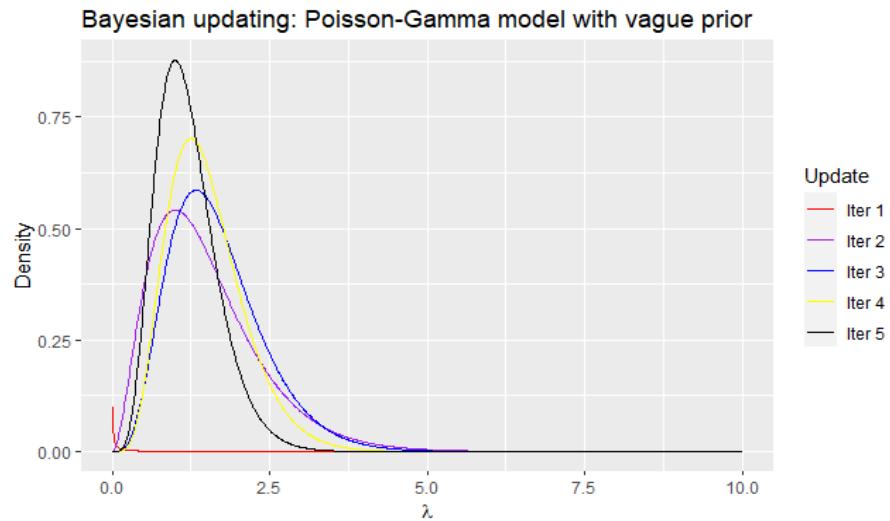


FIGURE 1.6
Bayesian updating: Posterior densities.

see that iteration 5 is based on all the sample information in our example, as a consequence, the posterior density in iteration 5 is equal to the posterior density in Figure 1.3.

R code. Health insurance, Bayes updating

```

1 # Bayesian updating
2 BayUp <- function(y, lambda, a0, b0){
3   N <- length(y)
4   #sample size
5   an <- a0 + sum(y)
6   # Posterior shape parameter
7   bn <- b0 / ((b0 * N) + 1)
8   # Posterior scale parameter
9   p <- dgamma(lambda, shape = an, scale = bn)
10  # Posterior density
11  return(list(Post = p, a0New = an, b0New = bn))
12 }
13 PostUp <- NULL
14 for(i in 1:N){
15   if(i == 1){
16     PostUp <- BayUp(y[i], lambda, a0 = 0.001, b0 = 1/0.001)
17   }
18   else{
19     PostUp <- BayUp(y[i], lambda, a0 = PostUp$a0New, b0 =
20     PostUp$b0New)
21   }
22 PostUp <- cbind(PostUp, PostUp$Post)
23 DataUp <- data.frame(cbind(rep(lambda, 5), c(PostUp), rep
24   (1:5, each = 1000))) #Data frame
25 colnames(DataUp) <- c("Lambda", "Density", "Factor")
26 DataUp$Factor <- factor(DataUp$Factor, levels=c("1", "2", "3",
27   "4", "5"),
28   labels=c("Iter 1", "Iter 2", "Iter 3", "Iter 4", "Iter 5"))
29 ggplot(data = DataUp, aes_string(x = "Lambda", y = "Density"
30   , group = "Factor")) + geom_line(aes(color = Factor)) +
31   xlab(TeX("\lambda")) + ylab("Density") + ggtitle("Bayesian updating: Poisson-Gamma model with vague prior") + guides(color=guide_legend(title="Update")) + scale_color_manual(values = c("red", "purple", "blue", "yellow", "black"))
32 S <- 100000 # Posterior draws
33 PostMeanLambdaUps <- sapply(1:N, function(i) {mean(sample(
34   lambda, S, replace = TRUE, prob = PostUp[, i]))}) #
35   Posterior mean update i
36 paste("Posterior means using all information and sequential
37   updating are:", round(PostMeanV, 2), "and", round(
38   PostMeanLambdaUps[5], 2), sep = " ")
39 Posterior means using all information and sequential
40   updating are: 1.2 and 1.2
41 PostVarLambdaUps <- sapply(1:N, function(i) {var(sample(
42   lambda, S, replace = TRUE, prob = PostUp[, i]))}) #
43   Posterior variance update i
44 paste("Posterior variances using all information and
45   sequential updating are:", round(PostVarV, 2), "and",
46   round(PostVarLambdaUps[5], 2), sep = " ")
47 Posterior variances using all information and sequential
48   updating are: 0.24 and 0.24

```

1.3 Bayesian reports: Decision theory under uncertainty

The Bayesian framework allows reporting the full posterior distributions. However, some situations demand to report a specific value of the posterior distribution (point estimate), an informative interval (set), point or interval predictions and/or selecting a specific model. Decision theory offers an elegant framework to make a decision regarding what are the optimal posterior values to report [16].

The point of departure is a *loss function*, which is a non-negative real value function whose arguments are the unknown *state of nature* (Θ), and a set of *actions* to be made (\mathcal{A}), that is,

$$L(\boldsymbol{\theta}, a) : \Theta \times \mathcal{A} \rightarrow \mathcal{R}^+.$$

This function is a mathematical expression of the loss of making mistakes. In particular, selecting action $a \in \mathcal{A}$ when $\boldsymbol{\theta} \in \Theta$ is the true. In our case, the unknown state of nature can be parameters, functions of them, future or unknown realizations, models, etc.

From a Bayesian perspective, we should choose the action that minimizes the posterior expected loss ($a^*(\mathbf{y})$), that is, the *posterior risk function* ($\mathbb{E}[L(\boldsymbol{\theta}, a)|\mathbf{y}]$),

$$a^*(\mathbf{y}) = \operatorname{argmin}_{a \in \mathcal{A}} \mathbb{E}[L(\boldsymbol{\theta}, a)|\mathbf{y}],$$

$$\text{where } \mathbb{E}[L(\boldsymbol{\theta}, a)|\mathbf{y}] = \int_{\Theta} L(\boldsymbol{\theta}, a) \pi(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}.^{12}$$

Different loss functions imply different optimal decisions. We illustrate this assuming $\theta \in \mathcal{R}$.

- The quadratic loss function, $L(\theta, a) = [\theta - a]^2$, gives as optimal decision the posterior mean, $a^*(\mathbf{y}) = \mathbb{E}[\theta|\mathbf{y}]$, that is

$$\mathbb{E}[\theta|\mathbf{y}] = \operatorname{argmin}_{a \in \mathcal{A}} \int_{\Theta} [\theta - a]^2 \pi(\theta|\mathbf{y}) d\theta.$$

To get this results, let's use the first condition order, differentiate the risk function with respect to a , interchange differential and integral order, and set this equal to zero, $-2 \int_{\Theta} [\theta - a^*] \pi(\theta|\mathbf{y}) d\theta = 0$ implies that $a^* \int_{\Theta} \pi(\theta|\mathbf{y}) d\theta = a^*(\mathbf{y}) = \int_{\Theta} \theta \pi(\theta|\mathbf{y}) d\theta = \mathbb{E}[\theta|\mathbf{y}]$, that is, the posterior mean is the Bayesian optimal action. This means that we should report the posterior mean as a point estimate of θ when facing the quadratic loss function.

¹²[36] propose Laplace type estimators (LTE) based on the *quasi-posterior*, $p(\boldsymbol{\theta}) = \frac{\exp\{L_n(\boldsymbol{\theta})\}\pi(\boldsymbol{\theta})}{\int_{\Theta} \exp\{L_n(\boldsymbol{\theta})\}\pi(\boldsymbol{\theta})d\boldsymbol{\theta}}$ where $L_n(\boldsymbol{\theta})$ is not necessarily a log-likelihood function. The LTE minimizes the *quasi-posterior risk*.

- The generalized quadratic loss function, $L(\theta, a) = w(\theta)[\theta - a]^2$, where $w(\theta) > 0$ is a weighting function, gives as optimal decision rule the weighted mean. We should follow same steps as the previous result to get $a^*(\mathbf{y}) = \frac{\mathbb{E}[w(\theta) \times \theta | \mathbf{y}]}{\mathbb{E}[w(\theta) | \mathbf{y}]}$. Observe that the weighted average is driven by the weighting function $w(\theta)$.
- The absolute error loss function, $L(\theta, a) = |\theta - a|$, gives as optimal action the posterior median (Exercise 5).
- The generalized absolute error function,

$$L(\theta, a) = \begin{cases} K_0(\theta - a), & \theta - a \geq 0 \\ K_1(a - \theta), & \theta - a < 0 \end{cases}, \quad K_0, K_1 > 0,$$

implies the following risk function,

$$\mathbb{E}[L(\theta, a) | \mathbf{y}] = \int_{-\infty}^a K_1(a - \theta) \pi(\theta | \mathbf{y}) d\theta + \int_a^\infty K_0(\theta - a) \pi(\theta | \mathbf{y}) d\theta.$$

Differentiating with respect to a , interchanging differentials and integrals, and equating to zero,

$$K_1 \int_{-\infty}^{a^*} \pi(\theta | \mathbf{y}) d\theta - K_0 \int_{a^*}^\infty \pi(\theta | \mathbf{y}) d\theta = 0,$$

then, $\int_{-\infty}^{a^*} \pi(\theta | \mathbf{y}) d\theta = \frac{K_0}{K_0 + K_1}$, that is, any $K_0/(K_0 + K_1)$ -percentile of $\pi(\theta | \mathbf{y})$ is an optimal Bayesian estimate of θ .

We can also use decision theory under uncertainty in hypothesis testing. In particular, testing $H_0 : \theta \in \Theta_0$ versus $H_1 : \theta \in \Theta_1$, $\Theta = \Theta_0 \cup \Theta_1$ and $\emptyset = \Theta_0 \cap \Theta_1$, there are two actions of interest, a_0 and a_1 , where a_j denotes no rejecting H_j , $j = \{0, 1\}$.

Given the $0 - K_j$ loss function,

$$L(\theta, a_j) = \begin{cases} 0, & \theta \in \Theta_j \\ K_j, & \theta \in \Theta_j, j \neq i \end{cases},$$

where there is no loss if the right decision is made, for instance, no rejecting H_0 when $\theta \in \Theta_0$, and the loss is K_j when an error is made, for instance, type I error, rejecting the null hypothesis (H_0) when it is true ($\theta \in \Theta_0$), implies a loss equal to K_1 due to picking a_1 , no rejecting H_1 .

The posterior expected loss associated with decision a_j , that is, no rejecting H_j , is $\mathbb{E}[L(\theta, a_j) | \mathbf{y}] = 0 \times P(\Theta_j | \mathbf{y}) + K_j P(\Theta_i | \mathbf{y}) = K_j P(\Theta_i | \mathbf{y})$, $j \neq i$. Therefore, the Bayes optimal decision is the one that gives the smallest posterior expected loss, that is, the null hypothesis is rejected (a_1 is not rejected), when $K_0 P(\Theta_1 | \mathbf{y}) > K_1 P(\Theta_0 | \mathbf{y})$. Given our framework ($\Theta = \Theta_0 \cup \Theta_1$, $\emptyset = \Theta_0 \cap \Theta_1$),

then $P(\Theta_0|\mathbf{y}) = 1 - P(\Theta_1|\mathbf{y})$, and as a consequence, $P(\Theta_1|\mathbf{y}) > \frac{K_1}{K_1+K_0}$, that is, the rejection region of the Bayesian test is $R = \left\{ \mathbf{y} : P(\Theta_1|\mathbf{y}) > \frac{K_1}{K_1+K_0} \right\}$.

Decision theory also helps to construct interval (region) estimates. Let $\Theta_{C(\mathbf{y})} \subset \Theta$ a *credible set* for θ , and $L(\theta, \Theta_{C(\mathbf{y})}) = 1 - \mathbb{1}\{\theta \in \Theta_{C(\mathbf{y})}\}$, where

$$\mathbb{1}\{\theta \in \Theta_{C(\mathbf{y})}\} = \begin{cases} 1, & \theta \in \Theta_{C(\mathbf{y})} \\ 0, & \theta \notin \Theta_{C(\mathbf{y})} \end{cases}.$$

Then,

$$L(\theta, \Theta_{C(\mathbf{y})}) = \begin{cases} 0, & \theta \in \Theta_{C(\mathbf{y})} \\ 1, & \theta \notin \Theta_{C(\mathbf{y})} \end{cases},$$

where the 0–1 loss function is equal to zero if $\theta \in \Theta_{C(\mathbf{y})}$, and one if $\theta \notin \Theta_{C(\mathbf{y})}$. Then, the risk function is $1 - P(\theta \in \Theta_{C(\mathbf{y})})$.

Given a *measure of credibility* ($\alpha(\mathbf{y})$) that defines the level of trust that $\theta \in \Theta_{C(\mathbf{y})}$; then, we can measure the accuracy of the report by $L(\theta, \alpha(\mathbf{y})) = [\mathbb{1}\{\theta \in \Theta_{C(\mathbf{y})}\} - \alpha(\mathbf{y})]^2$. This loss function could be used to suggest a choice of the report $\alpha(\mathbf{y})$. Given that this is a quadratic loss function, the optimal action is the posterior mean, that is $\mathbb{E}[\mathbb{1}\{\theta \in \Theta_{C(\mathbf{y})}\} | \mathbf{y}] = P(\theta \in \Theta_{C(\mathbf{y})} | \mathbf{y})$. This probability can be calculated given the posterior distribution, that is, $P(\theta \in \Theta_{C(\mathbf{y})} | \mathbf{y}) = \int_{\Theta_{C(\mathbf{y})}} \pi(\theta | \mathbf{y}) d\theta$. This is a measure of the belief that $\theta \in \Theta_{C(\mathbf{y})}$ given the prior beliefs and sample information.

The set $\Theta_{C(\mathbf{y})} \in \Theta$ is a $100(1 - \alpha)\%$ credible set with respect to $\pi(\theta | \mathbf{y})$ if $P(\theta \in \Theta_{C(\mathbf{y})} | \mathbf{y}) = \int_{\Theta_{C(\mathbf{y})}} \pi(\theta | \mathbf{y}) = 1 - \alpha$.

Two alternatives to report credible sets are the *symmetric credible set* and the *highest posterior density set* (HPD). The former is based on $\frac{\alpha}{2}\%$ and $(1 - \frac{\alpha}{2})\%$ percentiles of the posterior distribution, and the latter is a $100(1 - \alpha)\%$ credible interval for θ with the property that it has the smallest distance compared to any other $100(1 - \alpha)\%$ credible interval for θ based on the posterior distribution. That is, $C(\mathbf{y}) = \{\theta : \pi(\theta | \mathbf{y}) \geq k(\alpha)\}$, where $k(\alpha)$ is the largest number such that $\int_{\theta: \pi(\theta | \mathbf{y}) \geq k(\alpha)} \pi(\theta | \mathbf{y}) d\theta = 1 - \alpha$. The HPD set can be a collection of disjoint intervals when working with multimodal posterior densities. In addition, they have the limitation of not necessarily being invariant under transformations.

Decision theory can also be used to perform prediction (point, sets or probabilistic). Suppose that there is a loss function $L(Y_0, a)$ involving the prediction of Y_0 . Then, $\mathbb{E}_{Y_0}[L(Y_0, a)] = \int_{Y_0} L(Y_0, a) \pi(Y_0 | \mathbf{y}) dY_0$, where $\pi(Y_0 | \mathbf{y})$ is the predictive density function. Thus, we make an optimal choice for prediction that minimizes the risk function given a specific loss function.

Although BMA allows incorporating model uncertainty in a regression framework, sometimes it is desirable to select just one model. A compelling alternative is the model with the highest posterior model probability. This model is the best alternative for prediction in the case of a 0–1 loss function [46].

1.3.1 Example: Health insurance continues

We show some optimal rules in the health insurance example. In particular, the best point estimates of λ given the quadratic, absolute and generalized absolute loss functions. For the third, we assume that underestimating λ is twice as costly as overestimating it, that is, $K_0 = 2$ and $K_1 = 1$.

Taking into account that the posterior distribution of λ is $G(\alpha_0 + \sum_{i=1}^N y_i, \beta_0/(\beta_0 N + 1))$, using the hyperparameters from empirical Bayes, we have that $\mathbb{E}[\lambda|\mathbf{y}] = \alpha_n\beta_n = 1.2$, the median is 1.19, and the 2/3-th quantile is 1.26. Those are the optimal point estimates for the quadratic, absolute and generalized absolute loss functions.

In addition, we test the null hypothesis H_0 . $\lambda \in [0, 1)$ versus H_1 . $\lambda \in [1, \infty)$ setting $K_0 = K_1 = 1$ we should reject the null hypothesis due to $P(\lambda \in [0, 1)) = 0.9 > K_1/(K_0 + K_1) = 0.5$.

We get that the 95% symmetric credible interval is (0.91, 1.53), and the highest posterior density interval is (0.9, 1.51). Finally, the optimal point prediction under a quadratic loss function is 1.2, which is the mean value of the posterior predictive distribution, and the optimal model assuming a 0-1 loss function is the model using the hyperparameters from the empirical Bayes procedure due to the posterior model probability of this model being approximately 1, whereas the posterior model probability of the model using vague hyperparameters is approximately 0.

R code. Health insurance, Bayesian reports

```

1 an <- sum(y) + a0EB
2 # Posterior shape parameter
3 bn <- b0EB / (N*b0EB + 1)
4 # Posterior scale parameter
5 S <- 1000000
6 # Number of posterior draws
7 Draws <- rgamma(1000000, shape = an, scale = bn)
8 # Posterior draws
9 ##### Point estimation #####
10 OptQua <- an*bn
11 # Mean: Optimal choice quadratic loss function
12 OptQua
13 1.200952
14 OptAbs <- qgamma(0.5, shape = an, scale = bn)
15 # Median: Optimal choice absolute loss function
16 OptAbs
17 1.194034
18 # Setting K0 = 2 and K1 = 1, that is, to underestimate
# lambda is twice as costly as to overestimate it.
19 K0 <- 2; K1 <- 1
20 OptGenAbs <- quantile(Draws, K0/(K0 + K1))
21 # Median: Optimal choice generalized absolute loss function
22 OptGenAbs
23 66.66667%
24 1.262986
25 ##### Hypothesis test #####
26 # H0: lambda in [0,1) vs H1: lambda in [1, Inf]
27 K0 <- 1; K1 <- 1
28 ProbH0 <- pgamma(1, shape = an, scale = bn)
29 ProbH0 # Posterior probability H0
30 0.09569011
31 ProbH1 <- 1 -ProbH0
32 ProbH1 # Posterior probability H1
33 0.9043099
34 # We should reject H0 given ProbH1 > K1 / (K0 + K1)
35 ##### Credible intervals #####
36 LimInf <- qgamma(0.025, shape = an, scale = bn) # Lower
bound
37 LimInf
38 0.9114851
39 LimSup <- qgamma(0.975, shape = an, scale = bn) # Upper
bound
40 LimSup
41 1.529724
42 HDI <- HDInterval::hdi(Draws, credMass = 0.95) # Highest
posterior density credible interval
43 HDI
44   lower      upper
45 0.8971505 1.5125911
46 attr(,"credMass")
47 [1] 0.95
48 ##### Predictive optimal choices #####
49 p <- bn / (bn + 1) # Probability negative binomial density
50 OptPred <- p/(1-p)*an # Optimal point prediction given a
quadratic loss function in prediction
51 OptPred
52 1.200952

```

1.4 Summary

We introduce the Bayes' rule to update probabilistic statements using funny examples. Then, we study the three probabilistic objects of main relevance in Bayesian inference: the posterior distribution, the marginal likelihood and the predictive density. The first allows performing inference regarding parameters, the second is required to perform hypothesis test for model selection using the Bayes factor, and the third to perform probabilistic predictions. We also review some sampling properties of Bayesian estimators, and Bayes update. All those concepts were developed using a simple example in R software. Finally, we introduce some concepts of decision theory that can be used to report summary statistics minimizing posterior expected losses.

1.5 Exercises

1. *The court case: the blue or green cap*

A cab was involved in a hit and run accident at night. There are two cab companies in the town: blue and green. The former has 150 cabs, and the latter 850 cabs. A witness said that a blue cab was involved in the accident; the court tested his/her reliability under the same circumstances, and got that 80% of the times the witness correctly identified the color of the cab. *What is the probability that the color of the cab involved in the accident was blue given that the witness said it was blue?*

2. *The Monty Hall problem*

What is the probability of winning a car in the *Monty Hall problem* switching the decision if there are four doors, where there are three goats and one car? Solve this problem analytically and computationally. What if there are n doors, $n - 1$ goats and one car?

3. Solve the health insurance example using a Gamma prior in the rate parametrization, that is, $\pi(\lambda) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \lambda^{\alpha_0-1} \exp\{-\lambda\beta_0\}$.
4. Suppose that you are analyzing to buy a car insurance next year. To make a better decision you want to know *what is the probability that you have a car claim next year?* You have the records of your car claims in the last 15 years, $\mathbf{y} = \{0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0\}$.

Assume that this is a random sample from a data generating process (statistical model) that is Bernoulli, $Y_i \sim Ber(p)$, and your probabilistic prior beliefs about p are well described by a beta distribution with parameters α_0 and β_0 , $p \sim B(\alpha_0, \beta_0)$, then, you are

interested in calculating the probability of a claim the next year $P(Y_0 = 1|y)$.

Solve this using an empirical Bayes approach and a non-informative approach where $\alpha_0 = \beta_0 = 1$ (uniform distribution).

5. Show that given the loss function, $L(\theta, a) = |\theta - a|$, then the optimal decision rule minimizing the risk function, $a^*(y)$, is the median.



2

Conceptual differences of the Bayesian and Frequentist approaches

We give some of the conceptual differences between the Bayesian and Frequentist inferential approaches. We emphasize in the Bayesian concepts as most of the readers can be familiarized with the Frequentist statistical framework.

2.1 The concept of probability

Let's begin with the following thought experiment: Assume that you are watching the international game show "Who wants to be a millionaire?", the contestant is asked to answer a very simple question: **What is the last name of the brothers who are credited with inventing the world's first successful motor-operated airplane?**

- What is the probability that the contestant answers this question correctly?

Unless you have:

1. watched this particular contestant participating in this show many times,
2. seen him asked this same question each time,
3. and computed the relative frequency with which he gives the correct answer,

you need to answer this question as a Bayesian!

Uncertainty about the event *answer this question* needs to be expressed as a "degree of belief" informed both by information coming from data on the skill of the particular participant, and how much he knows about inventors, and possibly prior knowledge on his performance in other game shows. Of course, your prior knowledge of the contestant may be minimal, or it may be very informed. Either way, your final answer remains a degree of belief held about an uncertain, and inherently unrepeatable state of nature.

The point of this hypothetical, light-hearted scenario is simply to highlight that a key distinction between the Frequentist and Bayesian approaches to inference is not the use (or nature) of prior information, but simply the manner

in which probability is used. To the Bayesian, probability is the mathematical construct used to quantify uncertainty about an unknown state of nature, conditional on observed data and prior knowledge about the context in which that state of nature occurs. To the Frequentist, probability is linked intrinsically to the concept of a repeated experiment, and the relative frequency with which a particular outcome occurs, conditional on that unknown state. This distinction remains key whether the Bayesian chooses to be *informative or subjective* in the specification of prior information, or chooses to be *non-informative or objective*.

Frequentists consider probability as a physical phenomenon, like mass or wavelength, whereas Bayesians stipulate that probability lives in the mind of scientists as any scientific construct [161].

It seems that the understanding of the concept of probability for the common human being is more associated with “degrees of belief” rather than relative frequency. Peter Diggle, President of The Royal Statistical Society between 2014 and 2016, was asked in an interview “A different trend which has surged upwards in statistics during Peter’s career is the popularity of Bayesian statistics. Does Peter consider himself a Bayesian?”, and he replied “... you can’t not believe in Bayes’ theorem because it’s true. But that doesn’t make you a Bayesian in the philosophical sense. When people are making personal decisions – even if they don’t formally process Bayes’ theorem in their mind – they are adapting what they think they should believe in response to new evidence as it comes in. Bayes’ theorem is just the formal mathematical machinery for doing that.”

However, we should say that psychological experiments suggest that human beings suffer from *anchoring*, that is, a cognitive bias that causes us to rely too heavily on the previous information (prior) such that the updating process (posterior) due to new information (likelihood) being low compared to the Bayes’ rule [114].

2.2 Subjectivity is not the key

The concepts of *subjectivity* and *objectivity* indeed characterize both statistical paradigms in differing ways. Among Bayesians there are those who are immersed in *subjective* rationality [181, 50, 192, 136], but others who adopt *objective* prior distributions such as Jeffreys’, reference, empirical or robust [11, 130, 106, 15] to operationalize Bayes’ rule, and thereby weight quantitative (data-based) evidence. Among Frequentists, there are choices made about significance levels which, if not explicitly subjective, are typically not grounded in any objective and documented assessment of the relative losses

of Type I and Type II errors.¹ In addition, both Frequentist and Bayesian statisticians make decisions about the form of the data generating process, or “model”, which – if not subject to rigorous diagnostic assessment – retains a subjective element that potentially influences the final inferential outcome. Although we all know that by definition a model is a schematic and simplified approximation to reality,

“Since all models are wrong the scientist cannot obtain a *correct* one by excessive elaboration. On the contrary following William of Occam he should seek an economical description of natural phenomena.” [22].

We also know that “All models are wrong, but some are useful” [24], that is why model diagnostics are important. This task can be performed in both approaches. Particularly, the Bayesian framework can use predictive p -values for absolute testing [73, 9] or posterior odds ratios for relative statements [105, 120]. This is because the marginal likelihood, conditional on data, is interpreted as the evidence of the prior distribution [14].

In addition, what is objectivity in a Frequentist approach? For example, why should we use a 5% or 1% significance level rather than any other value? As someone said, the apparent objectivity is really a consensus [136]. In fact “Student” (William Gosset) saw statistical significance at any level as being “nearly valueless” in itself [229]. But, this is not just a situation in the Frequentist approach. The cut-offs given to “establish” scientific evidence against a null hypothesis in terms of \log_{10} scale [106] or \log_e scale [120] in Table 1.1 are also *ad hoc*.

Although the true state of nature in Bayesian inference is expressed in “degrees of belief”, the distinction between the two paradigms does not reside in one being more, or less, *subjective* than the other. Rather, the differences are philosophical, pedagogical, and methodological.

2.3 Estimation, hypothesis testing and prediction

All what is required to perform estimation, hypothesis testing (model selection) and prediction in the Bayesian approach is to apply the Bayes’ rule. This means coherence under a probabilistic view. But, there is no free lunch, coherence reduces flexibility. On the other hand, the Frequentist approach may be not coherent from a probabilistic point of view, but it is very flexible. This approach can be seen as a tool kit that offers inferential solutions under the umbrella of understanding probability as relative frequency. For instance, a point estimator in a Frequentist approach is found such that satisfies good

¹Type I error is rejecting the null hypothesis when this is true, and the Type II error is not rejecting the null hypothesis when this is false.

sampling properties like unbiasness, efficiency, or a large sample property as consistency.

A remarkable difference is that optimal Bayesian decisions are calculated minimizing the expected value of the loss function with respect to the posterior distribution, that is, it is conditional on observed data. On the other hand, Frequentist “optimal” actions are base on the expected values over the distribution of the estimator (a function of data) conditional on the unknown parameters, that is, it considers sampling variability.

The Bayesian approach allows to obtain the posterior distribution of any unknown object such as parameters, latent variables, future or unobserved variables or models. A nice advantage is that prediction can take into account estimation error, and predictive distributions (probabilistic forecasts) can be easily recovered.

Hypothesis testing (model selection) is based on *inductive logic* reasoning (*inverse probability*); on the basis of what we see, we evaluate what hypothesis is most tenable, and is performed using posterior odds, which in turn are based on Bayes factors that evaluate evidence in favor of a null hypothesis taking explicitly the alternative [120], following the rules of probability [136], comparing how well the hypothesis predicts data [89], minimizing the weighted sum of type I and type II error probabilities [52, 162], and taking the implicit balance of losses [106, 18] into account. Posterior odds allows to use the same framework to analyze nested and non-nested models and perform model average. However, Bayes factors cannot be based on improper or vague priors [126], the practical interplay between model selection and posterior distributions is not as easy as it maybe in the Frequentist approach, and the computational burden can be more demanding due to solving potentially difficult integrals.

On the other hand, the Frequentist approach establishes most of its estimators as the solution of a system of equations. Observe that optimization problems reduce to solve systems. We can potentially get the distribution of these estimators, but most of the time it is needed asymptotic arguments or resampling techniques. Hypothesis testing requires pivotal quantities and/or also resampling, and prediction most of the time is based on a *plug-in approach*, which means not taking estimation error into account.² In addition, ancillary statistics can be used to build prediction intervals.³ Comparing models depends on their structure, for instance, there are different Frequentist statistical approaches to compare nested and non-nested models. A nice feature in some situations is that there is a practical interplay between hypothesis testing and confidence intervals, for instance in the normal population mean hypothesis framework you cannot reject at α significance level (Type I error) any null hypothesis H_0 . $\mu = \mu^0$ if μ^0 is in the $1 - \alpha$ confidence interval

²A pivot quantity is a function of unobserved parameters and observations whose probability distribution does not depend on the unknown parameters.

³An ancillary statistic is a pivotal quantity that is also a statistic.

$P(\mu \in [\hat{\mu} - |t_{N-1}^{\alpha/2}| \times \hat{\sigma}_{\hat{\mu}}, \hat{\mu} + |t_{N-1}^{\alpha/2}| \times \hat{\sigma}_{\hat{\mu}}]) = 1 - \alpha$, where $\hat{\mu}$ and $\hat{\sigma}_{\hat{\mu}}$ are the maximum likelihood estimators of the mean and standard error, and $t_{N-1}^{\alpha/2}$ is the quantile value of the Student's t distribution at $\alpha/2$ probability and $N-1$ degrees of freedom, N is the sample size.

A remarkable difference between the Bayesian and the Frequentist inferential frameworks is the interpretation of credible/confidence intervals. Observe that once we have estimates, such that for example the previous interval is $[0.2, 0.4]$ given a 95% confidence level, we cannot say that $P(\mu \in [0.2, 0.4]) = 0.95$ in the Frequentist framework. In fact, this probability is 0 or 1 under this approach, as μ can be there or not, the problem is that we will never know in applied settings. This due to that $P(\mu \in [\hat{\mu} - |t_{N-1}^{0.025}| \hat{\sigma}_{\hat{\mu}}, \hat{\mu} + |t_{N-1}^{0.025}| \hat{\sigma}_{\hat{\mu}}]) = 0.95$ being in the sense of repeated sampling. On the other hand, once we have the posterior distribution, we can say that $P(\mu \in [0.2, 0.4]) = 0.95$ under the Bayesian framework.

Following common practice, most of researchers and practitioners do hypothesis testing based on the p -value in the Frequentist framework. But, **what is a p -value?** Most of the users do not know the answer due to many time statistical inference is not performed by statisticians [15].⁴ A p -value is the probability of obtaining a statistical summary of the data equal to or *more extreme* than what was actually observed, assuming that the null hypothesis is true.

Therefore, p -value calculations involve not just the observed data, but also more *extreme* hypothetical observations. So,

“What the use of p implies, therefore, is that a hypothesis that may be true may be rejected because it has not predicted observable results that have not occurred.”[106]

It seems that common Frequentist inferential practice intertwined two different logic reasoning arguments: the p -value [67] and *significance level* [157]. The former is an informal short-run criterion, whose philosophical foundation is *reduction to absurdity*, which measures the discrepancy between the data and the null hypothesis. So, the p -value is not a direct measure of the probability that the null hypothesis is false. The latter, whose philosophical foundations is *deduction*, is based on a long-run performance such that controls the overall number of incorrect inferences in the repeated sampling without care of individual cases. The p -value fallacy consists in interpreting the p -value as the strength of evidence against the null hypothesis, and using it simultaneously with the frequency of type I error under the null hypothesis [89].

The American Statistical Association has several concerns regarding the use of the p -value as a cornerstone to perform hypothesis testing in science. This concern motivates the ASA's statement on p -values [217], which can be summarized in the following principles:

⁴<https://fivethirtyeight.com/features/not-even-scientists-can-easily-explain-p-values/>

- “P–values can indicate how incompatible the data are with a specified statistical model.”
- “P–values do not measure the probability that the studied hypothesis is true, or the probability that the data were produced by random chance alone.”
- “Scientific conclusions and business or policy decisions should not be based only on whether a p –value passes a specific threshold.”
- “Proper inference requires full reporting and transparency.”
- “A p –value, or statistical significance, does not measure the size of an effect or the importance of a result.”
- “By itself, a p –value does not provide a good measure of evidence regarding a model or hypothesis.”

To sum up, Fisher proposed the p -value as a witness rather than a judge. So, a p -value lower than the significance level means more inspection of the null hypothesis, but it is not a final conclusion about it.

Another difference between the Frequentists and the Bayesians is the way how scientific hypothesis are tested. The former use the p -value, whereas the latter use the Bayes factor. Observe that the p -value is associated with the probability of the data given the hypothesis, whereas the Bayes factor is associated with the probability of the hypothesis given the data. However, there is an approximate link between the t statistic and the Bayes factor for regression coefficients [168]. In particular, $|t| > (\log(N) + 6)^{1/2}$, corresponds to strong evidence in favor of rejecting the not relevance of a control in a regression. Observe that in this setting the threshold of the t statistic, and as a consequence the significant level, depends on the sample size. Observe that this setting agrees with the idea in experimental designs of selecting the sample size such that we control Type I and Type II errors. In observational studies we cannot control the sample size, but we can select the significance level.

See also [195] and [13] for nice exercises to reveal potential flaws of the p -value (p) due to $p \sim U[0, 1]$ under the null hypothesis,⁵ and calibrations of the p -value to interpret them as the odds ratio and the error probability. In particular, $B(p) = -e \times p \times \log(p)$ when $p < e^{-1}$, and interpret this as the Bayes factor of H_0 to H_1 , where H_1 denotes the unspecified alternative to H_0 , and $\alpha(p) = (1 + [-e \times p \times \log(p)]^{-1})^{-1}$ as the error probability α in rejecting H_0 . Take into account that $B(p)$ and $\alpha(p)$ are lower bounds.

Logic of argumentation in the Frequentist approach is based on *deductive logic*, this means that it starts from a statement about the true state of nature (null hypothesis), and predicts what should be seen if this statement were true. On the other hand, the Bayesian approach is based on *inductive logic*, this means that it defines what hypothesis is more consistent with what is

⁵<https://joyeuserrance.wordpress.com/2011/04/22/proof-that-p-values-under-the-null-are-uniformly-distributed/> for a simple proof.

seen. The former inferential approach establishes that the true of the premises implies the true of the conclusion, that is why we reject or not reject hypothesis. The latter establishes that the premises supply some evidence, but not full assurance, of the true of the conclusion, that is why we get probabilistic statements.

Here, there is a difference between *effects of causes* (forward causal inference) and *causes of effects* (reverse causal inference) [78, 49]. To illustrate this point, imagine that a firm increases the price of a specific good, then economic theory would say that its demand decreases. The premise (null hypothesis) is a price increase, and the consequence is a demand reduction. Another view would be to observe a demand reduction, and try to identify which cause is more tenable. For instance, demand reduction can be caused by any positive supply shocks or any negative demand shocks. The Frequentist logic sees the first view, and the Bayesian reasoning gives the probability associated with possible causes.

2.4 The likelihood principle

The **likelihood principle** states that in making inference or decisions about the state of the nature all the relevant *experimental* information is given by the *likelihood function*. The Bayesian framework follows this statement, that is, it is conditional on observed data.

We follow [14], who in turns followed [137], to illustrate the likelihood principle. We are given a coin such that we are interested in the probability, θ , of having it come up heads when flipped. It is desired to test $H_0: \theta = 1/2$ versus $H_1: \theta > 1/2$. An experiment is conducted by flipping the coin (independently) in a series of trials, the results of which is the observation of 9 heads and 3 tails.

This is not yet enough information to specify $p(y|\theta)$, since the series of trials was not explained. Two possibilities:

1. The experiment consisted of a predetermine 12 flips, so that $Y = [\text{Heads}]$ would be $B(12, \theta)$, then $p_1(y|\theta) = \binom{n}{y} \theta^y (1-\theta)^{n-y} = 220 \times \theta^9 (1-\theta)^3$.
2. The experiment consisted of flipping the coin until 3 tails were observed ($r = 3$). Then, Y , the number of failures (heads) until getting 3 tails, is $NB(3, 1-\theta)$. Then, $p_2(y|\theta) = \binom{y+r-1}{r-1} (1-(1-\theta)^y (1-\theta)^r = 55 \times \theta^9 (1-\theta)^3$.

Using a Frequentist approach, the significance level of $y = 9$ using the Binomial model against $\theta = 1/2$ would be:

$$\alpha_1 = P_{1/2}(Y \geq 9) = p_1(9|1/2) + p_1(10|1/2) + p_1(11|1/2) + p_1(12|1/2) = 0.073.$$

R code. The likelihood principle: Binomial model

```

1 success <- 9
2 # Number of observed success in n trials
3 n <- 12
4 # Number of trials
5 siglevel <- sum(sapply(9:n, function(y) dbinom(y, n, 0.5)))
6 siglevel
7 0.073

```

For the Negative Binomial model, the significance level would be:

$$\alpha_2 = P_{1/2}(Y \geq 9) = p_2(9|1/2) + p_2(10|1/2) + \dots = 0.0327.$$

R code. The likelihood principle: Negative Binomial model

```

1 success <- 3
2 # Number of target success (tails)
3 failures <- 9
4 # Number of failures
5 siglevel <- 1 - pnbinom((failures - 1), success, 0.5)
6 siglevel
7 0.0327

```

We arrive to different conclusions using a significance level equal to 5%, whereas we obtain the same outcomes using a Bayesian approach because the kernels of both distributions are the same ($\theta^9 \times (1 - \theta)^3$).

2.5 Why is not the Bayesian approach that popular?

At this stage, we may wonder why the Bayesian statistical framework is not the dominant inferential approach despite that it has its historical origin in 1763 [12], whereas the Frequentist statistical framework was largely developed in the early 20th century. The scientific battle over the Bayesian inferential approach lasted for 150 years, and this maybe explained by some of the following facts.

There is an issue regarding *apparent subjectivity* as the Bayesian inferential approach runs counter the strong conviction that science demands objectivity, and Bayesian probability is a measure of degrees of belief, where the initial prior maybe just a guess; this was not accepted as objective and rigorous science. Initial critics said that Bayes was quantifying ignorance as he set equal probabilities to any potential result. As a consequence, prior distributions were damned [151].

Bayes himself seemed not to have believed in his idea. Although, it seems that Bayes achieved his breakthrough during the late 1740s, he did not send it off to the Royal Society for publication. It was his friend, Richard Price, another Presbyterian minister, who rediscovered Bayes' idea, polished it and published.

However, it was Laplace who independently generalized Bayes' theorem in 1781. He used it initially in gambling problems, and soon after in astronomy, mixing different sources of information in order to leverage research in specific situations where data was scarce. Then, he wanted to use his discovery to find the probability of causes, and thought that this required large data sets, and turned into demography. In this field, he had to perform large calculations that demanded to develop smart approximations, creating the Laplace's approximation and the central limit theorem [130]; although, at the cost of apparently leaving his research on Bayesian inference.

Once *Laplace was gone in 1827*, the Bayes' rule disappeared from the scientific spectrum for almost a century. In part, personal attacks against Laplace made the rule be forgotten, and also, the old fashion thought that statistics does not have to say anything about causation, and that the prior is very subjective to be compatible with science. Although, practitioners used it to solve problems in astronomy, communication, medicine, military and social issues with remarkable results.

Thus, the concept of degrees of belief to operationalize probability was gone in name of scientific objectivity, and probability as the frequency an event occurs in many repeatable trials became the rule. Laplace critics argued that those concepts were diametric opposites, although, Laplace considered them as basically equivalent when large sample sizes are involved [151].

The era of the Frequentists or sampling theorists began, lead by Karl Pearson, and his nemesis, Ronald Fisher, both brilliant, against the inverse prob-

ability approach, persuasive and dominant characters that made impossible to argue against their ideas. Karl Pearson legacy was taken by his son Egon, and Egon's friend Neyman, both inherited the anti-Bayesian and anti-Fisher legacy.

Despite the anti-Bayesian campaign among statisticians, there were some independent characters developing Bayesian ideas, Borel, Ramsey and de Finetti, all of them isolated in different countries, France, England and Italy. However, the anti-Bayesian trio of Fisher, Neyman and Egon Person got all the attention during the 1920s and 1930s. Only, a geophysicist, Harold Jeffreys, kept alive Bayesian inference in the 1930s and 1940s. Jeffreys was a very quiet, shy, uncommunicative gentleman working at Cambridge in the astronomy department. He was Fisher's friend thanks to his character, although they were diametric opposites regarding the Bayesian inferential approach, facing very high intellectual battles. Unfortunately for the Bayesian approach, *Jeffreys lost*, he was very technical using confusing high level mathematics, worried about inference from scientific evidence, not guiding future actions based on decision theory, which was very important in that era for mathematical statistics due to the Second World War. On the other hand, Fisher was a very dominant character, persuasive in public and a master of practice, his techniques were written in a popular style with minimum mathematics.

However, Bayes' rule achieved remarkable results in applied settings like the AT&T company or the social security system in USA. Bayesian inference also had a relevant role during the second World War and the Cold War. Alan Turing used inverse probability at Bletchley Park to crack German messages called Enigma code used by U-boats, Andrei Kolmogorov used it to improved firing tables of Russia's artillery, Bernard Koopman applied it for searching targets in the open sea and the RAND Corporation used it in the Cold War. Unfortunately, *these Bayesian developments were top secrets for almost 40 years* that keep classified the contribution of inverse probability in modern human history.

During 1950s and 1960s three mathematicians lead the rebirth of the Bayesian approach, Good, Savage and Lindley. However, it seems that they were unwilling to apply their theories to real problems, and despite that the Bayesian approach proved its worth, for instance, in business decisions, navy search, lung cancer, etc, it was applied to simple models due to its *mathematical complexity and requirement of large computations*. But, there were some breakthrough that change this. First, hierarchical models introduced by Lindley and Smith, where a complex model is decomposed into many easy to solve models, and second, Markov chain Monte Carlo methods developed by Hastings in the 1970s [96] and the Geman brothers in the 1980s [79]. These methods were introduced into the Bayesian inferential framework in the 1990s by Gelfand and Smith [71], and Tierney [210], when desktop computers got enough computational power to solve complex models. Since then, the Bayesian inferential framework has gained increasing popularity among practitioners and scientists.

2.6 A simple working example

We will illustrate some conceptual differences between the Bayesian and Frequentist statistical approaches performing inference given a random sample $\mathbf{y} = [y_1, y_2, \dots, y_N]$, where $y_i \stackrel{iid}{\sim} N(\mu, \sigma^2)$, $i = 1, 2, \dots, N$.

In particular, we set $\pi(\mu, \sigma) = \pi(\mu)\pi(\sigma) \propto \frac{1}{\sigma}$. This is a standard *non-informative improper* prior (Jeffreys prior, see Chapter 3), that is, this prior is perfectly compatible with sample information. In addition, we are assuming independent priors for μ and σ . Then,

$$\begin{aligned}\pi(\mu, \sigma | \mathbf{y}) &\propto \frac{1}{\sigma} \times (\sigma^2)^{-N/2} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu)^2 \right\} \\ &= \frac{1}{\sigma} \times (\sigma^2)^{-N/2} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N ((y_i - \bar{y}) - (\mu - \bar{y}))^2 \right\} \\ &= \frac{1}{\sigma} \exp \left\{ -\frac{N}{2\sigma^2} (\mu - \bar{y})^2 \right\} \times (\sigma)^{-N} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \bar{y})^2 \right\} \\ &= \frac{1}{\sigma} \exp \left\{ -\frac{N}{2\sigma^2} (\mu - \bar{y})^2 \right\} \times (\sigma)^{-(\alpha_n+1)} \exp \left\{ -\frac{\alpha_n \hat{\sigma}^2}{2\sigma^2} \right\},\end{aligned}$$

where $\bar{y} = \frac{\sum_{i=1}^N y_i}{N}$, $\alpha_n = N - 1$ and $\hat{\sigma}^2 = \frac{\sum_{i=1}^N (y_i - \bar{y})^2}{N-1}$.

The first term in the last expression is the kernel of a normal density, $\mu | \sigma, \mathbf{y} \sim N(\bar{y}, \sigma^2/N)$. The second term is the kernel of an inverted gamma density [227], $\sigma | \mathbf{y} \sim IG(\alpha_n, \hat{\sigma}^2)$. Therefore, $\pi(\mu | \sigma, \mathbf{y}) = (2\pi\sigma^2/N)^{-1/2} \exp \left\{ -\frac{N}{2\sigma^2} (\mu - \bar{y})^2 \right\}$ and $\pi(\sigma | \mathbf{y}) = \frac{2}{\Gamma(\alpha_n/2)} \left(\frac{\alpha_n \hat{\sigma}^2}{2} \right)^{\alpha_n/2} \frac{1}{\sigma^{\alpha_n+1}} \times \exp \left\{ -\frac{\alpha_n \hat{\sigma}^2}{2\sigma^2} \right\}$.

Observe that $\mathbb{E}[\mu | \sigma, \mathbf{y}] = \bar{y}$, this is also the maximum likelihood (Frequentist) point estimate of μ in this setting. In addition, the Frequentist $(1 - \alpha)\%$ confidence interval and the Bayesian $(1 - \alpha)\%$ credible interval have exactly the same form, $\bar{y} \pm |z_{\alpha/2}| \frac{\sigma}{\sqrt{N}}$, where $z_{\alpha/2}$ is the $\alpha/2$ percentile of a standard normal distribution. However, the interpretations are totally different. The confidence interval has a probabilistic interpretation under sampling variability of \bar{Y} , that is, in repeated sampling $(1 - \alpha)\%$ of the intervals $\bar{Y} \pm |z_{\alpha/2}| \frac{\sigma}{\sqrt{N}}$ would include μ , but given an observed realization of \bar{Y} , say \bar{y} , the probability of $\bar{y} \pm |z_{\alpha/2}| \frac{\sigma}{\sqrt{N}}$ including μ is 1 or 0, that is why we say a $(1 - \alpha)\%$ confidence interval. On the other hand, $\bar{y} \pm |z_{\alpha/2}| \frac{\sigma}{\sqrt{N}}$ has a simple probabilistic interpretation in the Bayesian framework, there is a $(1 - \alpha)\%$ probability that μ lies in this interval.

If we want to get the marginal posterior density of μ ,

$$\begin{aligned}
\pi(\mu|\mathbf{y}) &= \int_0^\infty \pi(\mu, \sigma|\mathbf{y}) d\sigma \\
&\propto \int_0^\infty \frac{1}{\sigma} \times (\sigma^2)^{-N/2} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu)^2 \right\} d\sigma \\
&= \int_0^\infty \left(\frac{1}{\sigma} \right)^{N+1} \exp \left\{ -\frac{N}{2\sigma^2} \frac{\sum_{i=1}^N (y_i - \mu)^2}{N} \right\} d\sigma \\
&= \left[\frac{2}{\Gamma(N/2)} \left(\frac{N \sum_{i=1}^N (y_i - \mu)^2}{2N} \right)^{N/2} \right]^{-1} \\
&\propto \left[\sum_{i=1}^N (y_i - \mu)^2 \right]^{-N/2} \\
&= \left[\sum_{i=1}^N ((y_i - \bar{y}) - (\mu - \bar{y}))^2 \right]^{-N/2} \\
&= [\alpha_n \hat{\sigma}^2 + N(\mu - \bar{y})^2]^{-N/2} \\
&\propto \left[1 + \frac{1}{\alpha_n} \left(\frac{\mu - \bar{y}}{\hat{\sigma}/\sqrt{N}} \right)^2 \right]^{-(\alpha_n+1)/2}.
\end{aligned}$$

The fourth line is due to having the kernel of a inverted gamma density with N degrees of freedom in the integral [227].

The last expression is the kernel of a Student's t density function with $\alpha_n = N - 1$ degrees of freedom, expected value equal to \bar{y} , and variance $\frac{\hat{\sigma}^2}{N} \left(\frac{\alpha_n}{\alpha_n - 2} \right)$. Then, $\mu|\mathbf{y} \sim t \left(\bar{y}, \frac{\hat{\sigma}^2}{N} \left(\frac{\alpha_n}{\alpha_n - 2} \right), \alpha_n \right)$.

Observe that a $(1 - \alpha)\%$ confidence interval and $(1 - \alpha)\%$ credible interval have exactly the same expression, $\bar{y} \pm |t_{\alpha/2}^{\alpha_n}| \frac{\hat{\sigma}}{\sqrt{N}}$, where $t_{\alpha/2}^{\alpha_n}$ is the $\alpha/2$ percentile of a Student's t distribution. But again, the interpretations are totally different.

The mathematical similarity between the Frequentist and Bayesian expressions in this example is due to using an improper prior.

2.6.1 Example: Math test

You have a random sample of math scores of size $N = 50$ from a normal distribution, $Y_i \sim N(\mu, \sigma)$. The sample mean and variance are equal to 102 and 10, respectively. Assuming an improper prior equal to $1/\sigma$,

- Get 95% confidence and credible intervals for μ .
- What is the posterior probability that $\mu > 103$?

Using $\mu|\mathbf{y} \sim t\left(\bar{y}, \frac{\hat{\sigma}^2}{N} \left(\frac{\alpha_n}{\alpha_n-2}\right), \alpha_n\right)$, which implies that $\bar{y} \pm |t_{\alpha/2}^{\alpha_n}| \frac{\hat{\sigma}}{\sqrt{N}}$, where $\bar{y} = 102$, $\hat{\sigma}^2 = 10$ and $\alpha_n = 49$, the 95% confidence and credible intervals for μ are the same (101.1, 102.9), and $P(\mu > 103) = 1.49\%$ given the sample information.

R code. Example: Math test

```

1 N <- 50 # Sample size
2 y_bar <- 102 # Sample mean
3 s2 <- 10 # Sample variance
4 alpha <- N - 1
5 serror <- (s2/N)^0.5
6 LimInf <- y_bar - abs(qt(0.025, alpha)) * serror
7 LimInf
8 101.101
9 # Lower bound
10 LimSup <- y_bar + abs(qt(0.025, alpha)) * serror
11 LimSup
12 102.898
13 # Upper bound
14 y.cut <- 103
15 P <- 1-metRology::pt.scaled(y.cut, df = alpha, mean = y_bar,
16                               sd = serror)
17 P
18 # Probability of mu greater than y.cut

```

2.7 Summary

The differences between the Bayesian and Frequentist inferential approaches are philosophical, including as pertains to the role of probability; pedagogical, in particular as relates to the use of inference to inform decision making; and methodological, as having differences in their mathematical and computational frameworks. Although at methodological level, the debate has become considerably muted, except for some aspects of inference, with the recognition that each approach has a great deal to contribute to statistical practice [88, 10, 119]. As Bradley Efron said “Computer-age statistical inference at its most successful **combines** elements of the two philosophies” [64].

2.8 Exercises

1. Jeffreys-Lindley's paradox

The **Jeffreys-Lindley's paradox** [106, 138] is an apparent disagreement between the Bayesian and Frequentist frameworks to a hypothesis testing situation.

In particular, assume that in a city 49,581 boys and 48,870 girls have been born in 20 years. Assume that the male births is distributed Binomial with probability θ . We want to test the null hypothesis H_0 . $\theta = 0.5$ versus H_1 . $\theta \neq 0.5$.

- Show that the posterior model probability for the model under the null is approximately 0.95. Assume $\pi(H_0) = \pi(H_1) = 0.5$, and $\pi(\theta)$ equal to $U(0, 1)$ under H_1 .
 - Show that the p -value for this hypothesis test is equal to 0.0235 using the normal approximation, $Y \sim N(N \times \theta, N \times \theta \times (1-\theta))$.
2. We want to test H_0 . $\mu = \mu_0$ vs H_1 . $\mu \neq \mu_0$ given $y_i \stackrel{iid}{\sim} N(\mu, \sigma^2)$.

Assume $\pi(H_0) = \pi(H_1) = 0.5$, and $\pi(\mu, \sigma) \propto 1/\sigma$ under the alternative hypothesis.

Show that

$$p(\mathbf{y}|\mathcal{M}_1) = \frac{\pi^{-N/2}}{2} \Gamma(N/2) 2^{N/2} \left(\frac{1}{\alpha_n \hat{\sigma}^2} \right)^{N/2} \left(\frac{N}{\alpha_n \hat{\sigma}^2} \right)^{-1/2} \frac{\Gamma(1/2)\Gamma(\alpha_n/2)}{\Gamma((\alpha_n+1)/2)}$$

and $p(\mathbf{y}|\mathcal{M}_0) = (2\pi)^{-N/2} \left[\frac{2}{\Gamma(N/2)} \left(\frac{N \sum_{i=1}^N (y_i - \mu_0)^2}{N} \right)^{N/2} \right]^{-1}$. Then,

$$\begin{aligned} PO_{01} &= \frac{p(\mathbf{y}|\mathcal{M}_0)}{p(\mathbf{y}|\mathcal{M}_1)} \\ &= \frac{\Gamma((\alpha_n+1)/2)}{\Gamma(1/2)\Gamma(\alpha_n/2)} (\alpha_n \hat{\sigma}^2 / N)^{-1/2} \left[1 + \frac{(\mu_0 - \bar{y})^2}{\alpha_n \hat{\sigma}^2 / N} \right]^{-\left(\frac{\alpha_n+1}{2}\right)}, \end{aligned}$$

where $\alpha_n = N - 1$ and $\hat{\sigma}^2 = \frac{\sum_{i=1}^N (y_i - \bar{y})^2}{N-1}$.

Find the relationship between the posterior odds and the classical test statistic for the null hypothesis.

3. Math test continues

Using the setting of the **Example: Math test** in subsection 2.6.1, test H_0 . $\mu = \mu_0$ vs H_1 . $\mu \neq \mu_0$ where $\mu_0 = \{100, 100.5, 101, 101.5, 102\}$.

- What is the p -value for these hypothesis tests?

- Find the posterior model probability of the null model for each μ_0 .



3

Cornerstone models: Conjugate families

We will introduce conjugate families in basic statistical models with examples, solving them analytically and computationally using R. We will have some mathematical, and computational exercises in R.

3.1 Motivation of conjugate families

Observing the three fundamental pieces of Bayesian analysis: the posterior distribution (parameter inference), the marginal likelihood (hypothesis testing), and the predictive distribution (prediction), equations 3.1, 3.2 and 3.3, respectively,

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta})}{p(\mathbf{y})}, \quad (3.1)$$

$$p(\mathbf{y}) = \int_{\Theta} p(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}, \quad (3.2)$$

and

$$p(\mathbf{Y}_0|\mathbf{y}) = \int_{\Theta} p(\mathbf{Y}_0|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}, \quad (3.3)$$

we can understand that some of the initial limitations of the application of the Bayesian analysis were associated with the absence of algorithms to draw from non-standard posterior distributions (equation 3.1), and the lack of analytical solutions of the marginal likelihood (equation 3.2) and the predictive distribution (equation 3.3). Both issues requiring computational power.

Although there were algorithms to sample from non-standard posterior distributions since the second half of the last century [153, 96, 79], their particular application in the Bayesian framework emerged later [71, 210], maybe until the increasing computational power of desktop computers. However, it is also common practice nowadays to use models that have standard conditional posterior distributions to mitigate computational requirements. In addition, nice mathematical tricks plus computational algorithms [72, 40, 43]

and approximations [211, 111] are used to obtain the marginal likelihood (prior predictive).

Despite these advances, there are two potentially conflicting desirable model specification features that we can see from equations 3.1, 3.2 and 3.3: (1) analytical solutions and (2) the posterior distribution in the same family as the prior distribution for a given likelihood. The latter is called *conjugate priors*, a family of priors that is closed under sampling [193].

These features are desirable as the former implies facility to perform hypothesis testing and predictive analysis, and the latter means invariance of the prior-to-posterior updating. Both features imply less computational burden.

We can easily achieve each of these features independently, for instance using improper priors for analytical tractability, and defining in a broad sense the family of prior distributions for prior conjugacy. However, these features are in conflict.

Fortunately, we can achieve these two nice characteristics if we assume that the data generating process is given by a distribution function in the *exponential family*. That is, given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$, a probability density function $p(\mathbf{y}|\boldsymbol{\theta})$ belongs to the exponential family if it has the form

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\theta}) &= \prod_{i=1}^N h(y_i)C(\boldsymbol{\theta}) \exp\{\eta(\boldsymbol{\theta})^\top \mathbf{T}(y_i)\} \\ &= h(\mathbf{y})C(\boldsymbol{\theta})^N \exp\{\eta(\boldsymbol{\theta})^\top \mathbf{T}(\mathbf{y})\} \\ &= h(\mathbf{y}) \exp\{\eta(\boldsymbol{\theta})^\top \mathbf{T}(\mathbf{y}) - A(\boldsymbol{\theta})\}, \end{aligned} \quad (3.4)$$

where $h(\mathbf{y}) = \prod_{i=1}^N h(y_i)$ is a non-negative function, $\eta(\boldsymbol{\theta})$ is a known function of the parameters, $A(\boldsymbol{\theta}) = \log\{\int_Y h(\mathbf{y}) \exp\{\eta(\boldsymbol{\theta})^\top \mathbf{T}(\mathbf{y})\} d\mathbf{y}\} = -N \log(C(\boldsymbol{\theta}))$ is a normalization factor, and $\mathbf{T}(\mathbf{y}) = \sum_{i=1}^N \mathbf{T}(y_i)$ is the vector of sufficient statistics of the distribution (by the factorization theorem).

If the support of \mathbf{y} is independent of $\boldsymbol{\theta}$, then the family is said to be *regular*, otherwise it is irregular. In addition, if we set $\eta = \eta(\boldsymbol{\theta})$, then the exponential family is said to be in the *canonical form*

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\eta}) &= h(\mathbf{y})D(\boldsymbol{\eta})^N \exp\{\boldsymbol{\eta}^\top \mathbf{T}(\mathbf{y})\} \\ &= h(\mathbf{y}) \exp\{\boldsymbol{\eta}^\top \mathbf{T}(\mathbf{y}) - B(\boldsymbol{\eta})\}. \end{aligned}$$

A nice feature of this representation is that $\mathbb{E}[\mathbf{T}(\mathbf{y})|\boldsymbol{\eta}] = \nabla B(\boldsymbol{\eta})$ and $Var[\mathbf{T}(\mathbf{y})|\boldsymbol{\eta}] = \nabla^2 B(\boldsymbol{\eta})$.

3.1.1 Examples of exponential family distributions

1. Discrete distributions

Let's show that some of the most common distributions for random variables that can take values on a finite or countably infinite set are part of the exponential family.

Poisson distribution

Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from a *Poisson distribution* let's show that $p(\mathbf{y}|\lambda)$ is in the exponential family.

$$\begin{aligned} p(\mathbf{y}|\lambda) &= \prod_{i=1}^N \frac{\lambda^{y_i} \exp(-\lambda)}{y_i!} \\ &= \frac{\lambda^{\sum_{i=1}^N y_i} \exp(-N\lambda)}{\prod_{i=1}^N y_i!} \\ &= \frac{\exp(-N\lambda) \exp(\sum_{i=1}^N y_i \log(\lambda))}{\prod_{i=1}^N y_i!}, \end{aligned}$$

then $h(\mathbf{y}) = \left[\prod_{i=1}^N y_i! \right]^{-1}$, $\eta(\lambda) = \log(\lambda)$, $T(\mathbf{y}) = \sum_{i=1}^N y_i$ (sufficient statistic) and $C(\lambda) = \exp(-\lambda)$.

If we set $\eta = \log(\lambda)$, then

$$p(\mathbf{y}|\eta) = \frac{\exp(\eta \sum_{i=1}^N y_i - N \exp(\eta))}{\prod_{i=1}^N y_i!},$$

such that $B(\eta) = N \exp(\eta)$, then $\nabla(B(\eta)) = N \exp(\eta) = N\lambda = \mathbb{E} \left[\sum_{i=1}^N y_i \middle| \lambda \right]$, that is, $\mathbb{E} \left[\frac{\sum_{i=1}^N y_i}{N} \middle| \lambda \right] = \mathbb{E}[\bar{y}|\lambda] = \lambda$, and $\nabla^2(B(\eta)) = N \exp(\eta) = N\lambda = \text{Var} \left[\sum_{i=1}^N y_i \middle| \lambda \right] = N^2 \times \text{Var} [\bar{y}|\lambda]$, then $\text{Var} [\bar{y}|\lambda] = \frac{\lambda}{N}$.

Bernoulli distribution

Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from a *Bernoulli distribution* let's show that $p(\mathbf{y}|\theta)$ is in the exponential family.

$$\begin{aligned} p(\mathbf{y}|\theta) &= \prod_{i=1}^N \theta^{y_i} (1-\theta)^{1-y_i} \\ &= \theta^{\sum_{i=1}^N y_i} (1-\theta)^{N-\sum_{i=1}^N y_i} \\ &= (1-\theta)^N \exp \left\{ \sum_{i=1}^N y_i \log \left(\frac{\theta}{1-\theta} \right) \right\}, \end{aligned}$$

then $h(\mathbf{y}) = \mathbb{I}[y_i \in \{0, 1\}]$ (indicator function), $\eta(\theta) = \log\left(\frac{\theta}{1-\theta}\right)$, $T(\mathbf{y}) = \sum_{i=1}^N y_i$ and $C(\theta) = 1 - \theta$.

Write this distribution in the canonical form, and find the mean and variance of the sufficient statistic (Exercise 1).

Multinomial distribution

Given a random sample $\mathbf{y} = [\mathbf{y}_1 \ \mathbf{y}_2 \ \dots \ \mathbf{y}_N]$ from a *m-dimensional multinomial distribution*, where $\mathbf{y}_i = [y_{i1} \ y_{i2} \ \dots \ y_{im}]$, $\sum_{l=1}^m y_{il} = n$, n independent trials each of which leads to a success for exactly one of m categories with probabilities $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \dots \ \theta_m]$, $\sum_{l=1}^m \theta_l = 1$. Let's show that $p(\mathbf{y}|\boldsymbol{\theta})$ is in the exponential family.

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\theta}) &= \prod_{i=1}^N \frac{n!}{\prod_{l=1}^m y_{il}!} \prod_{l=1}^m \theta_l^{y_{il}} \\ &= \frac{(n!)^N}{\prod_{i=1}^N \prod_{l=1}^m y_{il}!} \exp \left\{ \sum_{i=1}^N \sum_{l=1}^m y_{il} \log(\theta_l) \right\} \\ &= \frac{(n!)^N}{\prod_{i=1}^N \prod_{l=1}^m y_{il}!} \exp \left\{ \left(N \times n - \sum_{i=1}^N \sum_{l=1}^{m-1} y_{il} \right) \log(\theta_m) \right. \\ &\quad \left. + \sum_{i=1}^N \sum_{l=1}^{m-1} y_{il} \log(\theta_l) \right\} \\ &= \frac{(n!)^N}{\prod_{i=1}^N \prod_{l=1}^m y_{il}!} \theta_m^{N \times n} \exp \left\{ \sum_{i=1}^N \sum_{l=1}^{m-1} y_{il} \log(\theta_l / \theta_m) \right\}, \end{aligned}$$

then $h(\mathbf{y}) = \frac{(n!)^N}{\prod_{i=1}^N \prod_{l=1}^m y_{il}!}$, $\eta(\boldsymbol{\theta}) = \left[\log\left(\frac{\theta_1}{\theta_m}\right) \dots \log\left(\frac{\theta_{m-1}}{\theta_m}\right) \right]$, $T(\mathbf{y}) = \left[\sum_{i=1}^N y_{i1} \dots \sum_{i=1}^N y_{im-1} \right]$ and $C(\boldsymbol{\theta}) = \theta_m^n$.

2. Continuous distributions

Let's show that some of the most common distributions for random variables that can take any value within a certain range or interval, often an infinite number of possible values, are part of the exponential family.

Normal distribution

Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from a *normal distribution* let's show that $p(\mathbf{y}|\mu, \sigma^2)$ is in the exponential family.

$$\begin{aligned}
p(\mathbf{y}|\mu, \sigma^2) &= \prod_{i=1}^N \frac{1}{2\pi\sigma^2} \exp \left\{ -\frac{1}{2\sigma^2} (y_i - \mu)^2 \right\} \\
&= (2\pi)^{-N/2} (\sigma^2)^{-N/2} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu)^2 \right\} \\
&= (2\pi)^{-N/2} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N y_i^2 + \frac{\mu}{\sigma^2} \sum_{i=1}^N y_i \right. \\
&\quad \left. - N \frac{\mu^2}{2\sigma^2} - \frac{N}{2} \log(\sigma^2) \right\},
\end{aligned}$$

then $h(\mathbf{y}) = (2\pi)^{-N/2}$, $\eta(\mu, \sigma^2) = \left[\frac{\mu}{\sigma^2}, \frac{-1}{2\sigma^2} \right]$, $T(\mathbf{y}) = \left[\sum_{i=1}^N y_i, \sum_{i=1}^N y_i^2 \right]$ and $C(\mu, \sigma^2) = \exp \left\{ -\frac{\mu^2}{2\sigma^2} - \frac{\log(\sigma^2)}{2} \right\}$.

Observe that

$$p(\mathbf{y}|\mu, \sigma^2) = (2\pi)^{-N/2} \exp \left\{ \eta_1 \sum_{i=1}^N y_i + \eta_2 \sum_{i=1}^N y_i^2 - \frac{N}{2} \log(-2\eta_2) + \frac{N}{4} \frac{\eta_1^2}{\eta_2} \right\},$$

where $B(\boldsymbol{\eta}) = \frac{N}{2} \log(-2\eta_2) - \frac{N}{4} \frac{\eta_1^2}{\eta_2}$. Then,

$$\nabla B(\boldsymbol{\eta}) = \begin{bmatrix} -\frac{N}{2} \frac{\eta_1}{\eta_2} \\ -\frac{N}{2} \frac{1}{\eta_2} + \frac{N}{4} \frac{\eta_1^2}{\eta_2^2} \end{bmatrix} = \begin{bmatrix} N \times \mu \\ N \times (\mu^2 + \sigma^2) \end{bmatrix} = \begin{bmatrix} \mathbb{E} \left[\sum_{i=1}^N y_i | \mu, \sigma^2 \right] \\ \mathbb{E} \left[\sum_{i=1}^N y_i^2 | \mu, \sigma^2 \right] \end{bmatrix}.$$

Multivariate normal distribution

Given $\mathbf{Y} = [\mathbf{y}_1 \ \mathbf{y}_2 \ \dots \ \mathbf{y}_p]$ a $N \times p$ matrix such that $\mathbf{y}_i \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $i = 1, 2, \dots, N$, that is, each i -th row of \mathbf{Y} follows a *multivariate normal distribution*. Then, assuming independence between rows, let's show that $p(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is in the exponential family.

$$\begin{aligned}
p(\mathbf{y}_1, \dots, \mathbf{y}_N | \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \prod_{i=1}^N (2\pi)^{-p/2} |\boldsymbol{\Sigma}|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}) \right\} \\
&= (2\pi)^{-pN/2} |\boldsymbol{\Sigma}|^{-N/2} \exp \left\{ -\frac{1}{2} \text{tr} \left[\sum_{i=1}^N (\mathbf{y}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}) \right] \right\} \\
&= (2\pi)^{-pN/2} |\boldsymbol{\Sigma}|^{-N/2} \exp \left\{ -\frac{1}{2} \text{tr} \left[(\mathbf{S} + N(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^\top) \boldsymbol{\Sigma}^{-1} \right] \right\} \\
&= (2\pi)^{-pN/2} \exp \left\{ -\frac{1}{2} \left[\left(\text{vec}(\mathbf{S})^\top + N \text{vec}(\hat{\boldsymbol{\mu}} \hat{\boldsymbol{\mu}}^\top)^\top \right) \text{vec}(\boldsymbol{\Sigma}^{-1}) \right. \right. \\
&\quad \left. \left. - 2N \hat{\boldsymbol{\mu}}^\top \text{vec}(\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}) + N \text{tr}(\boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}) + N \log(|\boldsymbol{\Sigma}|) \right] \right\},
\end{aligned}$$

where the second line uses the trace operator (tr), and its invariance under cyclic permutation is used in the third line. In addition, we add and subtract $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_i$ in each parenthesis such that we get $\mathbf{S} = \sum_{i=1}^N (\mathbf{y}_i - \hat{\boldsymbol{\mu}})(\mathbf{y}_i - \hat{\boldsymbol{\mu}})^\top$. We get the fourth line after collecting terms, and using some properties of the trace operator to introduce the vectorization operator (vec), that is, $\text{tr}(\mathbf{AB}) = \text{vec}(\mathbf{A}^\top)^\top \text{vec}(\mathbf{B})$, and $\text{vec}(\mathbf{A} + \mathbf{B}) = \text{vec}(\mathbf{A}) + \text{vec}(\mathbf{B})$.

$$\begin{aligned}
\text{Then } h(\mathbf{y}) &= (2\pi)^{-pN/2}, \eta(\boldsymbol{\mu}, \boldsymbol{\Sigma})^\top = \left[(\text{vec}(\boldsymbol{\Sigma}^{-1}))^\top \quad (\text{vec}(\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}))^\top \right], \\
T(\mathbf{y}) &= \left[-\frac{1}{2} \left(\text{vec}(\mathbf{S})^\top + N \text{vec}(\hat{\boldsymbol{\mu}} \hat{\boldsymbol{\mu}}^\top)^\top \right) \quad -N \hat{\boldsymbol{\mu}}^\top \right]^\top \text{ and } C(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \\
&\exp \left\{ -\frac{1}{2} \left(\text{tr}(\boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}) + \log(|\boldsymbol{\Sigma}|) \right) \right\}.
\end{aligned}$$

3.2 Conjugate prior to exponential family

Theorem 4.2.1

The prior distribution $\pi(\boldsymbol{\theta}) \propto C(\boldsymbol{\theta})^{b_0} \exp \{ \eta(\boldsymbol{\theta})^\top \mathbf{a}_0 \}$ is conjugate to the exponential family (equation 3.4).

Proof

$$\begin{aligned}
\pi(\boldsymbol{\theta} | \mathbf{y}) &\propto C(\boldsymbol{\theta})^{b_0} \exp \{ \eta(\boldsymbol{\theta})^\top \mathbf{a}_0 \} \times h(\mathbf{y}) C(\boldsymbol{\theta})^N \exp \{ \eta(\boldsymbol{\theta})^\top T(\mathbf{y}) \} \\
&\propto C(\boldsymbol{\theta})^{N+b_0} \exp \{ \eta(\boldsymbol{\theta})^\top (T(\mathbf{y}) + \mathbf{a}_0) \}.
\end{aligned}$$

Observe that the posterior is in the exponential family, $\pi(\boldsymbol{\theta} | \mathbf{y}) \propto C(\boldsymbol{\theta})^{\beta_n} \exp \{ \eta(\boldsymbol{\theta})^\top \boldsymbol{\alpha}_n \}$, $\beta_n = N + b_0$ and $\boldsymbol{\alpha}_n = T(\mathbf{y}) + \mathbf{a}_0$.

Remarks

We see comparing the prior and the likelihood that b_0 plays the role of a hypothetical sample size, and \mathbf{a}_0 plays the role of hypothetical sufficient statistics. This view helps the elicitation process.

We established the result in the *standard form* of the exponential family. We can also establish this result in the *canonical form* of the exponential family. Observe that given $\boldsymbol{\eta} = \boldsymbol{\eta}(\boldsymbol{\theta})$, another way to get a prior for $\boldsymbol{\eta}$ is to use the change of variable theorem given a bijective function.

In the setting where there is a regular conjugate prior, [54] show that we obtain a posterior expectation of the sufficient statistics that is a weighted average between the prior expectation and the likelihood estimate.

3.2.1 Examples: Theorem 4.2.1

1. Likelihood functions from discrete distributions

The Poisson-gamma model

Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from a Poisson distribution then a conjugate prior density for λ has the form

$$\begin{aligned}\pi(\lambda) &\propto (\exp(-\lambda))^{b_0} \exp\{a_0 \log(\lambda)\} \\ &= \exp(-\lambda b_0) \lambda^{a_0} \\ &= \exp(-\lambda \beta_0) \lambda^{\alpha_0 - 1}.\end{aligned}$$

This is the kernel of a gamma density in the *rate parametrization*, $G(\alpha_0, \beta_0)$, $\alpha_0 = a_0 + 1$ and $\beta_0 = b_0$.¹ Then, a prior conjugate distribution for the Poisson likelihood is a gamma distribution.

Taking into account that $\sum_{i=1}^N y_i$ is a sufficient statistic for the Poisson distribution, then we can think about a_0 as the number of occurrences in b_0 experiments. Observe that

$$\begin{aligned}\pi(\lambda|\mathbf{y}) &\propto \exp(-\lambda \beta_0) \lambda^{\alpha_0 - 1} \times \exp(-N\lambda) \lambda^{\sum_{i=1}^N y_i} \\ &= \exp(-\lambda(N + \beta_0)) \lambda^{\sum_{i=1}^N y_i + \alpha_0 - 1}.\end{aligned}$$

As expected, this is the kernel of a gamma distribution, which means $\lambda|\mathbf{y} \sim G(\alpha_n, \beta_n)$, $\alpha_n = \sum_{i=1}^N y_i + \alpha_0$ and $\beta_n = N + \beta_0$.

Observe that α_0/β_0 is the prior mean, and α_0/β_0^2 is the prior variance. Then, $\alpha_0 \rightarrow 0$ and $\beta_0 \rightarrow 0$ imply a non-informative prior such that the posterior mean converges to the maximum likelihood estimator $\bar{y} = \frac{\sum_{i=1}^N y_i}{N}$,

$$\begin{aligned}\mathbb{E}[\lambda|\mathbf{y}] &= \frac{\alpha_n}{\beta_n} \\ &= \frac{\sum_{i=1}^N y_i + \alpha_0}{N + \beta_0} \\ &= \frac{N\bar{y}}{N + \beta_0} + \frac{\alpha_0}{N + \beta_0}.\end{aligned}$$

¹Another parametrization of the gamma density is the *scale parametrization* where $\kappa_0 = 1/\beta_0$. See the health insurance example in Chapter 1.

The posterior mean is a weighted average between sample and prior information. This is a general result from regular conjugate priors [54]. Observe that $\mathbb{E}[\lambda|\mathbf{y}] = \bar{y}, \lim N \rightarrow \infty$.

In addition, $\alpha_0 \rightarrow 0$ and $\beta_0 \rightarrow 0$ corresponds to $\pi(\lambda) \propto \frac{1}{\lambda}$, which is an improper prior. Improper priors may have bad consequences on Bayes factors (hypothesis testing), see below a discussion of this in the linear regression framework. In this example, we can get analytical solutions for the marginal likelihood and the predictive distribution (see the health insurance example and Exercise 3 in Chapter 1).

The Bernoulli-beta model

Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from a Bernoulli distribution then a conjugate prior density for θ has the form

$$\begin{aligned}\pi(\theta) &\propto (1-\theta)^{b_0} \exp \left\{ a_0 \log \left(\frac{\theta}{1-\theta} \right) \right\} \\ &= (1-\theta)^{b_0-a_0} \theta^{a_0} \\ &= \theta^{\alpha_0-1} (1-\theta)^{\beta_0-1}.\end{aligned}$$

This is the kernel of a beta density, $B(\alpha_0, \beta_0)$, $\alpha_0 = a_0 + 1$ and $\beta_0 = b_0 - a_0 + 1$. A prior conjugate distribution for the Bernoulli likelihood is a beta distribution. Given that b_0 is the hypothetical sample size, and a_0 is the hypothetical sufficient statistic, which is the number of successes, then $b_0 - a_0$ is the number of failures. This implies that α_0 is the number of prior successes plus one, and β_0 is the number of prior failures plus one. Given that the mode of a beta distributed random variable is $\frac{\alpha_0-1}{\alpha_0+\beta_0-2} = \frac{a_0}{b_0}$, then we have the prior probability of success. Setting $\alpha_0 = 1$ and $\beta_0 = 1$, which implies a 0-1 uniform distribution, corresponds to a setting with 0 successes (and 0 failures) in 0 experiments.

Observe that

$$\begin{aligned}\pi(\theta|\mathbf{y}) &\propto \theta^{\alpha_0-1} (1-\theta)^{\beta_0-1} \times \theta^{\sum_{i=1}^N y_i} (1-\theta)^{N-\sum_{i=1}^N y_i} \\ &= \theta^{\alpha_0+\sum_{i=1}^N y_i-1} (1-\theta)^{\beta_0+N-\sum_{i=1}^N y_i-1}.\end{aligned}$$

The posterior distribution is beta, $\theta|\mathbf{y} \sim B(\alpha_n, \beta_n)$, $\alpha_n = \alpha_0 + \sum_{i=1}^N y_i$ and $\beta_n = \beta_0 + N - \sum_{i=1}^N y_i$, where the posterior mean $\mathbb{E}[\theta|\mathbf{y}] = \frac{\alpha_n}{\alpha_n+\beta_n} = \frac{\alpha_0+N\bar{y}}{\alpha_0+\beta_0+N} = \frac{\alpha_0+\beta_0}{\alpha_0+\beta_0+N} \frac{\alpha_0}{\alpha_0+\beta_0} + \frac{N}{\alpha_0+\beta_0+N} \bar{y}$. The posterior mean is a weighted average between the prior mean and the maximum likelihood estimate.

The marginal likelihood in this setting is

$$\begin{aligned} p(\mathbf{y}) &= \int_0^1 \frac{\theta^{\alpha_0-1}(1-\theta)^{\beta_0-1}}{B(\alpha_0, \beta_0)} \times \theta^{\sum_{i=1}^N y_i} (1-\theta)^{N-\sum_{i=1}^N y_i} d\theta \\ &= \frac{B(\alpha_n, \beta_n)}{B(\alpha_0, \beta_0)}, \end{aligned}$$

where $B(\cdot, \cdot)$ is the beta function.

In addition, the predictive density is

$$\begin{aligned} p(Y_0|\mathbf{y}) &= \int_0^1 \theta^{y_0} (1-\theta)^{1-y_0} \times \frac{\theta^{\alpha_n-1}(1-\theta)^{\beta_n-1}}{B(\alpha_n, \beta_n)} d\theta \\ &= \frac{B(\alpha_n + y_0, \beta_n + 1 - y_0)}{B(\alpha_n, \beta_n)} \\ &= \frac{\Gamma(\alpha_n + \beta_n) \Gamma(\alpha_n + y_0) \Gamma(\beta_n + 1 - y_0)}{\Gamma(\alpha_n + \beta_n + 1) \Gamma(\alpha_n) \Gamma(\beta_n)} \\ &= \begin{cases} \frac{\alpha_n}{\alpha_n + \beta_n}, & y_0 = 1 \\ \frac{\beta_n}{\alpha_n + \beta_n}, & y_0 = 0 \end{cases}. \end{aligned}$$

This is a Bernoulli distribution with probability of success equal to $\frac{\alpha_n}{\alpha_n + \beta_n}$.

The multinomial-Dirichlet model

Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from a multinomial distribution then a conjugate prior density for $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \dots \ \theta_m]$ has the form

$$\begin{aligned} \pi(\boldsymbol{\theta}) &\propto \theta_m^{b_0} \exp \{ \boldsymbol{\eta}(\boldsymbol{\theta})^\top \mathbf{a}_0 \} \\ &= \prod_{l=1}^{m-1} \theta_l^{a_{0l}} \theta_m^{b_0 - \sum_{l=1}^{m-1} a_{0l}} \\ &= \prod_{l=1}^m \theta_l^{\alpha_{0l}-1}, \end{aligned}$$

where $\boldsymbol{\eta}(\boldsymbol{\theta}) = \left[\log\left(\frac{\theta_1}{\theta_m}\right) \ \dots \ \log\left(\frac{\theta_{m-1}}{\theta_m}\right) \right]^\top$, $\mathbf{a}_0 = [a_{01} \ \dots \ a_{am-1}]^\top$, $\boldsymbol{\alpha}_0 = [\alpha_{01} \ \alpha_{02} \ \dots \ \alpha_{0m}]$, $\alpha_{0l} = a_{0l} + 1$, $l = 1, 2, \dots, m-1$ and $\alpha_{0m} = b_0 - \sum_{l=1}^{m-1} a_{0l} + 1$.

This is the kernel of a Dirichlet distribution, that is, the prior distribution is $D(\boldsymbol{\alpha}_0)$.

Observe that a_{0l} is the number of hypothetical number of times outcome l is observed over the hypothetical b_0 trials. Setting $\alpha_{0l} =$

1, that is a uniform distribution over the open standard simplex, implicitly we set $a_{0l} = 0$, which means that there are 0 occurrences of category l in $b_0 = 0$ experiments.

The posterior distribution of the multinomial-Dirichlet model is given by

$$\begin{aligned}\pi(\boldsymbol{\theta}|\mathbf{y}) &\propto \prod_{l=1}^m \theta_l^{\alpha_{0l}-1} \times \prod_{l=1}^m \theta_l^{\sum_{i=1}^N y_{il}} \\ &= \prod_{l=1}^m \theta_l^{\alpha_{0l} + \sum_{i=1}^N y_{il}-1}.\end{aligned}$$

This is the kernel of a Dirichlet distribution $D(\boldsymbol{\alpha}_n)$, $\boldsymbol{\alpha}_n = [\alpha_{n1} \alpha_{n2} \dots \alpha_{nm}]$, $\alpha_{nl} = \alpha_{0l} + \sum_{i=1}^N y_{il}$, $l = 1, 2, \dots, m$. Observe that

$$\begin{aligned}\mathbb{E}[\theta_j|\mathbf{y}] &= \frac{\alpha_{nj}}{\sum_{l=1}^m [\alpha_{0l} + \sum_{i=1}^N y_{il}]} \\ &= \frac{\sum_{l=1}^m \alpha_{0l}}{\sum_{l=1}^m [\alpha_{0l} + \sum_{i=1}^N y_{il}]} \frac{\alpha_{0j}}{\sum_{l=1}^m \alpha_{0l}} \\ &\quad + \frac{\sum_{l=1}^m \sum_{i=1}^N y_{il}}{\sum_{l=1}^m [\alpha_{0l} + \sum_{i=1}^N y_{il}]} \frac{\sum_{i=1}^N y_{ij}}{\sum_{l=1}^m \sum_{i=1}^N y_{il}}.\end{aligned}$$

We have again that the posterior mean is a weighted average between the prior mean and the maximum likelihood estimate.

The marginal likelihood is

$$\begin{aligned}p(\mathbf{y}) &= \int_{\Theta} \frac{\prod_{l=1}^m \theta_l^{\alpha_{0l}-1}}{B(\boldsymbol{\alpha}_0)} \times \prod_{i=1}^N \frac{n!}{\prod_{l=1}^m y_{il}} \prod_{l=1}^m \theta_l^{y_{il}} d\boldsymbol{\theta} \\ &= \frac{N \times n!}{B(\boldsymbol{\alpha}_0) \prod_{i=1}^N \prod_{l=1}^m y_{il}!} \int_{\Theta} \prod_{l=1}^m \theta_l^{\alpha_{0l} + \sum_{i=1}^N y_{il}-1} d\boldsymbol{\theta} \\ &= \frac{N \times n!}{B(\boldsymbol{\alpha}_0) \prod_{i=1}^N \prod_{l=1}^m y_{il}!} B(\boldsymbol{\alpha}_n) \\ &= \frac{N \times n! \Gamma(\sum_{l=1}^m \alpha_{0l})}{\Gamma(\sum_{l=1}^m \alpha_{0l} + N \times n) \prod_{l=1}^m \Gamma(\alpha_{0l}) \prod_{i=1}^N \prod_{l=1}^m y_{il}!},\end{aligned}$$

where $B(\boldsymbol{\alpha}) = \frac{\prod_{l=1}^m \Gamma(\alpha_l)}{\Gamma(\sum_{l=1}^m \alpha_l)}$.

Following similar steps we get the predictive density

$$p(Y_0|\mathbf{y}) = \frac{n! \Gamma(\sum_{l=1}^m \alpha_{nl})}{\Gamma(\sum_{l=1}^m \alpha_{nl} + n)} \prod_{l=1}^m \frac{\Gamma(\alpha_{nl} + y_{0l})}{\Gamma(\alpha_{nl}) y_{0l}!}.$$

This is a Dirichlet-multinomial distribution with parameters α_n .

Example: English premier league, Liverpool vs Manchester city

Let's see an example based on data from the English Premier league. In particular, we want to get the probability that in the following five matches Liverpool versus Manchester city, the former wins two games, and the latter three game. This is done based on the historical records of the last five matches where Liverpool was local between January 14th, 2018 and April tenth, 2022. There were two wins for Liverpool, two draws, and one win for Manchester city.²

We use two strategies to get the hyperparameters. First, we estimate the hyperparameters of the Dirichlet distribution using betting odds from bookmakers at 19:05 hours October sixth, 2022 (Colombia time). We got information from 24 bookmakers (see file *DataOddsLIVvsMAN.csv*),³ and transform these odds in probabilities using a simple standardization approach, then we use maximum likelihood to estimate the hyperparameters. Second, we use empirical Bayes, that is, we estimate the hyperparameters optimizing the marginal likelihood.

²<https://www.11v11.com/teams/manchester-city/tab/opposingTeams/opposition/Liverpool/>.

³<https://www.oddsportal.com/soccer/england/premier-league/liverpool-manchester-city-WrqgEz5S/>

R code. Multinomial-Dirichlet model: Liverpool vs Manchester city

```

1 # Multinomial-Dirichlet example: Liverpool vs Manchester
2 # city
3 Data <- read.csv("https://raw.githubusercontent.com/
4   besmarter/BSTApp/refs/heads/master/DataApp/
5   DataOddsLIVvsMAN.csv", sep = ",", header = TRUE, quote =
6   "")
7
8 attach(Data)
9 library(dplyr)
10 Probs <- Data %>%
11   mutate(pns1 = 1/home, pns2 = 1/draw, pns3 = 1/away)%>%
12   mutate(SumInvOdds = pns1 + pns2 + pns3) %>%
13   mutate(p1 = pns1/SumInvOdds, p2 = pns2/SumInvOdds, p3 =
14     pns3/SumInvOdds) %>%
15   select(p1, p2, p3)
16 # We get probabilities using simple standardization. There
17 # are more technical approaches to do this. See for
18 # instance Shin (1993) and Strumbelj (2014).
19 DirMLE <- sirt::dirichlet.mle(Probs)
20 # Use maximum likelihood to estimate parameters of the
21 # Dirichlet distribution
22 alphaOdds <- DirMLE$alpha
23 alphaOdds
24   p1          p2          p3
25 1599.122 1342.703 2483.129
26
27 y <- c(2, 2, 1)
28 # Historical records last five matches
29 # Liverpool wins (2), draws (2) and Manchester
30 # city wins (1)
31
32 # Marginal likelihood
33 MarLik <- function(a0){
34   n <- sum(y)
35   Res1 <- sum(sapply(1:length(y),
36     function(i){lgamma(a0[1]+y[i])-lgamma(a0[1]))})
37   Res <- lgamma(sum(a0))-lgamma(sum(a0)+n)+Res1
38   return(-Res)
39 }
40 EmpBay <- optim(alphaOdds, MarLik, method = "BFGS")
41 alphaOEB <- EmpBay$par
42 alphaOEB
43   p1          p2          p3
44 2362.622 2660.153 1279.510
45
46 # Bayes factor empirical Bayes vs betting odds.
47 # This is greater than 1 by construction
48 BF <- exp(-MarLik(alphaOEB))/exp(-MarLik(alphaOdds))
49 BF
50 2.085819
51
52 # Posterior distribution based on empirical Bayes
53 alphan <- alphaOEB + y
54 # Posterior parameters
55 S <- 100000
56 # Simulation draws from the Dirichlet distribution
57 thetas <- MCMCpack::rdirichlet(S, alphan)
58 colnames(thetas) <- c("Liverpool", "Draw", "Manchester")

```

R code. Multinomial-Dirichlet model: Liverpool vs Manchester city

```

1 # Predictive distribution based on simulations
2 y0 <- c(2, 0, 3)
3 # Liverpool two wins and Manchester city three wins in next
4 # five matches
5 Pred <- apply(thetas, 1, function(p) {rmultinom(1, size =
6   sum(y0), prob = p)})
7 ProYo <- sum(sapply(1:S, function(s){sum(Pred[,s]==y0)==3}))/S
8 ProYo
9 0.0832
10 # Probability of y0
11
12 # Predictive distribution using analytical expression
13 PredY0 <- function(y0){
14   n <- sum(y0)
15   Res1 <- sum(sapply(1:length(y), function(l){lgamma(alphan[
16     1]+y0[1]) - lgamma(alphan[1])-lfactorial(y0[1]))})
17   Res <- lfactorial(n) + lgamma(sum(alphan)) - lgamma(sum(
18     alphan)+n) + Res1
19   return(exp(Res))
20 }
21 PredY0(y0)
22 0.0833

```

We see that the Bayes factor gives evidence in favor of the hyperparameters based on empirical Bayes, this is by construction, as these hyperparameters maximize the marginal likelihood.

We observe that using the hyperparameters from empirical Bayes, the probability that in the next five games Liverpool wins two games and Manchester city wins three games is 8.33%. The result using the predictive distribution based on simulations is similar to the probability using the exact predictive.

2. Likelihood functions from continuous distributions

The normal-normal/inverse-gamma model

Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from a normal distri-

bution, then the conjugate prior density has the form

$$\begin{aligned}
\pi(\mu, \sigma^2) &\propto \exp \left\{ b_0 \left(-\frac{\mu^2}{2\sigma^2} - \frac{\log \sigma^2}{2} \right) \right\} \exp \left\{ a_{01} \frac{\mu}{\sigma^2} - a_{02} \frac{1}{\sigma^2} \right\} \\
&= \exp \left\{ b_0 \left(-\frac{\mu^2}{2\sigma^2} - \frac{\log \sigma^2}{2} \right) \right\} \exp \left\{ a_{01} \frac{\mu}{\sigma^2} - a_{02} \frac{1}{\sigma^2} \right\} \\
&\quad \times \exp \left\{ -\frac{a_{01}^2}{2\sigma^2 b_0} \right\} \exp \left\{ \frac{a_{01}^2}{2\sigma^2 b_0} \right\} \\
&= \exp \left\{ -\frac{b_0}{2\sigma^2} \left(\mu - \frac{a_{01}}{b_0} \right)^2 \right\} \left(\frac{1}{\sigma^2} \right)^{\frac{b_0+1-1}{2}} \\
&\quad \times \exp \left\{ \frac{1}{\sigma^2} \frac{-2b_0 a_{02} + a_{01}^2}{2b_0} \right\} \\
&= \underbrace{\left(\frac{1}{\sigma^2} \right)^{\frac{1}{2}} \exp \left\{ -\frac{b_0}{2\sigma^2} \left(\mu - \frac{a_{01}}{b_0} \right)^2 \right\}}_1 \\
&\quad \times \underbrace{\left(\frac{1}{\sigma^2} \right)^{\frac{b_0-1}{2}} \exp \left\{ -\frac{1}{\sigma^2} \frac{2b_0 a_{02} - a_{01}^2}{2b_0} \right\}}_2.
\end{aligned}$$

The first part is the kernel of a normal density with mean $\mu_0 = a_{01}/\beta_0$ and variance σ^2/β_0 , $\beta_0 = b_0$ that is, $\mu|\sigma^2 \sim N(\mu_0, \sigma^2/\beta_0)$. The second part is the kernel of an inverse gamma density with shape parameter $\alpha_0/2 = \frac{\beta_0-3}{2}$, and scale parameter $\delta_0/2 = \frac{2\beta_0 a_{02} - a_{01}^2}{2\beta_0}$, $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$. Observe that $b_0 = \beta_0$ is the hypothetical sample size, and a_{01} is the hypothetical sum of prior observations, then, it makes sense that a_{01}/β_0 and σ^2/β_0 are the prior mean and variance, respectively.

Therefore, the posterior distribution is also a normal-inverse gamma

distribution,

$$\begin{aligned}
\pi(\mu, \sigma^2 | \mathbf{y}) &\propto \left(\frac{1}{\sigma^2} \right)^{1/2} \exp \left\{ -\frac{\beta_0}{2\sigma^2} (\mu - \mu_0)^2 \right\} \left(\frac{1}{\sigma^2} \right)^{\alpha_0/2+1} \exp \left\{ -\frac{\delta_0}{2\sigma^2} \right\} \\
&\quad \times (\sigma^2)^{-N/2} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu)^2 \right\} \\
&= \left(\frac{1}{\sigma^2} \right)^{1/2} \exp \left\{ -\frac{1}{2\sigma^2} \left(\beta_0(\mu - \mu_0)^2 + \sum_{i=1}^N (y_i - \bar{y})^2 + N(\mu - \bar{y})^2 + \delta_0 \right) \right\} \\
&\quad \times \left(\frac{1}{\sigma^2} \right)^{\frac{\alpha_0+N}{2}+1} + \frac{(\beta_0\mu_0 + N\bar{y})^2}{\beta_0 + N} - \frac{(\beta_0\mu_0 + N\bar{y})^2}{\beta_0 + N} \\
&= \underbrace{\left(\frac{1}{\sigma^2} \right)^{1/2} \exp \left\{ -\frac{1}{2\sigma^2} \left((\beta_0 + N) \left(\mu - \left(\frac{\beta_0\mu_0 + N\bar{y}}{\beta_0 + N} \right) \right)^2 \right) \right\}}_1 \\
&\quad \times \underbrace{\left(\frac{1}{\sigma^2} \right)^{\frac{\alpha_0+N}{2}+1} \exp \left\{ -\frac{1}{2\sigma^2} \left(\sum_{i=1}^N (y_i - \bar{y})^2 + \delta_0 + \frac{\beta_0 N}{\beta_0 + N} (\bar{y} - \mu_0)^2 \right) \right\}}_2.
\end{aligned}$$

The first term is the kernel of a normal density, $\mu | \sigma^2, \mathbf{y} \sim N(\mu_n, \sigma_n^2)$, where $\mu_n = \frac{\beta_0\mu_0 + N\bar{y}}{\beta_0 + N}$ and $\sigma_n^2 = \frac{\sigma^2}{\beta_n}$, $\beta_n = \beta_0 + N$. The second term is the kernel of an inverse gamma density, $\sigma^2 | \mathbf{y} \sim IG(\alpha_n/2, \delta_n/2)$ where $\alpha_n = \alpha_0 + N$ and $\delta_n = \sum_{i=1}^N (y_i - \bar{y})^2 + \delta_0 + \frac{\beta_0 N}{\beta_0 + N} (\bar{y} - \mu_0)^2$. Observe that the posterior mean is a weighted average between prior and sample information. The weights depends on the sample sizes (β_0 and N).

The marginal posterior for σ^2 is inverse gamma with shape and scale parameters $\alpha_n/2$ and $\delta_n/2$, respectively. The marginal posterior of μ is

$$\begin{aligned}
\pi(\mu | \mathbf{y}) &\propto \int_0^\infty \left\{ \left(\frac{1}{\sigma^2} \right)^{\frac{\alpha_n+1}{2}+1} \exp \left\{ -\frac{1}{2\sigma^2} (\beta_n(\mu - \mu_n)^2 + \delta_n) \right\} \right\} d\sigma^2 \\
&= \frac{\Gamma(\frac{\alpha_n+1}{2})}{\left[\frac{\beta_n(\mu - \mu_n)^2 + \delta_n}{2} \right]^{\frac{\alpha_n+1}{2}}} \\
&\propto \left[\frac{\beta_n(\mu - \mu_n)^2 + \delta_n}{2} \right]^{-\frac{\alpha_n+1}{2}} \left(\frac{\delta_n}{\beta_n} \right)^{-\frac{\alpha_n+1}{2}} \\
&\propto \left[\frac{\alpha_n \beta_n (\mu - \mu_n)^2}{\alpha_n \delta_n} + 1 \right]^{-\frac{\alpha_n+1}{2}},
\end{aligned}$$

where the second line due to having the kernel of an inverse gamma density with parameters $(\alpha_n + 1)/2$ and $-\frac{1}{2\sigma^2} (\beta_n(\mu - \mu_n)^2 + \delta_n)$.

This is the kernel of a Student's t distribution, $\mu | \mathbf{y} \sim$

$t(\mu_n, \delta_n/\beta_n \alpha_n, \alpha_n)$, where $\mathbb{E}[\mu|\mathbf{y}] = \mu_n$ and $Var[\mu|\mathbf{y}] = \frac{\alpha_n}{\alpha_n-2} \left(\frac{\delta_n}{\beta_n \alpha_n} \right) = \frac{\delta_n}{(\alpha_n-2)\beta_n}$, $\alpha_n > 2$. Observe that the marginal posterior distribution for μ has heavier tails than the conditional posterior distribution due to incorporating uncertainty regarding σ^2 .

The marginal likelihood is

$$\begin{aligned}
p(\mathbf{y}) &= \int_{-\infty}^{\infty} \int_0^{\infty} \left\{ (2\pi\sigma^2/\beta_0)^{-1/2} \exp \left\{ -\frac{1}{2\sigma^2/\beta_0} (\mu - \mu_0)^2 \right\} \frac{(\delta_0/2)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} \left(\frac{1}{\sigma^2} \right)^{\alpha_0/2+1} \right. \\
&\quad \times \left. \exp \left\{ -\frac{\delta_0}{2\sigma^2} \right\} (2\pi\sigma^2)^{-N/2} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu)^2 \right\} \right\} d\sigma^2 d\mu \\
&= \frac{(\delta_0/2)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} (2\pi)^{-\left(\frac{N+1}{2}\right)} \beta_0^{1/2} \int_{-\infty}^{\infty} \int_0^{\infty} \left\{ \left(\frac{1}{\sigma^2} \right)^{\frac{\alpha_0+N+1}{2}+1} \right. \\
&\quad \times \left. \exp \left\{ -\frac{1}{2\sigma^2} (\beta_0(\mu - \mu_0)^2 + \sum_{i=1}^N (y_i - \mu)^2 + \delta_0) \right\} \right\} d\sigma^2 d\mu \\
&= \frac{(\delta_0/2)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} (2\pi)^{-\left(\frac{N+1}{2}\right)} \beta_0^{1/2} \Gamma \left(\frac{N+1+\alpha_0}{2} \right) \\
&\quad \times \int_{-\infty}^{\infty} \left[\frac{\beta_0(\mu - \mu_0)^2 + \sum_{i=1}^N (y_i - \mu)^2 + \delta_0}{2} \right]^{-\frac{\alpha_0+N+1}{2}} d\mu \\
&= \frac{(\delta_0/2)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} (2\pi)^{-\left(\frac{N+1}{2}\right)} \beta_0^{1/2} \Gamma \left(\frac{N+1+\alpha_0}{2} \right) \\
&\quad \times \int_{-\infty}^{\infty} \left[\frac{\beta_n(\mu - \mu_n)^2 + \delta_n}{2} \right]^{-\frac{\alpha_n+1}{2}} d\mu \left(\frac{\delta_n/2}{\delta_n/2} \right)^{-\frac{\alpha_n+1}{2}} \\
&= \frac{(\delta_0/2)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} (2\pi)^{-\left(\frac{N+1}{2}\right)} \beta_0^{1/2} \Gamma \left(\frac{\alpha_n+1}{2} \right) \left(\frac{\delta_n}{2} \right)^{-\frac{\alpha_n+1}{2}} \frac{\left(\frac{\delta_n \pi}{\beta_n} \right)^{1/2} \Gamma \left(\frac{\alpha_n}{2} \right)}{\Gamma \left(\frac{\alpha_n+1}{2} \right)} \\
&= \frac{\Gamma \left(\frac{\alpha_n}{2} \right)}{\Gamma \left(\frac{\alpha_0}{2} \right)} \frac{(\delta_0/2)^{\alpha_0/2}}{(\delta_n/2)^{\alpha_n/2}} \left(\frac{\beta_0}{\beta_n} \right)^{1/2} (\pi)^{-N/2},
\end{aligned}$$

where we take into account that $\int_{-\infty}^{\infty} \left[\frac{\beta_n(\mu - \mu_n)^2 + \delta_n}{2} \right]^{-\frac{\alpha_n+1}{2}} d\mu \left(\frac{\delta_n/2}{\delta_n/2} \right)^{-\frac{\alpha_n+1}{2}} = \int_{-\infty}^{\infty} \left[\frac{\beta_n \alpha_n (\mu - \mu_n)^2}{\delta_n \alpha_n} + 1 \right]^{-\frac{\alpha_n+1}{2}} d\mu \left(\frac{\delta_n}{2} \right)^{-\frac{\alpha_n+1}{2}}$. The term in the integral is the kernel of a Student's t density, this means that the integral is equal to $\frac{\left(\frac{\delta_n \pi}{\beta_n} \right)^{1/2} \Gamma \left(\frac{\alpha_n}{2} \right)}{\Gamma \left(\frac{\alpha_n+1}{2} \right)}$.

The predictive density is

$$\begin{aligned}
\pi(Y_0|\mathbf{y}) &\propto \int_{-\infty}^{\infty} \int_0^{\infty} \left\{ \left(\frac{1}{\sigma^2} \right)^{1/2} \exp \left\{ -\frac{1}{2\sigma^2} (y_0 - \mu)^2 \right\} \left(\frac{1}{\sigma^2} \right)^{1/2} \exp \left\{ -\frac{\beta_n}{2\sigma^2} (\mu - \mu_n)^2 \right\} \right. \\
&\quad \times \left. \left(\frac{1}{\sigma^2} \right)^{\alpha_n/2+1} \exp \left\{ -\frac{\delta_n}{2\sigma^2} \right\} \right\} d\sigma^2 d\mu \\
&= \int_{-\infty}^{\infty} \int_0^{\infty} \left\{ \left(\frac{1}{\sigma^2} \right)^{\frac{\alpha_n+2}{2}+1} \exp \left\{ -\frac{1}{2\sigma^2} ((y_0 - \mu)^2 + \beta_n(\mu - \mu_n)^2 + \delta_n) \right\} \right\} d\sigma^2 d\mu \\
&\propto \int_{-\infty}^{\infty} [\beta_n(\mu - \mu_n)^2 + (y_0 - \mu)^2 + \delta_n]^{-(\frac{\alpha_n}{2}+1)} d\mu \\
&= \int_{-\infty}^{\infty} \left[(\beta_n + 1) \left(\mu - \left(\frac{\beta_n \mu_n + y_0}{\beta_n + 1} \right) \right)^2 + \frac{\beta_n(y_0 - \mu_n)^2}{\beta_n + 1} + \delta_n \right]^{-(\frac{\alpha_n}{2}+1)} d\mu \\
&= \int_{-\infty}^{\infty} \left[1 + \frac{(\beta_n + 1)^2 \left(\mu - \left(\frac{\beta_n \mu_n + y_0}{\beta_n + 1} \right) \right)^2}{\beta_n(y_0 - \mu_n)^2 + (\beta_n + 1)\delta_n} \right]^{-(\frac{\alpha_n}{2}+1)} d\mu \\
&\quad \times \left(\frac{\beta_n(y_0 - \mu_n)^2 + (\beta_n + 1)\delta_n}{\beta_n + 1} \right)^{-(\frac{\alpha_n}{2}+1)} \\
&\propto \left(\frac{\beta_n(y_0 - \mu_n)^2 + (\beta_n + 1)\delta_n}{(\beta_n + 1)^2(\alpha_n + 1)} \right)^{\frac{1}{2}} \left(\frac{\beta_n(y_0 - \mu_n)^2 + (\beta_n + 1)\delta_n}{\beta_n + 1} \right)^{-(\frac{\alpha_n}{2}+1)} \\
&\propto (\beta_n(y_0 - \mu_n)^2 + (\beta_n + 1)\delta_n)^{\left(\frac{\alpha_n+1}{2} \right)} \\
&\propto \left[1 + \frac{\beta_n \alpha_n}{(\beta_n + 1)\delta_n \alpha_n} (y_0 - \mu_n)^2 \right]^{-(\frac{\alpha_n+1}{2})},
\end{aligned}$$

where we have that $\left[1 + \frac{(\beta_n + 1)^2 \left(\mu - \left(\frac{\beta_n \mu_n + y_0}{\beta_n + 1} \right) \right)^2}{\beta_n(y_0 - \mu_n)^2 + (\beta_n + 1)\delta_n} \right]^{-(\frac{\alpha_n}{2}+1)}$ is the kernel of a Student's t density with degrees of freedom $\alpha_n + 1$ and scale $\frac{\beta_n(y_0 - \mu_n)^2 + (\beta_n + 1)\delta_n}{(\beta_n + 1)^2(\alpha_n + 1)}$.

The last expression is the kernel of a Student's t density, that is, $Y_0|\mathbf{y} \sim t\left(\mu_n, \frac{(\beta_n + 1)\delta_n}{\beta_n \alpha_n}, \alpha_n\right)$.

The multivariate normal-normal/inverse-Wishart model

We show in subsection 3.1 that the multivariate normal distribution is in the exponential family where

$$C(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp \left\{ -\frac{1}{2} (tr(\boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}) + \log(|\boldsymbol{\Sigma}|)) \right\},$$

$$\eta(\boldsymbol{\mu}, \boldsymbol{\Sigma})^\top = \left[(vec(\boldsymbol{\Sigma}^{-1}))^\top \quad (vec(\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}))^\top \right]^\top,$$

$$T(\mathbf{y}) = \left[-\frac{1}{2} \left(vec(\mathbf{S})^\top + N vec(\hat{\boldsymbol{\mu}} \hat{\boldsymbol{\mu}}^\top)^\top \right) \quad -N \hat{\boldsymbol{\mu}}^\top \right]^\top$$

and

$$h(\mathbf{y}) = (2\pi)^{-pN/2}.$$

Then, its conjugate prior distribution should have the form

$$\begin{aligned} \pi(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &\propto \exp \left\{ -\frac{b_0}{2} (\text{tr}(\boldsymbol{\mu}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}) + \log(|\boldsymbol{\Sigma}|)) \right\} \\ &\quad \times \exp \left\{ \mathbf{a}_{01}^\top \text{vec}(\boldsymbol{\Sigma}^{-1}) + \mathbf{a}_{02}^\top \text{vec}(\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}) \right\} \\ &= |\boldsymbol{\Sigma}|^{-b_0/2} \exp \left\{ -\frac{b_0}{2} (\text{tr}(\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu})) + \text{tr}(\mathbf{a}_{02}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}) \right\} \\ &\quad \times \exp \left\{ \mathbf{a}_{01}^\top \text{vec}(\boldsymbol{\Sigma}^{-1}) + \frac{\mathbf{a}_{02}^\top \boldsymbol{\Sigma}^{-1} \mathbf{a}_{02}}{2b_0} - \frac{\mathbf{a}_{02}^\top \boldsymbol{\Sigma}^{-1} \mathbf{a}_{02}}{2b_0} \right\} \\ &= |\boldsymbol{\Sigma}|^{-b_0/2} \exp \left\{ -\frac{b_0}{2} \left(\boldsymbol{\mu} - \frac{\mathbf{a}_{02}}{b_0} \right)^\top \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\mu} - \frac{\mathbf{a}_{02}}{b_0} \right) \right\} \\ &\quad \times \exp \left\{ -\frac{1}{2} \text{tr} \left(\left(\mathbf{A}_{01} - \frac{\mathbf{a}_{02} \mathbf{a}_{02}^\top}{b_0} \right) \boldsymbol{\Sigma}^{-1} \right) \right\} \\ &= \underbrace{|\boldsymbol{\Sigma}|^{-1/2} \exp \left\{ -\frac{b_0}{2} \left(\boldsymbol{\mu} - \frac{\mathbf{a}_{02}}{b_0} \right)^\top \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\mu} - \frac{\mathbf{a}_{02}}{b_0} \right) \right\}}_1 \\ &\quad \times \underbrace{|\boldsymbol{\Sigma}|^{-(\alpha_0+p+1)/2} \exp \left\{ -\frac{1}{2} \text{tr} \left(\left(\mathbf{A}_{01} - \frac{\mathbf{a}_{02} \mathbf{a}_{02}^\top}{b_0} \right) \boldsymbol{\Sigma}^{-1} \right) \right\}}_2, \end{aligned}$$

where b_0 is the hypothetical sample size, and \mathbf{a}_{01} and \mathbf{a}_{02} are p^2 and p dimensional vectors of prior sufficient statistics, where $\mathbf{a}_{01} = -\frac{1}{2} \text{vec}(\mathbf{A}_{01})$ such that \mathbf{A}_{01} is a $p \times p$ positive semi-definite matrix. Setting $b_0 = 1 + \alpha_0 + p + 1$, we have that the first part in the last expression is the kernel of a multivariate normal density with mean $\boldsymbol{\mu}_0 = \mathbf{a}_{02}/b_0$ and covariance $\frac{\boldsymbol{\Sigma}}{b_0}$, that is, $\boldsymbol{\mu} | \boldsymbol{\Sigma} \sim N_p \left(\boldsymbol{\mu}_0, \frac{\boldsymbol{\Sigma}}{b_0} \right)$, $b_0 = \beta_0$. It makes sense these hyperparameters because \mathbf{a}_{02} is the hypothetical sum of prior observations and b_0 is the hypothetical prior sample size. In addition, the second expression in the last line is the kernel of a inverse Wishart distribution with scale matrix $\boldsymbol{\Psi}_0 = \left(\mathbf{A}_{01} - \frac{\mathbf{a}_{02} \mathbf{a}_{02}^\top}{b_0} \right)$ and α_0 degrees of freedom, that is, $\boldsymbol{\Sigma} \sim IW_p(\boldsymbol{\Psi}_0, \alpha_0)$. Observe that $\boldsymbol{\Psi}_0$ has the same structure as the first part of the sufficient statistics in $T(\mathbf{y})$, just that it should be understood as coming from prior hypothetical observations.

Therefore, the prior distribution in this setting is normal/inverse-Wishart, and given conjugacy, the posterior distribution is in the

same family.

$$\begin{aligned}\pi(\boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{Y}) &\propto (2\pi)^{-pN/2} |\boldsymbol{\Sigma}|^{-N/2} \exp \left\{ -\frac{1}{2} \text{tr} \left[(\mathbf{S} + N(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^\top) \boldsymbol{\Sigma}^{-1} \right] \right\} \\ &\quad \times |\boldsymbol{\Sigma}|^{-1/2} \exp \left\{ -\frac{\beta_0}{2} \text{tr} \left[(\boldsymbol{\mu} - \boldsymbol{\mu}_0)(\boldsymbol{\mu} - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1} \right] \right\} |\boldsymbol{\Sigma}|^{-(\alpha_0+p+1)/2} \\ &\quad \times \exp \left\{ -\frac{1}{2} \text{tr}(\boldsymbol{\Psi}_0 \boldsymbol{\Sigma}^{-1}) \right\}.\end{aligned}$$

Taking into account that

$$\begin{aligned}N(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^\top + \beta_0(\boldsymbol{\mu} - \boldsymbol{\mu}_0)(\boldsymbol{\mu} - \boldsymbol{\mu}_0)^\top &= (N + \beta_0)(\boldsymbol{\mu} - \boldsymbol{\mu}_n)(\boldsymbol{\mu} - \boldsymbol{\mu}_n)^\top \\ &\quad + \frac{N\beta_0}{N + \beta_0}(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_0)(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_0)^\top,\end{aligned}$$

where $\boldsymbol{\mu}_n = \frac{N}{N+\beta_0}\hat{\boldsymbol{\mu}} + \frac{\beta_0}{N+\beta_0}\boldsymbol{\mu}_0$ is the posterior mean. We have

$$\begin{aligned}\pi(\boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{Y}) &\propto |\boldsymbol{\Sigma}|^{-1/2} \exp \left\{ -\frac{N + \beta_0}{2} \text{tr} \left[((\boldsymbol{\mu} - \boldsymbol{\mu}_n)(\boldsymbol{\mu} - \boldsymbol{\mu}_n)^\top) \boldsymbol{\Sigma}^{-1} \right] \right\} \\ &\quad \times |\boldsymbol{\Sigma}|^{-(N+\alpha_0+p+1)/2} \\ &\quad \times \exp \left\{ -\frac{1}{2} \text{tr} \left[\left(\boldsymbol{\Psi}_0 + \mathbf{S} + \frac{N\beta_0}{N + \beta_0}(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_0)(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_0)^\top \right) \boldsymbol{\Sigma}^{-1} \right] \right\}.\end{aligned}$$

Then, $\boldsymbol{\mu} | \boldsymbol{\Sigma}, \mathbf{Y} \sim N_p \left(\boldsymbol{\mu}_n, \frac{1}{\beta_n} \boldsymbol{\Sigma} \right)$, and $\boldsymbol{\Sigma} | \mathbf{Y} \sim IW(\boldsymbol{\Psi}_n, \alpha_n)$ where $\beta_n = N + \beta_0$, $\alpha_n = N + \alpha_0$ and $\boldsymbol{\Psi}_n = \boldsymbol{\Psi}_0 + \mathbf{S} + \frac{N\beta_0}{N + \beta_0}(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_0)(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_0)^\top$.

The marginal posterior of $\boldsymbol{\mu}$ is given by $\int_{\mathcal{S}} \pi(\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\boldsymbol{\Sigma}$ where \mathcal{S} is the space of positive semi-definite matrices. Then,

$$\begin{aligned}\pi(\boldsymbol{\mu} | \mathbf{Y}) &\propto \int_{\mathcal{S}} \left\{ |\boldsymbol{\Sigma}|^{-(\alpha_n+p+2)/2} \right. \\ &\quad \left. \exp \left\{ -\frac{1}{2} \text{tr} \left[(\beta_n(\boldsymbol{\mu} - \boldsymbol{\mu}_n)(\boldsymbol{\mu} - \boldsymbol{\mu}_n)^\top + \boldsymbol{\Psi}_n) \boldsymbol{\Sigma}^{-1} \right] \right\} \right\} d\boldsymbol{\Sigma} \\ &\propto \left| (\beta_n(\boldsymbol{\mu} - \boldsymbol{\mu}_n)(\boldsymbol{\mu} - \boldsymbol{\mu}_n)^\top + \boldsymbol{\Psi}_n) \right|^{-(\alpha_n+1)/2} \\ &= \left[|\boldsymbol{\Psi}_n| \times \left| 1 + \beta_n(\boldsymbol{\mu} - \boldsymbol{\mu}_n)^\top \boldsymbol{\Psi}_n^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_n) \right| \right]^{-(\alpha_n+1)/2} \\ &\propto \left(1 + \frac{1}{\alpha_n + 1 - p} (\boldsymbol{\mu} - \boldsymbol{\mu}_n)^\top \left(\frac{\boldsymbol{\Psi}_n}{(\alpha_n + 1 - p)\beta_n} \right)^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_n) \right)^{-(\alpha_n+1-p+p)/2},\end{aligned}$$

where the second line uses properties of the inverse Wishart distribution, and the third line uses a particular case of the Sylvester's determinant theorem.

We observe that the last line is the kernel of a multivariate t distribution, that is, $\boldsymbol{\mu} | \mathbf{Y} \sim t_p(v_n, \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ where $v_n = \alpha_n + 1 - p$ and $\boldsymbol{\Sigma}_n = \frac{\boldsymbol{\Psi}_n}{(\alpha_n + 1 - p)\beta_n}$.

The marginal likelihood is given by

$$p(\mathbf{Y}) = \frac{\Gamma_p \left(\frac{v_n}{2} \right)}{\Gamma_p \left(\frac{\alpha_0}{2} \right)} \frac{|\boldsymbol{\Psi}_0|^{\alpha_0/2}}{|\boldsymbol{\Psi}_n|^{\alpha_n/2}} \left(\frac{\beta_0}{\beta_n} \right)^{p/2} (2\pi)^{-Np/2},$$

where Γ_p is the multivariate gamma function (see Exercise 5).

The posterior predictive distribution is $\mathbf{Y}_0 | \mathbf{Y} \sim t_p(v_n, \boldsymbol{\mu}_n, (\beta_n + 1)\boldsymbol{\Sigma}_n)$ (see Exercise 6).

Example: Tangency portfolio of US tech stocks

The tangency portfolio is the portfolio that maximizes the Sharpe ratio, where this is the excess of return of a portfolio standardized by its risk.

We want to find the shares \mathbf{w} of a portfolio that maximizes the Sharpe ratio, where $\mu_{i,T+\kappa} = \mathbb{E}(R_{i,T+\kappa} - R_{f,T+\kappa} | \mathcal{I}_T)$, $R_{i,T+\kappa}$ and $R_{f,T+\kappa}$ are the returns of stock i and a risk-free asset. Observe that we have the expected value at period $T+\kappa$ of the excess return conditional on information up to T (\mathcal{I}_T), and $\boldsymbol{\Sigma}_{T+\kappa}$ is the covariance of the excess returns, which is a measure of risk. In particular,

$$\arg \max_{\mathbf{w} \in \mathbb{R}^p} \frac{\mathbf{w}^\top \boldsymbol{\mu}_{T+\kappa}}{\sqrt{\mathbf{w}^\top \boldsymbol{\Sigma}_{T+\kappa} \mathbf{w}}}; \quad \text{s.t.} \quad \mathbf{w}^\top \mathbf{1} = 1,$$

where the solution is

$$\mathbf{w}^* = \frac{\boldsymbol{\Sigma}_{T+\kappa}^{-1} \boldsymbol{\mu}_{T+\kappa}}{\mathbf{1}^\top \boldsymbol{\Sigma}_{T+\kappa}^{-1} \boldsymbol{\mu}_{T+\kappa}}.$$

If we want to find the optimal portfolio for the next period under the assumption that the excess of returns follow a multivariate normal distribution, which is a common assumption in these applications, we can set $\kappa = 1$, and use the predictive distribution of the excess of returns such that $\boldsymbol{\mu}_{T+1} = \boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_{T+1} = \frac{v_n}{v_n - 2}(\beta_n + 1)\boldsymbol{\Sigma}_n$ given the previous predictive result.

We apply this framework to ten tech stocks of the US market between January first, 2021, and September ninth, 2022. In particular, we use information from Yahoo Finance for Apple (AAPL), Netflix (NFLX), Amazon (AMZN), Microsoft (MSFT), Google (GOOG), Meta (META), Tesla (TSLA), NVIDIA Corporation (NVDA), Intel (INTC), and PayPal (PYPL).

R code. Optimal tangency portfolio: Tech shares

```

1 library(quantmod)
2 library(xts)
3 library(ggplot2)
4 library(gridExtra)
5 # grid.arrange
6 graphics.off()
7 rm(list=ls())
8 # Data Range
9 sdate <- as.Date("2021-01-01")
10 edate <- as.Date("2022-09-30")
11 Date <- seq(sdate, edate, by = "day")
12 tickers <- c("AAPL", "NFLX", "AMZN", "GOOG", "INTC", "META",
13             "MSFT", "TSLA", "NVDA", "PYPL")
14 p <- length(tickers)
15 # AAPL: Apple, NFLX: Netflix, AMZN: Amazon,
16 # MSFT: Microsoft, GOOG: Google, META: Meta,
17 # TSLA: Tesla, NVDA: NVIDIA Corporation
18 # INTC: Intel, PYPL: PayPal
19 ss_stock <- getSymbols(tickers, from=sdate, to=edate, auto.
20                         assign = T)
21 ss_stock <- purrr::map(tickers, function(x) Ad(get(x)))
22 ss_stock <- as.data.frame(purrr::reduce(ss_stock, merge))
23 colnames(ss_stock) <- tickers
24 # This is to get stock prices
25 ss_rtn <- as.data.frame(apply(ss_stock, 2, function(x) {diff
26     (log(x), 1)}))
27 # Daily returns
28 t10yr <- getSymbols(Symbols = "DGS10", src = "FRED", from=
29             sdate, to=edate, auto.assign = F)
30 # To get 10-Year US Treasury yield data from the
31 Federal Reserve Electronic Database (FRED)
32 t10yrd <- (1 + t10yr/100)^(1/365)-1
33 # Daily returns
34 t10yrd <- t10yrd[row.names(ss_rtn)]
35 Exc_rtn <- as.matrix(ss_rtn) - kronecker(t(rep(1, p)), as.
36                                         matrix(t10yrd))
37 # Excesses of return
38 df <- as.data.frame(Exc_rtn)
39 df$Date <- as.Date(rownames(df))
40 # Get months
41 df$Month <- months(df$Date)
42 # Get years
43 df$Year <- format(df$Date, format="%y")
44 # Aggregate on months and year and get mean
45 Data <- sapply(1:p, function(i) {
46     aggregate(df[, i] ~ Month + Year, df, mean)})
47 DataExcRtn <- matrix(0, length(Data[, 1]$Month), p)
48 for(i in 1:p){
49     DataExcRtn[, i] <- as.numeric(Data[, i]$`df[, i]`)
50 }
51 colnames(DataExcRtn) <- tickers
52 head(DataExcRtn)

```

R code. Optimal tangency portfolio: Tech shares

```

1 # Hyperparameters #
2 N <- dim(DataExcRtn)[1]
3 mu0 <- rep(0, p)
4 beta0 <- 1
5 Psi0 <- 100 * diag(p)
6 alpha0 <- p + 2
7 # Posterior parameters #
8 alphan <- N + alpha0
9 vn <- alphan + 1 - p
10 muhat <- colMeans(DataExcRtn)
11 mun <- N/(N + beta0) * muhat + beta0/(N + beta0) * mu0
12 S <- t(DataExcRtn - rep(1, N) %*% t(muhat)) %*% (DataExcRtn -
   rep(1, N) %*% t(muhat))
13 Psin <- Psi0 + S + N*beta0/(N + beta0)*(muhat - mu0) %*% t(
   muhat - mu0)
14 betan <- N + beta0
15 Sigman <- Psin/((alphan + 1 - p)*betan)
16 Covarn <- (Sigman * (1 + betan)) * vn / (vn - 2)
17 Covari <- solve(Covarn)
18 OptShare <- t(Covari %*% mun/as.numeric((t(rep(1, p)) %*% Covari
   %*% mun)))
19 colnames(OptShare) <- tickers
20 OptShare
21 AAPL    NFLX    AMZN    GOOG    INTC    META    MSFT    TSLA    NVDA
   PYPL
22 -0.019  0.248  0.102  -0.034  0.173  0.23  -0.022  -0.016  0.035
   0.301

```

We find that the optimal tangency portfolio is composed by 24.8%, 10.2%, 17.3%, 23%, 3.5% and 30.1% weights of Netflix, Amazon, Intel, Meta, NVIDIA and PayPal, and -1.9%, -3.4%, -2.2% and -1.6% weights of Apple, Google, Microsoft and Tesla. A negative weight means being short in financial jargon, that is, borrowing a stock to sell it.

3.3 Linear regression: The conjugate normal-normal/inverse gamma model

In this setting we analyze the conjugate normal-normal/inverse gamma model which is the workhorse in econometrics. In this model, the dependent variable

y_i is related to a set of regressors $\mathbf{x}_i = [x_{i1} \ x_{i2} \ \dots \ x_{iK}]^\top$ in a linear way, that is, $y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_K x_{iK} + \mu_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \mu_i$ where $\boldsymbol{\beta} = [\beta_1 \ \beta_2 \ \dots \ \beta_K]^\top$ and $\mu_i \stackrel{iid}{\sim} N(0, \sigma^2)$ is a stochastic error such that $\mathbb{E}[\mu_i | \mathbf{x}_i] = 0$.

$$\text{Defining } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22} & \dots & x_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NK} \end{bmatrix} \text{ and } \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{bmatrix},$$

we can write the model in matrix form: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\mu}$, where $\boldsymbol{\mu} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$. This implies that $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I})$. Then, the likelihood function is

$$\begin{aligned} p(\mathbf{y} | \boldsymbol{\beta}, \sigma^2, \mathbf{X}) &= (2\pi\sigma^2)^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\} \\ &\propto (\sigma^2)^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\}. \end{aligned}$$

The conjugate priors for the parameters are

$$\begin{aligned} \boldsymbol{\beta} | \sigma^2 &\sim N(\boldsymbol{\beta}_0, \sigma^2 \mathbf{B}_0), \\ \sigma^2 &\sim IG(\alpha_0/2, \delta_0/2). \end{aligned}$$

Then, the posterior distribution is

$$\begin{aligned} \pi(\boldsymbol{\beta}, \sigma^2 | \mathbf{y}, \mathbf{X}) &\propto (\sigma^2)^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\} \\ &\times (\sigma^2)^{-\frac{K}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{B}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right\} \\ &\times \frac{(\delta_0/2)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} \left(\frac{1}{\sigma^2} \right)^{\alpha_0/2+1} \exp \left\{ -\frac{\delta_0}{2\sigma^2} \right\} \\ &\propto (\sigma^2)^{-\frac{K}{2}} \exp \left\{ -\frac{1}{2\sigma^2} [\boldsymbol{\beta}^\top (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X}) \boldsymbol{\beta} - 2\boldsymbol{\beta}^\top (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \mathbf{X}^\top \mathbf{X} \hat{\boldsymbol{\beta}})] \right\} \\ &\times \left(\frac{1}{\sigma^2} \right)^{(\alpha_0+N)/2+1} \exp \left\{ -\frac{\delta_0 + \mathbf{y}^\top \mathbf{y} + \boldsymbol{\beta}_0^\top \mathbf{B}_0^{-1} \boldsymbol{\beta}_0}{2\sigma^2} \right\}, \end{aligned}$$

where $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ is the maximum likelihood estimator.

Adding and subtracting $\boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n$ to complete the square, where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}$ and $\boldsymbol{\beta}_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \mathbf{X}^\top \mathbf{X} \hat{\boldsymbol{\beta}})$,

$$\pi(\boldsymbol{\beta}, \sigma^2 | \mathbf{y}, \mathbf{X}) \propto \underbrace{(\sigma^2)^{-\frac{K}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \mathbf{B}_n^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_n) \right\}}_1 \\ \times \underbrace{(\sigma^2)^{-(\frac{\alpha_n}{2} + 1)} \exp \left\{ -\frac{\delta_n}{2\sigma^2} \right\}}_2.$$

The first expression is the kernel of a normal density function, $\boldsymbol{\beta} | \sigma^2, \mathbf{y}, \mathbf{X} \sim N(\boldsymbol{\beta}_n, \sigma^2 \mathbf{B}_n)$. The second expression is the kernel of a inverse gamma density, $\sigma^2 | \mathbf{y}, \mathbf{X} \sim IG(\alpha_n/2, \delta_n/2)$, where $\alpha_n = \alpha_0 + N$ and $\delta_n = \delta_0 + \mathbf{y}^\top \mathbf{y} + \boldsymbol{\beta}_0^\top \mathbf{B}_0^{-1} \boldsymbol{\beta}_0 - \boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n$.

Taking into account that

$$\begin{aligned} \boldsymbol{\beta}_n &= (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \mathbf{X}^\top \mathbf{X} \hat{\boldsymbol{\beta}}) \\ &= (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X} \hat{\boldsymbol{\beta}}, \end{aligned}$$

where $(\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{B}_0^{-1} = \mathbf{I}_K - (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X}$ [203]. Setting $\mathbf{W} = (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X}$ we have $\boldsymbol{\beta}_n = (\mathbf{I}_K - \mathbf{W}) \boldsymbol{\beta}_0 + \mathbf{W} \hat{\boldsymbol{\beta}}$, that is, the posterior mean of $\boldsymbol{\beta}$ is a weighted average between the sample and prior information, where the weights depend on the precision of each piece of information. Observe that when the prior covariance matrix is highly vague (non-informative), such that $\mathbf{B}_0^{-1} \rightarrow \mathbf{0}_K$, we obtain $\mathbf{W} \rightarrow \mathbf{I}_K$, such that $\boldsymbol{\beta}_n \rightarrow \hat{\boldsymbol{\beta}}$, that is, the posterior mean location parameter converges to the maximum likelihood estimator.

In addition, we know that the posterior conditional covariance matrix of the location parameters $\sigma^2 (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1} - \sigma^2 ((\mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{B}_0 + (\mathbf{X}^\top \mathbf{X})^{-1})^{-1} (\mathbf{X}^\top \mathbf{X})^{-1})$ is positive semi-definite.⁴ Given that $\sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}$ is the covariance matrix of the maximum likelihood estimator, we observe that prior information reduces estimation uncertainty.

Another way to see this model is considering that \mathbf{y} and $\boldsymbol{\beta}$ are random objects under the Bayesian framework. Thus, we can have the joint distribution of these two vectors,

$$\begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{y} \end{bmatrix} \sim N \left[\begin{pmatrix} \boldsymbol{\beta}_0 \\ \mathbf{X} \boldsymbol{\beta}_0 \end{pmatrix}, \sigma^2 \begin{pmatrix} \mathbf{B}_0 & \mathbf{B}_0 \mathbf{X}^\top \\ \mathbf{X} \mathbf{B}_0^\top & \mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N \end{pmatrix} \right],$$

where we use that $Cov[\boldsymbol{\beta}, \mathbf{y}] = \mathbb{E}[\boldsymbol{\beta} \mathbf{y}^\top] - \mathbb{E}[\boldsymbol{\beta}] \mathbb{E}[\mathbf{y}^\top] = \mathbb{E}[\boldsymbol{\beta} (\mathbf{X} \boldsymbol{\beta} + \boldsymbol{\mu})^\top] - \mathbb{E}[\boldsymbol{\beta}] \mathbb{E}[\mathbf{y}^\top] = [Var[\boldsymbol{\beta}] + \mathbb{E}[\boldsymbol{\beta}] \mathbb{E}[\boldsymbol{\beta}^\top]] \mathbf{X}^\top - \mathbb{E}[\boldsymbol{\beta}] \mathbb{E}[\mathbf{y}^\top] = \sigma^2 \mathbf{B}_0 \mathbf{X}^\top + \boldsymbol{\beta}_0 \boldsymbol{\beta}_0^\top \mathbf{X}^\top - \boldsymbol{\beta}_0 \boldsymbol{\beta}_0^\top \mathbf{X}^\top = \sigma^2 \mathbf{B}_0 \mathbf{X}^\top$.

Then, we can get the conditional distribution of $\boldsymbol{\beta} | \mathbf{y}$ using the properties of the multivariate normal distribution, this is normal with mean

⁴A particular case of the Woodbury matrix identity, $(\mathbf{A} + \mathbf{U} \mathbf{C} \mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{U} (\mathbf{C}^{-1} + \mathbf{V} \mathbf{A}^{-1} \mathbf{U})^{-1} \mathbf{V} \mathbf{A}^{-1}$.

equal to $\beta_0 + \mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1} (\mathbf{y} - \mathbf{X} \beta_0)$ and covariance matrix $\sigma^2 (\mathbf{B}_0 - \mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1} \mathbf{X} \mathbf{B}_0^\top)$. Observe that in this representation, the posterior mean is equal to the prior mean plus a correction term that takes into account the deviation between the observations and the prior expected value ($\mathbf{X} \beta_0$). The weight of this correction is given by the matrix $\mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1}$.

This form of expressing the posterior distribution is relevant to get some intuition of the Bayesian inference of time series models in the *Gaussian linear state-space representation* in Chapter 8, also known as the Kalman filter in time series literature.

We can show that both conditional posterior distributions are the same. In particular, the posterior mean in this representation is $[\mathbf{I}_K - \mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1} \mathbf{X}] \beta_0 + \mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1} \mathbf{y}$, where

$$\begin{aligned}\mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1} &= \mathbf{B}_0 \mathbf{X}^\top [\mathbf{I}_N - \mathbf{I}_N \mathbf{X} (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{I}_N \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{I}_N] \\ &= \mathbf{B}_0 [\mathbf{I}_K - \mathbf{X}^\top \mathbf{I}_N \mathbf{X} (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{I}_N \mathbf{X})^{-1}] \mathbf{X}^\top \\ &= \mathbf{B}_0 [\mathbf{I}_K - [\mathbf{I}_K - \mathbf{B}_0^{-1} (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{I}_N \mathbf{X})^{-1}]] \mathbf{X}^\top \\ &= (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top,\end{aligned}$$

where the first equality uses the Woodbury matrix identity (matrix inversion lemma), and the third equality uses $\mathbf{D}(\mathbf{D} + \mathbf{E})^{-1} = \mathbf{I} - \mathbf{E}(\mathbf{D} + \mathbf{E})^{-1}$.

Thus, $[\mathbf{I}_K - \mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1} \mathbf{X}] \beta_0 + \mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1} \mathbf{y} = [\mathbf{I}_K - (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X}] \beta_0 + (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = [\mathbf{I}_K - (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X}] \beta_0 + (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X} \hat{\beta}$. Again, we see that the posterior mean is a weighted average between the prior mean, and the maximum likelihood estimator.

The equality of variances of both approaches is as follows:

$$\begin{aligned}Var[\beta | \mathbf{y}] &= \sigma^2 (\mathbf{B}_0 - \mathbf{B}_0 \mathbf{X}^\top (\mathbf{X} \mathbf{B}_0 \mathbf{X}^\top + \mathbf{I}_N)^{-1} \mathbf{X} \mathbf{B}_0) \\ &= \sigma^2 (\mathbf{B}_0 - \mathbf{B}_0 \mathbf{X}^\top (\mathbf{I}_N - \mathbf{I}_N \mathbf{X} (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{I}_N \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{I}_N) \mathbf{X} \mathbf{B}_0) \\ &= \sigma^2 (\mathbf{B}_0 - \mathbf{B}_0 \mathbf{X}^\top \mathbf{X} \mathbf{B}_0 + \mathbf{B}_0 \mathbf{X}^\top \mathbf{X} (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X} \mathbf{B}_0) \\ &= \sigma^2 (\mathbf{B}_0 - \mathbf{B}_0 \mathbf{X}^\top \mathbf{X} \mathbf{B}_0 + \mathbf{B}_0 \mathbf{X}^\top \mathbf{X} [\mathbf{I}_K - (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{B}_0^{-1}] \mathbf{B}_0) \\ &= \sigma^2 (\mathbf{B}_0 - \mathbf{B}_0 \mathbf{X}^\top \mathbf{X} (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}) \\ &= \sigma^2 (\mathbf{B}_0 [\mathbf{I}_K - \mathbf{X}^\top \mathbf{X} (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}]) \\ &= \sigma^2 (\mathbf{B}_0 [\mathbf{I}_K - (\mathbf{I}_K - \mathbf{B}_0^{-1} (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1})]) \\ &= \sigma^2 (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1},\end{aligned}$$

where the second equality uses the Woodbury matrix identity, the fourth equality uses $(\mathbf{D} + \mathbf{E})^{-1} \mathbf{D} = \mathbf{I} - (\mathbf{D} + \mathbf{E})^{-1} \mathbf{E}$, and the seventh equality uses $\mathbf{D}(\mathbf{D} + \mathbf{E})^{-1} = \mathbf{I} - \mathbf{E}(\mathbf{D} + \mathbf{E})^{-1}$.

Now, we calculate the posterior marginal distribution of β following the standard approach,

$$\begin{aligned}\pi(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) &= \int_0^\infty \pi(\boldsymbol{\beta}, \sigma^2|\mathbf{y}, \mathbf{X}) d\sigma^2 \\ &= \int_0^\infty \left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_n+K}{2}+1} \exp\left\{-\frac{s}{2\sigma^2}\right\} d\sigma^2,\end{aligned}$$

where $s = \delta_n + (\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \mathbf{B}_n^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_n)$. Then we can write

$$\begin{aligned}\pi(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) &= \int_0^\infty \left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_n+K}{2}+1} \exp\left\{-\frac{s}{2\sigma^2}\right\} d\sigma^2 \\ &= \frac{\Gamma((\alpha_n+K)/2)}{(s/2)^{(\alpha_n+K)/2}} \int_0^\infty \frac{(s/2)^{(\alpha_n+K)/2}}{\Gamma((\alpha_n+K)/2)} (\sigma^2)^{-(\alpha_n+K)/2-1} \exp\left\{-\frac{s}{2\sigma^2}\right\} d\sigma^2.\end{aligned}$$

The right term is the integral of the probability density function of an inverse gamma distribution with parameters $\nu = (\alpha_n+K)/2$ and $\tau = s/2$. Since we are integrating over the whole support of σ^2 , the integral is equal to 1, and therefore

$$\begin{aligned}\pi(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) &= \frac{\Gamma((\alpha_n+K)/2)}{(s/2)^{(\alpha_n+K)/2}} \\ &\propto s^{-(\alpha_n+K)/2} \\ &= [\delta_n + (\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \mathbf{B}_n^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_n)]^{-(\alpha_n+K)/2} \\ &= \left[1 + \frac{(\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \left(\frac{\delta_n}{\alpha_n} \mathbf{B}_n\right)^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_n)}{\alpha_n}\right]^{-(\alpha_n+K)/2} (\delta_n)^{-(\alpha_n+K)/2} \\ &\propto \left[1 + \frac{(\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \mathbf{H}_n^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_n)}{\alpha_n}\right]^{-(\alpha_n+K)/2},\end{aligned}$$

where $\mathbf{H}_n = \frac{\delta_n}{\alpha_n} \mathbf{B}_n$. This last expression is a multivariate t distribution for $\boldsymbol{\beta}$, $\boldsymbol{\beta}|\mathbf{y}, \mathbf{X} \sim t_K(\alpha_n, \boldsymbol{\beta}_n, \mathbf{H}_n)$.

Observe that as we have incorporated the uncertainty of the variance, the posterior for $\boldsymbol{\beta}$ changes from a normal to a t distribution, which has heavier tails, indicating more uncertainty.

The marginal likelihood of this model is

$$p(\mathbf{y}) = \int_0^\infty \int_{R^K} \pi(\boldsymbol{\beta}|\sigma^2, \mathbf{B}_0, \boldsymbol{\beta}_0) \pi(\sigma^2|\alpha_0/2, \delta_0/2) p(\mathbf{y}|\boldsymbol{\beta}, \sigma^2, \mathbf{X}) d\sigma^2 d\boldsymbol{\beta}.$$

Taking into account that $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{B}_0^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_0) = (\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \mathbf{B}_n^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_n) + m$, where $m = \mathbf{y}^\top \mathbf{y} + \boldsymbol{\beta}_0^\top \mathbf{B}_0^{-1} \boldsymbol{\beta}_0 - \boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n$, we

have that

$$\begin{aligned}
 p(\mathbf{y}) &= \int_0^\infty \int_{R^K} \pi(\boldsymbol{\beta}|\sigma^2) \pi(\sigma^2) p(\mathbf{y}|\boldsymbol{\beta}, \sigma^2, \mathbf{X}) d\sigma^2 d\boldsymbol{\beta} \\
 &= \int_0^\infty \pi(\sigma^2) \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}m\right\} \frac{1}{(2\pi\sigma^2)^{K/2} |\mathbf{B}_0|^{1/2}} \\
 &\quad \times \int_{R^K} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \mathbf{B}_n^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_n)\right\} d\sigma^2 d\boldsymbol{\beta} \\
 &= \int_0^\infty \pi(\sigma^2) \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}m\right\} \frac{|\mathbf{B}_n|^{1/2}}{|\mathbf{B}_0|^{1/2}} d\sigma^2 \\
 &= \int_0^\infty \frac{(\delta_0/2)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} \left(\frac{1}{\sigma^2}\right)^{\alpha_0/2+1} \exp\left\{\left(-\frac{\delta_0}{2\sigma^2}\right)\right\} \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}m\right\} \frac{|\mathbf{B}_n|^{1/2}}{|\mathbf{B}_0|^{1/2}} d\sigma^2 \\
 &= \frac{1}{(2\pi)^{N/2}} \frac{(\delta_0/2)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} \frac{|\mathbf{B}_n|^{1/2}}{|\mathbf{B}_0|^{1/2}} \int_0^\infty \left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_0+N}{2}+1} \exp\left\{\left(-\frac{\delta_0+m}{2\sigma^2}\right)\right\} d\sigma^2 \\
 &= \frac{1}{\pi^{N/2}} \frac{\delta_0^{\alpha_0/2}}{\delta_n^{\alpha_n/2}} \frac{|\mathbf{B}_n|^{1/2}}{|\mathbf{B}_0|^{1/2}} \frac{\Gamma(\alpha_n/2)}{\Gamma(\alpha_0/2)}.
 \end{aligned}$$

We can show that $\delta_n = \delta_0 + \mathbf{y}^\top \mathbf{y} + \boldsymbol{\beta}_0^\top \mathbf{B}_0^{-1} \boldsymbol{\beta}_0 - \boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n = \delta_0 + (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) + (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)^\top ((\mathbf{X}^\top \mathbf{X})^{-1} + \mathbf{B}_0)^{-1} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)$ (see Exercise 7). Therefore, if we want to compare two models under this setting, the Bayes factor is

$$\begin{aligned}
 BF_{12} &= \frac{p(\mathbf{y}|\mathcal{M}_1)}{p(\mathbf{y}|\mathcal{M}_2)} \\
 &= \frac{\frac{\delta_0^{\alpha_{10}/2}}{\delta_{1n}^{\alpha_{1n}/2}} \frac{|\mathbf{B}_{1n}|^{1/2}}{|\mathbf{B}_{10}|^{1/2}} \frac{\Gamma(\alpha_{1n}/2)}{\Gamma(\alpha_{10}/2)}}{\frac{\delta_{20}^{\alpha_{20}/2}}{\delta_{2n}^{\alpha_{2n}/2}} \frac{|\mathbf{B}_{2n}|^{1/2}}{|\mathbf{B}_{20}|^{1/2}} \frac{\Gamma(\alpha_{2n}/2)}{\Gamma(\alpha_{20}/2)}},
 \end{aligned}$$

where subscripts 1 and 2 refer to each model, respectively.

Observe that *ceteris paribus*, the model having better fit, coherence between sample and prior information regarding location parameters, higher prior to posterior precision and less parameters is favored by the Bayes factor. Observe that the Bayes factor rewards model fit as the sum of squared errors is in δ_n , the better fit (lower sum of squared errors), the better the Bayes factor. In addition, a weighted distance between sample and prior location parameters also appears in δ_n , the greater this distance, the worse is model support. The ratio of determinants between posterior and prior covariance matrices is also present, the higher this ratio, the better for the Bayes factor supporting a model due to information gains. To see the effect of model's parsimony, let's take the common situation in applications where $\mathbf{B}_{j0} = c\mathbf{I}_{K_j}$ then $|\mathbf{B}_{j0}| = c^{K_j}$

such that $\left(\frac{|\mathbf{B}_{20}|}{|\mathbf{B}_{10}|}\right)^{1/2} = \left(\frac{c^{K_2/2}}{c^{K_1/2}}\right)$, if $K_2/K_1 > 1$ and $c \rightarrow \infty$, the latter implying a non-informative prior, then $BF_{12} \rightarrow \infty$, this means infinite evidence supporting the parsimonious model no matter what sample information says. Comparing models having the same number of regressors ($K_1 = K_2$) is not a safe ground as $|\mathbf{B}_0|$ depending on measure units of the regressors such that

conclusions regarding model selection depending on this, which is not a nice property. This prevents against using non-informative priors when performing model selection in the Bayesian framework. Observe that this is not the case when $\alpha_0 \rightarrow 0$ and $\delta_0 \rightarrow 0$, which implies a non-informative prior for the variance parameter.⁵ We observe here that $\Gamma(\alpha_{j0})$ cancels out, $\alpha_{jn} \rightarrow N$ and $\delta_{jn} \rightarrow (\mathbf{y} - \mathbf{X}_j \hat{\boldsymbol{\beta}}_j)^\top (\mathbf{y} - \mathbf{X}_j \hat{\boldsymbol{\beta}}_j) + (\hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_{j0})^\top ((\mathbf{X}_j^\top \mathbf{X}_j)^{-1} + \mathbf{B}_{j0})^{-1} (\hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_{j0})$, therefore there is no effect. This is due to σ^2 being a common parameter in both models.

In general, we can use non-informative priors for common parameters to all models, but we cannot use non-informative priors for non-common parameters when performing model selection using the Bayes factor. This issue raises the question of how to set informative priors. On one hand, we have those who advocate for *subjective* priors [181, 50, 192, 136]; on the other hand, those who prefer *objective* priors [11, 130, 106, 15]. Regarding the former, eliciting *subjective* priors, that is, "... formulating a person's knowledge and beliefs about one or more uncertain quantities into a (joint) probability distribution for those quantities" [70], is a very difficult task due to human beings' heuristics and biases associated with representativeness, information availability, conservatism, overconfidence, anchoring and adjustment issues [212]. However, there are good efforts using predictive and structural elicitation procedures [112, 113]. Regarding the latter, there are the *reference priors* that are designed to have minimal impact on the posterior distribution and being invariant to different parametrizations of the model [19, Chap. 5]. A remarkable example of *reference priors* is the *Jeffreys' prior* [107], whose origin is the critic to *non-informative priors* that were not invariant to transformations of the parameters' space. In particular, the *Jeffreys' prior* is $\pi(\boldsymbol{\theta}) \propto |I(\boldsymbol{\theta})|^{1/2}$, where $I(\boldsymbol{\theta}) = \mathbb{E} \left(-\frac{\partial^2 p(\mathbf{y}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \right)$, that is, $I(\boldsymbol{\theta})$ is the Fisher's information matrix. However, *Jeffreys' prior* is often improper, which means that it does not work well for model selection. Thus, a standard *objective* approach is to use the *intrinsic priors* [17], where a *minimal training* dataset is used with a *reference prior* to get a proper posterior distribution, and then, use this proper distribution as a prior, and proceed in the standard way using the remaining dataset. In this way, we end up with meaningful Bayes factors for model selection.

Regardless of using a *subjective* or *objective* approach to define a prior distribution, it is always a good idea to assess the sensitivity of the posterior results to the prior assumptions. This is commonly done using local or pointwise assessments, such as partial derivatives [87, 104, 94] or, more often, in terms of multiple evaluations (*scenario analysis*) [184, 122, 4]. Recently, [103] extend these approaches to perform sensitivity analysis in high-dimensional hyperparameter settings.

Returning to the linear model, the posterior predictive is equal to

⁵[77] prevents against this common practice.

$$\begin{aligned}\pi(\mathbf{Y}_0|\mathbf{y}) &= \int_0^\infty \int_{R^K} p(\mathbf{Y}_0|\boldsymbol{\beta}, \sigma^2, \mathbf{y}) \pi(\boldsymbol{\beta}|\sigma^2, \mathbf{y}) \pi(\sigma^2|\mathbf{y}) d\boldsymbol{\beta} d\sigma^2 \\ &= \int_0^\infty \int_{R^K} p(\mathbf{Y}_0|\boldsymbol{\beta}, \sigma^2) \pi(\boldsymbol{\beta}|\sigma^2, \mathbf{y}) \pi(\sigma^2|\mathbf{y}) d\boldsymbol{\beta} d\sigma^2,\end{aligned}$$

where we take into account independence between \mathbf{Y}_0 and \mathbf{Y} . Given \mathbf{X}_0 , which is the $N_0 \times K$ matrix of regressors associated with \mathbf{Y}_0 , Then,

$$\begin{aligned}\pi(\mathbf{Y}_0|\mathbf{y}) &= \int_0^\infty \int_{R^K} \left\{ (2\pi\sigma^2)^{-\frac{N_0}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{Y}_0 - \mathbf{X}_0\boldsymbol{\beta})^\top (\mathbf{Y}_0 - \mathbf{X}_0\boldsymbol{\beta}) \right\} \right. \\ &\quad \times (2\pi\sigma^2)^{-\frac{K}{2}} |\mathbf{B}_n|^{-1/2} \exp \left\{ -\frac{1}{2\sigma^2} (\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \mathbf{B}_n^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_n) \right\} \\ &\quad \left. \times \frac{(\delta_n/2)^{\alpha_n/2}}{\Gamma(\alpha_n/2)} \left(\frac{1}{\sigma^2} \right)^{\alpha_n/2+1} \exp \left\{ -\frac{\delta_n}{2\sigma^2} \right\} \right\} d\boldsymbol{\beta} d\sigma^2.\end{aligned}$$

Setting $\mathbf{M} = (\mathbf{X}_0^\top \mathbf{X}_0 + \mathbf{B}_n^{-1})$ and $\boldsymbol{\beta}_* = \mathbf{M}^{-1}(\mathbf{B}_n^{-1}\boldsymbol{\beta}_n + \mathbf{X}_0^\top \mathbf{Y}_0)$, we have $(\mathbf{Y}_0 - \mathbf{X}_0\boldsymbol{\beta})^\top (\mathbf{Y}_0 - \mathbf{X}_0\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_n)^\top \mathbf{B}_n^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_n) = (\boldsymbol{\beta} - \boldsymbol{\beta}_*)^\top \mathbf{M} (\boldsymbol{\beta} - \boldsymbol{\beta}_*) + \boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n + \mathbf{Y}_0^\top \mathbf{Y}_0 - \boldsymbol{\beta}_*^\top \mathbf{M} \boldsymbol{\beta}_*$. Thus,

$$\begin{aligned}\pi(\mathbf{Y}_0|\mathbf{y}) &\propto \int_0^\infty \left\{ \left(\frac{1}{\sigma^2} \right)^{-\frac{K+N_0+\alpha_n}{2}+1} \exp \left\{ -\frac{1}{2\sigma^2} (\boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n + \mathbf{Y}_0^\top \mathbf{Y}_0 - \boldsymbol{\beta}_*^\top \mathbf{M} \boldsymbol{\beta}_* + \delta_n) \right\} \right. \\ &\quad \left. \times \int_{R^K} \exp \left\{ -\frac{1}{2\sigma^2} (\boldsymbol{\beta} - \boldsymbol{\beta}_*)^\top \mathbf{M} (\boldsymbol{\beta} - \boldsymbol{\beta}_*) \right\} d\boldsymbol{\beta} \right\} d\sigma^2,\end{aligned}$$

where the term in the second integral is the kernel of a multivariate normal density with mean $\boldsymbol{\beta}_*$ and covariance matrix $\sigma^2 \mathbf{M}^{-1}$. Then,

$$\pi(\mathbf{Y}_0|\mathbf{y}) \propto \int_0^\infty \left(\frac{1}{\sigma^2} \right)^{\frac{N_0+\alpha_n}{2}+1} \exp \left\{ -\frac{1}{2\sigma^2} (\boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n + \mathbf{Y}_0^\top \mathbf{Y}_0 - \boldsymbol{\beta}_*^\top \mathbf{M} \boldsymbol{\beta}_* + \delta_n) \right\} d\sigma^2,$$

which is the kernel of an inverse gamma density. Thus,

$$\pi(\mathbf{Y}_0|\mathbf{y}) \propto \left[\frac{\boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n + \mathbf{Y}_0^\top \mathbf{Y}_0 - \boldsymbol{\beta}_*^\top \mathbf{M} \boldsymbol{\beta}_* + \delta_n}{2} \right]^{-\frac{\alpha_n+N_0}{2}}.$$

Setting $\mathbf{C}^{-1} = \mathbf{I}_{N_0} + \mathbf{X}_0 \mathbf{B}_n \mathbf{X}_0^\top$ such that $\mathbf{C} = \mathbf{I}_{N_0} - \mathbf{X}_0 (\mathbf{B}_n^{-1} +$

$\mathbf{X}_0^\top \mathbf{X}_0)^{-1} \mathbf{X}_0^\top = \mathbf{I}_{N_0} - \mathbf{X}_0 \mathbf{M}^{-1} \mathbf{X}_0^\top$,⁶ and $\boldsymbol{\beta}_{**} = \mathbf{C}^{-1} \mathbf{X}_0 \mathbf{M}^{-1} \mathbf{B}_n^{-1} \boldsymbol{\beta}_n$, then

$$\begin{aligned} \boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n + \mathbf{Y}_0^\top \mathbf{Y}_0 - \boldsymbol{\beta}_*^\top \mathbf{M} \boldsymbol{\beta}_* &= \boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} \boldsymbol{\beta}_n + \mathbf{Y}_0^\top \mathbf{Y}_0 - (\boldsymbol{\beta}_n^\top \mathbf{B}_n^{-1} + \mathbf{Y}_0^\top \mathbf{X}_0) \mathbf{M}^{-1} (\mathbf{B}_n^{-1} \boldsymbol{\beta}_n + \mathbf{X}_0^\top \mathbf{Y}_0) \\ &= \boldsymbol{\beta}_n^\top (\mathbf{B}_n^{-1} - \mathbf{B}_n^{-1} \mathbf{M}^{-1} \mathbf{B}_n^{-1}) \boldsymbol{\beta}_n + \mathbf{Y}_0^\top \mathbf{C} \mathbf{Y}_0 \\ &\quad - 2 \mathbf{Y}_0^\top \mathbf{C} \mathbf{C}^{-1} \mathbf{X}_0 \mathbf{M}^{-1} \mathbf{B}_n^{-1} \boldsymbol{\beta}_n + \boldsymbol{\beta}_{**}^\top \mathbf{C} \boldsymbol{\beta}_{**} - \boldsymbol{\beta}_{**}^\top \mathbf{C} \boldsymbol{\beta}_{**} \\ &= \boldsymbol{\beta}_n^\top (\mathbf{B}_n^{-1} - \mathbf{B}_n^{-1} \mathbf{M}^{-1} \mathbf{B}_n^{-1}) \boldsymbol{\beta}_n + (\mathbf{Y}_0 - \boldsymbol{\beta}_{**})^\top \mathbf{C} (\mathbf{Y}_0 - \boldsymbol{\beta}_{**}) \\ &\quad - \boldsymbol{\beta}_{**}^\top \mathbf{C} \boldsymbol{\beta}_{**}, \end{aligned}$$

where $\boldsymbol{\beta}_n^\top (\mathbf{B}_n^{-1} - \mathbf{B}_n^{-1} \mathbf{M}^{-1} \mathbf{B}_n^{-1}) \boldsymbol{\beta}_n = \boldsymbol{\beta}_{**}^\top \mathbf{C} \boldsymbol{\beta}_{**}$ and $\boldsymbol{\beta}_{**} = \mathbf{X}_0 \boldsymbol{\beta}_n$ (see Exercise 8).

Then,

$$\begin{aligned} \pi(\mathbf{Y}_0 | \mathbf{y}) &\propto \left[\frac{(\mathbf{Y}_0 - \mathbf{X}_0 \boldsymbol{\beta}_n)^\top \mathbf{C} (\mathbf{Y}_0 - \mathbf{X}_0 \boldsymbol{\beta}_n) + \delta_n}{2} \right]^{-\frac{\alpha_n + N_0}{2}} \\ &\propto \left[\frac{(\mathbf{Y}_0 - \mathbf{X}_0 \boldsymbol{\beta}_n)^\top \left(\frac{\mathbf{C} \alpha_n}{\delta_n} \right) (\mathbf{Y}_0 - \mathbf{X}_0 \boldsymbol{\beta}_n)}{\alpha_n} + 1 \right]^{-\frac{\alpha_n + N_0}{2}}. \end{aligned}$$

The posterior predictive is a multivariate t distribution, $\mathbf{Y}_0 | \mathbf{y} \sim t \left(\mathbf{X}_0 \boldsymbol{\beta}_n, \frac{\delta_n (\mathbf{I}_{N_0} + \mathbf{X}_0 \mathbf{B}_n \mathbf{X}_0^\top)}{\alpha_n}, \alpha_n \right)$ centered at $\mathbf{X}_0 \boldsymbol{\beta}_n$.

Example: Demand of electricity

We study in this example the determinants of monthly demand of electricity by Colombian households. There is information of 2103 households, particularly, average price (USD/kWh), indicators of socioeconomic conditions of the neighborhood where the household is located (IndSocio1 is the lowest and IndSocio3 is the highest), an indicator if the household is located in a municipality that is above 1000 meters above the sea level, the number of rooms in the house, the number of members of the households, presence of children in the household (1 is yes), and monthly income (USD). The specification is

$$\begin{aligned} \log(\text{Electricity}_i) &= \beta_1 \log(\text{price}_i) + \beta_2 \text{IndSocio1}_i + \beta_3 \text{IndSocio2}_i + \beta_4 \text{Altitude}_i \\ &\quad + \beta_5 \text{Nrooms}_i + \beta_6 \text{HouseholdMem}_i + \beta_7 \text{Children}_i \\ &\quad + \beta_8 \log(\text{Income}_i) + \beta_9 + \mu. \end{aligned}$$

We use a non-informative vague prior setting such that $\alpha_0 = \delta_0 = 0.001$, $\boldsymbol{\beta}_0 = \mathbf{0}$ and $\mathbf{B}_0 = c_0 \mathbf{I}_k$, where $c_0 = 1000$ and k is the number of regressors.

The results from the R code (see below) is that the posterior mean of the own-price of electricity demand is -1.09, and the 95% symmetric credible interval is (-1.47, -0.71). Households in neighborhoods of low socioeconomic

⁶Using $(\mathbf{A} + \mathbf{BDC})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{D}^{-1} + \mathbf{C} \mathbf{A}^{-1} \mathbf{B})^{-1} \mathbf{C} \mathbf{A}^{-1}$

conditions and located in municipalities 1000 meters above the sea level consume less electricity, 32.7% and 19.7% on average, respectively. An additional room implies a 8.7% increase in electricity consumption, and another household member increases consumption in 5.9% on average. The income elasticity mean estimate is 0.074, which means that 10% increase of income increases 0.74% electricity demand.

We want to check the results of the Bayes factor comparing the previous specification (model 1) with other specification without considering the price of electricity (model 2), that is,

$$\begin{aligned}\log(\text{Electricity}_i) = & \beta_1 \text{IndSocio1}_i + \beta_2 \text{IndSocio2}_i + \beta_3 \text{Altitude}_i + \beta_4 \text{Nrooms}_i \\ & + \beta_5 \text{HouseholdMem}_i + \beta_6 \text{Children}_i + \beta_7 \log(\text{Income}_i) \\ & + \beta_8 + \mu.\end{aligned}$$

In particular, we check what happens as c_0 increases from 10^0 to 10^{20} . We see that when $c_0 = 1$, $BF_{12} = 8.68 \times 10^{+16}$, which means very strong evidence in favor of the model including the price of electricity. However, as c_0 increases, the Bayes factor decreases, which means evidence supporting the model 2, for instance, $BF_{12} = 3.11 \times 10^{-4}$ when $c_0 = 10^{20}$. This is an example of the problem of using non-informative priors to calculate the Bayes factor; there is very strong evidence to support the parsimonious model when $c_0 \rightarrow \infty$.

We can get the posterior predictive distribution of the monthly electricity demand of a household located in the lowest socioeconomic condition in a municipality located below 1000 meters above the sea level, 2 rooms, 3 members with children, a monthly income equal to USD 500, and an electricity price equal to USD/kWh 0.15. Figure 3.1 shows the histogram of the predictive posterior distribution, the highest posterior density credible interval at 95% is between kWh 44.4 and kWh 373.9, and the posterior mean is kWh 169.4.

R code. Demand of electricity, posterior predictive distribution

```

1 rm(list = ls())
2 set.seed(010101)
3 # Electricity demand
4 DataUt <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/Utilities.csv"
  , sep = ",", header = TRUE, quote = "")
5 library(dplyr)
6 DataUtEst <- DataUt %>%
  filter(Electricity != 0)
8 attach(DataUtEst)
9 # Dependent variable: Monthly consumption (kWh) in log
10 Y <- log(Electricity)
11 # Regressors quantity including intercept
12 X <- cbind(LnPriceElect, IndSocio1, IndSocio2, Altitude,
  Nrooms, HouseholdMem, Children, Lnincome, 1)
13 # LnPriceElect: Price per kWh (USD) in log
14 # IndSocio1, IndSocio2, IndSocio3: Indicators socio-economic
  condition (1) is the lowest and (3) the highest
15 # Altitude: Indicator of household location (1 is more than
  1000 meters above sea level)
16 # Nrooms: Number of rooms in house
17 # HouseholdMem: Number of household members
18 # Children: Indicator por presence of children in household
  (1)
19 # Lnincome: Monthly income (USD) in log
20 k <- dim(X)[2]
21 N <- dim(X)[1]
22 # Hyperparameters
23 d0 <- 0.001
24 a0 <- 0.001
25 b0 <- rep(0, k)
26 B0 <- 1000*diag(k)
27 # Posterior parameters
28 bhat <- solve(t(X)%*%X)%*%t(X)%*%Y
29 Bn <- as.matrix(Matrix:::forceSymmetric(solve(B0) + t(X)
  )%*%X)) # Force this matrix to be symmetric
30 bn <- Bn%*%(solve(B0)%*%b0 + t(X)%*%X%*%bhat)
31 dn <- as.numeric(d0 + t(Y)%*%Y+t(b0)%*%solve(B0)%*%b0-t(bn)%
  *%solve(Bn)%*%bn)
32 an <- a0 + N
33 Hn <- Bn*dn/an
34 # Posterior draws
35 S <- 10000 # Number of draws from posterior distributions
36 sig2 <- MCMCpack::rinvgamma(S,an/2,dn/2)
37 summary(coda::mcmc(sig2))

```

R code. Demand of electricity, posterior distribution

```

1 Iterations = 1:10000
2 Thinning interval = 1
3 Number of chains = 1
4 Sample size per chain = 10000
5
6 1. Empirical mean and standard deviation for each
7 variable, plus standard error of the mean:
8
9 Mean           SD      Naive SE Time-series SE
10 2.361e-01     7.617e-03 7.617e-05 7.617e-05
11
12 2. Quantiles for each variable:
13
14 2.5%    25%    50%    75%   97.5%
15 0.2217  0.2309  0.2360  0.2412  0.2513
16
17 Betas <- LaplacesDemon::rmvt(S, bn, Hn, an)
18 summary(coda::mcmc(Betas))
19 Iterations = 1:10000
20 Thinning interval = 1
21 Number of chains = 1
22 Sample size per chain = 10000
23
24 1. Empirical mean and standard deviation for each
25 variable, plus standard error of the mean:
26
27          Mean        SD      Naive SE Time-series SE
28 LnPriceElect -1.09043 0.19459 0.0019459 0.0019459
29 IndSocio1    -0.32783 0.05294 0.0005294 0.0005294
30 IndSocio2    -0.05737 0.04557 0.0004557 0.0004557
31 Altitude     -0.19780 0.02386 0.0002386 0.0002429
32 Nrooms       0.08731 0.01094 0.0001094 0.0001119
33 HouseholdMem 0.05987 0.01334 0.0001334 0.0001334
34 Children     0.05696 0.03043 0.0003043 0.0003043
35 Lnincome     0.07447 0.01223 0.0001223 0.0001223
36                  2.52296 0.35077 0.0035077 0.0035077
37
38 2. Quantiles for each variable:
39
40          2.5%    25%    50%    75%   97.5%
41 LnPriceElect -1.472069 -1.22432 -1.08961 -0.95703 -0.71429
42 IndSocio1    -0.435957 -0.36228 -0.32731 -0.29133 -0.22588
43 IndSocio2    -0.147252 -0.08744 -0.05757 -0.02650  0.03254
44 Altitude     -0.244759 -0.21372 -0.19783 -0.18164 -0.15094
45 Nrooms       0.066432 0.07985 0.08709 0.09480 0.10864
46 HouseholdMem 0.033623 0.05089 0.05975 0.06889 0.08596
47 Children     -0.002259 0.03637 0.05698 0.07736 0.11681
48 Lnincome     0.050536 0.06614 0.07449 0.08283 0.09852
49                  1.835507 2.28703 2.52165 2.76364 3.21199
50

```

R code. Demand of electricity, Bayes factor

```

1 # Log marginal function (multiply by -1 due to minimization)
2 LogMarLikLM <- function(X, c0){
3   k <- dim(X)[2]
4   N <- dim(X)[1]
5   # Hyperparameters
6   B0 <- c0*diag(k)
7   b0 <- rep(0, k)
8   # Posterior parameters
9   bhat <- solve(t(X)%*%X)%*%t(X)%*%Y
10  # Force this matrix to be symmetric
11  Bn <- as.matrix(Matrix::forceSymmetric(solve(solve(B0) + t
12    (X)%*%X)))
13  bn <- Bn%*%(solve(B0)%*%b0 + t(X)%*%X%*%bhat)
14  dn <- as.numeric(d0 + t(Y)%*%Y+t(b0)%*%solve(B0)%*%b0-t(bn
15    )%*%solve(Bn)%*%bn)
16  an <- a0 + N
17  # Log marginal likelihood
18  logpy <- (N/2)*log(1/pi)+(a0/2)*log(d0)-(an/2)*log(dn) +
19    0.5*log(det(Bn)/det(B0)) + lgamma(an/2)-lgamma(a0/2)
20  return(-logpy)
21 }
22 cs <- c(10^0, 10^3, 10^6, 10^10, 10^12, 10^15, 10^20)
23 # Observe -1 to recover the right sign
24 LogML <- sapply(cs, function(c) {-LogMarLikLM(c0=c, X = X)})
25 # Regressor without price
26 Xnew <- cbind(IndSocio1, IndSocio2, Altitude, Nrooms,
27   HouseholdMem, Children, Lnincome, 1)
28 # Observe -1 to recover the right sign
29 LogMLnew <- sapply(cs, function(c) {-LogMarLikLM(c0=c, X =
30   Xnew)})
31 # Bayes factor
32 BF <- exp(LogML - LogMLnew)
33 BF
34 8.687567e+16 1.006679e+05 3.108415e+03 3.108340e+01 3.108343
35   e+00 9.829443e-02 3.108343e-04
36 # Empirical Bayes: Obtain c0 maximizing the log
37 marginal likelihood
38 c0 <- c0
39 EB <- optim(c0, fn = LogMarLikLM, method = "Brent", lower =
40   0.0001, upper = 10^6, X = X)
41 EB$par
42 3.254822
43 EB$value
44 1404.108
45 EBnew <- optim(c0, fn = LogMarLikLM, method = "Brent", lower
46   = 0.0001, upper = 10^6, X = Xnew)
47 EBnew$par
48 10.00597
49 EBnew$value
50 1422.199
51 # Change of order to take into account the -1 in the
52   LogMarLikLM function
53 BFEM <- exp(EBnew$value - EB$value)
54 BFEM
55 71897938

```

R code. Demand of electricity, predictive distribution

```

1 # Predictive distribution
2 Xpred <- c(log(0.15), 1, 0, 0, 2, 3, 1, log(500), 1)
3 Mean <- Xpred%*%bn
4 Hn <- dn*(1+t(Xpred)%*%Bn%*%Xpred)/an
5 ExpKwH <- exp(LaplacesDemon::rmvt(S, Mean, Hn, an))
6 summary(ExpKwH)
7 Min. : 24.06
8 1st Qu.: 121.70
9 Median : 169.37
10 Mean : 189.60
11 3rd Qu.: 234.19
12 Max. : 1243.68
13 HDI <- HDInterval::hdi(ExpKwH, credMass = 0.95) # Highest
       posterior density credible interval
14 HDI
15 lower 44.40203
16 upper 373.86494
17 hist(ExpKwH, main = "Histogram: Monthly demand of
       electricity", xlab = "Monthly kWh", col = "blue", breaks
       = 50)

```

Histogram: Monthly demand of electricity

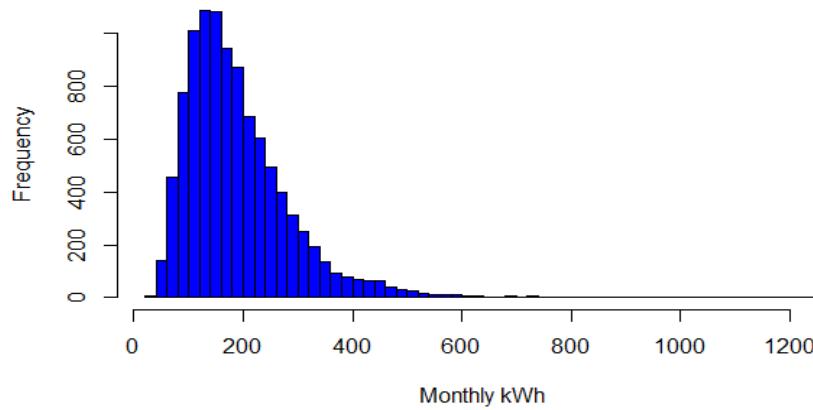


FIGURE 3.1

Histogram using the posterior predictive distribution of electricity demand

3.4 Multivariate linear regression: The conjugate normal-normal/inverse Wishart model

Let's study the multivariate regression setting where there are N -dimensional vectors \mathbf{y}_m , $m = 1, 2, \dots, M$ such that $\mathbf{y}_m = \mathbf{X}\boldsymbol{\beta}_m + \boldsymbol{\mu}_m$, \mathbf{X} is the set of common regressors, and $\boldsymbol{\mu}_m$ is the N -dimensional vector of stochastic errors for each equation such that $\mathbf{U} = [\boldsymbol{\mu}_1 \ \boldsymbol{\mu}_2 \ \dots \ \boldsymbol{\mu}_M] \sim MN_{N,M}(\mathbf{0}, \mathbf{I}_N, \boldsymbol{\Sigma})$, that is, a matrix variate normal distribution where $\boldsymbol{\Sigma}$ is the covariance matrix of each i -th row of \mathbf{U} , $i = 1, 2, \dots, N$, and we are assuming independence between the rows. Then, $\text{vec}(\mathbf{U}) \sim N_{N \times M}(\mathbf{0}, \boldsymbol{\Sigma} \otimes \mathbf{I}_N)$.⁷

This framework can be written in matrix form

$$\underbrace{\begin{bmatrix} y_{11} & y_{12} & \dots & y_{1M} \\ y_{21} & y_{22} & \dots & y_{2M} \\ \vdots & \vdots & \dots & \vdots \\ y_{N1} & y_{N2} & \dots & y_{NM} \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22} & \dots & x_{2K} \\ \vdots & \vdots & \dots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NK} \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1M} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2M} \\ \vdots & \vdots & \dots & \vdots \\ \beta_{K1} & \beta_{K2} & \dots & \beta_{KM} \end{bmatrix}}_{\boldsymbol{B}} + \underbrace{\begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1M} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2M} \\ \vdots & \vdots & \dots & \vdots \\ \mu_{N1} & \mu_{N2} & \dots & \mu_{NM} \end{bmatrix}}_{\mathbf{U}}$$

Therefore, $\mathbf{Y} \sim N_{N \times M}(\mathbf{X}\boldsymbol{B}, \boldsymbol{\Sigma} \otimes \mathbf{I}_N)$,⁸

$$\begin{aligned} p(\mathbf{Y} | \boldsymbol{B}, \boldsymbol{\Sigma}, \mathbf{X}) &\propto |\boldsymbol{\Sigma}|^{-N/2} \exp \left\{ -\frac{1}{2} \text{tr} [(\mathbf{Y} - \mathbf{X}\boldsymbol{B})^\top (\mathbf{Y} - \mathbf{X}\boldsymbol{B}) \boldsymbol{\Sigma}^{-1}] \right\} \\ &= |\boldsymbol{\Sigma}|^{-N/2} \exp \left\{ -\frac{1}{2} \text{tr} [(\mathbf{S} + (\boldsymbol{B} - \widehat{\boldsymbol{B}})^\top \mathbf{X}^\top \mathbf{X} (\boldsymbol{B} - \widehat{\boldsymbol{B}})) \boldsymbol{\Sigma}^{-1}] \right\}, \end{aligned}$$

where $\mathbf{S} = (\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{B}})^\top (\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{B}})$, $\widehat{\boldsymbol{B}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$ (see Exercise 9).

The conjugate prior for this models is $\pi(\boldsymbol{B}, \boldsymbol{\Sigma}) = \pi(\boldsymbol{B} | \boldsymbol{\Sigma})\pi(\boldsymbol{\Sigma})$ where $\boldsymbol{B} | \boldsymbol{\Sigma} \sim N_{K \times M}(\boldsymbol{B}_0, \mathbf{V}_0, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma} \sim IW(\boldsymbol{\Psi}_0, \alpha_0)$, that is,

⁷ vec denotes the vectorization operation, and \otimes denotes the kronecker product.

⁸We can write down the former expression in a more familiar way using vectorization properties, $\underbrace{\text{vec}(\mathbf{Y})}_{\mathbf{y}} = \underbrace{(\mathbf{I}_M \otimes \mathbf{X})}_{\mathbf{Z}} \underbrace{\text{vec}(\boldsymbol{B})}_{\boldsymbol{\beta}} + \underbrace{\text{vec}(\mathbf{U})}_{\boldsymbol{\mu}}$, where $\mathbf{y} \sim N_{N \times M}(\mathbf{Z}\boldsymbol{\beta}, \boldsymbol{\Sigma} \otimes \mathbf{I}_N)$.

$$\begin{aligned}\pi(\mathbf{B}, \boldsymbol{\Sigma}) &\propto |\boldsymbol{\Sigma}|^{-K/2} \exp \left\{ -\frac{1}{2} \operatorname{tr} [(\mathbf{B} - \mathbf{B}_0)^\top \mathbf{V}_0^{-1} (\mathbf{B} - \mathbf{B}_0) \boldsymbol{\Sigma}^{-1}] \right\} \\ &\quad \times |\boldsymbol{\Sigma}|^{-(\alpha_0 + M + 1)/2} \exp \left\{ -\frac{1}{2} \operatorname{tr} [\boldsymbol{\Psi}_0 \boldsymbol{\Sigma}^{-1}] \right\}.\end{aligned}$$

The posterior distribution is given by

$$\begin{aligned}\pi(\mathbf{B}, \boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{X}) &\propto p(\mathbf{Y} | \mathbf{B}, \boldsymbol{\Sigma}, \mathbf{X}) \pi(\mathbf{B} | \boldsymbol{\Sigma}) \pi(\boldsymbol{\Sigma}) \\ &\propto |\boldsymbol{\Sigma}|^{-\frac{N+K+\alpha_0+M+1}{2}} \\ &\quad \times \exp \left\{ -\frac{1}{2} \operatorname{tr} [(\boldsymbol{\Psi}_0 + \mathbf{S} + (\mathbf{B} - \mathbf{B}_0)^\top \mathbf{V}_0^{-1} (\mathbf{B} - \mathbf{B}_0) \right. \\ &\quad \left. + (\mathbf{B} - \hat{\mathbf{B}})^\top \mathbf{X}^\top \mathbf{X} (\mathbf{B} - \hat{\mathbf{B}})) \boldsymbol{\Sigma}^{-1}] \right\}.\end{aligned}$$

Completing the squares on \mathbf{B} and collecting the remaining terms in the bracket yields

$$\boldsymbol{\Psi}_0 + \mathbf{S} + (\mathbf{B} - \mathbf{B}_0)^\top \mathbf{V}_0^{-1} (\mathbf{B} - \mathbf{B}_0) + (\mathbf{B} - \hat{\mathbf{B}})^\top \mathbf{X}^\top \mathbf{X} (\mathbf{B} - \hat{\mathbf{B}}) = (\mathbf{B} - \mathbf{B}_n)^\top \mathbf{V}_n^{-1} (\mathbf{B} - \mathbf{B}_n) + \boldsymbol{\Psi}_n,$$

where

$$\begin{aligned}\mathbf{B}_n &= (\mathbf{V}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{V}_0^{-1} \mathbf{B}_0 + \mathbf{X}^\top \mathbf{Y}) = (\mathbf{V}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{V}_0^{-1} \mathbf{B}_0 + \mathbf{X}^\top \mathbf{X} \hat{\mathbf{B}}), \\ \mathbf{V}_n &= (\mathbf{V}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}, \\ \boldsymbol{\Psi}_n &= \boldsymbol{\Psi}_0 + \mathbf{S} + \mathbf{B}_0^\top \mathbf{V}_0^{-1} \mathbf{B}_0 + \hat{\mathbf{B}}^\top \mathbf{X}^\top \mathbf{X} \hat{\mathbf{B}} - \mathbf{B}_n^\top \mathbf{V}_n^{-1} \mathbf{B}_n.\end{aligned}$$

Thus, the posterior distribution can be written as

$$\begin{aligned}\pi(\mathbf{B}, \boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{X}) &\propto |\boldsymbol{\Sigma}|^{-K/2} \exp \left\{ -\frac{1}{2} \operatorname{tr} [(\mathbf{B} - \mathbf{B}_n)^\top \mathbf{V}_n^{-1} (\mathbf{B} - \mathbf{B}_n) \boldsymbol{\Sigma}^{-1}] \right\} \\ &\quad \times |\boldsymbol{\Sigma}|^{-\frac{N+\alpha_0+M+1}{2}} \exp \left\{ -\frac{1}{2} \operatorname{tr} [\boldsymbol{\Psi}_n \boldsymbol{\Sigma}^{-1}] \right\}.\end{aligned}$$

That is $\pi(\mathbf{B}, \boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{X}) = \pi(\mathbf{B} | \boldsymbol{\Sigma}, \mathbf{Y}, \mathbf{X}) \pi(\boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{X})$ where $\mathbf{B} | \boldsymbol{\Sigma}, \mathbf{Y}, \mathbf{X} \sim N_{K \times M}(\mathbf{B}_n, \mathbf{V}_n, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{X} \sim IW(\boldsymbol{\Psi}_n, \alpha_n)$, $\alpha_n = N + \alpha_0$. Observe again that we can write down the posterior mean as a weighted average between prior and sample information such that $\mathbf{V}_0 \rightarrow \infty$ implies $\mathbf{B}_n \rightarrow \hat{\mathbf{B}}$, as we show in the univariate linear model.

The marginal posterior for \mathbf{B} is given by

$$\begin{aligned}\pi(\mathbf{B} | \mathbf{Y}, \mathbf{X}) &\propto \int_{\boldsymbol{\Sigma}} |\boldsymbol{\Sigma}|^{-(\alpha_n + K + M + 1)/2} \\ &\quad \times \exp \left\{ -\frac{1}{2} \operatorname{tr} [(\mathbf{B} - \mathbf{B}_n)^\top \mathbf{V}_n^{-1} (\mathbf{B} - \mathbf{B}_n) + \boldsymbol{\Psi}_n] \boldsymbol{\Sigma}^{-1} \right\} d\boldsymbol{\Sigma} \\ &\propto |(\mathbf{B} - \mathbf{B}_n)^\top \mathbf{V}_n^{-1} (\mathbf{B} - \mathbf{B}_n) + \boldsymbol{\Psi}_n|^{-(K + \alpha_n)/2} \\ &= [|\boldsymbol{\Psi}_n| \times |\mathbf{I}_K + \mathbf{V}_n^{-1} (\mathbf{B} - \mathbf{B}_n) \boldsymbol{\Psi}_n^{-1} (\mathbf{B} - \mathbf{B}_n)^\top|]^{-(\alpha_n + 1 - M + K + M - 1)/2} \\ &\propto |\mathbf{I}_K + \mathbf{V}_n^{-1} (\mathbf{B} - \mathbf{B}_n) \boldsymbol{\Psi}_n^{-1} (\mathbf{B} - \mathbf{B}_n)^\top|^{-(\alpha_n + 1 - M + K + M - 1)/2}.\end{aligned}$$

The second line uses the inverse Wishart distribution, the third line the Sylvester's theorem, and the last line is the kernel of a matrix t distribution, that is, $\mathbf{B}|\mathbf{Y}, \mathbf{X} \sim T_{K \times M}(\mathbf{B}_n, \mathbf{V}_n, \boldsymbol{\Psi}_n)$ with $\alpha_n + 1 - M$ degrees of freedom.

Observe that $\text{vec}(\mathbf{B})$ has mean $\text{vec}(\mathbf{B}_n)$ and variance $(\mathbf{V}_n \otimes \boldsymbol{\Psi}_n)/(\alpha_n - M - 1)$ based on its marginal distribution. On the other hand, the variance based on the conditional distribution is $\mathbf{V}_n \otimes \boldsymbol{\Sigma}$, where the mean of $\boldsymbol{\Sigma}$ is $\boldsymbol{\Psi}_n/(\alpha_n - M - 1)$.

The marginal likelihood is the following,

$$\begin{aligned}
 p(\mathbf{Y}) &= \int_{\mathcal{B}} \int_{\mathcal{S}} \left\{ (2\pi)^{-NM/2} |\Sigma|^{-N/2} \exp \left\{ -\frac{1}{2} \text{tr} \left[\mathbf{S} + (\mathbf{B} - \widehat{\mathbf{B}})^\top \mathbf{X}^\top \mathbf{X} (\mathbf{B} - \widehat{\mathbf{B}}) \right] \Sigma^{-1} \right\} \right. \\
 &\quad \times (2\pi)^{-KM/2} |\mathbf{V}_0|^{-M/2} |\Sigma|^{-K/2} \exp \left\{ -\frac{1}{2} \text{tr} \left[(\mathbf{B} - \mathbf{B}_0)^\top \mathbf{V}_0^{-1} (\mathbf{B} - \mathbf{B}_0) \Sigma^{-1} \right] \right\} \\
 &\quad \times \frac{|\Psi_0|^{\alpha_0/2}}{2^{\alpha_0 M/2} \Gamma_M(\alpha_0/2)} |\Sigma|^{-(\alpha_0+M+1)/2} \exp \left\{ -\frac{1}{2} \text{tr} [\Psi_0 \Sigma^{-1}] \right\} d\Sigma d\mathbf{B} \\
 &= (2\pi)^{-M(N+K)/2} |\mathbf{V}_0|^{-M/2} \frac{|\Psi_0|^{\alpha_0/2}}{2^{\alpha_0 M/2} \Gamma_M(\alpha_0/2)} \\
 &\quad \times \int_{\mathcal{B}} \int_{\mathcal{S}} \left\{ |\Sigma|^{-(\alpha_0+N+K+M+1)/2} \right. \\
 &\quad \left. \exp \left\{ -\frac{1}{2} \text{tr} \left[\mathbf{S} + (\mathbf{B} - \widehat{\mathbf{B}})^\top \mathbf{X}^\top \mathbf{X} (\mathbf{B} - \widehat{\mathbf{B}}) + (\mathbf{B} - \mathbf{B}_0)^\top \mathbf{V}_0^{-1} (\mathbf{B} - \mathbf{B}_0) + \Psi_0 \right] \Sigma^{-1} \right\} \right\} d\Sigma d\mathbf{B} \\
 &= (2\pi)^{-M(N+K)/2} |\mathbf{V}_0|^{-M/2} \frac{|\Psi_0|^{\alpha_0/2}}{2^{\alpha_0 M/2} \Gamma_M(\alpha_0/2)} 2^{M(\alpha_n+K)/2} \Gamma_M((\alpha_n+K)/2) \\
 &\quad \times \int_{\mathcal{B}} \left| \mathbf{S} + (\mathbf{B} - \widehat{\mathbf{B}})^\top \mathbf{X}^\top \mathbf{X} (\mathbf{B} - \widehat{\mathbf{B}}) + (\mathbf{B} - \mathbf{B}_0)^\top \mathbf{V}_0^{-1} (\mathbf{B} - \mathbf{B}_0) + \Psi_0 \right|^{-(\alpha_n+K)/2} d\mathbf{B} \\
 &= (2\pi)^{-M(N+K)/2} |\mathbf{V}_0|^{-M/2} \frac{|\Psi_0|^{\alpha_0/2}}{2^{\alpha_0 M/2} \Gamma_M(\alpha_0/2)} 2^{M(\alpha_n+K)/2} \Gamma_M((\alpha_n+K)/2) \\
 &\quad \times \int_{\mathcal{B}} \left| (\mathbf{B} - \widehat{\mathbf{B}}_n)^\top \mathbf{V}_n^{-1} (\mathbf{B} - \widehat{\mathbf{B}}_n) + \Psi_n \right|^{-(\alpha_n+K)/2} d\mathbf{B} \\
 &= (2\pi)^{-M(N+K)/2} |\mathbf{V}_0|^{-M/2} \frac{|\Psi_0|^{\alpha_0/2}}{2^{\alpha_0 M/2} \Gamma_M(\alpha_0/2)} 2^{M(\alpha_n+K)/2} \Gamma_M((\alpha_n+K)/2) \\
 &\quad \times \int_{\mathcal{B}} \left[|\Psi_n| \times |\mathbf{I}_K + \mathbf{V}_n^{-1} (\mathbf{B} - \widehat{\mathbf{B}}_n) \Psi_n^{-1} (\mathbf{B} - \widehat{\mathbf{B}}_n)^\top| \right]^{-(\alpha_n+K)/2} d\mathbf{B} \\
 &= |\Psi_n|^{-(\alpha_n+K)/2} (2\pi)^{-M(N+K)/2} |\mathbf{V}_0|^{-M/2} \frac{|\Psi_0|^{\alpha_0/2} 2^{M(\alpha_n+K)/2} \Gamma_M((\alpha_n+K)/2)}{2^{\alpha_0 M/2} \Gamma_M(\alpha_0/2)} \\
 &\quad \times \int_{\mathcal{B}} \left| \mathbf{I}_K + \mathbf{V}_n^{-1} (\mathbf{B} - \widehat{\mathbf{B}}_n) \Psi_n^{-1} (\mathbf{B} - \widehat{\mathbf{B}}_n)^\top \right|^{-(\alpha_n+1-M+K+M-1)/2} d\mathbf{B} \\
 &= |\Psi_n|^{-(\alpha_n+K)/2} (2\pi)^{-M(N+K)/2} |\mathbf{V}_0|^{-M/2} \frac{|\Psi_0|^{\alpha_0/2} 2^{M(\alpha_n+K)/2} \Gamma_M((\alpha_n+K)/2)}{2^{\alpha_0 M/2} \Gamma_M(\alpha_0/2)} \\
 &\quad \times \pi^{MK/2} \frac{\Gamma_M((\alpha_n+1-M+K+M-1)/2)}{\Gamma_M((\alpha_n+1-M+K+M-1)/2)} |\Psi_n|^{K/2} |\mathbf{V}_n|^{M/2} \\
 &= \frac{|\mathbf{V}_n|^{M/2} |\Psi_0|^{\alpha_0/2}}{|\mathbf{V}_0|^{M/2} |\Psi_n|^{\alpha_n/2}} \frac{\Gamma_M(\alpha_n/2)}{\Gamma_M(\alpha_0/2)} \pi^{-MN/2}.
 \end{aligned}$$

The third equality follows from having the kernel of a inverse Wishart distribution, the fifth from the Sylvester's theorem, and the seventh from having the kernel of a matrix t distribution.

Observe that this last expression is the multivariate case of the marginal likelihood of the univariate regression model. Taking into account that

$$\begin{aligned} (\mathbf{A} + \mathbf{B})^{-1} &= \mathbf{A}^{-1} - (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1} \mathbf{A}^{-1} \\ &= \mathbf{B}^{-1} - (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1} \mathbf{B}^{-1} \\ &= \mathbf{A}^{-1}(\mathbf{A}^{-1} + \mathbf{B}^{-1})\mathbf{B}^{-1}, \end{aligned}$$

we can show that $\Psi_n = \Psi_0 + \mathbf{S} + (\hat{\mathbf{B}} - \mathbf{B}_0)^\top \mathbf{V}_n (\hat{\mathbf{B}} - \mathbf{B}_0)$ (see Exercise 7). Therefore, the marginal likelihood rewards fit (smaller sum of squares, \mathbf{S}), similarity between prior and sample information regarding location parameters, and information gains in variability from \mathbf{V}_0 to \mathbf{V}_n .

Given a matrix of regressors \mathbf{X}_0 for N_0 unobserved units, the predictive density of \mathbf{Y}_0 given \mathbf{Y} , $\pi(\mathbf{Y}_0|\mathbf{Y})$ is a matrix t distribution $T_{N_0, M}(\alpha_n - M + 1, \mathbf{X}_0 \mathbf{B}_n, \mathbf{I}_{N_0} + \mathbf{X}_0 \mathbf{V}_n \mathbf{X}_0^\top, \Psi_n)$ (see Exercise 6). Observe that the prediction is centered at $\mathbf{X}_0 \mathbf{B}_n$, and the covariance matrix of $\text{vec}(\mathbf{Y}_0)$ is $\frac{(\mathbf{I}_{N_0} + \mathbf{X}_0 \mathbf{V}_n \mathbf{X}_0^\top) \otimes \Psi_n}{\alpha_n - M - 1}$.

3.5 Summary

We introduce the conjugate family models for discrete and continuous data. These models are the basic Bayesian framework due to its mathematical tractability as we get closed-form expressions for the posterior distributions, the marginal likelihood, and the predictive distribution. We also present the Bayesian linear univariate and multivariate regression frameworks under conjugate families. This is the cornerstone to perform regression analysis in the Bayesian setting.

3.6 Exercises

1. Write in the canonical form the distribution of the Bernoulli example, and find the mean and variance of the sufficient statistic.
2. Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from N binomial experiments each having known size n_i and same unknown probability θ . Show that $p(\mathbf{y}|\theta)$ is in the exponential family, and find the posterior distribution, the marginal likelihood and the predictive distribution of the binomial-beta model assuming the number of trials is known.
3. Given a random sample $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^\top$ from a exponential distribution. Show that $p(\mathbf{y}|\lambda)$ is in the exponential family, and find

the posterior distribution, marginal likelihood and predictive distribution of the exponential-gamma model.

4. Given $\mathbf{y} \sim N_N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, that is, a *multivariate normal distribution* show that $p(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is in the exponential family.
5. Find the marginal likelihood in the normal/inverse-Wishart model.
6. Find the posterior predictive distribution in the normal/inverse-Wishart model, and show that $\mathbf{Y}_0|\mathbf{Y} \sim T_{N_0, M}(\alpha_n - M + 1, \mathbf{X}_0 \mathbf{B}_n, \mathbf{I}_{N_0} + \mathbf{X}_0 \mathbf{V}_n \mathbf{X}_0^\top, \boldsymbol{\Psi}_n)$.
7. Show that $\delta_n = \delta_0 + (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) + (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)^\top((\mathbf{X}^\top \mathbf{X})^{-1} + \mathbf{B}_0)^{-1}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)$ in the linear regression model, and that $\boldsymbol{\Psi}_n = \boldsymbol{\Psi}_0 + \mathbf{S} + (\hat{\mathbf{B}} - \mathbf{B}_0)^\top \mathbf{V}_n (\hat{\mathbf{B}} - \mathbf{B}_0)$ in the linear multivariate regression model.
8. Show that in the linear regression model $\boldsymbol{\beta}_n^\top(\mathbf{B}_n^{-1} - \mathbf{B}_n^{-1} \mathbf{M}^{-1} \mathbf{B}_n^{-1})\boldsymbol{\beta}_n = \boldsymbol{\beta}_{**}^\top \mathbf{C} \boldsymbol{\beta}_{**}$ and $\boldsymbol{\beta}_{**} = \mathbf{X}_0 \boldsymbol{\beta}_n$.
9. Show that $(\mathbf{Y} - \mathbf{X}\mathbf{B})^\top(\mathbf{Y} - \mathbf{X}\mathbf{B}) = \mathbf{S} + (\mathbf{B} - \hat{\mathbf{B}})^\top \mathbf{X}^\top \mathbf{X} (\mathbf{B} - \hat{\mathbf{B}})$ where $\mathbf{S} = (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})^\top(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})$, $\hat{\mathbf{B}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$ in the multivariate regression model.
10. **What is the probability that the Sun will rise tomorrow?**
This is the most famous Richard Price's example developed in the Appendix of the Bayes' theorem paper [12]. Here, we implicitly use *Laplace's Rule of Succession* to solve this question. In particular, if we were a priori uncertain about the probability the Sun will rise on a specified day, we can assume a prior uniform distribution over $(0,1)$, that is, a beta $(1,1)$ distribution. Then, what is the probability that the Sun will rise?
11. Using information from Public Policy Polling in September 27th-28th for the 2016 presidential five-way race in USA, there are 411, 373 and 149 sampled people supporting Hillary Clinton, Donald Trump and other, respectively.
 - Find the posterior probability of the percentage difference of people supporting Hillary versus Trump according to this data using a non-informative prior, that is, $\alpha_0 = [1 \ 1 \ 1]$ in the multinomial-Dirichlet model. What is the probability of having more supports of Hillary vs Trump?
 - What is the probability that sampling one hundred independent individuals 44, 40 and 16 support Hillary, Trump and other, respectively?
12. **Math test example continues**
You have a random sample of math scores of size $N = 50$ from a normal distribution, $Y_i \sim N(\mu, \sigma^2)$. The sample mean and variance are

equal to 102 and 10, respectively. Using the normal-normal/inverse-gamma model where $\mu_0 = 100$, $\beta_0 = 1$, $\alpha_0 = \delta_0 = 0.001$

- Get a 95% confidence and credible interval for μ .
- What is the posterior probability that $\mu > 103$?

13. Demand of electricity example continues

Set c_0 such that maximizes the marginal likelihood in the specifications with and without electricity price in the example of demand of electricity (empirical Bayes). Then, calculate the Bayes factor, and conclude if there is evidence supporting the inclusion of the price of electricity in the demand equation.

14. Utility demand

Use the file *Utilities.csv* to estimate a multivariate linear regression model where $\mathbf{Y}_i = [\log(\text{electricity}_i) \log(\text{water}_i) \log(\text{gas}_i)]$ as function of $\log(\text{electricity price}_i)$, $\log(\text{water price}_i)$, $\log(\text{gas price}_i)$, IndSocio1_i , IndSocio2_i , Altitude_i , Nrooms_i , HouseholdMem_i , Children_i , and $\log(\text{Income}_i)$, where electricity, water and gas are monthly consumption of electricity (kWh), water (m^3) and gas (m^3), and other definitions are given in the Example of Section 3.3. Omit households that do not consume any of the utilities in this exercise.

Set a non-informative prior framework, $\mathbf{B}_0 = [0]_{11 \times 3}$, $\mathbf{V}_0 = 1000\mathbf{I}_{11}$, $\Psi_0 = 1000\mathbf{I}_3$ and $\alpha_0 = 3$, where we have $K = 11$ (regressors plus intercept) and $M = 3$ (equations) in this exercise.

- Find the posterior mean estimates and the highest posterior density intervals at 95% of \mathbf{B} and Σ . Use the marginal distribution and the conditional distribution to obtain the posterior estimates of \mathbf{B} , and compare the results.
- Find the Bayes factor comparing the baseline model in this exercise with the same specification but using the income in dollars. Now, calculate the Bayes factor using the income in thousand dollars. Is there any difference?
- Find the predictive distribution for the monthly demand of electricity, water and gas in the baseline specification of a household located in the lowest socioeconomic condition in a municipality located below 1000 meters above the sea level, 2 rooms, 3 members with children, a monthly income equal to USD 500, an electricity price equal to USD/kWh 0.15, a water price equal to USD/ M^3 0.70, and a gas price equal to USD/ M^3 0.75.

15. Ph.D. students sleeping hours [3, Chap. 2]

We are interested in learning about the proportion of Ph.D. students who sleep at least 6 hours per day. We have a sample of 52 students, where 15 report sleeping at least 6 hours, and the remaining 37 report not sleeping at least 6 hours. The prior distribution is a Beta distribution, with hyperparameters calibrated so that the prior probabilities of the proportion of students who sleep least than 6 hours being less than 0.4 and 0.75 are 0.6 and 0.95, respectively. Estimate the 95% posterior credible interval for the proportion of Ph.D. students. Then, assume there is a group of experts whose beliefs about the proportion of Ph.D. students sleeping at least 6 hours are represented in the following table:

TABLE 3.1

Probability distribution: Ph.D students that sleep at least 6 hours per day.

h	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55
$P(p = h)$	0.05	0.07	0.10	0.12	0.15	0.17	0.15	0.11	0.06	0.01	0.01

Use Table 3.1 as prior information, and find the posterior distribution of the proportion of students that sleep at least 6 hours.



4

Simulation methods

In the previous chapters, we focused on conjugate families, where the posterior and predictive distributions have standard analytical forms (e.g., normal, Student's t, gamma, binomial, Poisson, etc.) and where the marginal likelihood has a closed-form analytical solution. However, realistic models are often more complex and lack such closed-form solutions.

To address this complexity, we rely on simulation (stochastic) methods to draw samples from posterior and predictive distributions. This chapter introduces posterior simulation, a cornerstone of Bayesian inference. We discuss Markov Chain Monte Carlo (MCMC) methods, including Gibbs sampling, Metropolis-Hastings, and Hamiltonian Monte Carlo, as well as other techniques like importance sampling and particle filtering (sequential Monte Carlo).

The simulation methods discussed in this chapter are specifically applied throughout this book. However, we do not delve into deterministic methods, such as numerical integration (quadrature), or other simulation methods, including discrete approximation, the probability integral transform, the method of composition, accept-reject sampling, and slice sampling algorithms. While these methods are also widely used, they are not as common as the approaches explicitly employed in this book.

For readers interested in these alternative methods, we recommend exploring [186, Chaps. 2 and 3], [185, Chaps. 2, 3, and 8], [92, Chap. 5], and [76, Chap. 10].

4.1 Markov chain Monte Carlo methods

Markov Chain Monte Carlo (MCMC) methods are algorithms used to approximate complex probability distributions by constructing a Markov chain. This chain is a sequence of random samples where each sample depends only on the previous one. The goal of MCMC methods is to obtain draws from the posterior distribution as the equilibrium distribution. The key point in MCMC methods is the transition kernel or density, $q(\boldsymbol{\theta}^{(s)}|\boldsymbol{\theta}^{(s-1)})$, which generates a draw $\boldsymbol{\theta}^{(s)}$ at stage s that depends solely on $\boldsymbol{\theta}^{(s-1)}$. This transition distribution must be designed such that the Markov chain converges to a unique

stationary distribution, which, in our case, is the posterior distribution, that is, $\pi(\boldsymbol{\theta}^{(s)}|\mathbf{y}) = \int_{\Theta} q(\boldsymbol{\theta}^{(s)}|\boldsymbol{\theta}^{(s-1)})\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})d\boldsymbol{\theta}^{(s-1)}$.

Given that we start at an arbitrary point, $\boldsymbol{\theta}^{(0)}$, the algorithm requires that the Markov chain be *irreducible*, meaning that the process can reach any other state with positive probability. Additionally, the process must be *aperiodic*, meaning that for each state, the greatest common divisor of the number of steps it takes to return to the state is 1, ensuring that there are no cycles forcing the system to return to a state only after a fixed number of steps. Furthermore, the process must be *recurrent*, meaning that it will return to any state an infinite number of times with probability one. However, to ensure convergence to the stationary distribution, a stronger condition is required: the process must be *positive recurrent*, meaning that the expected return time to a state is finite. Given an *irreducible*, *aperiodic*, and *positive recurrent* transition density, the Markov chain algorithm will asymptotically converge to the stationary posterior distribution we are seeking. For more details, see [185, chap. 6].

4.1.1 Gibbs sampler

This Gibbs sampler algorithm is one of the most widely used MCMC methods for sampling from non-standard distributions in Bayesian analysis. While it is a special case of the Metropolis-Hastings (MH) algorithm, it originated from a different theoretical background [79, 71]. The key requirement for implementing the Gibbs sampling algorithm is the availability of conditional posterior distributions. The algorithm works by cycling through the conditional posterior distributions corresponding to different blocks of the parameter space under inference.

Two simplify concepts let's focus on a parameter space composed by two blocks, $\boldsymbol{\theta} = [\boldsymbol{\theta}_1 \ \boldsymbol{\theta}_2]^\top$, the Gibbs sampling algorithm uses as transition kernel $q(\boldsymbol{\theta}_1^{(s)}, \boldsymbol{\theta}_2^{(s)}|\boldsymbol{\theta}_1^{(s-1)}, \boldsymbol{\theta}_2^{(s-1)}) = \pi(\boldsymbol{\theta}_1^{(s)}|\boldsymbol{\theta}_2^{(s-1)}, \mathbf{y})\pi(\boldsymbol{\theta}_2^{(s)}|\boldsymbol{\theta}_1^{(s)}, \mathbf{y})$. Thus,

$$\begin{aligned} \int_{\Theta} q(\boldsymbol{\theta}^{(s)}|\boldsymbol{\theta}^{(s-1)})\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})d\boldsymbol{\theta}^{(s-1)} &= \int_{\Theta_2} \int_{\Theta_1} \pi(\boldsymbol{\theta}_1^{(s)}|\boldsymbol{\theta}_2^{(s-1)}, \mathbf{y})\pi(\boldsymbol{\theta}_2^{(s)}|\boldsymbol{\theta}_1^{(s)}, \mathbf{y})\pi(\boldsymbol{\theta}_1^{(s-1)}, \boldsymbol{\theta}_2^{(s-1)}|\mathbf{y})d\boldsymbol{\theta}_1^{(s-1)}d\boldsymbol{\theta}_2^{(s-1)} \\ &= \pi(\boldsymbol{\theta}_2^{(s)}|\boldsymbol{\theta}_1^{(s)}, \mathbf{y}) \int_{\Theta_2} \int_{\Theta_1} \pi(\boldsymbol{\theta}_1^{(s)}|\boldsymbol{\theta}_2^{(s-1)}, \mathbf{y})\pi(\boldsymbol{\theta}_1^{(s-1)}, \boldsymbol{\theta}_2^{(s-1)}|\mathbf{y})d\boldsymbol{\theta}_1^{(s-1)}d\boldsymbol{\theta}_2^{(s-1)} \\ &= \pi(\boldsymbol{\theta}_2^{(s)}|\boldsymbol{\theta}_1^{(s)}, \mathbf{y}) \int_{\Theta_2} \pi(\boldsymbol{\theta}_1^{(s)}|\boldsymbol{\theta}_2^{(s-1)}, \mathbf{y})\pi(\boldsymbol{\theta}_2^{(s-1)}|\mathbf{y})d\boldsymbol{\theta}_2^{(s-1)} \\ &= \pi(\boldsymbol{\theta}_2^{(s)}|\boldsymbol{\theta}_1^{(s)}, \mathbf{y}) \int_{\Theta_2} \pi(\boldsymbol{\theta}_1^{(s)}, \boldsymbol{\theta}_2^{(s-1)}|\mathbf{y})d\boldsymbol{\theta}_2^{(s-1)} \\ &= \pi(\boldsymbol{\theta}_2^{(s)}|\boldsymbol{\theta}_1^{(s)}, \mathbf{y})\pi(\boldsymbol{\theta}_1^{(s)}|\mathbf{y}) \\ &= \pi(\boldsymbol{\theta}_1^{(s)}, \boldsymbol{\theta}_2^{(s)}|\mathbf{y}). \end{aligned}$$

Then, $\pi(\boldsymbol{\theta}|\mathbf{y})$ is the stationary distribution for the Gibbs transition kernel.

A word of caution! Even if we have well-defined conditional posterior distributions $\pi(\boldsymbol{\theta}_1^{(s)}|\boldsymbol{\theta}_2^{(s-1)}, \mathbf{y})$ and $\pi(\boldsymbol{\theta}_2^{(s)}|\boldsymbol{\theta}_1^{(s)}, \mathbf{y})$, and we can simulate from

them, the joint posterior distribution $\pi(\boldsymbol{\theta}_1^{(s)}, \boldsymbol{\theta}_2^{(s)} | \mathbf{y})$ may not correspond to any proper distribution. We should be mindful of this situation, especially when dealing with improper prior distributions (see [185, Chap. 10] for details).

Algorithm A1 demonstrates the implementation of a Gibbs sampler with d blocks. The number of iterations (S) is chosen to ensure convergence to the stationary distribution. In Section 4.4, we review several convergence diagnostics to assess whether the posterior draws have reached convergence.

Algorithm A1 Gibbs sampling

```

1: Set  $\boldsymbol{\theta}_2^{(0)}, \boldsymbol{\theta}_3^{(0)}, \dots, \boldsymbol{\theta}_d^{(0)}$ 
2: for  $s = 1, \dots, S$  do
3:   Draw  $\boldsymbol{\theta}_1^{(s)}$  from  $\pi(\boldsymbol{\theta}_1^{(s)} | \boldsymbol{\theta}_2^{(s-1)}, \dots, \boldsymbol{\theta}_d^{(s-1)}, \mathbf{y})$ 
4:   Draw  $\boldsymbol{\theta}_2^{(s)}$  from  $\pi(\boldsymbol{\theta}_2^{(s)} | \boldsymbol{\theta}_1^{(s)}, \dots, \boldsymbol{\theta}_d^{(s-1)}, \mathbf{y})$ 
5:   :
6:   Draw  $\boldsymbol{\theta}_d^{(s)}$  from  $\pi(\boldsymbol{\theta}_d^{(s)} | \boldsymbol{\theta}_1^{(s)}, \dots, \boldsymbol{\theta}_{d-1}^{(s)}, \mathbf{y})$ 
7: end for
```

Example: Mining disaster change point

Let's use the dataset *Mining.csv* provided by [29]. This dataset records the number of mining disasters per year from 1851 to 1962 in British coal mines.

We assume there is an unknown structural change point in the number of mining disasters, where the parameters of the Poisson distributions change. In particular,

$$p(y_t) = \begin{cases} \frac{\exp(-\lambda_1)\lambda_1^{y_t}}{y_t!}, & t = 1, 2, \dots, H \\ \frac{\exp(-\lambda_2)\lambda_2^{y_t}}{y_t!}, & t = H + 1, \dots, T \end{cases},$$

where H is the changing point.

We use conjugate families for λ_l , $l = 1, 2$, where $\lambda_l \sim G(\alpha_{l0}, \beta_{l0})$, and set $\pi(H) = 1/T$, which corresponds to a discrete uniform distribution for the change point. This implies that, a priori, we assume equal probability for any time to be the change point.

The posterior distribution is

$$\begin{aligned} \pi(\lambda_1, \lambda_2, H | \mathbf{y}) &\propto \prod_{t=1}^H \frac{\exp(-\lambda_1)\lambda_1^{y_t}}{y_t!} \prod_{t=H+1}^T \frac{\exp(-\lambda_2)\lambda_2^{y_t}}{y_t!} \\ &\times \exp(-\beta_{10}\lambda_1)\lambda_1^{\alpha_{10}-1} \exp(-\beta_{20}\lambda_2)\lambda_2^{\alpha_{20}-1} 1/T \\ &\propto \exp(-H\lambda_1)\lambda_1^{\sum_{t=1}^H y_t} \exp(-(T-H)\lambda_2)\lambda_2^{\sum_{t=H+1}^T y_t} \\ &\times \exp(-\beta_{10}\lambda_1)\lambda_1^{\alpha_{10}-1} \exp(-\beta_{20}\lambda_2)\lambda_2^{\alpha_{20}-1}. \end{aligned}$$

Then, the conditional posterior distribution of $\lambda_1 | \lambda_2, H, \mathbf{y}$ is

$$\pi(\lambda_1 | \lambda_2, H, \mathbf{y}) \propto \exp(-(H + \beta_{10})\lambda_1)\lambda_1^{\sum_{t=1}^H y_t + \alpha_{10}-1},$$

that is, $\lambda_1|\lambda_2, H, \mathbf{y} \sim G(\alpha_{1n}, \beta_{1n})$, $\beta_{1n} = H + \beta_{10}$ and $\alpha_{1n} = \sum_{t=1}^H y_t + \alpha_{10}$.

The conditional posterior distribution of $\lambda_2|\lambda_1, H, \mathbf{y}$ is

$$\pi(\lambda_2|\lambda_1, H, \mathbf{y}) \propto \exp(-((T-H) + \beta_{20})\lambda_2)\lambda_2^{\sum_{t=H+1}^T y_t + \alpha_{20} - 1},$$

that is, $\lambda_2|\lambda_1, H, \mathbf{y} \sim G(\alpha_{2n}, \beta_{2n})$, $\beta_{2n} = (T-H) + \beta_{20}$ and $\alpha_{2n} = \sum_{t=H+1}^T y_t + \alpha_{20}$.

The conditional posterior distribution of the change point is

$$\begin{aligned} \pi(H|\lambda_1, \lambda_2, \mathbf{y}) &\propto \exp(-H\lambda_1)\lambda_1^{\sum_{t=1}^H y_t} \exp(-(T-H)\lambda_2)\lambda_2^{\sum_{t=H+1}^T y_t} \\ &\propto \exp(-H(\lambda_1 - \lambda_2))\lambda_1^{\sum_{t=1}^H y_t} \lambda_2^{\sum_{t=H+1}^T y_t} \exp(-T\lambda_2) \frac{\lambda_2^{\sum_{t=1}^H y_t}}{\lambda_2^{\sum_{t=1}^H y_t}} \\ &\propto \exp(-H(\lambda_1 - \lambda_2)) \left(\frac{\lambda_1}{\lambda_2}\right)^{\sum_{t=1}^H y_t}. \end{aligned}$$

Thus, the conditional posterior distribution of H is

$$\pi(H|\lambda_1, \lambda_2, \mathbf{y}) = \frac{\exp(-H(\lambda_1 - \lambda_2)) \left(\frac{\lambda_1}{\lambda_2}\right)^{\sum_{t=1}^H y_t}}{\sum_{H=1}^T \exp(-H(\lambda_1 - \lambda_2)) \left(\frac{\lambda_1}{\lambda_2}\right)^{\sum_{t=1}^H y_t}}, \quad H = 1, 2, \dots, T.$$

The following code shows how to do a Gibbs sampling algorithm to perform inference of this model using the hyperparameters suggested by [92, Chap. 7], $\alpha_{l0} = 0.5$ and $\beta_{l0} = 1$, $l = 1, 2$.

R code. Gibbs sampler: The mining disaster changepoint

```

1 rm(list = ls()); set.seed(010101)
2 dataset<-read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/
  MiningDataCarlin.csv",header=T)
3 attach(dataset); str(dataset)
4 a10<-0.5; a20<-0.5
5 b10<-1; b20<-1
6 y<-Count
7 sumy<-sum(Count); N<-length(Count)
8 theta1<-NULL; theta2<-NULL
9 kk<-NULL; k<-60; S<-10000
10 for(i in 1:S){
11   a1<-a10+sum(y[1:k]); b1<-b10+k
12   theta11<-rgamma(1,a1,b1)
13   theta1<-c(theta1,theta11)
14   a2<-a20+sum(y[(1+k):N]); b2<-b20+N-k
15   theta22<-rgamma(1,a2,b2)
16   theta2<-c(theta2,theta22)
17   pp<-NULL
18   for(l in 1:N){
19     p<-exp(1*(theta22-theta11))*(theta11/theta22)^sum(y[1:l]
20     ))
21   pp<-c(pp,p)
22   }
23   prob<-pp/sum(pp); k<-sample(1:N,1,prob=prob)
24   kk<-c(kk,k)
25 }
26 library(coda); summary(mcmc(theta1)); summary(mcmc(theta2))
26 summary(mcmc(kk)); hist(HPost, main = "Histogram: Posterior
  mean change point", xlab = "Posterior mean", col = "blue
  ", breaks = 25)

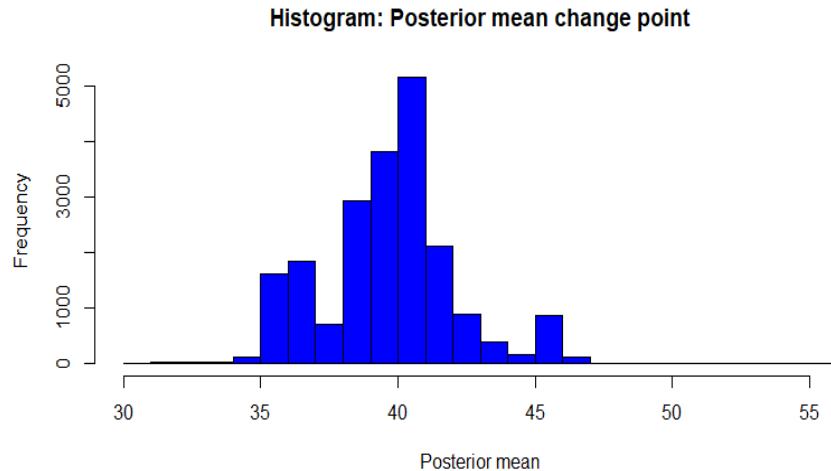
```

The posterior results indicate that the rate of disasters decrease from 3.1 to 0.92 per year in 1890.

Figure 4.1 shows the histogram of the posterior draws of the change point in mining disasters.

4.1.2 Metropolis-Hastings

The Metropolis-Hastings (M-H) algorithm [153, 96] is a general MCMC method that does not require standard closed-form solutions for the conditional posterior distributions. The key idea is to use a transition kernel whose unique invariant distribution is $\pi(\boldsymbol{\theta}|\mathbf{y})$. This kernel must satisfy the *balance*-

**FIGURE 4.1**

Histogram of posterior draws of change point: Mining disasters

ing condition, meaning that, given a realization $\boldsymbol{\theta}^{(s-1)}$ at stage $s - 1$ from the stationary distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$, we generate a candidate draw $\boldsymbol{\theta}^c$ from the proposal distribution $q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)})$ at stage s such that:

$$q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)})\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y}) = q(\boldsymbol{\theta}^{(s-1)}|\boldsymbol{\theta}^c)\pi(\boldsymbol{\theta}^c|\mathbf{y}),$$

which implies that the probability of moving from $\boldsymbol{\theta}^{(s-1)}$ to $\boldsymbol{\theta}^c$ is equal to the probability of moving from $\boldsymbol{\theta}^c$ to $\boldsymbol{\theta}^{(s-1)}$.

In general, the *balancing condition* is not automatically satisfied, and we must introduce an acceptance probability $\alpha(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\theta}^c)$ to ensure that the condition holds:

$$q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)})\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})\alpha(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\theta}^c) = q(\boldsymbol{\theta}^{(s-1)}|\boldsymbol{\theta}^c)\pi(\boldsymbol{\theta}^c|\mathbf{y}).$$

Thus, the acceptance probability is given by:

$$\alpha(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\theta}^c) = \min \left\{ \frac{q(\boldsymbol{\theta}^{(s-1)}|\boldsymbol{\theta}^c)\pi(\boldsymbol{\theta}^c|\mathbf{y})}{q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)})\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})}, 1 \right\},$$

where $q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)})$ and $\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})$ must be nonzero, as transitioning from $\boldsymbol{\theta}^{(s-1)}$ to $\boldsymbol{\theta}^c$ is only possible under these conditions.

Algorithm A2 shows how to implement a Metropolis-Hastings algorithm. The number of iterations (S) is chosen to ensure convergence to the stationary distribution.

Some remarks: First, we do not need to know the marginal likelihood to

Algorithm A2 Metropolis-Hastings algorithm

```

1: Set  $\boldsymbol{\theta}^{(0)}$  in the support of  $\pi(\boldsymbol{\theta}|\mathbf{y})$ 
2: for  $s = 1, \dots, S$  do
3:   Draw  $\boldsymbol{\theta}^c$  from  $q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)})$ 
4:   Calculate  $\alpha(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\theta}^c) = \min\left\{\frac{q(\boldsymbol{\theta}^{(s-1)}|\boldsymbol{\theta}^c)\pi(\boldsymbol{\theta}^c|\mathbf{y})}{q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)})\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})}, 1\right\}$ 
5:   Draw  $U$  from  $U(0, 1)$ 
6:    $\boldsymbol{\theta}^{(s)} = \begin{cases} \boldsymbol{\theta}^c & \text{if } U \leq \alpha(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\theta}^c) \\ \boldsymbol{\theta}^{(s-1)} & \text{otherwise} \end{cases}$ 
7: end for

```

implement the M-H algorithm, as it cancels out when calculating the acceptance probability. Specifically, given that $\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \pi(\boldsymbol{\theta}) \times p(\mathbf{y}|\boldsymbol{\theta})$, we can use the right-hand side expression to compute the acceptance probability. Second, the Gibbs sampling algorithm is a particular case of the M-H algorithm where the acceptance probability is equal to 1 ([75] and [185, Chap. 10], see Exercise 2). Third, we can combine the M-H and Gibbs sampling algorithms when dealing with relatively complex posterior distributions. Specifically, the Gibbs sampling algorithm can be used for blocks with conditional posterior distributions in standard closed forms, while the M-H algorithm is applied to sample from conditional posterior distributions that do not have standard forms. This approach is known as the M-H within Gibbs sampling algorithm. Fourth, we can note that the transition kernel in the M-H algorithm is a mixture of a continuous density ($q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)})$) and a probability mass function ($\alpha(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\theta}^c)$) [42].

Fifth, a crucial point associated with the proposal densities is the acceptance probability. Low or high acceptance probabilities are not ideal. A low rate implies poor mixing, meaning the chain does not move effectively through the support of the posterior distribution. Conversely, a high acceptance rate implies that the chain will converge too slowly. A sensible value depends on the dimension of the parameter space. A rule of thumb is that if the dimension is less than or equal to 2, the acceptance rate should be around 0.50. If the dimension is greater than 2, the acceptance rate should be approximately 0.25 [187]. For technical details of the Metropolis-Hastings algorithm, see [185, Chap. 7].

Regarding the proposal density, it must be positive everywhere the posterior distribution is positive. This ensures that the Markov chain can explore the entire support of the posterior distribution. Additionally, the proposal density must allow the Markov chain to reach any region of the posterior distribution's support. There are three standard approaches for choosing the proposal density: the independent proposal, the random walk proposal, and the tailored proposal.

In the independent proposal, $q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)}) = q(\boldsymbol{\theta}^c)$, which implies that

$$\alpha(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\theta}^c) = \min \left\{ \frac{q(\boldsymbol{\theta}^{(s-1)})\pi(\boldsymbol{\theta}^c|\mathbf{y})}{q(\boldsymbol{\theta}^c)\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})}, 1 \right\}.$$

In this case, a move from $\boldsymbol{\theta}^{(s-1)}$ to $\boldsymbol{\theta}^c$ is always accepted if $q(\boldsymbol{\theta}^{(s-1)})\pi(\boldsymbol{\theta}^c|\mathbf{y}) \geq q(\boldsymbol{\theta}^c)\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})$.

In the random walk proposal, $\boldsymbol{\theta}^c = \boldsymbol{\theta}^{(s-1)} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a random perturbation. If $p(\boldsymbol{\epsilon}) = p(-\boldsymbol{\epsilon})$, meaning the distribution of $p(\boldsymbol{\epsilon})$ is symmetric around zero, then $q(\boldsymbol{\theta}^c|\boldsymbol{\theta}^{(s-1)}) = q(\boldsymbol{\theta}^{(s-1)}|\boldsymbol{\theta}^c)$. This was the original Metropolis algorithm [153]. Thus, the acceptance rate is

$$\alpha(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\theta}^c) = \min \left\{ \frac{\pi(\boldsymbol{\theta}^c|\mathbf{y})}{\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})}, 1 \right\}.$$

In this case, a move from $\boldsymbol{\theta}^{(s-1)}$ to $\boldsymbol{\theta}^c$ is always accepted if $\pi(\boldsymbol{\theta}^c|\mathbf{y}) \geq \pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})$.

In the tailored proposal, the density is designed to have fat tails, is centered at the mode of the posterior distribution, and its scale matrix is given by the negative inverse Hessian matrix evaluated at the mode. Specifically, for two blocks, the log posterior distribution is maximized with respect to $\boldsymbol{\theta}_1$ given $\boldsymbol{\theta}_2$. This process is repeated at each iteration of the algorithm because $\boldsymbol{\theta}_2$ changes at different stages. As a result, the algorithm can be slow since the optimization process is computationally demanding (see [92, Chap. 7 and 9] for examples).

A sensible recommendation when performing M-H algorithm is to use a random walk proposal such that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, c^2 \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is the negative inverse Hessian matrix evaluated at the mode, that is, maximize with respect to all parameters, and set $c \approx 2.4/\sqrt{\dim\{\boldsymbol{\theta}\}}$, which is the most efficient scale compared to independent sampling [76, Chap. 12]. After some iterations of the algorithm, adjust the scale matrix $\boldsymbol{\Sigma}$ as before, and increase or decrease c if the acceptance rate of the simulations is too high or low, respectively. The objective is to bring the acceptance rate to the stated rule of thumb, that is, if the dimension is less than or equal to 2, the acceptance rate should be around 0.50, and if the dimension is greater than 2, the acceptance rate should be around 0.25. Once this is achieved, we should run the algorithm without modifications and use this part of the algorithm to perform inference.

Example: Ph.D. students sleeping hours continues

In the Ph.D. students sleeping hours exercise of Chapter 3 we get a posterior distribution that is Beta with parameters 16.55 and 39.57. We can sample from this posterior distribution using the function `rbeta` from **R**. However, we want to compare the performance of a M-H algorithm using as proposal density a $U(0, 1)$ distribution.

The following code shows how to do a M-H algorithm to sample from the beta distribution using the uniform distribution.

R code. Metropolis-Hastings algorithm: Ph.D. students sleeping hours

```

1 rm(list = ls()); set.seed(010101)
2 an <- 16.55; bn <- 39.57
3 S <- 100000; p <- runif(S); accept <- rep(0, S)
4 for (s in 2:S){
5   pc <- runif(1) # Candidate
6   a <- dbeta(pc, an, bn)/dbeta(p[s-1], an, bn) # Acceptance
    rate
7   U <- runif(1)
8   if(U <= a){
9     p[s] <- pc
10    accept[s] <- 1
11  }else{
12    p[s] <- p[s-1]
13    accept[s] <- 0
14  }
15 }
16 mean(accept); mean(p); sd(p)
17 an/(an + bn); (an*bn/((an+bn)^2*(an+bn+1)))^0.5 # Population
  values
18 h <- hist(p, breaks=50, col="blue", xlab="Proportion Ph.D.
  students sleeping at least 6 hours", main="Beta draws
  from a Metropolis-Hastings algorithm")
19 pfit <- seq(min(p),max(p),length=50)
20 yfit<-dbeta(pfit, an, bn)
21 yfit <- yfit*diff(h$mids[1:2])*length(p)
22 lines(pfit, yfit, col="red", lwd=2)

```

The results indicate that the mean and standard deviation obtained from the posterior draws are similar to the population values. Furthermore, Figure 4.2 presents the histogram of the posterior draws alongside the density of the beta distribution, demonstrating a good match between them.

4.1.3 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) was proposed by [61] and later introduced to the statistical community by [154]. HMC extends the Metropolis algorithm to efficiently explore the parameter space by introducing *momentum variables*, which help overcome the random walk behavior of Gibbs sampling and the Metropolis-Hastings algorithm. Known also as hybrid Monte Carlo, HMC is particularly advantageous for high-dimensional posterior distributions, as it reduces the risk of getting stuck in local modes and significantly improves mixing [155].

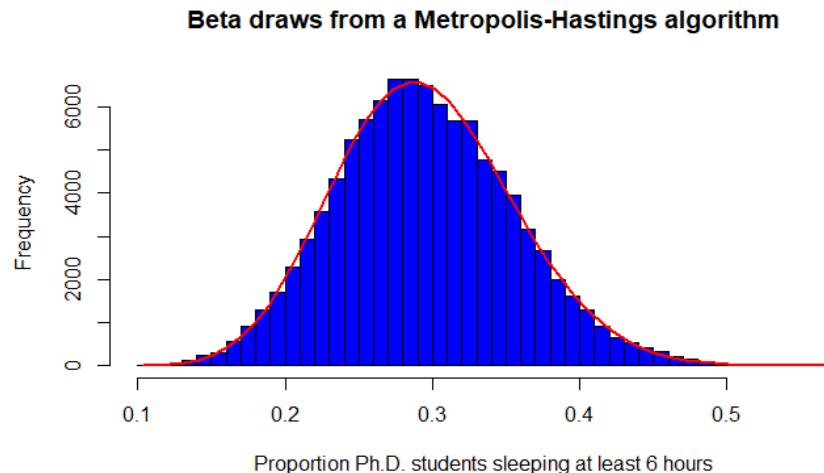


FIGURE 4.2

Histogram of posterior draws of beta distribution and the density of the beta distribution.

However, HMC is designed to work with strictly positive target densities. Therefore, transformations are required to handle bounded parameters, such as variances and proportions. For example, logarithmic and logit transformations can be applied. These transformations necessitate the use of the change-of-variable theorem to compute the log posterior density and its gradient, which are essential for implementing the HMC algorithm.

HMC leverages concepts from physics, specifically Hamiltonian mechanics, to propose transitions in the Markov chain. In Hamiltonian mechanics, two key variables define the total energy of the system: the *position* (θ) and the *momentum* (δ). The Hamiltonian represents the total energy of the system, consisting of *potential energy* (energy due to position) and *kinetic energy* (energy associated with motion). The objective is to identify trajectories that preserve the system's total energy, meaning the Hamiltonian remains invariant, while avoiding trajectories that do not. This approach enhances the acceptance rate of proposed transitions.

To implement HMC, we solve the differential equations derived from the Hamiltonian, which involve derivatives with respect to position and momentum. However, these equations rarely have analytical solutions, requiring numerical methods for approximation. This necessitates discretizing Hamilton's equations, which introduces errors. To mitigate these errors, HMC uses the *leapfrog integrator*, a numerical method with smaller errors compared to simpler approaches like the Euler method.

HMC uses a *momentum variable* (δ_k) for each θ_k , so that the transition

kernel of $\boldsymbol{\theta}$ is determined by $\boldsymbol{\delta}$. Both vectors are updated using a Metropolis algorithm at each stage such that the distribution of $\boldsymbol{\theta}$ remains invariant [155]. The joint density in HMC is given by $p(\boldsymbol{\theta}, \boldsymbol{\delta}|\mathbf{y}) = \pi(\boldsymbol{\theta}|\mathbf{y}) \times p(\boldsymbol{\delta})$, where $\boldsymbol{\delta} \sim N(\mathbf{0}, \mathbf{M})$, and \mathbf{M} is a diagonal matrix such that $\delta_k \sim N(0, M_{kk})$.

Algorithm A3 outlines the HMC implementation. The gradient vector $\frac{d \log(\pi(\boldsymbol{\theta}|\mathbf{y}))}{d\boldsymbol{\theta}}$ must be computed analytically, as using finite differences can be computationally expensive. However, it is advisable to verify the analytical calculations by evaluating the gradient at the maximum posterior estimate, where the function should return values close to 0, or by comparing results with finite differences at a few points.

Algorithm A3 Hamiltonian Monte Carlo

```

1: Initiate at  $\boldsymbol{\theta}^{(0)}$  in the support of  $\pi(\boldsymbol{\theta}|\mathbf{y})$ , and set step size  $\epsilon$ , number of
   leapfrog steps  $L$ , and total iterations  $S$ 
2: Draw  $\boldsymbol{\delta}^{(0)}$  from  $N(\mathbf{0}, \mathbf{M})$ 
3: for  $s = 1, \dots, S$  do
4:   for  $l = 1, \dots, L$  do
5:     if  $l = 1$  then
6:        $\boldsymbol{\delta}^c \leftarrow \boldsymbol{\delta}^{(s-1)} + \frac{1}{2}\epsilon \frac{d \log(\pi(\boldsymbol{\theta}|\mathbf{y}))}{d\boldsymbol{\theta}}$ 
7:        $\boldsymbol{\theta}^c \leftarrow \boldsymbol{\theta}^{(s-1)} + \epsilon \mathbf{M}^{-1} \boldsymbol{\delta}^c$ 
8:     else
9:       if  $l=2,\dots,L-1$  then
10:         $\boldsymbol{\delta}^c \leftarrow \boldsymbol{\delta}^c + \epsilon \frac{d \log(\pi(\boldsymbol{\theta}|\mathbf{y}))}{d\boldsymbol{\theta}}$ 
11:         $\boldsymbol{\theta}^c \leftarrow \boldsymbol{\theta}^c + \epsilon \mathbf{M}^{-1} \boldsymbol{\delta}^c$ 
12:      else
13:         $\boldsymbol{\delta}^c \leftarrow \boldsymbol{\delta}^c + \frac{1}{2}\epsilon \frac{d \log(\pi(\boldsymbol{\theta}|\mathbf{y}))}{d\boldsymbol{\theta}}$ 
14:         $\boldsymbol{\theta}^c \leftarrow \boldsymbol{\theta}^c + \epsilon \mathbf{M}^{-1} \boldsymbol{\delta}^c$ 
15:      end if
16:    end if
17:  end for
18:  Calculate  $\alpha([\boldsymbol{\theta} \; \boldsymbol{\delta}]^{(s-1)}, [\boldsymbol{\theta} \; \boldsymbol{\delta}]^c) = \min \left\{ \frac{p(\boldsymbol{\delta}^c)\pi(\boldsymbol{\theta}^c|\mathbf{y})}{p(\boldsymbol{\delta}^{(s-1)})\pi(\boldsymbol{\theta}^{(s-1)}|\mathbf{y})}, 1 \right\}$ 
19:  Draw  $U$  from  $U(0, 1)$ 
20:   $\boldsymbol{\theta}^{(s)} = \begin{cases} \boldsymbol{\theta}^c & \text{if } U \leq \alpha(\cdot, \cdot) \\ \boldsymbol{\theta}^{(s-1)} & \text{otherwise} \end{cases}$ 
21: end for

```

Note that HMC does not require the marginal likelihood, as neither the gradient vector $\frac{d \log(\pi(\boldsymbol{\theta}|\mathbf{y}))}{d\boldsymbol{\theta}}$ nor the acceptance rate depend on it. That is, we can use only $\pi(\boldsymbol{\theta}) \times p(\mathbf{y}|\boldsymbol{\theta})$ to implement HMC. In addition, we do not retain $\boldsymbol{\delta}$ after it is updated at the beginning of each iteration, as it is not required subsequently. To begin, the step size (ϵ) can be drawn randomly from a uniform distribution between 0 and $2\epsilon_0$, and the number of leapfrog steps (L) is set as the largest integer near $1/\epsilon$, ensuring $\epsilon \times L \approx 1$. We need to

set \mathbf{M} to be the inverse of the posterior covariance matrix evaluated at the maximum a posteriori estimate under this setting.

The acceptance rate should be checked, with the optimal rate around 65% [76, Chap. 12]. If the acceptance rate is much higher than 65%, increase ϵ_0 ; if it is much lower, decrease it. This strategy may not always work, and alternative strategies can be tested, such as setting $\mathbf{M} = \mathbf{I}$ and fine-tuning ϵ and L to achieve an acceptance rate near 65%. Finally, the number of iterations (S) is chosen to ensure convergence to the stationary distribution.

Example: Sampling from a bi-variate Gaussian distribution

As a toy example, let's compare the Gibbs sampling, M-H and HMC algorithms when the posterior distribution is a bi-variate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. Let's set $\rho = 0.98$.

The Gibbs sampler requires the conditional posterior distributions, which in this case are $\theta_1|\theta_2 \sim N(\rho\theta_2, 1 - \rho^2)$ and $\theta_2|\theta_1 \sim N(\rho\theta_1, 1 - \rho^2)$. We use the random walk proposal distribution for the M-H algorithm, where $\boldsymbol{\theta}^c \sim N(\boldsymbol{\theta}^{(s-1)}, \text{diag}\{0.18^2\})$. We set $\epsilon = 0.05$, $L = 20$ and $\mathbf{M} = \mathbf{I}_2$ for the HMC algorithm, and given that $\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{\theta}^\top \Sigma^{-1} \boldsymbol{\theta}\right\}$, then $\frac{d \log(\pi(\boldsymbol{\theta}|\mathbf{y}))}{d\boldsymbol{\theta}} = -\Sigma^{-1}\boldsymbol{\theta}$.

The following code shows how to implement the Gibbs sampler, the random walk M-H algorithm, and the HMC in this example such that the effective number of posterior draws is 400.

R code. Gibbs, M-H and HMC: Bi-variate normal distribution

```

1 rm(list = ls()); set.seed(010101)
2 # Gibbs sampler
3 Gibbs <- function(theta, rho){
4   thetal <- rnorm(1, mean = rho*theta, sd = (1- rho^2)^0.5)
5   return(thetal)
6 }
7 # Metropolis-Hastings
8 MH <- function(theta, rho, sig2){
9   SIGMA <- matrix(c(1, rho, rho, 1), 2, 2)
10  SIGMAC <- matrix(c(1, sig2, sig2, 1), 2, 2)
11  thetac <- MASS::mvrnorm(1, mu = theta, Sigma = SIGMAC)
12  a <- mvtnorm::dmvnorm(thetac, c(0, 0), SIGMA)/mvtnorm::
13    dmvnorm(theta, c(0, 0), SIGMA)
14  U <- runif(1)
15  if(U <= a){
16    theta <- thetac
17    accept <- 1
18  }else{
19    theta <- theta
20    accept <- 0
21  }
22  return(list(theta = theta, accept = accept))
23 }
24 # Hamiltonian Monte Carlo
25 HMC <- function(theta, rho, epsilon, M){
26   SIGMA <- matrix(c(1, rho, rho, 1), 2, 2)
27   L <- ceiling(1/epsilon)
28   Minv <- solve(M); thetac <- theta
29   K <- length(theta)
30   mom <- t(mvtnorm::rmvnorm(1, rep(0, K), M))
31   logPost_Mom_t <- mvtnorm::dmvnorm(t(theta), rep(0, K),
32     SIGMA, log = TRUE) + mvtnorm::dmvnorm(t(mom), rep(0, K)
33     , M, log = TRUE)
34   for(l in 1:L){
35     if(l == 1 | l == L){
36       mom <- mom + 0.5*epsilon*(-solve(SIGMA)%*%theta)
37       theta <- theta + epsilon*Minv%*%mom
38     }else{
39       mom <- mom + epsilon*(-solve(SIGMA)%*%theta)
40       theta <- theta + epsilon*Minv%*%mom
41     }
42   }
43   logPost_Mom_star <- mvtnorm::dmvnorm(t(theta), rep(0, K),
44     SIGMA, log = TRUE) + mvtnorm::dmvnorm(t(mom), rep(0, K)
45     , M, log = TRUE)
46   alpha <- min(1, exp(logPost_Mom_star-logPost_Mom_t))
47   u <- runif(1)
48   if(u <= alpha){
49     thetaNew <- c(theta)
50   }else{
51     thetaNew <- thetac
52   }
53   rest <- list(theta = thetaNew, Prob = alpha)
54   return(rest)
55 }
```

R code. Gibbs, M-H and HMC: Bi-variate normal distribution

```

1 # Hyperparameters
2 rho <- 0.98; sig2 <- 0.18^2
3 # Posterior draws Gibbs and M-H
4 S <- 8000; thin <- 20; K <- 2
5 thetaPostGibbs <- matrix(NA, S, K)
6 thetaPostMH <- matrix(NA, S, K)
7 AcceptMH <- rep(NA, S)
8 thetaGibbs <- c(-2, 3); thetaMH <- c(-2, 3)
9 for(s in 1:S){
10   theta1 <- Gibbs(thetaGibbs[2], rho)
11   theta2 <- Gibbs(theta1, rho)
12   thetaGibbs <- c(theta1, theta2)
13   ResMH <- MH(thetaMH, rho, sig2)
14   thetaMH <- ResMH$theta
15   thetaPostGibbs[s,] <- thetaGibbs
16   thetaPostMH[s,] <- thetaMH
17   AcceptMH[s] <- ResMH$accept
18 }
19 keep <- seq(0, S, thin)
20 mean(AcceptMH[keep[-1]])
21 thetaPostGibbsMCMC <- coda::mcmc(thetaPostGibbs[keep[-1],])
22 summary(thetaPostGibbsMCMC)
23 coda::autocorr.plot(thetaPostGibbsMCMC)
24 thetaPostMHMCMC <- coda::mcmc(thetaPostMH[keep[-1],])
25 plot(thetaPostMHMCMC)
26 coda::autocorr.plot(thetaPostMHMCMC)
27 # Posterior draws HMC
28 S <- 400; epsilon <- 0.05; L <- ceiling(1/epsilon); M <-
29   diag(2)
30 thetaPostHMC <- matrix(NA, S, K)
31 ProbAcceptHMC <- rep(NA, S)
32 thetaHMC <- c(-2, 3)
33 for(s in 1:S){
34   ResHMC <- HMC(theta = thetaHMC, rho, epsilon, M)
35   thetaHMC <- ResHMC$theta
36   thetaPostHMC[s,] <- thetaHMC
37   ProbAcceptHMC[s] <- ResHMC$Prob
38 }
39 thetaPostHMCMCMC <- coda::mcmc(thetaPostHMC)
40 plot(thetaPostHMCMCMC); coda::autocorr.plot(thetaPostHMCMCMC)
41 summary(ProbAcceptHMC)
42 #Figure
43 df <- as.data.frame(cbind(1:S, thetaPostHMC[,1], thetaPostMH
44   [keep[-1],1], thetaPostGibbs[keep[-1],1]))
45 colnames(df) <- c("Iter", "HMC", "MH", "Gibbs")
46 library(latex2exp); library(ggpubr)
47 g1 <- ggplot(df, aes(x= Iter)) + geom_point(aes(y=HMC),
48   colour="black") + labs(x = "Iteration", y = TeX("$\\theta_{\\{1\\}}$"),
49   title = "HMC algorithm")
50 g2 <- ggplot(df, aes(x= Iter)) + geom_point(aes(y=MH),
51   colour="black") + labs(x = "Iteration", y = TeX("$\\theta_{\\{1\\}}$"),
52   title = "M-H algorithm")
53 g3 <- ggplot(df, aes(x= Iter)) + geom_point(aes(y=Gibbs),
54   colour="black") + labs(x = "Iteration", y = TeX("$\\theta_{\\{1\\}}$"),
55   title = "Gibbs sampling")
56 ggarrange(g3, g2, g1, labels = c("A", "B", "C"), ncol = 3,
57   nrow = 1)

```

Figure 4.3 shows the posterior draws of θ_1 using the Gibbs sampler (Panel A, left), the Metropolis-Hastings algorithm (Panel B, middle), and the Hamiltonian Monte Carlo (Panel C, right). The convergence diagnostic plots (no shown) suggests that the three algorithms perform a good job. Although, the acceptance rate in HMC is higher than the M-H due to the HMC producing larger changes in θ than a corresponding number of random-walk M-H iterations [155].

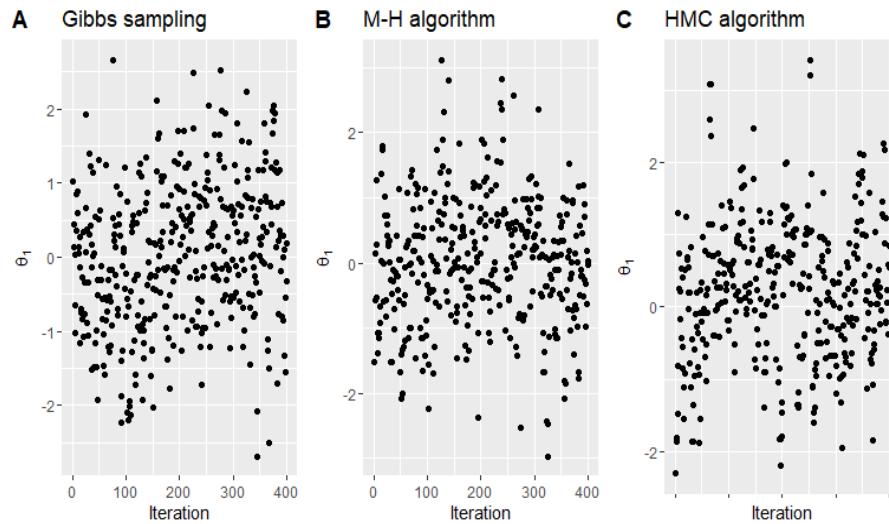


FIGURE 4.3

Posterior draws: Gibbs sampler (A), Metropolis-Hastings (B) and Hamiltonian Monte Carlo (C) in the bivariate normal example, $\rho = 0.98$.

4.2 Importance sampling

Up to this section, we have introduced MCMC methods for sampling from the posterior distribution when it does not have a standard closed form. However, MCMC methods have some limitations. First, the samples are generated sequentially, which complicates parallel computing. Although multiple MCMC chains can be run simultaneously, this approach—often referred to as brute-force parallelization—does not fully address the sequential nature of individual chains. Second, consecutive samples are correlated, which reduces the effective sample size and complicates convergence diagnostics.

Thus, in this section, we introduce *importance sampling* (IS), a simulation method for drawing samples from the posterior distribution that avoids

these limitations. Unlike MCMC, IS does not require satisfying the balancing condition, making it conceptually and mathematically simpler to implement in certain situations. Moreover, importance weights can be reused to analyze posterior quantities, compute marginal likelihoods, compare models, approximate new target distributions, and allow for straightforward parallelization in large-scale problems.

However, the critical challenge in IS lies in selecting an appropriate proposal distribution. This involves satisfying both support and stability conditions, which can be difficult to achieve, particularly in high-dimensional problems. In such cases, MCMC methods may be more suitable.

The starting point is evaluating the integral:

$$\mathbb{E}_\pi[h(\boldsymbol{\theta})] = \int_{\Theta} h(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}, \quad (4.1)$$

where \mathbb{E}_π denotes expected value under the posterior distribution. Thus, we can approximate Equation 4.1 by

$$\bar{h}(\boldsymbol{\theta})_S = \frac{1}{S} \sum_{s=1}^S h(\boldsymbol{\theta}^{(s)}), \quad (4.2)$$

where $\boldsymbol{\theta}^{(s)}$ are draws from $\pi(\boldsymbol{\theta}|\mathbf{y})$. The *strong law of large numbers* shows that $\bar{h}(\boldsymbol{\theta})_S$ converges (almost surely) to $\mathbb{E}_\pi[h(\boldsymbol{\theta})]$ as $S \rightarrow \infty$.

The challenge arises when we do not know how to obtain samples from $\pi(\boldsymbol{\theta}|\mathbf{y})$. The ingenious idea is to express Equation 4.1 in a different way using the *importance sampling fundamental identity* [185, Chap. 3]:

$$\begin{aligned} \mathbb{E}_\pi[h(\boldsymbol{\theta})] &= \int_{\Theta} h(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{y})\frac{q(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}d\boldsymbol{\theta} \\ &= \mathbb{E}_q \left[\frac{h(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{y})}{q(\boldsymbol{\theta})} \right], \end{aligned} \quad (4.3)$$

where $q(\boldsymbol{\theta})$ is the proposal distribution.

Thus, we have

$$\frac{1}{S} \sum_{s=1}^S \left[\frac{h(\boldsymbol{\theta}^{(s)})\pi(\boldsymbol{\theta}^{(s)}|\mathbf{y})}{q(\boldsymbol{\theta}^{(s)})} \right] = \frac{1}{S} \sum_{s=1}^S h(\boldsymbol{\theta}^{(s)})w(\boldsymbol{\theta}^{(s)}),$$

where $w(\boldsymbol{\theta}^{(s)}) = \left[\frac{\pi(\boldsymbol{\theta}^{(s)}|\mathbf{y})}{q(\boldsymbol{\theta}^{(s)})} \right]$ are called the *importance weights*, and $\boldsymbol{\theta}^{(s)}$ are samples from the proposal distribution. This expression converges to $\mathbb{E}_\pi[h(\boldsymbol{\theta})]$ given that the support of $q(\boldsymbol{\theta})$ includes the support of $\pi(\boldsymbol{\theta}^{(s)}|\mathbf{y})$.

There are many proposal distributions that satisfy the support condition. However, the stability of the method depends heavily on the variability of the importance weights. In particular, the variance of

$$\frac{1}{S} \sum_{s=1}^S h(\boldsymbol{\theta}^{(s)})w(\boldsymbol{\theta}^{(s)})$$

can be large if the proposal distribution has lighter tails than the posterior distribution. In this case, the weights $w(\boldsymbol{\theta}^{(s)})$ will vary widely, assigning too much importance to a few values of $\boldsymbol{\theta}^{(s)}$. Thus, it is important to use proposals that have thicker tails than the posterior distribution. In any case, we should check the adequacy of the proposal distribution by analyzing the behavior of the importance weights. If they are distributed more or less uniformly over the support, it is a good sign. Consider, for instance, the extreme case where $q(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y})$, then $w(\boldsymbol{\theta}^{(s)}) = 1$ everywhere.

A natural choice in Bayesian inference is to use the prior distribution as the proposal, given that it is a proper density function. The prior distribution typically has heavier tails than the posterior by construction, and it is usually a distribution that allows for easy sampling.

The most relevant point for us is that importance sampling provides a way to simulate from the posterior distribution when there is no closed-form solution. The method generates samples $\boldsymbol{\theta}^{(s)}$ from $q(\boldsymbol{\theta})$ and computes the importance weights $w(\boldsymbol{\theta}^{(s)})$. Thus, if we *resample* with replacement from $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots, \boldsymbol{\theta}^{(S)}$, selecting $\boldsymbol{\theta}^{(s)}$ with probability proportional to $w(\boldsymbol{\theta}^{(s)})$, we would get a sample $\boldsymbol{\theta}^{*(1)}, \boldsymbol{\theta}^{*(2)}, \dots, \boldsymbol{\theta}^{*(L)}$ of size L from $\pi(\boldsymbol{\theta}|\mathbf{y})$ [204, 190]. This is named *sampling/importance resampling* (SIR) algorithm. Observe that the number of times $L^{(s)}$ each particular point $\boldsymbol{\theta}^{(s)}$ is selected follows a binomial distribution with size L , and probabilities proportional to $w^{(s)}$. Consequently, the vector $L_{\boldsymbol{\theta}} = \{L_{\boldsymbol{\theta}^1}, L_{\boldsymbol{\theta}^2}, \dots, L_{\boldsymbol{\theta}^S}\}$ follows a multinomial distribution with L trials and probabilities proportional to $w(\boldsymbol{\theta}^{(s)})$, $s = 1, 2, \dots, S$ [27]. Therefore, the resampling step ensures that points in the first-stage sample with small importance weights are more likely to be discarded, while points with high weights are replicated in proportion to their importance weights. In most applications, it is typical to have $S \gg L$.

The intuition is that importance weights are scaling factors that correct for the bias introduced by drawing from $q(\boldsymbol{\theta}^{(s)})$ instead of $\pi(\boldsymbol{\theta}^{(s)}|\mathbf{y})$; thus, when combined, the samples and weights effectively recreate the posterior distribution, ensuring the resampled data set reflects the posterior. Let's proof this

$$\begin{aligned} P(\boldsymbol{\theta}^* \in A) &= \frac{1}{S} \sum_{s=1}^S w^{(s)} \mathbb{1}_A(\boldsymbol{\theta}^{(s)}) \\ &\rightarrow \mathbb{E}_q \left[\mathbb{1}_{\in A}(\boldsymbol{\theta}) \frac{\pi(\boldsymbol{\theta}|\mathbf{y})}{q(\boldsymbol{\theta})} \right] \\ &= \int_A \left[\frac{\pi(\boldsymbol{\theta}|\mathbf{y})}{q(\boldsymbol{\theta})} \right] q(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \int_A \pi(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}. \end{aligned}$$

Thus, $\boldsymbol{\theta}^*$ is approximately distributed as an observation from $\pi(\boldsymbol{\theta}|\mathbf{y})$.

However, the weights $\pi(\boldsymbol{\theta}^{(s)}|\mathbf{y})/(Sq(\boldsymbol{\theta}^{(s)}))$ do not sum up to 1, and we

need to standardize them,

$$w^*(\boldsymbol{\theta}^{(s)}) = \frac{\frac{1}{S} w(\boldsymbol{\theta}^{(s)})}{\frac{1}{S} \sum_{s=1}^S w(\boldsymbol{\theta}^{(s)})}.$$

Note that we could alternatively arrive to these weights as follow

$$\begin{aligned} \mathbb{E}_\pi[h(\boldsymbol{\theta})] &= \int_{\Theta} \left[\frac{h(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{y})}{q(\boldsymbol{\theta})} \right] q(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \frac{\int_{\Theta} \left[\frac{h(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{y})}{q(\boldsymbol{\theta})} \right] q(\boldsymbol{\theta}) d\boldsymbol{\theta}}{\int_{\Theta} \left[\frac{\pi(\boldsymbol{\theta}|\mathbf{y})}{q(\boldsymbol{\theta})} \right] q(\boldsymbol{\theta}) d\boldsymbol{\theta}}. \end{aligned}$$

Then,

$$\frac{\frac{1}{S} \sum_{s=1}^S h(\boldsymbol{\theta}^{(s)}) w(\boldsymbol{\theta}^{(s)})}{\frac{1}{S} \sum_{s=1}^S w(\boldsymbol{\theta}^{(s)})} = \sum_{s=1}^S h(\boldsymbol{\theta}^{(s)}) w^*(\boldsymbol{\theta}^{(s)}).$$

This alternative expression also converges (almost surely) to $\mathbb{E}_\pi[h(\boldsymbol{\theta})]$. In addition, this expression is very useful because if we do not have the marginal likelihood in the posterior distribution, this constant cancels out in $w^*(\boldsymbol{\theta}^{(s)})$. And, although this estimator is biased, the bias is small, and provides good gains in variance reduction compared with the non-standardized option [185, Chap. 3].

A nice by-product of implementing IS is that it easily allows the calculation of the marginal likelihood. In particular, we know from Bayes' rule that

$$p(\mathbf{y})^{-1} = \frac{\pi(\boldsymbol{\theta}|\mathbf{y})}{p(\mathbf{y}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta})},$$

then,

$$\begin{aligned} \int_{\Theta} p(\mathbf{y})^{-1} q(\boldsymbol{\theta}) d\boldsymbol{\theta} &= \int_{\Theta} \frac{q(\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta})} \pi(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \\ &= \mathbb{E}_\pi \left[\frac{q(\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta})} \right]. \end{aligned}$$

Thus, an estimate of the marginal likelihood is $\left[\frac{1}{S} \sum_{s=1}^S \frac{q(\boldsymbol{\theta}^{*(s)})}{p(\mathbf{y}|\boldsymbol{\theta}^{*(s)}) \times \pi(\boldsymbol{\theta}^{*(s)})} \right]^{-1}$. This is the Gelfand-Dey method to calculate the marginal likelihood [72] (see subsection 10.5.3).

Example: Cauchy distribution

Let's assume that the posterior distribution is Cauchy with parameters 0 and 1. We perform an importance sampling algorithm using as proposals a standard normal distribution and a Student's t distribution with 3 degrees of freedom. The following code shows how to do this.

R code. Importance sampling: Cauchy distribution

```

1 rm(list = ls()); set.seed(010101)
2 S <- 20000 # Size proposal
3 # Importance sampling from standard normal proposal
4 thetaNs <- rnorm(S)
5 wNs <- dcauchy(thetaNs)/dnorm(thetaNs)
6 wNstars <- wNs/sum(wNs)
7 L <- 10000 # Size posterior
8 thetaCauchyN <- sample(thetaNs, L, replace = TRUE, prob =
  wNstars)
9 h <- hist(thetaCauchyN, breaks=50, col="blue", xlab="x",
  main="Cauchy draws from importance sampling: Normal
  standard proposal")
10 pfit <- seq(min(thetaCauchyN),max(thetaCauchyN),length=50)
11 yfit<-dcauchy(pfit)
12 yfit <- yfit*diff(h$mids[1:2])*length(thetaCauchyN)
13 lines(pfit, yfit, col="red", lwd=2)
14 # Importance sampling from Student's t proposal
15 df <- 3
16 thetaTs <- rt(S, df = df)
17 wTs <- dcauchy(thetaTs)/dt(thetaTs, df = df)
18 wTstars <- wTs/sum(wTs)
19 thetaCauchyT <- sample(thetaTs, L, replace = TRUE, prob =
  wTstars)
20 h <- hist(thetaCauchyT, breaks=50, col="blue", xlab="x",
  main="Cauchy draws from importance sampling: Student's t
  proposal")
21 pfit <- seq(min(thetaCauchyT),max(thetaCauchyT),length=50)
22 yfit<-dcauchy(pfit)
23 yfit <- yfit*diff(h$mids[1:2])*length(thetaCauchyT)
24 lines(pfit, yfit, col="red", lwd=2)
25 plot(wNstars, main = "Importance sampling: Cauchy
  distribution", ylab = "Weights", xlab = "Iterations")
26 points(wTstars, col = "blue")
27 legend("topright", legend = c("Normal", "Student's t"), col
  = c("black", "blue"), pch = c(1, 1))

```

Figure 4.4 shows the posterior draws of a Cauchy distribution using the standard normal distribution (black dots) and the Student's t-distribution with 3 degrees of freedom (blue dots) as proposals. We observe that a few draws carry too much weight when using the normal proposal; this occurs because the normal distribution has much lighter tails compared to the Cauchy distribution. In contrast, using the Student's t-distribution with 3 degrees of freedom improves this situation significantly.

Figures 4.5 and 4.6 show the histograms of the posterior draws using the normal and Student's t-distributions, respectively, along with the density of

the Cauchy distribution. The spike in the posterior draws from the standard normal proposal arises due to the lighter tails of the standard normal compared to the Cauchy distribution, consequently assigning too much weight to a specific draw from the normal distribution.

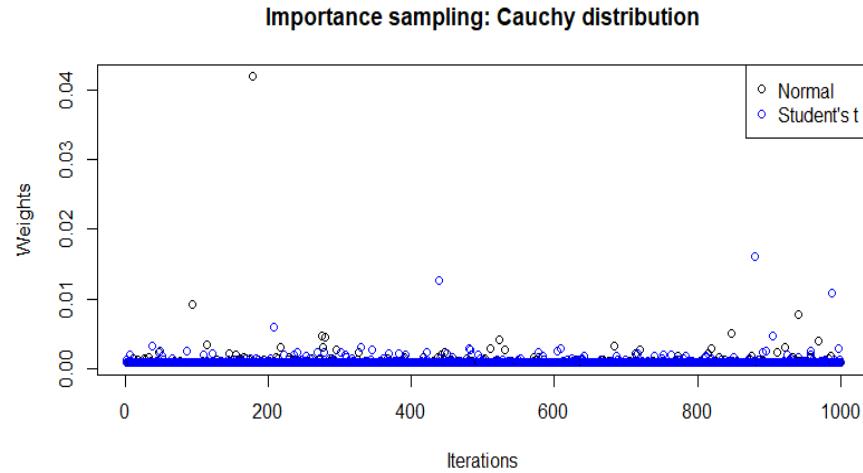


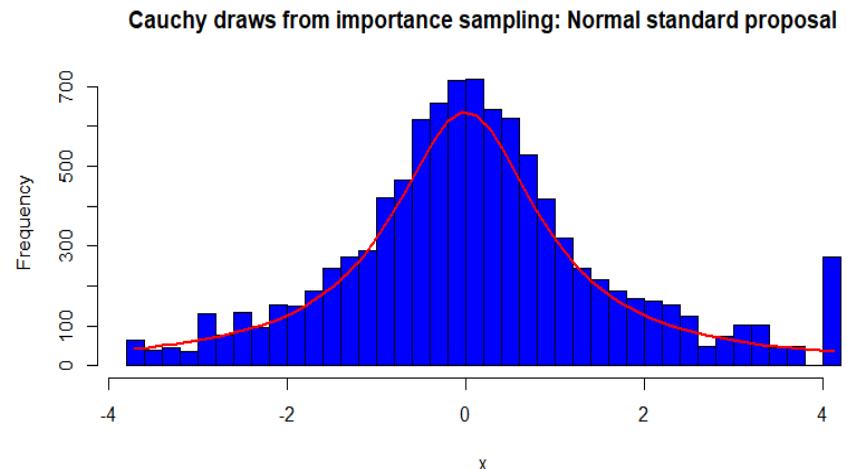
FIGURE 4.4

Importance sampling: 1000 draws of a Cauchy distribution using the standard normal and student's t with 3 degrees of freedom as proposals.

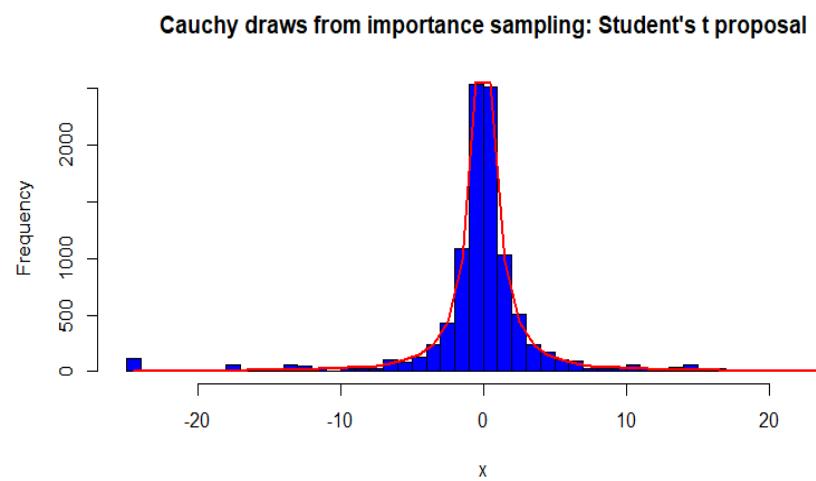
4.3 Particle filtering

Now, we consider the scenario where we need to sample from a posterior distribution whose dimension increases over time, $\pi(\theta_{0:t} | \mathbf{y}_{0:t})$, for $t = 0, 1, \dots$. The challenge arises from the fact that, even if this posterior distribution is known, the computational complexity of implementing a sampling scheme in this context increases linearly with t . This makes MCMC methods, which operate in batch mode and require a complete re-run whenever new information becomes available, less optimal. Consequently, we present sequential algorithms, which operate incrementally as new data becomes available, and are often a better alternative. These algorithms are typically faster and are well-suited for scenarios requiring real-time updates, commonly referred to as online mode.

Specifically, we consider the dynamic system in the *state-space* representation. This is a system where there is an *unobservable state vector* $\theta_t \in \mathbb{R}^K$,

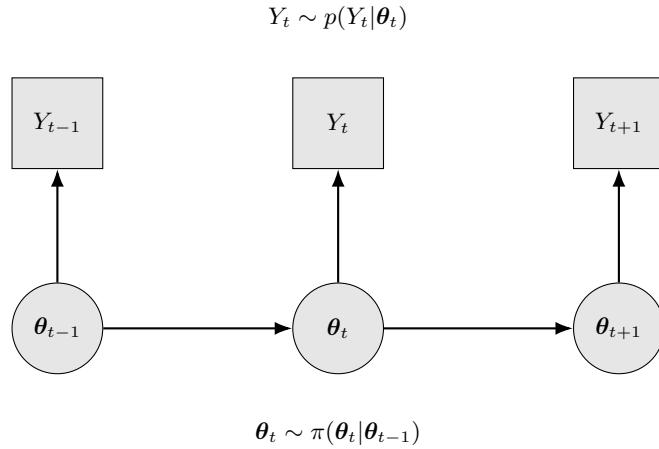
**FIGURE 4.5**

Importance sampling: Draws from a Cauchy distribution using the standard normal as proposal.

**FIGURE 4.6**

Importance sampling: Draws of a Cauchy distribution using the Student's t with 3 degrees of freedom as proposal.

and an observed variable \mathbf{Y}_t , $t = 0, 1, \dots$ such that (i) $\boldsymbol{\theta}_t$ is a *Markov process*, this is, $\pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}) = \pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1})$, $t = 1, 2, \dots$, all the relevant information to define $\boldsymbol{\theta}_t$ is in $\boldsymbol{\theta}_{t-1}$,¹ and ii) $\mathbf{Y}_t \perp \mathbf{Y}_s | \boldsymbol{\theta}_t$, $s < t$, there is independence between observable variables regarding their history conditional on the actual state vector. We can see in Figure 4.7 a graphical representation of the dynamic system.



Notes: The figure illustrates the structure of a *state-space* model where the latent states $\boldsymbol{\theta}_t$ evolve according to $\pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1})$, and the observations Y_t depend on the states via $p(Y_t | \boldsymbol{\theta}_t)$.

FIGURE 4.7
State-space model representation.

Formally,

$$\begin{aligned} \boldsymbol{\theta}_t &= h(\boldsymbol{\theta}_{t-1}, \mathbf{w}_t) && \text{(State equations)} \\ Y_t &= f(\boldsymbol{\theta}_t, \mu_t) && \text{(Observation equation),} \end{aligned}$$

where \mathbf{w}_t and μ_t are stochastic errors such that their probability distributions define the transition density $\pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1})$ and observation density $p(Y_t | \boldsymbol{\theta}_t)$.

We present *particle filtering*, a specific case of *sequential Monte Carlo* (SMC), which is one of the most commonly used algorithms for scenarios requiring sequential updates of the posterior distribution as described by the *state-space* model.

The starting point is *sequential importance sampling* (SIS), originally proposed by [95], which is a modification of IS to compute an estimate of $\pi(\boldsymbol{\theta}_{0:t} | \mathbf{y}_{0:t})$ without altering the past trajectories $\{\boldsymbol{\theta}_{1:t-1}^{(s)}, s = 1, 2, \dots, S\}$. The key idea is to use a proposal density that takes the form

¹ $\boldsymbol{\theta}_0$ comes from the given distribution $\pi(\boldsymbol{\theta}_0)$.

$$\begin{aligned} q(\boldsymbol{\theta}_{0:t} | \mathbf{y}_{0:t}) &= q(\boldsymbol{\theta}_{0:t-1} | \mathbf{y}_{1:t-1})q(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \mathbf{y}_t) \\ &= q(\boldsymbol{\theta}_0) \prod_{h=1}^t q(\boldsymbol{\theta}_h | \boldsymbol{\theta}_{h-1}, \mathbf{y}_h). \end{aligned}$$

This proposal density allows calculating the weights sequentially,

$$\begin{aligned} w_t(\boldsymbol{\theta}_{0:t}^{(s)}) &= \frac{\pi(\boldsymbol{\theta}_{0:t}^{(s)} | \mathbf{y}_{0:t})}{q(\boldsymbol{\theta}_{0:t}^{(s)} | \mathbf{y}_{0:t})} \\ &= \frac{p(\mathbf{y}_{0:t} | \boldsymbol{\theta}_{0:t}^{(s)})\pi(\boldsymbol{\theta}_{0:t}^{(s)})}{p(\mathbf{y}_{0:t})q(\boldsymbol{\theta}_{0:t}^{(s)} | \mathbf{y}_{0:t})} \\ &= \frac{p(\mathbf{y}_t | \boldsymbol{\theta}_t^{(s)})p(\mathbf{y}_{1:t-1} | \boldsymbol{\theta}_{0:t-1}^{(s)})\pi(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)})\pi(\boldsymbol{\theta}_{0:t-1}^{(s)})}{p(\mathbf{y}_{0:t})q(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)}, \mathbf{y}_t)q(\boldsymbol{\theta}_{0:t-1}^{(s)} | \mathbf{y}_{1:t-1})} \\ &\propto \frac{p(\mathbf{y}_{1:t-1} | \boldsymbol{\theta}_{0:t-1}^{(s)})\pi(\boldsymbol{\theta}_{0:t-1}^{(s)})}{q(\boldsymbol{\theta}_{0:t-1}^{(s)} | \mathbf{y}_{1:t-1})} \frac{p(\mathbf{y}_t | \boldsymbol{\theta}_t^{(s)})\pi(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)})}{q(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)}, \mathbf{y}_t)} \\ &\propto w_{t-1}(\boldsymbol{\theta}^{(s)}) \frac{p(\mathbf{y}_t | \boldsymbol{\theta}_t^{(s)})\pi(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)})}{q(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)}, \mathbf{y}_t)} \\ &\propto w_{t-1}^*(\boldsymbol{\theta}^{(s)}) \frac{p(\mathbf{y}_t | \boldsymbol{\theta}_t^{(s)})\pi(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)})}{q(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)}, \mathbf{y}_t)}. \end{aligned}$$

Take into account that $p(\mathbf{y}_{0:t})$ does not depend on $\boldsymbol{\theta}_{0:t}^{(s)}$. The term $\alpha_t(\boldsymbol{\theta}_{0:t}^{(s)}) = \frac{p(\mathbf{y}_t | \boldsymbol{\theta}_t^{(s)})\pi(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)})}{q(\boldsymbol{\theta}_t^{(s)} | \boldsymbol{\theta}_{t-1}^{(s)}, \mathbf{y}_t)}$ is called the *incremental importance weight*, and implies that

$$w_t(\boldsymbol{\theta}_{0:t}^s) = w_0(\boldsymbol{\theta}_0^s) \prod_{h=1}^t \alpha_h(\boldsymbol{\theta}_{1:h}^{(s)}).$$

This algorithm possesses the desirable property of maintaining fixed computational complexity. Consequently, we sequentially obtain draws $\boldsymbol{\theta}_t^{(s)}$, referred to as particles: $\boldsymbol{\theta}_0^{(s)}$ is drawn from $q(\boldsymbol{\theta}_0)$ at $t = 0$, and subsequently, $\boldsymbol{\theta}_h^{(s)}$ is drawn from $q(\boldsymbol{\theta}_h | \boldsymbol{\theta}_{h-1}, \mathbf{y}_h)$ at $t = h$ [59, 27].

A relevant case is when the proposal distribution takes the form of the prior distribution, that is, $q(\boldsymbol{\theta}_{0:t} | \mathbf{y}_{0:t}) = \pi(\boldsymbol{\theta}_{0:t}) = \pi(\boldsymbol{\theta}_0) \prod_{h=1}^t \pi(\boldsymbol{\theta}_h | \boldsymbol{\theta}_{h-1})$. This implies that

$$w_t(\boldsymbol{\theta}^{(s)}) \propto w_{t-1}^*(\boldsymbol{\theta}^{(s)}) p(\mathbf{y}_t | \boldsymbol{\theta}_t^{(s)}),$$

this means that the *incremental importance weight* is given by $p(\mathbf{y}_t | \boldsymbol{\theta}_t^{(s)})$.

Algorithm A4 shows how to perform SIS [27]. We set $w_t^{(s)} := w_t(\boldsymbol{\theta}_{0:t}^{(s)})$ to

Algorithm A4 Sequential importance sampling algorithm

```

1: for  $s = 1, \dots, S$  do
2:   Sample  $\boldsymbol{\theta}_0^{(s)}$  from  $q(\boldsymbol{\theta}_0|y_0)$ 
3:   Calculate the importance weights  $w_0^{(s)} \propto \frac{p(y_0|\boldsymbol{\theta}_0^{(s)})\pi(\boldsymbol{\theta}_0^{(s)})}{q(\boldsymbol{\theta}_0^{(s)}|y_0)}$ 
4: end for
5: for  $t = 1, \dots, T$  do
6:   for  $s = 1, \dots, S$  do
7:     Draw particles  $\boldsymbol{\theta}_t^{(s)}$  from  $q_t(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \mathbf{y}_t)$ 
8:     Compute the weights  $w_t^{(s)} = w_{t-1}^{*(s)} \frac{p(\mathbf{y}_t|\boldsymbol{\theta}_t^{(s)})\pi(\boldsymbol{\theta}_t^{(s)}|\boldsymbol{\theta}_{t-1}^{(s)})}{q(\boldsymbol{\theta}_t^{(s)}|\boldsymbol{\theta}_{t-1}^{(s)}, \mathbf{y}_t)}$ 
9:   end for
10:  Standardize the weights  $w_t^{*(s)} = \frac{w_t^{(s)}}{\sum_{h=1}^S w_t^{(h)}}, s = 1, 2, \dots, S$ 
11: end for

```

simplify notation.

Example: Dynamic linear model

Let's assume that the state-space representation is

$$\begin{aligned}\theta_t &= \theta_{t-1} + w_t && \text{(State equation)} \\ Y_t &= \phi\theta_t + \mu_t && \text{(Observation equation),}\end{aligned}$$

where $w_t \sim N(0, \sigma_w^2)$ and $\mu_t \sim N(0, \sigma_\mu^2)$, $t = 1, 2, \dots, 50$. In addition, we use the proposal distribution $q(\theta_t|y_t) = \pi(\theta_t)$, which is normal with mean θ_{t-1} and variance σ_w^2 . Then, the weights are given by the recursion

$$w_t^{(s)} \propto w_{t-1}^{*(s)} p(y_t|\theta_t, \sigma_\mu^2),$$

where $p(y_t|\theta_t, \sigma_\mu^2)$ is $N(\phi\theta_t, \sigma_\mu^2)$.

We can compute the mean and standard deviation of the state at each t using

$$\hat{\theta}_t = \sum_{s=1}^S w_t^{*(s)} \theta_t^{(s)}$$

and

$$\hat{\sigma}_\theta = \left(\sum_{s=1}^S w_t^{*(s)} \theta_t^{2(s)} - \hat{\theta}_t^2 \right)^{1/2}.$$

The following code demonstrates the implementation of this algorithm, setting $\sigma_w^2 = \sigma_\mu^2 = 1$ and $\phi = 0.5$. First, we simulate the process, and then we implement the SIS algorithm.

R code. Sequential importance sampling: Dynamic linear model

```

1 rm(list = ls()); set.seed(010101)
2 S <- 50000 # Number of particles
3 sigma_w <- 1 # State noise
4 sigma_mu <- 1 # Observation noise
5 phi <- 0.5 # Coefficient in observation equation
6 T <- 50 # Sample size
7 # Simulate true states and observations
8 theta_true <- numeric(T); y_obs <- numeric(T)
9 theta_true[1] <- rnorm(1, mean = 0, sd = sigma_w) # Initial
   state
10 for (t in 2:T) {
11   theta_true[t] <- rnorm(1, mean = theta_true[t-1], sd =
   sigma_w)
12 }
13 y_obs <- rnorm(T, mean = phi*theta_true, sd = sigma_mu)
14 # Sequential Importance Sampling (SIS)
15 particles <- matrix(0, nrow = S, ncol = T)
16 weights <- matrix(0, nrow = S, ncol = T)
17 weightsSt <- matrix(0, nrow = S, ncol = T)
18 # Initialization
19 particles[, 1] <- rnorm(S, mean = 0, sd = sigma_w) # Sample
   initial particles
20 weights[, 1] <- dnorm(y_obs[1], mean = phi*particles[, 1],
   sd = sigma_mu) # Importance weights
21 weightsSt[, 1] <- weights[, 1] / sum(weights[, 1]) # Standardized weights
22 # Sequential updating
23 for (t in 2:T) {
24   # Propagate particles
25   particles[, t] <- rnorm(S, mean = particles[, t-1], sd =
   sigma_w)
26   # Compute weights
27   weights[, t] <- weightsSt[, t-1] * dnorm(y_obs[t], mean =
   phi*particles[, t], sd = sigma_mu) # Recursive weight
   update
28   weightsSt[, t] <- weights[, t] / sum(weights[, t]) # Normalize weights
29 }
30 # Estimate the states (weighted mean)
31 FilterDist <- colSums(particles * weightsSt)
32 SDFilterDist <- (colSums(particles^2 * weightsSt) -
   FilterDist^2)^0.5
33 library(dplyr); library(ggplot2); library(latex2exp)
34 ggplot2::theme_set(theme_bw())
35 df <- tibble(t = 1:T, mean = FilterDist, lower = FilterDist -
   2*SDFilterDist, upper = FilterDist + 2*SDFilterDist,
   theta_true = theta_true)
36 # Function to plot
37 plot_filtering_estimates <- function(df) {
38   p <- ggplot(data = df, aes(x = t)) + geom_ribbon(aes(ymin =
   lower, ymax = upper), alpha = 1, fill = "lightblue") +
   geom_line(aes(y = theta_true), colour = "black", alpha =
   1, linewidth = 0.5) + geom_line(aes(y = mean), colour =
   "blue", linewidth = 0.5) + ylab(TeX("\theta_{\{t\}}"))
   + xlab("Time")
39   print(p)
40 }
41 plot_filtering_estimates(df)

```

Figure 4.8 shows the trajectory of the true state vector (black line), the posterior mean (blue line), and the area defined by $\pm 2\hat{\sigma}_\theta$ (light blue shaded area).

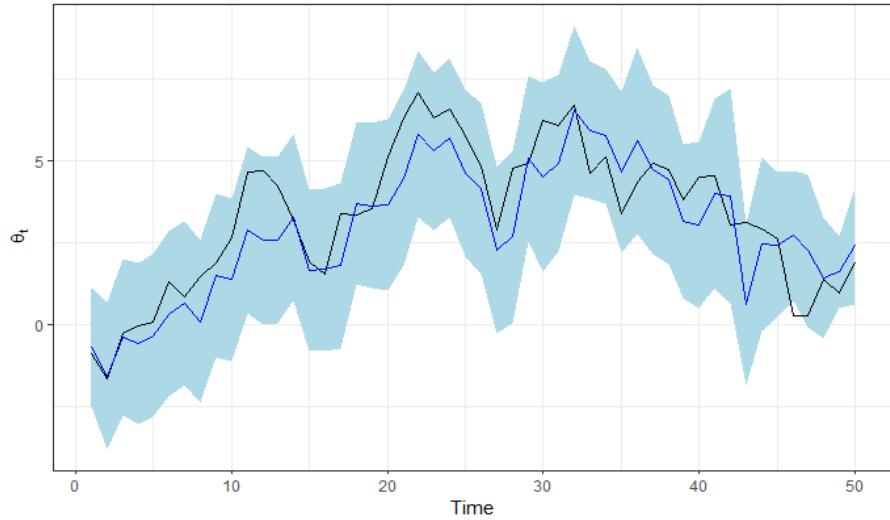


FIGURE 4.8

State in linear state-space model: True and mean estimate using sequential importance sampling, $T = 50$.

Sequential importance sampling is effective for sampling from the posterior distribution in the short term. However, it is important to note that SIS is a particular case of IS and, consequently, inherits the drawbacks of importance sampling. In particular, the variance of the weights increases exponentially with t [123]. This implies that, as t increases, the importance weights tend to degenerate in the long run; that is, all probability mass concentrates on a few weights, a phenomenon known as sample impoverishment or weight degeneracy. This is because it is impossible to accurately represent a distribution on a space of arbitrarily high dimension with a sample of fixed, finite size. This phenomenon can be observed, for instance, in the dynamic linear model example, where the highest standardized weight at $t = 50$ is 53%, and 7 out of 50,000 particles account for 87% of the total probability.

Given that, in practice, we are often interested in lower-dimensional marginal distributions, ideas from sampling/importance resampling can be employed. This strategy avoids the accumulation of errors due to resetting the system, although resampling introduces some additional Monte Carlo variation. [90] proposed the *Bootstrap filter*, where, at each time step, resampling is performed by drawing S particles from the current set using the standardized weights as probabilities of selection. This ensures that particles with small weights have a low probability of being selected. After resampling, the stan-

dardized weights are set equal to $1/S$. Note that the *Bootstrap filter* involves multiple iterations of the SIR algorithm, which implies that the resampled trajectories are no longer independent. This multinomial resampling provides an unbiased approximation to the posterior distribution obtained by SIS [60].

Algorithm A5 shows how to perform the *particle filter*. We set $w_t^{(s)} := w_t(\theta_{0:t}^{(s)})$ to simplify notation. [60].

Algorithm A5 The *particle filter* algorithm

```

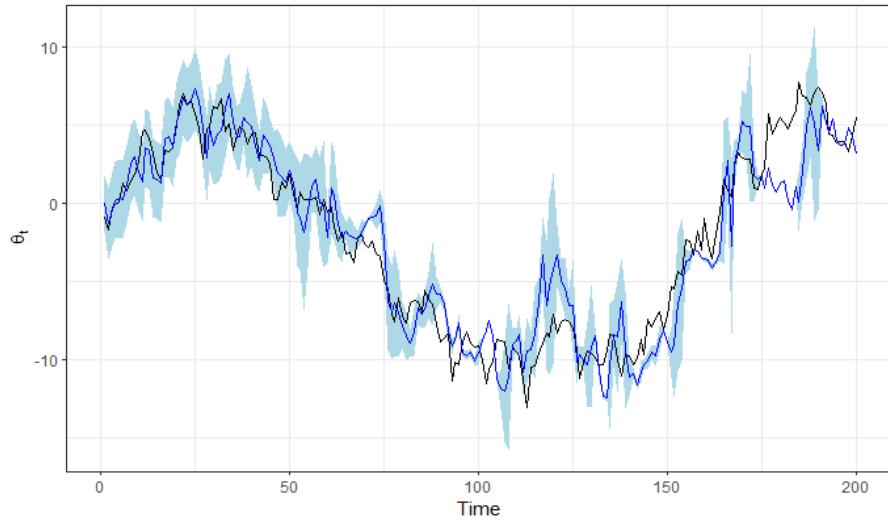
1: for  $s = 1, \dots, S$  do
2:   Sample  $\theta_0^{(s)}$  from  $q(\theta_0|y_0)$ 
3:   Calculate the importance weights  $w_0^{(s)} \propto \frac{p(y_0|\theta_0^{(s)})\pi(\theta_0^{(s)})}{q(\theta_0^{(s)}|y_0)}$ 
4: end for
5: Standardize the weights  $w_0^{*(s)} = \frac{w_0^{(s)}}{\sum_{h=1}^S w_0^{(h)}}, s = 1, 2, \dots, S$ 
6: Select  $S$  particles from  $\{\theta_0^{(s)}, w_0^{*(s)}\}$  to obtain  $\{\theta_0^{r(s)}, 1/S\}$ 
7: for  $t = 1, \dots, T$  do
8:   for  $s = 1, \dots, S$  do
9:     Draw particles  $\theta_t^{(s)}$  from  $q_t(\theta_t|\theta_{t-1}, y_t)$ 
10:    Set  $\theta_{1:t}^{(s)} \leftarrow (\theta_{t-1}^{r(s)}, \theta_t^{(s)})$ 
11:    Compute the weights  $\alpha_t^{(s)} = \frac{p(y_t|\theta_t^{(s)})\pi(\theta_t^{(s)}|\theta_{t-1}^{(s)})}{q(\theta_t^{(s)}|\theta_{t-1}^{(s)}, y_t)}$ 
12:   end for
13:   Standardize the weights  $w_t^{*(s)} = \frac{w_t^{(s)}}{\sum_{h=1}^S w_t^{(h)}}, s = 1, 2, \dots, S$ 
14:   Select  $S$  particles from  $\{\theta_{1:t}^{(s)}, w_t^{*(s)}\}$  to obtain  $\{\theta_{1:t}^{r(s)}, 1/S\}$ 
15: end for

```

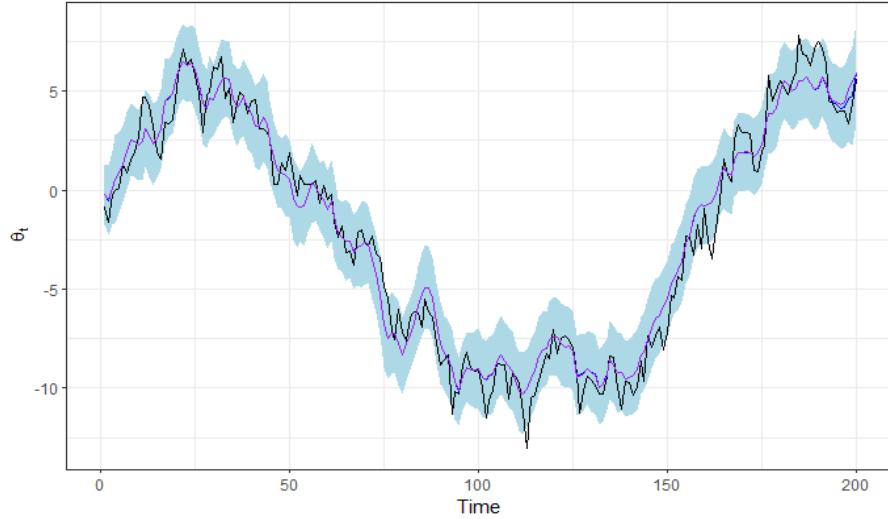
Example: Dynamic linear model continues

Let's apply the SIS algorithm to the dynamic linear model with a sample size of 200. Figure 4.9 illustrates the performance of sequential importance sampling. We observe that the algorithm's performance deteriorates as t increases. This is due to particle degeneration; at $t = 200$, a single particle holds a weight close to 100%.

Let's perform particle filtering in this example. The following code illustrate the procedure. Figure 4.10 show the performance of particle filtering in this example. There is the true state vector (black line), the means based on $\{\theta_{1:t}^{(s)}, w_t^{*(s)}\}$ (blue line) and $\{\theta_{1:t}^{r(s)}, 1/S\}$ (purple line), and the area defined by $\pm 2\hat{\sigma}_\theta$ based on the former (light blue shaded area). Note that the particle filtering algorithm has better performance than the SIS algorithm.

**FIGURE 4.9**

State in linear state-space model: True and mean estimate using sequential importance sampling, $T = 200$.

**FIGURE 4.10**

State in linear state-space model: True and mean estimate using particle filtering, $T = 200$.

R code. Particle filtering: Dynamic linear model

```

1 rm(list = ls()); set.seed(010101)
2 S <- 50000 # Number of particles
3 sigma_w <- 1; sigma_mu <- 1 # # State and observation
   noises
4 phi <- 0.5 # Coefficient in observation equation
5 T <- 200 # Sample size
6 # Simulate true states and observations
7 theta_true <- numeric(T); y_obs <- numeric(T)
8 theta_true[1] <- rnorm(1, mean = 0, sd = sigma_w)
9 for (t in 2:T) {
10   theta_true[t] <- rnorm(1, mean = theta_true[t-1], sd =
      sigma_w)
11 }
12 y_obs <- rnorm(T, mean = phi*theta_true, sd = sigma_mu)
13 # Particle filtering
14 particles <- matrix(0, nrow = S, ncol = T) # Store
   particles
15 particlesT <- matrix(0, nrow = S, ncol = T) # Store
   resampling particles
16 weights <- matrix(0, nrow = S, ncol = T) # Store weights
17 weightsSt <- matrix(0, nrow = S, ncol = T) # Store
   standardized weights
18 weightsSTT <- matrix(1/S, nrow = S, ncol = T) # Store
   standardized weights
19 logalphas <- matrix(0, nrow = S, ncol = T) # Store log
   incremental weights
20 particles[, 1] <- rnorm(S, mean = 0, sd = sigma_w)
21 weights[, 1] <- dnorm(y_obs[1], mean = phi*particles[, 1],
   sd = sigma_mu) # Importance weights
22 weightsSt[, 1] <- weights[, 1] / sum(weights[, 1]) #
   Normalize weights
23 ind <- sample(1:S, size = S, replace = TRUE, prob =
   weightsSt[, 1]) # Resample
24 particles[, 1] <- particles[ind, ] # Resampled particles
25 particlesT[, 1] <- particles[, 1] # Resampled particles
26 # Sequential updating
27 pb <- winProgressBar(title = "progress bar", min = 0, max =
   T, width = 300)
28 for (t in 2:T) {
29   particles[, t] <- rnorm(S, mean = particles[, t-1], sd =
      sigma_w)
30   logalphas[, t] <- dnorm(y_obs[t], mean = phi*particles[, t],
      sd = sigma_mu, log = TRUE)
31   weights[, t] <- exp(logalphas[, t])
32   weightsSt[, t] <- weights[, t] / sum(weights[, t])
33   if(t < T){
34     ind <- sample(1:S, size = S, replace = TRUE, prob =
      weightsSt[, t])
35     particles[, 1:t] <- particles[ind, 1:t]
36   }else{
37     ind <- sample(1:S, size = S, replace = TRUE, prob =
      weightsSt[, t])
38     particlesT[, 1:t] <- particles[ind, 1:t]
39   }
40   setWinProgressBar(pb, t, title=paste( round(t/T*100, 0), "
      % done"))
41 }
42 close(pb)

```

R code. Particle filtering: Dynamic linear model

```

1 FilterDist <- colSums(particles * weightsSt)
2 SDFilterDist <- (colSums(particles^2 * weightsSt) -
   FilterDist^2)^0.5
3 FilterDistT <- colSums(particlesT * weightsSTT)
4 SDFilterDistT <- (colSums(particlesT^2 * weightsSTT) -
   FilterDistT^2)^0.5
5 MargLik <- colMeans(weights)
6 plot(MargLik, type = "l")
7 library(dplyr)
8 library(ggplot2)
9 require(latex2exp)
10 ggplot2::theme_set(theme_bw())
11 df <- tibble(t = 1:T, mean = FilterDist, lower = FilterDist -
   - 2*SDFilterDist, upper = FilterDist + 2*SDFilterDist,
   meanT = FilterDistT, lowerT = FilterDistT - 2*
   SDFilterDistT,
12 upperT = FilterDistT + 2*SDFilterDistT, x_true = theta_true)
13 plot_filtering_estimates <- function(df) {
14   p <- ggplot(data = df, aes(x = t)) + geom_ribbon(aes(ymin =
   lower, ymax = upper), alpha = 1, fill = "lightblue") +
   geom_line(aes(y = x_true), colour = "black", alpha = 1,
   linewidth = 0.5) + geom_line(aes(y = mean), colour =
   "blue", linewidth = 0.5) +
   geom_line(aes(y = meanT), colour = "purple", linewidth =
   0.5) + ylab(TeX("\$\theta_{\cdot:t}\$")) + xlab("Time")
15   print(p)
16 }
17 }
18 plot_filtering_estimates(df)

```

Algorithm A5 performs resampling at every time step. However, it is common to perform resampling only when the effective sample size of the particles ($ESS = (\sum_{s=1}^S (w_t^{*(s)})^2)^{-1}$) falls below a specific threshold, such as 50% of the initial number of particles. Note that when $w_t^{*(s)} = 1/S$, the effective sample size is S , the total number of particles. Additionally, we should use $\{\theta_{1:t}^{(s)}, w_t^{*(s)}\}$ to estimate the posterior distribution, as it results in lower Monte Carlo error compared to calculations based on $\{\theta_{1:t}^{r(s)}, 1/S\}$ [27]. Finally, an estimate of the marginal likelihood can be obtained using

$$\hat{p}(y_t) = \frac{1}{S} \sum_{s=1}^S w_t^{(s)}.$$

Particle filtering offers several advantages, such as being quick and easy to implement, its modularity—allowing one to simply adjust the expressions for

the importance distribution and weights when changing the problem—and its suitability for parallel algorithms. Moreover, it enables straightforward sequential inference for very complex models.

However, there are also disadvantages. The resampling step introduces extra Monte Carlo variability. Using the state transition (prior) density as the importance distribution often leads to poor performance, manifested in a lack of robustness with respect to the observed sequence. For instance, performance deteriorates when outliers occur in the data or when the variance of the observation noise is small. Furthermore, the procedure is not well suited for sampling from $\pi(\boldsymbol{\theta}_{0:t}|y_{1:t})$ because most particles originate from the same ancestor.

Alternative resampling approaches, such as residual resampling [140] and systematic resampling [30], preserve unbiasedness while reducing variance. Additionally, auxiliary particle filtering [28] can help decrease Monte Carlo variability.

Lastly, estimating fixed parameters such as σ_w^2 , σ_μ^2 , and ϕ in the dynamic linear model poses a challenge. Various methods exist to address this issue; see [115, 116] for a comprehensive review and [5] for a seminal work in *particle MCMC* methods.

4.4 Convergence diagnostics

MCMC methods rely on *irreducibility*, *positive recurrence*, and *aperiodicity*, ensuring that after a sufficient burn-in period, the posterior draws are sampled from the invariant stationary posterior distribution. In this section, we present diagnostics to assess this property. First, we calculate the numerical standard error associated with the MCMC algorithm. Next, we review the effective number of simulation draws and some convergence tests. Finally, we examine potential errors in the posterior simulator.

4.4.1 Numerical standard error

Many times, the goal in Bayesian inference is to obtain a set of independent draws $\boldsymbol{\theta}^{(s)}$, $s = 1, 2, \dots, S$, from the posterior distribution, such that a measure of interest can be estimated with reasonable precision. In particular, we approximate (4.1) using (4.2). By the central limit theorem, we know that

$$\frac{\bar{h}(\boldsymbol{\theta})_S - \mathbb{E}_\pi[h(\boldsymbol{\theta})]}{\sigma_h(\boldsymbol{\theta})/\sqrt{S}} \xrightarrow{d} N(0, 1), \quad (4.4)$$

where $\sigma_h^2(\boldsymbol{\theta})$ is the variance of $h(\boldsymbol{\theta})$.

If we have independent draws, we can estimate $\sigma_h^2(\boldsymbol{\theta})$ using the posterior

draws as follows:

$$\hat{\sigma}_{Sh}^2(\boldsymbol{\theta}) = \frac{1}{S} \sum_{s=1}^S [h(\boldsymbol{\theta}^{(s)})]^2 - [\bar{h}(\boldsymbol{\theta})_S]^2.$$

However, if there are dependent draws, we have

$$\hat{\sigma}_{Sh}^{2*}(\boldsymbol{\theta}) = \frac{1}{S} \left\{ \sum_{s=1}^S [h(\boldsymbol{\theta}^{(s)}) - \bar{h}(\boldsymbol{\theta})_S]^2 + 2 \sum_{l=k+1}^K (h(\boldsymbol{\theta}^{(l)}) - \bar{h}(\boldsymbol{\theta})) (h(\boldsymbol{\theta}^{(l-k)}) - \bar{h}(\boldsymbol{\theta})) \right\}.$$

The *numerical standard error* is given by $\sigma_n(\boldsymbol{\theta})/\sqrt{S}$ and serves as a measure of the approximation error in the Monte Carlo integration. Note that this error can be decreased by increasing S . For instance, $S = 1000$ implies an error proportional to 3.2%, while $S = 10000$ reduces the error to approximately 1%.

4.4.2 Effective Number of Simulation Draws

MCMC posterior draws are not independent; therefore, the effective sample size of the posterior chains is not equal to S . To assess the effective sample size of the posterior draws, we use the following measure:

$$S_{\text{ef}} = \frac{S}{1 + 2 \sum_{k=1}^{\infty} \rho_k(h)},$$

where $\rho_k(h)$ is the autocorrelation of the sequence $h(\boldsymbol{\theta})$ at lag k .

The sample counterpart of this expression is:

$$\hat{S}_{\text{ef}} = \frac{S}{1 + 2 \sum_{k=1}^K \hat{\rho}_k(h)},$$

where

$$\hat{\rho}_k(h) = \frac{\sum_{l=k+1}^K (h(\boldsymbol{\theta}^{(l)}) - \bar{h}(\boldsymbol{\theta})) (h(\boldsymbol{\theta}^{(l-k)}) - \bar{h}(\boldsymbol{\theta}))}{\sum_{s=1}^K (h(\boldsymbol{\theta}^{(s)}) - \bar{h}(\boldsymbol{\theta}))^2}.$$

If $\hat{\rho}_k(h)$ declines to zero slowly as k increases, it indicates significant memory in the draws. Consequently, the effective sample size of the posterior draws is small, and it becomes necessary to either decrease the autocorrelation or increase the number of posterior draws.

Note that

$$\hat{\sigma}_{Sh}^{2*}(\boldsymbol{\theta}) = \hat{\sigma}_{Sh}^2(\boldsymbol{\theta}) (1 + 2 \sum_{k=1}^K \hat{\rho}_k(h)),$$

where $\hat{\sigma}_{Sh}^{2*}(\boldsymbol{\theta})$ and $\hat{\sigma}_{Sh}^2$ are the simulation variances using dependent and independent draws, and $\hat{\kappa}(h) = (1 + 2 \sum_{k=1}^K \hat{\rho}_k(h))$ is called the *inefficiency factor*, which represents the inflation of the simulation variance due to autocorrelation in the draws. Values near one indicate draws with little correlation.

4.4.3 Tests of convergence

Regarding convergence issues, there are several diagnostics to assess the adequacy of the posterior chains. In particular, graphical approaches such as trace plots and autocorrelation plots are widely used. Trace plots display the sampled values of a parameter (or multiple parameters) as a function of the iteration number, while autocorrelation plots graphically represent $\hat{\rho}_k$. The latter shows how correlated the values of $\boldsymbol{\theta}$, or functions of $\boldsymbol{\theta}$, are at different lags. Trace plots should fluctuate around a stable mean, exploring the entire parameter space without becoming stuck in any particular region. Autocorrelation plots, on the other hand, should exhibit values close to zero or diminish quickly as the lag increases.

Additionally, Geweke's test [82] provides a simple two-sample test of means. If the mean of the first window (10% of the chain) is not significantly different from the mean of the second window (50% of the chain), we fail to reject the null hypothesis that the two segments of the chain are drawn from the same stationary distribution.

The Raftery and Lewis test [173] is designed to calculate the approximate number of iterations (S), burn-in (b), and thinning parameter (d) required to estimate $p[H(\boldsymbol{\theta}) \leq h]$, where $H(\boldsymbol{\theta}) : \mathcal{R}^k \rightarrow \mathcal{R}$. This calculation is based on a specific quantile of interest (q), precision (r), and probability (p). The diagnostic is based on the dependence factor, $I = \frac{S+b}{S_{\text{Min}}}$, where $S_{\text{Min}} = \Phi^{-1} \left(\frac{1}{2}(p+1) \right)^2 q(1-q)/r^2$, and $\Phi(\cdot)$ is the standard normal cumulative distribution function. Values of I much greater than 5 indicate a high level of dependence.

Heidelberger and Welch's test [97] uses a Cramér-von Mises statistic to test the null hypothesis that the sampled values, $\boldsymbol{\theta}^{(s)}$, are drawn from a stationary distribution. The statistic is given by

$$\text{CVM}(B_S) = \int_0^1 B_S(t)^2 dt,$$

where $B_S(t) = \frac{S_{[St]} - [St]\bar{\boldsymbol{\theta}}^S}{\sqrt{Sp(0)}}$, $S_S = \sum_{s=1}^S \boldsymbol{\theta}^{(s)}$, $\bar{\boldsymbol{\theta}}^S = S_S/S$, and $p(0)$ is the spectral density at 0, with $0 \leq t \leq 1$. Under the null hypothesis, $B_S(t)$ converges in distribution to a Brownian bridge.

This test is recursively applied until either the null hypothesis is not rejected, or $s = 50\%$ of the chain has been discarded. Subsequently, the half-width test calculates a 95% confidence interval for the mean using the portion of the chain that passed the stationarity test. If the ratio of the half-width of this interval to the mean is less than 0.1, the test is considered passed. This indicates no evidence to reject the null hypothesis that the estimated mean is accurate and stable.

There are other diagnostics in Bayesian inference that we do not mention here, such as the Gelman and Rubin test [75]. This is because we focus on the available diagnostics in our Graphical User Interface (GUI).

4.4.4 Checking for errors in the posterior simulator

In this book, we provide basic code templates to get posterior draws for performing inference under the Bayesian framework when there is no closed-form solution. We are prone to making mistakes and greatly appreciate your feedback to help improve our code and identify any other potential issues. One way to check if our code works correctly is to perform simulations where the population parameters are known. If the code is functioning properly, the posterior estimates should converge to these values as the sample size increases due to the Bayesian consistency. This is an informal approach to identifying potential mistakes.

[84] offers a more formal method for code validation. The starting point is the joint density $p(\mathbf{y}, \boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$ and a test function $h(\mathbf{y}, \boldsymbol{\theta})$ such that $\sigma_h^2 = \text{Var}[h(\mathbf{y}, \boldsymbol{\theta})] < \infty$.

Assume that there is a *marginal-conditional simulator* for the joint distribution of \mathbf{y} and $\boldsymbol{\theta}$:

$$\begin{aligned}\boldsymbol{\theta}^{(s)} &\sim \pi(\boldsymbol{\theta}) \\ \mathbf{y}^{(s)} &\sim p(\mathbf{y}|\boldsymbol{\theta}^{(s)}) \\ h^{(s)} &= h(\mathbf{y}^{(s)}, \boldsymbol{\theta}^{(s)}).\end{aligned}$$

The sequence $\{\mathbf{y}^{(s)}, \boldsymbol{\theta}^{(s)}\}$ is i.i.d., \bar{h}_S converges almost surely to $\mathbb{E}[h(\mathbf{y}, \boldsymbol{\theta})]$, and there is convergence in distribution when \bar{h}_S is well standardized (see Equation 4.4) and $\hat{\sigma}_{Sh}^2(\boldsymbol{\theta})$ converges to $\sigma_h^2(\boldsymbol{\theta})$ almost surely.

A posterior simulator produces draws $\boldsymbol{\theta}^{(s)}$ given a particular realization \mathbf{y}_{Obs} , using the transition density $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s-1)}, \mathbf{y}_{\text{Obs}})$. Thus, a *successive-conditional simulator* consists of an initial draw $\boldsymbol{\theta}^{(0)}$ from $\pi(\boldsymbol{\theta})$ followed by:

$$\begin{aligned}\mathbf{y}^{(l)} &\sim p(\mathbf{y}|\boldsymbol{\theta}^{(l-1)}) \\ \boldsymbol{\theta}^{(l)} &\sim q(\boldsymbol{\theta}|\mathbf{y}^{(l)}, \boldsymbol{\theta}^{(l-1)}) \\ h^{(l)} &= h(\mathbf{y}^{(l)}, \boldsymbol{\theta}^{(l)}),\end{aligned}$$

where $\bar{h}_L = L^{-1} \sum_{l=1}^L h(\mathbf{y}^{(l)}, \boldsymbol{\theta}^{(l)})$ converges almost surely to $\mathbb{E}[h(\mathbf{y}, \boldsymbol{\theta})]$, and there is convergence in distribution when \bar{h}_L is well standardized, and $\hat{\sigma}_{Lh}^{*2}(\boldsymbol{\theta})$ converges to $\sigma_h^2(\boldsymbol{\theta})$ almost surely, for $l = 1, 2, \dots, L$. Thus,

$$\frac{\bar{h}_S - \bar{h}_L}{(S^{-1}\hat{\sigma}_{Sh}^2(\boldsymbol{\theta}) + L^{-1}\hat{\sigma}_{Lh}^{*2}(\boldsymbol{\theta}))^{1/2}} \xrightarrow{d} N(0, 1).$$

Thus, we can test $H_0: \bar{h}_S - \bar{h}_L = 0$ versus $H_1: \bar{h}_S - \bar{h}_L \neq 0$. Rejection of the null indicates potential errors in implementing the posterior simulator.

Example: Mining disaster change point continues

Let's revisit the mining disaster change point example from subsection 4.1.1 and examine some convergence diagnostics for the posterior draws of the

rate of disasters after the change point (λ_2). The following code demonstrates how to perform these diagnostics using the **R** package *coda*. For clarity and replicability of the results, we present the Gibbs sampler again.

Figures 4.11 and 4.12 show the trace and autocorrelation plots. We observe that the posterior draws of λ_2 appear stationary around their mean, and the autocorrelation decreases rapidly to zero.

The mean and standard deviation of the rate after the change point are 0.92 and 0.12, respectively. The naive and time series standard errors are 0.0008245 and 0.0008945, respectively. The naive standard error assumes iid posterior draws, whereas the time series standard error accounts for autocorrelation. Both standard errors are very similar, indicating a low level of autocorrelation, which is consistent with the results shown in Figure 4.12. The effective sample size of the posterior draws is 16,991, while the total number of posterior draws is 20,000 after a burn-in period of 1,000.

The Geweke test statistic is 1.43, which implies no statistical evidence to reject the null hypothesis of equal means in the two segments of the posterior draws. The Raftery and Lewis test yields a dependence factor near 1, indicating a low level of dependence. The Heidelberger and Welch test does not reject the null hypothesis of stationarity for the posterior draws and also confirms that the mean is accurate and stable.

In summary, all posterior diagnostics indicate that the posterior draws originate from an invariant stationary distribution.

The second part of the code implements the proposal by [84] to assess the reliability of the posterior simulator. The parameter vector is defined as $\boldsymbol{\theta} = [\lambda_1 \ \lambda_2 \ H]$, and the first moments of these parameters are used as test functions. We fail to reject the null hypothesis of equal means across the three test functions, indicating that the posterior simulator is functioning correctly.

To evaluate the effectiveness of the test, we run the *marginal-conditional simulator* with prior parameters $\alpha_{l0} = 0.5$ and $\beta_{l0} = 1$, $l = 1, 2$. In contrast, for the *successive-conditional simulator*, we use prior parameters $\alpha_{l0} = 1$ and $\beta_{l0} = 0.5$, $l = 1, 2$. In this case, we reject the null hypothesis in two out of three test functions, suggesting that the test performs well in this example.

R code. Posterior diagnostics: The mining disaster changepoint

```

1 rm(list = ls())
2 set.seed(010101)
3 dataset<-read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/
  MiningDataCarlin.csv",header=T)
4 attach(dataset)
5 str(dataset)
6 a10 <- 0.5; a20 <- 0.5
7 b10 <- 1; b20 <- 1
8 y <- Count
9 sumy <- sum(Count); T <- length(Count)
10 theta1 <- NULL; theta2 <- NULL
11 kk <- NULL; H <- 60
12 MCMC <- 20000; burnin <- 1000; S <- MCMC + burnin; keep <- (
  burnin+1):S
13 pb <- winProgressBar(title = "progress bar", min = 0, max =
  S, width = 300)
14 for(s in 1:S){
15   a1 <- a10 + sum(y[1:H])
16   b1 <- b10+H
17   theta11 <- rgamma(1,a1,b1)
18   theta1 <- c(theta1,theta11)
19   a2 <- a20 + sum(y[(1+H):T])
20   b2 <- b20 + T-H
21   theta22 <- rgamma(1,a2,b2)
22   theta2 <- c(theta2,theta22)
23   pp<-NULL
24   for(l in 1:T){
25     p <- exp(1*(theta22-theta11))*(theta11/theta22)^(sum(y
      [1:l]))
26     pp <- c(pp,p)
27   }
28   prob <- pp/sum(pp)
29   H <- sample(1:T,1,prob=prob)
30   kk <- c(kk,H)
31   setWinProgressBar(pb, s, title=paste( round(s/S*100, 0), "%"
    done"))
32 }
33 close(pb)
34 library(coda); library(latex2exp)
35 theta1Post <- mcmc(theta1[keep]); summary(theta1Post)
36 HPost <- mcmc(kk); summary(HPost)
37 theta2Post <- mcmc(theta2[keep]); summary(theta2Post)
38 plot(theta2Post, density = FALSE, main = "Trace plot", ylab
  = TeX("$\\theta_{\{2\}}$"))
39 autocorr.plot(theta2Post, main = "Autocorrelation plot")
40 raftery.diag(theta2Post, q = 0.025, r = 0.005, s = 0.95)
41 geweke.diag(theta2Post, frac1 = 0.1, frac2 = 0.5)
42 heidel.diag(theta2Post, eps = 0.1, pvalue = 0.05)
43 effectiveSize(theta2Post)

```

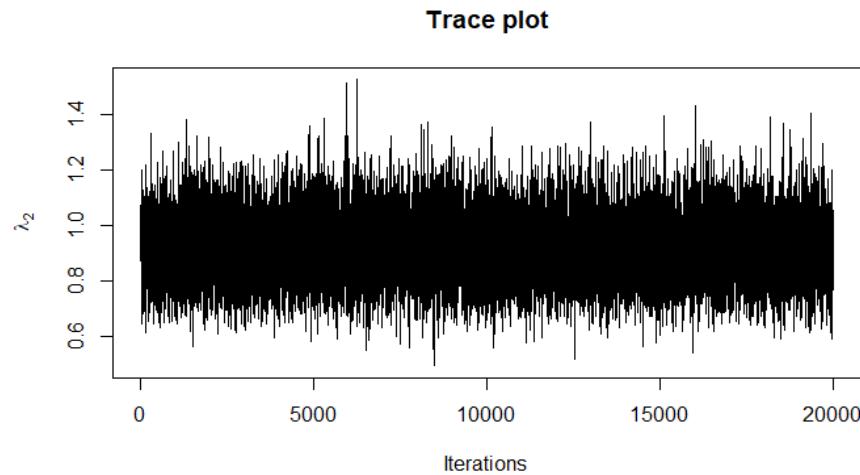


FIGURE 4.11
Mining disaster change point: Trace plot of λ_2 .

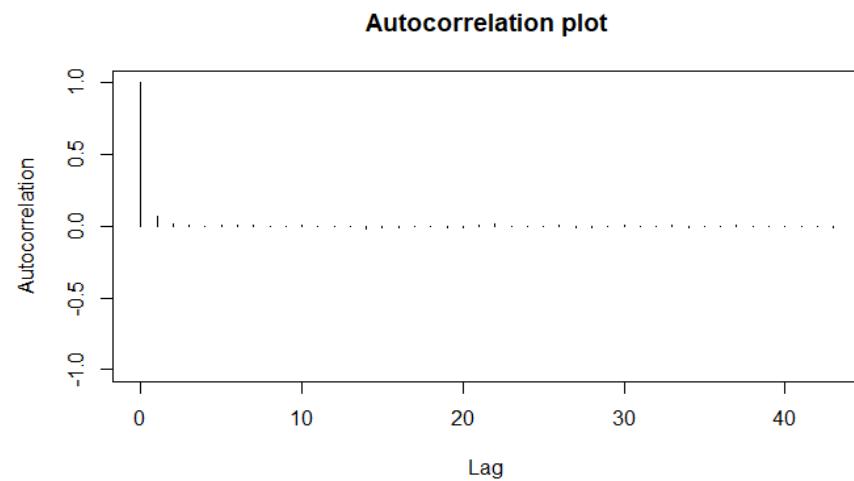


FIGURE 4.12
Mining disaster change point: Autocorrelation plot of λ_2 .

R code. Posterior diagnostics: The mining disaster changepoint

```

1 # Marginal-conditional simulator
2 Theta1Prior <- rgamma(MCMC,a10,b10); Theta2Prior <- rgamma(
  MCMC,a20,b20)
3 kPrior <- sample(1:T, MCMC, replace = TRUE, prob = rep(1/T,T))
4 ytPrior <- function(par){
5   y1t <- rpois(par[3], par[1])
6   if(par[3] == T){y2t <- NULL
7 }else{y2t <- rpois(T-par[3], par[2])
8 }
9 yt <- c(y1t, y2t)
10 return(yt)
11 }
12 pars1 <- cbind(Theta1Prior, Theta2Prior, kPrior); Yt <-
  apply(pars1, 1, ytPrior)
13 parsMcmc1 <- coda::mcmc(pars1); Summ1 <- summary(parsMcmc1)
14 # Successive-conditional simulator
15 SucConSim <- function(a10, b10, a20, b20, par){
16   y <- ytPrior(par)
17   theta1 <- par[1]; theta2 <- par[2]; H <- par[3]
18   a1 <- a10 + sum(y[1:H]); b1 <- b10+H
19   theta11 <- rgamma(1,a1,b1)
20   if(H == T){ a2 <- a20
21 }else{ a2 <- a20 + sum(y[(1+H):T])
22 }
23   b2 <- b20 + T-H; theta22 <- rgamma(1,a2,b2); pp<-NULL
24   for(l in 1:T){
25     p <- l*(theta22-theta11) + (sum(y[1:l]))*log(theta11/
      theta22)
26     pp <- c(pp,p)
27   }
28   pps <- exp(pp - max(pp)); prob <- pps/sum(pps)
29   H <- sample(1:T, 1, prob=prob)
30   parNew <- list(y = y, pars = c(theta11, theta22, H))
31   return(parNew)
32 }
33 a10 <- 0.5; b10 <- 1; a20 <- 0.5; b20 <- 1
34 # a10 <- 1; b10 <- 0.5; a20 <- 1; b20 <- 0.5
35 par1 <- rgamma(1,a10,b10); par2 <- rgamma(1,a20,b20)
36 par3 <- sample(1:T, 1, replace = TRUE, prob = rep(1/T,T))
37 pars2 <- matrix(NA, MCMC, 3); pars2[1,] <- c(par1, par2,
  par3)
38 for(s in 2:MCMC){
39   Res <- SucConSim(a10 = a10, b10 = b10, a20 = a20, b20 =
    b20, par = pars2[s-1,])
40   pars2[s, ] <- Res$pars
41 }
42 parsMcmc2 <- coda::mcmc(pars2); Summ2 <- summary(parsMcmc2)
43 TestGeweke <- function(j){
44   Test <- (Summ1[["statistics"]][j,1] - Summ2[["statistics"
    ]][j,1])/((Summ1[["statistics"]][j,4]+Summ2[["statistics"
    ]][j,4])^0.5
45   Reject <- abs(Test) > qnorm(0.975)
46   return(list(Test = Test, Reject = Reject))
47 }
48 TestGeweke(1); TestGeweke(2); TestGeweke(3)

```

4.5 Summary

In this chapter, we present the most popular methods for obtaining posterior draws when the posterior distribution does not have a standard closed-form solution. In particular, Markov chain Monte Carlo (MCMC) methods, such as Gibbs sampling and the Metropolis-Hastings algorithm, are the most commonly used approaches in this book. However, Hamiltonian Monte Carlo is gaining particular relevance in high-dimensional problems, while particle filtering (Sequential Monte Carlo) is widely applied in time series models.

Each problem requires careful consideration to determine the most appropriate method, and in many cases, a combination of methods is necessary. For instance, estimating fixed parameters in *state-space* models typically requires MCMC methods, while recursion of the state vector requires particle filtering. Additionally, convergence diagnostics are crucial because MCMC methods rely on technical assumptions that must be verified.

4.6 Exercises

1. Example: The normal model with independent priors

Let's recap the math test exercise in Chapter 3, this time assuming independent priors. Specifically, let $Y_i \sim N(\mu, \sigma^2)$, where $\mu \sim N(\mu_0, \sigma_0^2)$ and $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$. The sample size is 50, and the mean and standard deviation of the math scores are 102 and 10, respectively. We set $\mu_0 = 100$, $\sigma_0^2 = 100$, and $\alpha_0 = \delta_0 = 0.001$.

- Find the posterior distribution of μ and σ^2 .
 - Program a Gibbs sampler algorithm and plot the histogram of the posterior draws of μ
2. Show that the Gibbs sampler is a particular case of the Metropolis-Hastings where the acceptance probability is equal to 1.
 3. Implement a Metropolis-Hastings to sample from the Cauchy distribution, $C(0, 1)$, using as proposals a standard normal distribution and a Student's t distribution with 5 degrees of freedom.
 4. This exercise was proposed by Professor Hedibert Freitas Lopes, who cites [209] as a useful reference for an introduction to Hamiltonian Monte Carlo in **R** and the *hmclearn* package. The task is to obtain posterior draws using the Metropolis-Hastings and Hamiltonian Monte Carlo algorithms for the posterior distribution given

by

$$\pi(\theta_1, \theta_2 | \mathbf{y}) \propto \exp \left\{ -\frac{1}{2} (\theta_1^2 \theta_2^2 + \theta_1^2 + \theta_2^2 - 8\theta_1 - 8\theta_2) \right\}.$$

5. Ph.D. students sleeping hours continues

- (a) Use importance sampling based on a $U(0, 1)$ proposal to obtain draws of $\boldsymbol{\theta} | \mathbf{y} \sim B(16.55, 39.57)$ in the Ph.D. students' sleeping hours example in Chapter 3. Note that, based on Exercise 15 in Chapter 3, $\alpha_0 = 1.44$ and $\beta_0 = 2.57$.
- (b) Compute the marginal likelihood in this context (Bernoulli-Beta model) and compare it to the result obtained using the Gelfand-Dey method.

6. Example 4.1 in [90] is

$$\begin{aligned}\theta_t &= 0.5\theta_{t-1} + 25 \frac{\theta_{t-1}}{1 + \theta_{t-1}^2} + 8\cos(1.2t) + w_t \\ y_t &= \frac{\theta_t^2}{20} + \mu_t,\end{aligned}$$

where $\theta_0 \sim N(0, \sqrt{10})$, $w_t \sim \mathcal{N}(0, \sqrt{10})$ and $\mu_t \sim N(0, \sqrt{1})$.

- Perform sequential importance sampling in this example.
- Perform particle (Bootstrap) filtering in this example.
- Estimate the marginal likelihood in this example.

7. Ph.D. students sleeping hours continues

- Perform the diagnostics of Section 4.4 in this example
- Check if there are errors in the posterior simulator of the Metropolis-Hastings algorithm in this example.

— | | —

Part II

Regression models: A GUIDed toolkit



5

Graphical user interface

This chapter presents our graphical user interface (GUI) to carry out Bayesian regression analysis in a very friendly environment without any programming skills (drag and drop). Our GUI is based on an interactive web application using *shiny* [35], and packages like *MCMCpack* [146] and *bayesm* [188] from **R** software [167], and is designed for teaching and applied purposes at an introductory level. In the next chapters of the second part of this book we carry out some applications to highlight the potential of our GUI for applied researchers and practitioners.

5.1 Introduction

Our GUI allows performing inference using Bayesian regression analysis without requiring programming skills. The latter seems to be a significant impediment to increasing the use of the Bayesian framework [220, 117].

There are other available graphical user interfaces for carrying out Bayesian regression analysis. *ShinyStan* [205] is a very flexible open source program, but users are required to have some programming skills. *BugsXLA* [220] is open source, but less flexible. However, users do not need to have programming skills. *Bayesian regression: Nonparametric and parametric models* [117] is a very flexible and friendly GUI that is based on *MATLAB Compiler* for a 64-bit Windows computer. Its focus is on Bayesian nonparametric regressions, and it can be thought of for users who have mastered basic parametric models, such as the ones that we show in our GUI. There are also *MATLAB toolkit*, *Stata* and *BayES*, but these are not open sources.

We developed our GUI based on an interactive web application using *shiny* [35], and some libraries in **R** [166]. The specific libraries and commands that are used in our GUI can be seen in Table 14.1. It has ten univariate models, four multivariate, **time series models**, three hierarchical longitudinal, and seven Bayesian model averaging frameworks. In addition, it gives basic summaries and diagnostics of the posterior chains, as well as the posterior chains themselves, and different plots, such as trace, autocorrelation and densities.

In terms of its flexibility and possibilities, our GUI lies between *ShinyStan* and *BugsXLA*: users are not required to have any programming skills, but it

is not as advanced as [117]’s software. However, our GUI can be run in any operating system. Our GUI, which we call BEsmarter,¹ is freely available at <https://github.com/besmarter/BSTApp>; so users have access to all our code and datasets.

Simulated and applied datasets are in the folders **DataSim** (see Table 14.2 for details), and **DataApp** (see Table 14.3 for details) of our **GitHub** repository. The former folder also includes the files that were used to simulate different processes, so, the population parameters are available, and as a consequence these files can be used as a pedagogical tool to show some statistical properties of the inferential frameworks available in our GUI. The latter folder contains the datasets used in the applications of this second part of the book. Users should use these datasets as templates to structure their own datasets.

There are three ways to install our GUI. The easiest way, but that requires installation of **R**, and potentially a **R** code editor, is to type

R code. How to display our graphical user interface

```
1 shiny::runGitHub("besmarter/BSTApp", launch.browser = T)
```

in the **R** package console or any **R** code editor. We strongly recommend to type this directly, rather than copy and paste. This is due to a potential issue with the quotation mark.

The second option is to visit <https://posit.cloud/content/4328505>, log in or sign up for **Posit Cloud**, and access the project titled **GUIDed Bayesian regression app BSTApp**. In the right-bottom window, click on the **BSTApp-master** folder under **Files**, open the **app.R** file, and finally, click the **Run App** button. However, inactivity will cause the window to close.

The third approach, and our recommendation, is using a **Docker** image by running:

1. docker pull magralo95/besmartergui:latest
2. docker run -rm -p 3838:3838 magralo95/besmartergui

in your **Command Prompt**. This creates an isolated environment for our GUI, ensuring consistent performance across systems. Note that **Docker** must be installed to deploy our GUI this way.

After implementing any of the three ways to run our GUI, users can see

¹Bayesian econometrics: Simulations, models and applications to research, teaching and encoding with responsibility.

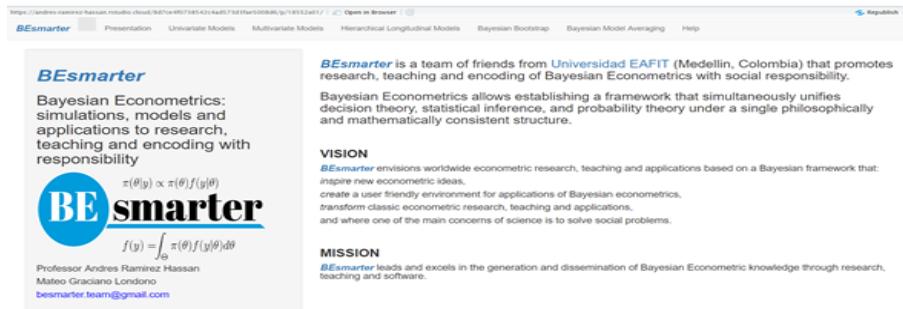


FIGURE 5.1
Display of our graphical user interface.

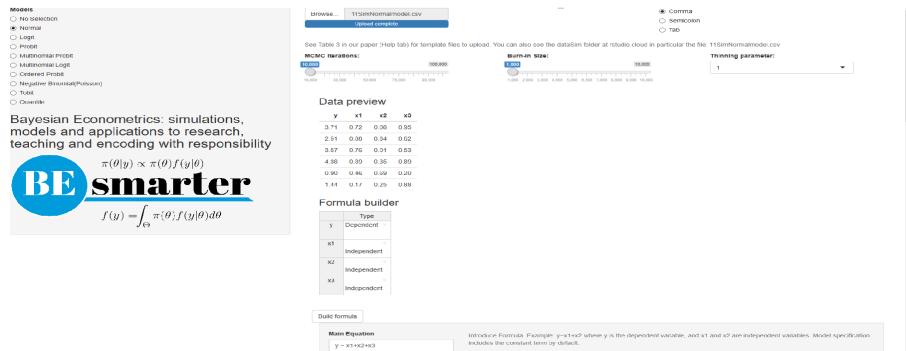


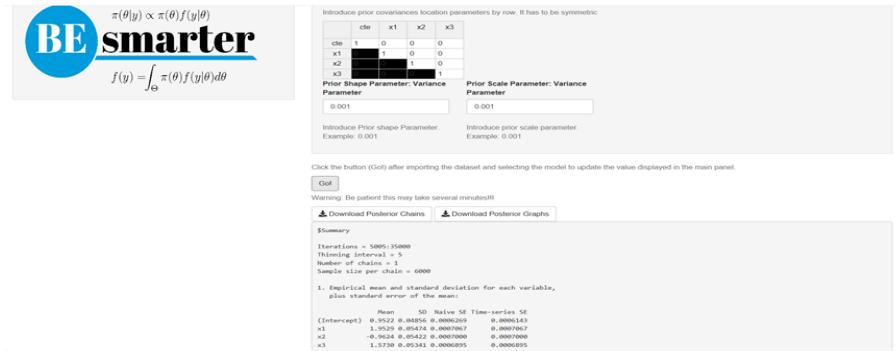
FIGURE 5.2
Univariate models: Specification.

a new window where a presentation of our research team is displayed (see Figure 5.1). In addition, the top panel in Figure 5.1 shows the class of models that can be estimated in our GUI.

5.2 Univariate models

After our GUI is deployed (see Figure 5.1), the user should select *Univariate Models* in the top panel. Then, the Figure 5.2 is displayed, and the user can see the radio button on the left hand side that shows the specific models inside this generic class. In particular, users can see that the normal model is selected from inside the class of univariate models.

Then, the right hand side panel displays a widget to upload the input

**FIGURE 5.3**

Univariate models: Results.

dataset, which should be a *csv* file with headers in the first row. Users also should select the kind of separator used in the input file: comma, semicolon, or tab (use the folders **DataSim** and **DataApp** for the input file templates). Once users upload the dataset, they can see a data preview. Range sliders help to set the number of iterations of the Markov chain Monte Carlo algorithm, the amount of burn-in, and the thinning parameter can be selected as well (see next chapters of this second part of the book for technical details). After this, users should specify the equation. This can be done with the formula builder, where users can select the dependent variable, and the independent variables, and then click on the *Build formula* tab. Users can see in the *Main Equation* space the formula expressed in the format used by **R** software (see Main equation box in Figure 5.2, $y \sim x_1 + x_2 + x_3$). Users can modify this if necessary, for instance, including higher order or interaction terms, other transformations are also allowed. This is done directly in the *Main Equation* space taking into account that this extra terms should follow formula command structure.² Note that the class of univariate models includes the intercept by default, except ordered probit, where the specification has to do this explicitly, that is, ordered probit models do not admit an intercept, for *identification* issues (see details below).³ Hence, users should write down specifically this fact ($y \sim x_1 + x_2 + x_3 - 1$). Finally, users should define the hyperparameters of the prior; for instance, in the normal-inverse gamma model, these are the mean vector, covariance matrix, shape, and scale parameters (see Figure 5.3). However, users should take into account that our GUI has *non-informative* hyperparameters by default in all our modelling frameworks, so the last part is not a requirement.

After this specification process, users should click the *Go!* button to initi-

²See <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/formula>

³An *identification* issue means that multiple values for the model parameters give rise to the same value for the likelihood function.

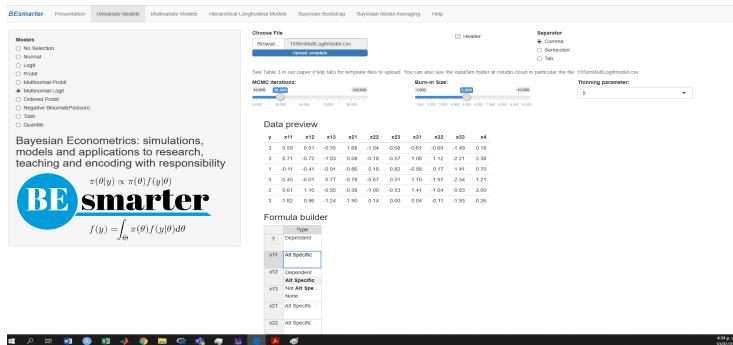


FIGURE 5.4
Univariate models: Multinomial.

ate the estimation. Our GUI displays the summary statistics and convergence diagnostics after this process is finished (see Figure 5.3). There are also widgets to download posterior chains (*csv* file) and graphs (*pdf* and *eps* files). Note that the order of the coefficients in the results (summary, posterior chains, and graphs) is first for the location parameters, and then for the scale parameters.

Multinomial models (probit and logit) require a dataset file to have in the first column the dependent variable, then alternative specific regressors (for instance alternatives' prices), and finally, non-alternative regressors (for instance, income). The formula builder specifies the dependent variable, and independent variables that are alternative specific and non-alternative specific (see technical details in next chapter). Specification also requires defining the base category, number of alternatives (this is also required in ordered probit), number of alternative specific regressors, and number of non-alternative regressors (see Figure 5.4). Multinomial logit also allows defining a tuning parameter, the number of degrees of freedom in this case, for the Metropolis-Hastings algorithm (see technical details in next chapter). This is a feature in our GUI when the estimation of the models is based on the Metropolis-Hastings algorithm. The order of the coefficients in the results of these models is first the intercepts (cte_l appearing in the summary display, l -th alternative), and then the non-alternative specific regressors (NAS_{jl} appearing in the summary display, l -th alternative and j -th non-alternative regressor), and lastly, the coefficients for the alternative specific regressors (AS_j appearing in the summary display, j -th alternative specific regressor). Note that the non-alternative specific regressors associated with the base category are equal to zero (they do not appear in the results). In addition, some coefficients of the main diagonal of the covariance matrix are constant due to identification issues in multinomial and multivariate probit models.

In the case of the negative binomial model, users should set a dispersion parameter (see the negative binomial model in the next chapter). User should

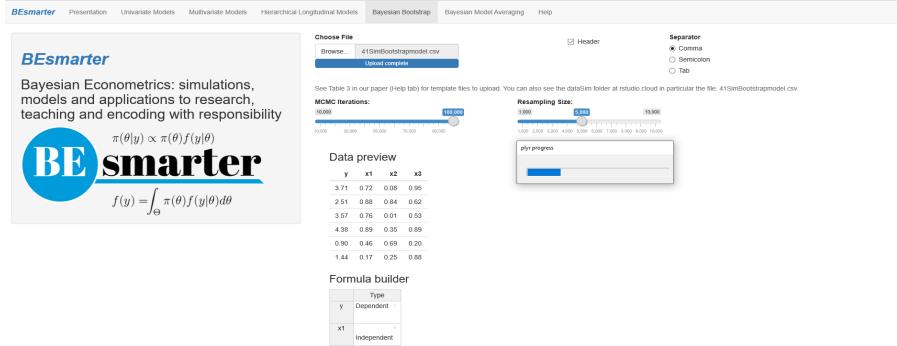


FIGURE 5.5
Univariate models: Bootstrap.

also set the censorship points and quantiles in the Tobit and quantile models, respectively.

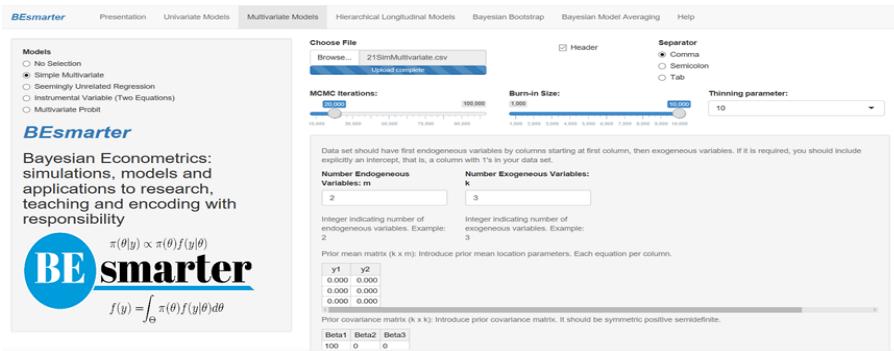
Bayesian bootstrap only requires uploading a dataset, specifying the number of iterations of the MCMC, the resampling size, and the equation (see Figure 5.5). The input file has the same structure as the file used in the univariate normal model.

5.3 Multivariate models

After our GUI is deployed (see Figure 5.1), the user should select *Multivariate Models* in the top panel. Then, the Figure 5.6 is displayed, and the user can see the radio button on the left hand side that shows the specific models inside this generic class.

Figure 5.6 displays the multivariate regression setting. In this case, the input file should have first the dependent variables, and then the regressors. If there are intercepts in each equation, there should be a column of 1's after the dependent variables in the input file. The user also has to set the number of dependent variables, the number of regressors, if necessary include the intercept, and the values of the hyperparameters (see Figure 5.6).

The input file in seemingly unrelated regressions should have first the dependent variables, and then the regressors by equation, including the intercept in each equation if necessary (column of 1's). Users should define the number of dependent variables (equations), the number of total regressors, that is, the sum of all regressors associated with the equation (if necessary include intercepts, each intercept is an additional regressor), and the number of regressors

**FIGURE 5.6**

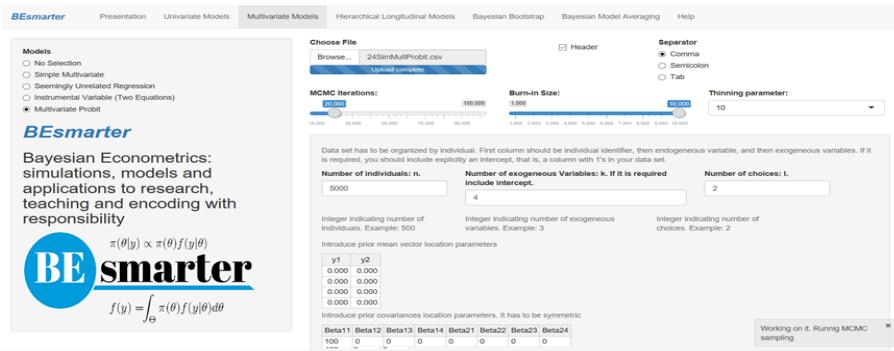
Multivariate models: Simple multivariate.

by equation (if necessary include the intercept). Users can also set the values of the hyperparameters if there is prior information.

The results of the simple multivariate and seemingly unrelated regressions show first the posterior location parameters by equation, and then the posterior covariance matrix.

In the instrumental variable setting, users should specify the main equation and the instrumental equation. This setting includes intercepts by default. The first variable on the right hand side in the main equation has to be the variable with endogeneity issues. In the instrumental equation box, the dependent variable is the variable with endogeneity issues as a function of the instruments. Users can also specify the values of the hyperparameters if they have prior information. The input file should have the dependent variable, the endogenous regressor, the instruments, and the exogenous regressors. The results first list the posterior estimates of the endogenous regressor, then the location parameters of the auxiliary regression (instrumental equation), and the location parameters of the exogenous regressors. Last is the posterior covariance matrix.

The multivariate probit model requires an input dataset ordered by unit, for instance three choices implies repeat each unit three times. The first column has to be the identification of each unit; users should use ordered integers, then the dependent variable, just one vector, composed of 0's and 1's, then the regressors, which should include a column of 1's for the intercepts. Users should set the number of units, number of regressors, and number of choices (see Figure 5.7). The results first display the posterior location parameters by equation, and then the posterior covariance matrix.

**FIGURE 5.7**

Multivariate models: Multivariate probit.

5.4 Time series model

5.5 Longitudinal/panel models

After our GUI is deployed (see Figure 5.1), the user should select *Hierarchical Longitudinal Models* in the top panel. Then, the Figure 5.8 is displayed, and the user can see the radio button on the left hand side that shows the specific models inside this generic class.

The hierarchical longitudinal models tab allows for estimating models that account for within-subject correlation when the dependent variable is continuous (Normal), binary (Logit), or a count (Poisson).

The input files for hierarchical longitudinal models should have first the dependent variable, then the regressors and a cross sectional identifier ($i = 1, 2, \dots, N$). It is not a requirement to have a balanced dataset: T_i can be different for each i (see Chapter 9 for technical details). Users can see templates of datasets in the folders **DataSim** (see Table 14.2 for details), and **DataApp** (see Table 14.3 for details) of our *GitHub* repository. Users should also specify the fixed part equation and the random part equation, both in **R** format. In case of only requiring random intercepts, do not introduce anything in the latter part (see Figure 5.8). Users should also type the name of the cross sectional identifier variable. The results displayed and the posterior graphs are associated with the fixed effects and covariance matrix. However, users can download the posterior chains of all posterior estimates: fixed and random effects, and covariance matrix.

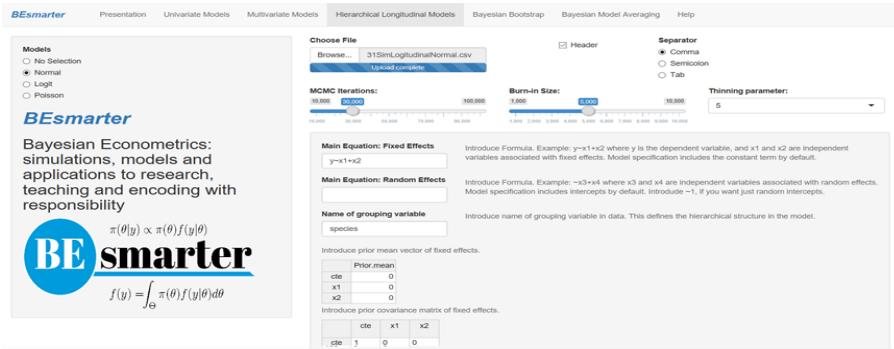


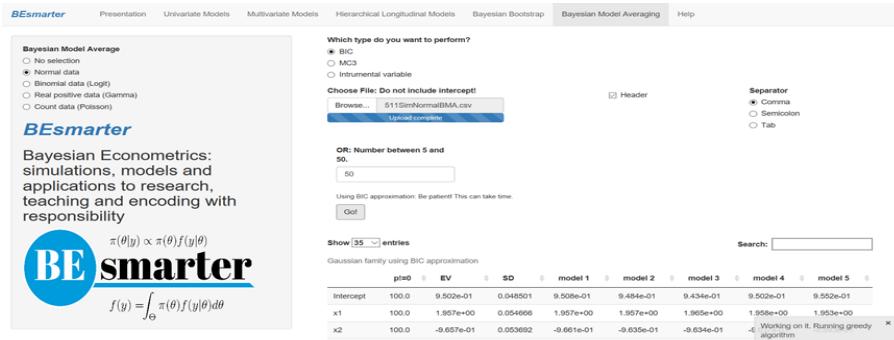
FIGURE 5.8
Hierarchical longitudinal models: Specification.

5.6 Bayesian model average

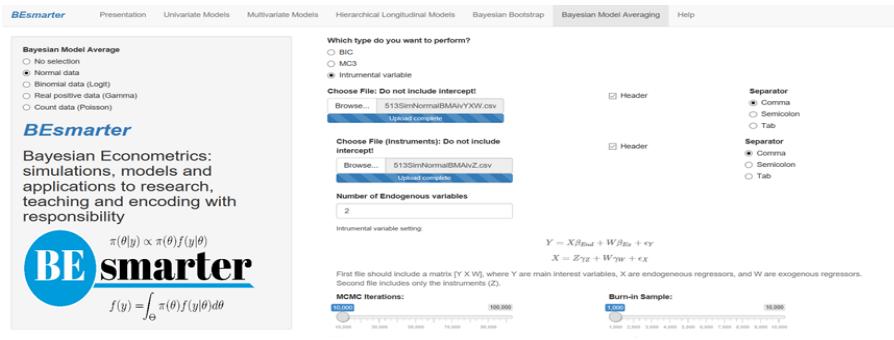
After our GUI is deployed (see Figure 5.1), the user should select *Bayesian Model Averaging* in the top panel. Then, the Figure 5.9 is displayed, and the user can see the radio button on the left hand side that shows the specific models inside this generic class.

Bayesian model averaging based on a Gaussian distribution can be carried out using the Bayesian information criterion (BIC) approximation, Markov chain Monte Carlo model composition (MC3), or instrumental variables (see Figure 5.9). The former two approaches require an input dataset where the first column is the dependent variable, and then, the potentially important regressors. Users should set the band width model selection parameter (O_R) and number of iterations for BIC and MC3, respectively (see Chapter 10 for technical details). The results include the posterior inclusion probability ($p! = 0$), expected value (EV), and standard deviation (SD) of the coefficients associated with each regressor. The BIC framework also displays the most relevant models, including the number of regressors, the coefficient of determination (R^2), the BIC, and the posterior model probability. Users can download two csv files: *Best models* and *Descriptive statistics coefficients*. The former is a 0-1 matrix such that the columns are the regressors and the rows are the models; a 1 indicates the presence of a specific regressor in a specific model, 0 otherwise. Note that the last column of this file is the posterior model probability for each model (row). The latter file shows the posterior inclusion probabilities, expected values, and standard deviations associated with each regressor, taking into account the BMA procedure based on the best models.

Bayesian model averaging with endogeneity issues requires two input files. The first one has the dependent variable in the first column, the next columns are the regressors with endogeneity issues, and then the exogenous regressors.

**FIGURE 5.9**

Bayesian model averaging: Specification and results.

**FIGURE 5.10**

Bayesian model averaging: Instrumental variable specification.

The user should include a column of 1's if an intercept is required. The second input file has all the instruments. Users should also introduce the number of regressors with endogeneity issues (see Figure 5.10).

The results include the posterior inclusion probabilities and expected values for each regressor. The user can find the results of the main equation, and then of the auxiliary equations. Users can download *csv* files of BMA results for both the second stage (main equation) and the first stage (auxiliary equations). In addition, users can download the posterior chains of the location parameters of the main equation, β_l , $l = 1, 2, \dots, \dim \{\beta\}$, the location parameters of the auxiliary equations, $\gamma_{j,i}$, $j = 1, 2, \dots, \dim \{\beta_s\}$ where $\dim \{\beta_s\}$ is the number of regressors with endogeneity issues, $i = 1, 2, \dots, \dim \{\gamma\}$, where $\dim \{\gamma\}$ is the number of regressors in the auxiliary regressors (exogeneous regressors + instruments), and the elements of the covariance matrix $\sigma_{j,k}$ (see Chapter 10 for technical details).

Bayesian model averaging based on BIC approximation for non-linear mod-

els, Logit, Gamma, and Poisson, requires an input dataset where the first column is the dependent variable, and the other columns are the potentially relevant regressors. Users should specify the band width model selection parameters, which are also referred to as Occam’s window parameters (O_R and O_L). Our GUI displays the posterior inclusion probabilities ($p! = 0$), the expected value of the posterior coefficients (EV), and the standard deviation (SD). In addition, users can see the results associated with the models with the highest posterior model probabilities, and download *csv* files with the results of specifications of the best models, and descriptive statistics of the posterior coefficients from the BMA procedure. These files are similar to the results of the BIC approximation of the Gaussian model.

5.7 Warning

Users should also note that sometimes our GUI shuts down. In our experience, this is due to computational issues using the implicit commands that we call when estimating some models, for instance, computationally singular systems, missing values where TRUE/FALSE needed, L-BFGS-B needs finite values of “fn”, NA/NaN/Inf values, or Error in backsolve. Sometimes these issues can be solved by adjusting the dataset, for instance, avoiding high levels of multicollinearity. It should also be taken into account that when warning messages are displayed in our GUI, there is a high chance that there are convergence issues of the posterior chains. So, the results are not trustworthy. Users can identify these problems by checking the console of their *RStudio* sections, where the specific folder/file where the issue happened is specified. In any case, we would appreciate your feedback to improve and enhance our GUI.

We also should say there are many ways to improve the codes that we present in the following five chapters. For instance, the *MCMCpack* and *bayesm* packages perform most of the matrix operations in C++ using the *rcpp* package. This substantially speeds up the algorithms compared with the codes that we present in the next chapters. We could improve the computational times of our codes using parallel computing and the *rcpp* package, but this requires more advanced skills that we do not cover in this book.



6

Univariate models

We describe how to perform Bayesian inference in some of the most common univariate models: normal-inverse gamma, logit, probit, multinomial probit and logit, ordered probit, negative binomial, tobit, quantile regression, and Bayesian bootstrap in linear models. The point of departure is assuming a random sample of cross-sectional units. Then, we show the posterior distributions of the parameters and some applications. In addition, we show how to perform inference in various models using three levels of programming skills: our graphical user interface (GUI), packages from **R**, and programming the posterior distributions. The first requires no programming skills, the second requires an intermediate level, and the third demands more advanced skills. We also include mathematical and computational exercises.

We can run our GUI typing

R code. How to display our graphical user interface

```
1 shiny::runGitHub("besmarter/BSTApp", launch.browser = T)
```

in the **R** package console or any **R** code editor. However, users should see Chapter 5 for details.

6.1 The Gaussian linear model

The Gaussian linear model specifies $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\mu}$ such that $\boldsymbol{\mu} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_N)$ is an stochastic error, \mathbf{X} is a $N \times K$ matrix of regressors, $\boldsymbol{\beta}$ is a K -dimensional vector of location coefficients, σ^2 is the variance of the model (scale parameter), \mathbf{y} is a N -dimensional vector of a dependent variable, and N is the sample size. We describe this model using the conjugate family in Section 3.3, that

is, $\pi(\boldsymbol{\beta}, \sigma^2) = \pi(\boldsymbol{\beta}|\sigma^2) \times \pi(\sigma^2)$, and this allowed to get the posterior marginal distribution for $\boldsymbol{\beta}$ and σ^2 .

We assume independent prior in this section, that is, $\pi(\boldsymbol{\beta}, \sigma^2) = \pi(\boldsymbol{\beta}) \times \pi(\sigma^2)$, where $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$, $\alpha_0/2$ and $\delta_0/2$ are the shape and rate parameters. This setting allows getting the posterior conditional distributions, that is, $\pi(\boldsymbol{\beta}|\sigma^2, \mathbf{y}, \mathbf{X})$ and $\pi(\sigma^2|\boldsymbol{\beta}, \mathbf{y}, \mathbf{X})$, which in turn allows to use the Gibbs sampler algorithm to perform posterior inference of $\boldsymbol{\beta}$ and σ^2 .

The likelihood function in this model is

$$p(\mathbf{y}|\boldsymbol{\beta}, \sigma^2, \mathbf{X}) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\}.$$

Then, the conditional posterior distributions are

$$\boldsymbol{\beta}|\sigma^2, \mathbf{y}, \mathbf{X} \sim N(\boldsymbol{\beta}_n, \mathbf{B}_n),$$

and

$$\sigma^2|\boldsymbol{\beta}, \mathbf{y}, \mathbf{X} \sim IG(\alpha_n/2, \delta_n/2),$$

where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \sigma^{-2} \mathbf{X}^\top \mathbf{X})^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \sigma^{-2} \mathbf{X}^\top \mathbf{y})$, $\alpha_n = \alpha_0 + N$ and $\delta_n = \delta_0 + (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ (see Exercise 1 in this chapter).¹

Example: The market value of soccer players in Europe

Let's analyze the determinants of the market value of soccer players in Europe. In particular, we use the dataset `1ValueFootballPlayers.csv` which is in folder **DataApp** in our github repository <https://github.com/besmarter/BSTApp>. This dataset was used by [198] to finding the determinants of high performance soccer players in the five most important national leagues in Europe.

The specification of the model is

$$\begin{aligned} \log(\text{Value}_i) = & \beta_1 + \beta_2 \text{Perf}_i + \beta_3 \text{Age}_i + \beta_4 \text{Age}_i^2 + \beta_5 \text{NatTeam}_i \\ & + \beta_6 \text{Goals}_i + \beta_7 \text{Exp}_i + \beta_8 \text{Exp}_i^2 + \mu_i, \end{aligned}$$

where *Value* is the market value in Euros (2017), *Perf* is a measure of performance, *Age* is the players' age in years, *NatTeam* is an indicator variable that takes the value of 1 if the player has been on the national team, *Goals* is the number of goals scored by the player during his career, and *Exp* is his experience in years.

We assume that the dependent variable distributes normal, then we use a normal-inverse gamma model using vague conjugate priors where $\boldsymbol{\beta}_0 =$

¹This model can be extended to consider heteroskedasticity such that $y_i \sim N(\mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2/\tau_i)$, where $\tau_i \sim G(v/2, v/2)$. See exercise 2 for details.

$1000\mathbf{I}_8$, $\beta_0 = \mathbf{0}_8$, $\alpha_0 = 0.001$ and $\delta_0 = 0.001$. We perform a Gibbs sampler with 5,000 MCMC iterations plus a burn-in equal to 5,000, and a thinning parameter equal to 1.

Once our GUI is displayed (see beginning of this chapter), we should follow Algorithm A6 to run linear Gaussian models in our GUI (see Chapter 5 for details):

Algorithm A6 Linear Gaussian model

- 1: Select *Univariate Models* on the top panel
 - 2: Select *Normal* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Select dependent and independent variables using the *Formula builder* table
 - 6: Click the *Build formula* button to generate the formula in **R** syntax. You can modify the formula in the **Main equation** box using valid arguments of the *formula* command structure in **R**
 - 7: Set the hyperparameters: mean vector, covariance matrix, shape and scale parameters. This step is not necessary as by default our GUI uses non-informative priors
 - 8: Click the *Go!* button
 - 9: Analyze results
 - 10: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

We can see in the next **R** codes how to perform the linear Gaussian model using the command *MCMCregress* of the *MCMCpack* package, and programming the Gibbs sampler ourselves. We should get similar results using the three approaches: GUI, package and our function. In fact, our GUI relies on the *MCMCregress* command. For instance, the value of a top soccer player in Europe increases 134% ($\exp(0.85) - 1$) on average when he has played in the national team, the credible interval at 95% is (86%, 197%).

R code. The value of soccer players, using R packages

```

1 rm(list = ls())
2 set.seed(010101)
3 ##### Linear regression: Value of
4 # soccer players #####
5 Data <- read.csv("https://raw.githubusercontent.com/
6   besmarter/BSTApp/refs/heads/master/DataApp/1
7   ValueFootballPlayers.csv", sep = ",", header = TRUE,
8   quote = "")
9 attach(Data)
10 y <- log(Value)
11 # Value: Market value in Euros (2017) of soccer players
12 # Regressors quantity including intercept
13 X <- cbind(1, Perf, Age, Age2, NatTeam, Goals, Exp, Exp2)
14 # Perf: Performance. Perf2: Performance squared. Age: Age;
15 # Age: Age squared.
16 # NatTeam: Indicator of national team. Goals: Scored goals.
17 # Goals2: Scored goals squared
18 # Exp: Years of experience. Exp2: Years of experience
19 # squared. Assists: Number of assists
20 k <- dim(X)[2]
21 N <- dim(X)[1]
22 # Hyperparameters
23 d0 <- 0.001/2
24 a0 <- 0.001/2
25 b0 <- rep(0, k)
26 c0 <- 1000
27 B0 <- c0*diag(k)
28 B0i <- solve(B0)
29 # MCMC parameters
30 mcmc <- 5000
31 burnin <- 5000
32 tot <- mcmc + burnin
33 thin <- 1
34 # Posterior distributions using packages: MCMCpack sets the
35 # model in terms of the precision matrix
36 posterior <- MCMCpack::MCMCregress(y~X-1, b0=b0, B0 = B0i,
37   c0 = a0, d0 = d0, burnin = burnin, mcmc = mcmc, thin =
38   thin)
39 summary(coda::mcmc(posterior))
40 Iterations = 1:5000
41 Thinning interval = 1
42 Number of chains = 1
43 Sample size per chain = 5000
44 1. Empirical mean and standard deviation for each variable,
45 plus standard error of the mean:
46
47      Mean        SD  Naive SE Time-series SE
48 X       3.695499 2.228060 3.151e-02     3.151e-02
49 XPerf    0.035445 0.004299 6.079e-05     6.079e-05
50 XAge     0.778410 0.181362 2.565e-03     2.565e-03
51 XAge2    -0.016617 0.003380 4.781e-05     4.781e-05
52 XNatTeam 0.850362 0.116861 1.653e-03     1.689e-03
53 XGoals   0.009097 0.001603 2.266e-05     2.266e-05
54 XExp     0.206208 0.062713 8.869e-04     8.428e-04
55 XExp2    -0.006992 0.002718 3.844e-05     3.719e-05
56 sigma2   0.969590 0.076091 1.076e-03     1.076e-03

```

R. code. The value of soccer players, programming our Gibbs sampler

```

1 # Posterior distributions programming the Gibbs sampling
2 # Auxiliary parameters
3 Xtx <- t(X) %*% X
4 bhat <- solve(Xtx) %*% t(X) %*% y
5 an <- a0 + N
6 # Gibbs sampling functions
7 PostSig2 <- function(Beta){
8   dn <- d0 + t(y - X %*% Beta) %*% (y - X %*% Beta)
9   sig2 <- invgamma::rinvgamma(1, shape = an/2, rate = dn/2)
10  return(sig2)
11 }
12 PostBeta <- function(sig2){
13   Bn <- solve(B0i + sig2^(-1)*Xtx)
14   bn <- Bn %*% (B0i %*% b0 + sig2^(-1)*Xtx %*% bhat)
15   Beta <- MASS::mvrnorm(1, bn, Bn)
16   return(Beta)
17 }
18 PostBetas <- matrix(0, mcmc+burnin, k)
19 PostSigma2 <- rep(0, mcmc+burnin)
20 Beta <- rep(0, k)
21 for(s in 1:tot){
22   sig2 <- PostSig2(Beta = Beta)
23   PostSigma2[s] <- sig2
24   Beta <- PostBeta(sig2 = sig2)
25   PostBetas[s,] <- Beta
26 }
27 keep <- seq((burnin+1), tot, thin)
28 PosteriorBetas <- PostBetas[keep,]
29 colnames(PosteriorBetas) <- c("Intercept", "Perf", "Age", "
  Age2", "NatTeam", "Goals", "Exp", "Exp2")
30 summary(coda::mcmc(PosteriorBetas))
31 Iterations = 1:5000
32 Thinning interval = 1
33 Number of chains = 1
34 Sample size per chain = 5000
35 1. Empirical mean and standard deviation for each variable,
36 plus standard error of the mean:
37               Mean        SD Naive SE Time-series SE
38 Intercept 3.663230 2.194363 3.103e-02    3.103e-02
39 Perf      0.035361 0.004315 6.102e-05    6.102e-05
40 Age       0.780374 0.178530 2.525e-03    2.525e-03
41 Age2     -0.016641 0.003332 4.713e-05    4.713e-05
42 NatTeam   0.850094 0.119093 1.684e-03    1.684e-03
43 Goals     0.009164 0.001605 2.270e-05    2.270e-05
44 Exp       0.205965 0.062985 8.907e-04    8.596e-04
45 Exp2     -0.007006 0.002731 3.862e-05    3.701e-05
46 PosteriorSigma2 <- PostSigma2[keep]
47 summary(coda::mcmc(PosteriorSigma2))
48 Iterations = 1:5000
49 Thinning interval = 1
50 Number of chains = 1
51 Sample size per chain = 5000
52 1. Empirical mean and standard deviation for each variable,
53 plus standard error of the mean:
54               Mean        SD Naive SE Time-series SE
55 0.973309    0.077316   0.001093   0.001116

```

6.2 The logit model

In the logit model the dependent variable is binary, $Y_i = \{1, 0\}$, then it follows a Bernoulli distribution, $Y_i \stackrel{ind}{\sim} B(\pi_i)$, that is, $p(Y_i = 1) = \pi_i$, such that $\pi_i = \frac{\exp\{\boldsymbol{x}_i^\top \boldsymbol{\beta}\}}{1 + \exp\{\boldsymbol{x}_i^\top \boldsymbol{\beta}\}}$.

The likelihood function of the logit model is

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{X}) &= \prod_{i=1}^N \pi_i^{y_i} (1 - \pi_i)^{1-y_i} \\ &= \prod_{i=1}^N \left(\frac{\exp\{\boldsymbol{x}_i^\top \boldsymbol{\beta}\}}{1 + \exp\{\boldsymbol{x}_i^\top \boldsymbol{\beta}\}} \right)^{y_i} \left(\frac{1}{1 + \exp\{\boldsymbol{x}_i^\top \boldsymbol{\beta}\}} \right)^{1-y_i}. \end{aligned}$$

We can specify a Normal distribution as prior, $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$. Then, the posterior distribution is

$$\begin{aligned} \pi(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) &\propto \prod_{i=1}^N \left(\frac{\exp\{\boldsymbol{x}_i^\top \boldsymbol{\beta}\}}{1 + \exp\{\boldsymbol{x}_i^\top \boldsymbol{\beta}\}} \right)^{y_i} \left(\frac{1}{1 + \exp\{\boldsymbol{x}_i^\top \boldsymbol{\beta}\}} \right)^{1-y_i} \\ &\times \exp \left\{ -\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{B}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right\}. \end{aligned}$$

The logit model does not have a standard posterior distribution. Then, a random walk Metropolis–Hastings algorithm can be used to obtain draws from the posterior distribution. A potential proposal is a multivariate Normal centered at the current value, with covariance matrix $\tau^2 (\mathbf{B}_0^{-1} + \hat{\Sigma}^{-1})^{-1}$, where $\tau > 0$ is a tuning parameter and $\hat{\Sigma}$ is the sample covariance matrix from the maximum likelihood estimation [145].²

Observe that $\log(p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{X})) = \sum_{i=1}^N y_i \boldsymbol{x}_i^\top \boldsymbol{\beta} - \log(1 + \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta}))$. We can use this expression when calculating the acceptance parameter in the computational implementation of the Metropolis–Hastings algorithm. In particular, the acceptance parameter is

$$\alpha = \min \left\{ 1, \exp(\log(p(\mathbf{y}|\boldsymbol{\beta}^c, \mathbf{X})) + \log(\pi(\boldsymbol{\beta}^c)) - (\log(p(\mathbf{y}|\boldsymbol{\beta}^{(s-1)}, \mathbf{X})) + \log(\pi(\boldsymbol{\beta}^{(s-1)})))) \right\},$$

where $\boldsymbol{\beta}^c$ and $\boldsymbol{\beta}^{(s-1)}$ are the draws from the proposal distribution and the previous iteration of the Markov chain, respectively.³

Example: Simulation exercise

²Tuning parameters should be set in a way such that one obtains reasonable diagnostic criteria and acceptance rates.

³Formulating the acceptance rate using log helps to mitigate computational problems.

Let's do a simulation exercise to check the performance of the algorithm. Set $\beta = [0.5 \ 0.8 \ -1.2]^\top$, $x_{ik} \sim N(0, 1)$, $k = 2, 3$ and $i = 1, 2, \dots, 10000$.

We set as hyperparameters $\beta_0 = [0 \ 0 \ 0]^\top$ and $B_0 = 1000I_3$. The tune parameter for the Metropolis-Hastings algorithm is equal to 1.

Once our GUI is displayed (see beginning of this chapter), we should follow Algorithm A7 to run logit models in our GUI (see Chapter 5 for details):

Algorithm A7 Logit model

- 1: Select *Univariate Models* on the top panel
 - 2: Select *Logit* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Select dependent and independent variables using the *Formula builder* table
 - 6: Click the *Build formula* button to generate the formula in **R** syntax. You can modify the formula in the **Main equation** box using valid arguments of the *formula* command structure in **R**
 - 7: Set the hyperparameters: mean vector and covariance matrix. This step is not necessary as by default our GUI uses non-informative priors
 - 8: Select the tuning parameter for the Metropolis-Hastings algorithm
 - 9: Click the *Go!* button
 - 10: Analyze results
 - 11: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

We can see in the next **R** codes how to perform the logit model using the command *MCMClogit* of the *MCMCpack* package, and programming the Metropolis-Hastings algorithm ourselves.

We should get similar results using the three approaches: GUI, package and our function. Our GUI relies on the *MCMClogit* command. In particular, we obtain an acceptance rate of 0.46, and the diagnostics suggest that the posterior chains behave well. In general, the 95% credible intervals encompass the population values, and the mean and median are very close to these values.

R. code. Simulation of the logit model estimation using R packages

```

1 ##### Logit: Simulation
2 # Simulate data
3 rm(list = ls())
4 set.seed(010101)
5 N <- 10000 # Sample size
6 B <- c(0.5, 0.8, -1.2) # Population location parameters
7 x2 <- rnorm(N) # Regressor
8 x3 <- rnorm(N) # Regressor
9 X <- cbind(1, x2, x3) # Regressors
10 XB <- X%*%B
11 PY <- exp(XB)/(1 + exp(XB)) # Probability of Y = 1
12 Y <- rbinom(N, 1, PY) # Draw Y's
13 table(Y) # Frequency
14 # write.csv(cbind(Y, x2, x3), file = "DataSimulations/
15 # LogitSim.csv") # Export data
15 # MCMC parameters
16 iter <- 5000; burnin <- 1000; thin <- 5; tune <- 1
17 # Hyperparameters
18 K <- dim(X)[2]
19 b0 <- rep(0, K)
20 c0 <- 1000
21 B0 <- c0*diag(K)
22 B0i <- solve(B0)
23 # Posterior distributions using packages: MCMCpack sets the
# model in terms of the precision matrix
24 RegLog <- MCMCpack::MCMClogit(Y~X-1, mcmc = iter, burnin =
burnin, thin = thin, b0 = b0, B0 = B0i, tune = tune)
25 summary(RegLog)
26 Iterations = 1001:5996
27 Thinning interval = 5
28 Number of chains = 1
29 Sample size per chain = 1000
30 1. Empirical mean and standard deviation for each variable,
31 plus standard error of the mean:
32      Mean     SD  Naive SE Time-series SE
33 X    0.4896  0.02550  0.0008064    0.001246
34 Xx2  0.8330  0.02730  0.0008632    0.001406
35 Xx3 -1.2104  0.03049  0.0009643    0.001536
36 2. Quantiles for each variable:
37      2.5%    25%    50%    75%   97.5%
38 X    0.4424  0.4728  0.4894  0.5072  0.5405
39 Xx2  0.7787  0.8159  0.8327  0.8505  0.8852
40 Xx3 -1.2758 -1.2296 -1.2088 -1.1902 -1.1513

```

R. code. Simulation of the logit model estimation programming our M-H algorithm

```

1 # Posterior distributions programming the Metropolis-
2 # Hastings algorithm
3 MHfunc <- function(y, X, b0 = rep(0, dim(X)[2] + 1), B0 =
4   1000*diag(dim(X)[2] + 1), tau = 1, iter = 6000, burnin =
5   1000, thin = 5){
6   Xm <- cbind(1, X) # Regressors
7   K <- dim(Xm)[2] # Number of location parameters
8   BETAS <- matrix(0, iter + burnin, K) # Space for posterior
9   chains
10  Reg <- glm(y ~ Xm - 1, family = binomial(link = "logit"))
11  # Maximum likelihood estimation
12  BETA <- Reg$coefficients # Maximum likelihood parameter
13  estimates
14  tot <- iter + burnin # Total iterations M-H algorithm
15  COV <- vcov(Reg) # Maximum likelihood covariance matrix
16  COVt <- tau^2*solve(solve(B0) + solve(COV)) # Covariance
17  # matrix for the proposal distribution
18  Accep <- rep(0, tot) # Space for calculating the
19  acceptance rate
20  # Create progress bar in case that you want to see
21  # iterations progress
22  pb <- winProgressBar(title = "progress bar", min = 0,
23  max = tot, width = 300)
24  for(it in 1:tot){
25    BETAc <- BETA + MASS::mvrnorm(n = 1, mu = rep(0, K),
26    Sigma = COVt) # Candidate location parameter
27    likecand <- sum((Xm%*%BETAc) * Y - apply(Xm%*%BETAc, 1,
28    function(x) log(1 + exp(x)))) # Log likelihood for the
29    candidate
30    likepast <- sum((Xm%*%BETA) * Y - apply((Xm%*%BETA), 1,
31    function(x) log(1 + exp(x)))) # Log likelihood for the
32    actual draw
33    priorcand <- (-1/2)*crossprod((BETAc - b0), solve(B0))%*
34    %(BETAc - b0) # Log prior for candidate
35    priorpast <- (-1/2)*crossprod((BETA - b0), solve(B0))%*%
36    (BETA - b0) # Log prior for actual draw
37    alpha <- min(1, exp((likecand + priorcand) - (likepast +
38    priorpast))) #Probability of selecting candidate
39    u <- runif(1) # Decision rule for selecting candidate
40    if(u < alpha){
41      BETA <- BETAc # Changing reference for candidate if
42      selected
43      Accep[it] <- 1 # Indicator if the candidate is
44      accepted
45    }
46    BETAS[it, ] <- BETA # Saving draws
47    setWinProgressBar(pb, it, title=paste( round(it/tot*100,
48    0),
49    "% done"))
50  }
51  close(pb)
52  keep <- seq(burnin, tot, thin)
53  return(list(Bs = BETAS[keep[-1], ], AceptRate = mean(Accep
54  [keep[-1]])))
55 }
```

*R. code. Simulation of the logit model
programming our M-H algorithm, results*

```

1 Posterior <- MHfunc(y = Y, X = cbind(x2, x3), iter = iter,
2   burnin = burnin, thin = thin) # Running our M-H function
3   changing some default parameters.
4 paste("Acceptance rate equal to", round(Posterior$AceptRate,
5   2), sep = " ")
6 "Acceptance rate equal to 0.46"
7 PostPar <- coda::mcmc(Posterior$Bs)
8 # Names
9 colnames(PostPar) <- c("Cte", "x1", "x2")
10 # Summary posterior draws
11 summary(PostPar)
12 Iterations = 1:1000
13 Thinning interval = 1
14 Number of chains = 1
15 Sample size per chain = 1000
16 1. Empirical mean and standard deviation for each variable,
17 plus standard error of the mean:
18 Mean      SD  Naive SE Time-series SE
19 Cte  0.4893  0.02427  0.0007674    0.001223
20 x1   0.8309  0.02699  0.0008536    0.001440
21 x2  -1.2107  0.02943  0.0009308    0.001423
22 2. Quantiles for each variable:
23    2.5%     25%     50%     75%   97.5%
24 Cte  0.4431  0.4721  0.4899  0.5059  0.5344
25 x1   0.7817  0.8123  0.8305  0.8505  0.8833
26 x2  -1.2665 -1.2309 -1.2107 -1.1911 -1.1538
27 # Trace and density plots
28 plot(PostPar)
29 # Autocorrelation plots
30 coda::autocorr.plot(PostPar)
31 # Convergence diagnostics
32 coda::geweke.diag(PostPar)
33 Fraction in 1st window = 0.1
34 Fraction in 2nd window = 0.5
35 Cte      x1      x2
36 -0.975 -3.112  1.326
37 coda::raftery.diag(PostPar,q=0.5,r=0.05,s = 0.95)
38 Quantile (q) = 0.5
39 Accuracy (r) = +/- 0.05
40 Probability (s) = 0.95
41 Burn-in Total Lower bound Dependence
42 (M)       (N)   (Nmin)   factor (I)
43 Cte 6        731    385      1.90
44 x1  6        703    385      1.83
45 x2  6        725    385      1.88
46 coda::heidel.diag(PostPar)
47 Stationarity start      p-value
48 test      iteration
49 Cte passed      1      0.4436
50 x1  passed     101    0.3470
51 x2  passed      1      0.0872
52 Halfwidth Mean      Halfwidth
53 test
54 Cte passed      0.489  0.00240
55 x1  passed      0.832  0.00268
56 x2  passed     -1.211  0.00279

```

6.3 The probit model

The probit model also has as dependent variable a binary outcome. In this case, there is a latent variable (y_i^* , unobserved) that defines the structure of the estimation problem. In particular,

$$Y_i = \begin{cases} 0, & Y_i^* \leq 0 \\ 1, & Y_i^* > 0 \end{cases},$$

such that $Y_i^* = \mathbf{x}_i^\top \boldsymbol{\beta} + \mu_i$, $\mu_i \stackrel{i.i.d.}{\sim} N(0, 1)$.⁴ This implies $P(Y_i = 1) = \pi_i = \Phi(\mathbf{x}_i^\top \boldsymbol{\beta})$.

[2] implemented data augmentation [208] to apply a Gibbs sampling algorithm in this model. Augmenting this model with Y_i^* , we can have the likelihood contribution from observation i , $p(y_i|y_i^*) = \mathbb{1}_{y_i=0}\mathbb{1}_{y_i^*\leq 0} + \mathbb{1}_{y_i=1}\mathbb{1}_{y_i^*>0}$, where $\mathbb{1}_A$ is an indicator function that takes the value of 1 when condition A is satisfied.

The posterior distribution is $\pi(\boldsymbol{\beta}, \mathbf{y}^*|\mathbf{y}, \mathbf{X}) \propto \prod_{i=1}^N [\mathbb{1}_{y_i=0}\mathbb{1}_{y_i^*\leq 0} + \mathbb{1}_{y_i=1}\mathbb{1}_{y_i^*>0}] \times N_N(\mathbf{y}^*|\mathbf{X}\boldsymbol{\beta}, \mathbf{I}_n) \times N_K(\boldsymbol{\beta}|\boldsymbol{\beta}_0, \mathbf{B}_0)$ when taking a Gaussian distribution as prior $\boldsymbol{\beta} \sim N_k(\boldsymbol{\beta}_0, \mathbf{B}_0)$. This implies

$$y_i^*|\boldsymbol{\beta}, \mathbf{y}, \mathbf{X} \sim \begin{cases} TN_{(-\infty, 0]}(\mathbf{x}_i^\top \boldsymbol{\beta}, 1), & y_i = 0 \\ TN_{(0, \infty)}(\mathbf{x}_i^\top \boldsymbol{\beta}, 1), & y_i = 1 \end{cases}, \quad ^5$$

$$\boldsymbol{\beta}|\mathbf{y}^*, \mathbf{X} \sim N(\boldsymbol{\beta}_n, \mathbf{B}_n),$$

where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}$, and $\boldsymbol{\beta}_n = \mathbf{B}_n(\mathbf{B}_0^{-1}\boldsymbol{\beta}_0 + \mathbf{X}^\top \mathbf{y}^*)$.

Example: Determinants of hospitalization

We use the dataset named **2HealthMed.csv**, which is in folder **DataApp** in our github repository (<https://github.com/besmarter/BSTAApp>) and was used by [175]. Our dependent variable is a binary indicator with a value equal to 1 if an individual was hospitalized in 2007, and 0 otherwise.

The specification of the model is

$$\begin{aligned} \text{Hosp}_i = & \beta_1 + \beta_2 \text{SHI}_i + \beta_3 \text{Female}_i + \beta_4 \text{Age}_i + \beta_5 \text{Age}_i^2 + \beta_6 \text{Est2}_i + \beta_7 \text{Est3}_i \\ & + \beta_8 \text{Fair}_i + \beta_9 \text{Good}_i + \beta_{10} \text{Excellent}_i, \end{aligned}$$

where SHI is a binary variable equal to 1 if the individual is in a subsidized

⁴The variance in this model is set to 1 due to identification restrictions. Observe that $P(Y_i = 1|\mathbf{x}_i) = P(Y_i^* > 0|\mathbf{x}_i) = P(\mathbf{x}_i^\top \boldsymbol{\beta} + \mu_i > 0|\mathbf{x}_i) = P(\mu_i > -\mathbf{x}_i^\top \boldsymbol{\beta}|\mathbf{x}_i) = P(c \times \mu_i > -c \times \mathbf{x}_i^\top \boldsymbol{\beta}|\mathbf{x}_i) \forall c > 0$. Multiplying for a positive constant does not affect the probability of $Y_i = 1$.

⁵ TN denotes a truncated normal density.

health care program and 0 otherwise, *Female* is an indicator of gender, *Age* in years, *Est2* and *Est3* are indicators of socioeconomic status, the reference is *Est1*, which is the lowest, and self perception of health status where *bad* is the reference.

Let's set $\beta_0 = \mathbf{0}_{10}$, $B_0 = I_{10}$, iterations, burn-in and thinning parameters equal to 10000, 1000 and 1, respectively. We can use the Algorithm A6 to run the probit model in our GUI. We should select *Probit* model in stage 2. Our GUI relies in the command *rbprobitGibbs* from the package *bayesm* to perform inference in the Probit model. The following **R** code shows how to run this example using the command *rbprobitGibbs*. We asked to program a Gibbs sampler algorithm to perform inference in the probit model in the exercises.

We find evidence that gender and self-perceived health status affect the probability of hospitalization. Women have a higher probability of being hospitalized than men, and a better perception of health status decreases this probability.

R. code. Determinants of hospitalization

```

1 mydata <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTAApp/refs/heads/master/DataApp/2HealthMed.
  csv", sep = ",", header = TRUE, quote = "")
2 attach(mydata)
3 str(mydata)
4 K <- 10 # Number of regressors
5 b0 <- rep(0, K) # Prio mean
6 B0i <- diag(K) # Prior precision (inverse of covariance)
7 Prior <- list(betabar = b0, A = B0i) # Prior list
8 y <- Hosp # Dependent variables
9 X <- cbind(1, SHI, Female, Age, Age2, Est2, Est3, Fair, Good
  , Excellent) # Regressors
10 Data <- list(y = y, X = X) # Data list
11 Mcmc <- list(R = 10000, keep = 1, nprint = 0) # MCMC
  parameters
12 RegProb <- bayesm::rbprobitGibbs(Data = Data, Prior = Prior,
  Mcmc = Mcmc) # Inference using bayesm package
13 PostPar <- coda::mcmc(RegProb$betadraw) # Posterior draws
14 colnames(PostPar) <- c("Cte", "SHI", "Female", "Age", "Age2"
  , "Est2", "Est3", "Fair", "Good", "Excellent") # Names
15 summary(PostPar) # Posterior summary
16 Iterations = 1:10000
17 Thinning interval = 1
18 Number of chains = 1
19 Sample size per chain = 10000
20 2. Quantiles for each variable:
21      2.5%     25%     50%     75%    97.5%
22 Cte    -1.22e+00 -1.03e+00 -9.43e-01 -8.50e-01 -0.671744
23 SHI    -1.24e-01 -4.63e-02 -6.30e-03  3.26e-02  0.104703
24 Female  2.80e-02  9.65e-02  1.28e-01  1.60e-01  0.223123
25 Age    -7.55e-03 -2.50e-03  1.25e-04  2.80e-03  0.007646
26 Age2   -4.98e-05  9.05e-06  4.02e-05  7.07e-05  0.000128
27 Est2   -1.89e-01 -1.23e-01 -8.84e-02 -5.32e-02  0.012714
28 Est3   -2.13e-01 -1.03e-01 -4.73e-02  1.01e-02  0.109527
29 Fair    -7.09e-01 -5.69e-01 -4.93e-01 -4.16e-01 -0.269494
30 Good   -1.42e+00 -1.28e+00 -1.20e+00 -1.12e+00 -0.982533
31 Excellent -1.33e+00 -1.15e+00 -1.06e+00 -9.74e-01 -0.795881

```

6.4 The multinomial probit model

The multinomial probit model is used to model the choice of the l -th alternative over a set L mutually exclusive options. We observe

$$y_{il} = \begin{cases} 1, & y_{il}^* \geq \max\{\mathbf{y}_i^*\} \\ 0, & \text{otherwise} \end{cases},$$

such that $\mathbf{y}_i^* = \mathbf{X}_i \boldsymbol{\delta} + \boldsymbol{\mu}_i$, $\boldsymbol{\mu}_i \stackrel{i.i.d.}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma})$, \mathbf{y}_i^* is an unobserved latent L dimensional vector, $\mathbf{X}_i = [(1 \ \mathbf{c}_i^\top) \otimes \mathbf{I}_L \ \mathbf{A}_i]$ is an $L \times j$ matrix of regressors for each alternative, $l = 1, 2, \dots, L$, $j = L \times (1 + \dim\{\mathbf{c}_i\}) + a$, \mathbf{c}_i is a vector of the individuals' specific characteristics, \mathbf{A}_i is an $L \times a$ matrix of alternative-varying regressors, a is the number of alternative-varying regressors, and $\boldsymbol{\delta}$ is a j dimensional vector of parameters.

We take into account simultaneously the alternative-varying regressors (alternative attributes) and alternative-invariant regressors (individual characteristics).⁶ \mathbf{y}_i^* can be stacked up into a multiple regression with correlated stochastic errors, $\mathbf{y}^* = \mathbf{X}\boldsymbol{\delta} + \boldsymbol{\mu}$, where $\mathbf{y}^* = [\mathbf{y}_1^{*\top} \ \mathbf{y}_2^{*\top} \ \dots \ \mathbf{y}_N^{*\top}]^\top$, $\mathbf{X} = [\mathbf{X}_1^\top \ \mathbf{X}_2^\top \ \dots \ \mathbf{X}_N^\top]^\top$, and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^\top \ \boldsymbol{\mu}_2^\top \ \dots \ \boldsymbol{\mu}_N^\top]^\top$.

Following the practice of expressing y_{il}^* relative to y_{iL}^* by letting $\mathbf{w}_i = [w_{i1} \ w_{i2} \ \dots \ w_{iL-1}]^\top$, $w_{il} = y_{il}^* - y_{iL}^*$, we can write $\mathbf{w}_i = \mathbf{R}_i \boldsymbol{\beta} + \boldsymbol{\epsilon}_i$, $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \boldsymbol{\Omega})$, where $\mathbf{R}_i = [(1 \ \mathbf{c}_i^\top) \otimes \mathbf{I}_{L-1} \ \Delta \mathbf{A}_i]$ is an $L-1 \times k$ matrix where $\Delta \mathbf{A}_i = \mathbf{A}_{li} - \mathbf{A}_{Li}$, $l = 1, 2, \dots, L-1$, that is, the last row of \mathbf{A}_i is subtracted from each row of \mathbf{A}_i , and $\boldsymbol{\beta}$ is a k dimensional vector, $k = (L-1) \times (1 + \dim\{\mathbf{c}_i\}) + a$.

Observe that $\boldsymbol{\beta}$ contains the same last a elements as $\boldsymbol{\delta}$, that is, alternative specific attributes coefficients, but the first $(L-1) \times (1 + \dim\{\mathbf{c}_i\})$ -th elements are $\delta_{jl} - \delta_{jL}$, $j = 1 + \dim\{\mathbf{c}_i\}$, $l = 1, 2, \dots, L-1$, that is, the difference between the coefficients of each qualitative response and the L -th alternative for the individuals' characteristics. This makes it difficult to interpret the multinomial probit coefficients.

Note that in multinomial models, for each alternative specific attribute, it is only required to estimate one coefficient for all alternatives, whereas for individuals' characteristics (non-alternative specific regressors), it is necessary to estimate $L-1$ coefficients (the coefficient of the base alternative is set equal to 0).

The likelihood function in this model is $p(\boldsymbol{\beta}, \boldsymbol{\Omega} | \mathbf{y}, \mathbf{R}) = \prod_{i=1}^N \prod_{l=1}^L p_{il}^{y_{il}}$ where $p_{il} = p(y_{il}^* \geq \max(\mathbf{y}_i^*))$.

We assume independent priors, $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\boldsymbol{\Omega}^{-1} \sim W(\alpha_0, \boldsymbol{\Sigma}_0)$.⁷ We can employ Gibbs sampling in this model because this is a standard

⁶Note that this model is not identified if $\boldsymbol{\Sigma}$ is unrestricted. The likelihood function is the same if a scalar random variable is added to each of the L latent regressions.

⁷ W denotes the Wishart density.

Bayesian linear regression model when data augmentation in \mathbf{w} is used. The posterior conditional distributions are

$$\begin{aligned} \boldsymbol{\beta} | \boldsymbol{\Omega}, \mathbf{w} &\sim N(\boldsymbol{\beta}_n, \mathbf{B}_n), \\ \boldsymbol{\Omega}^{-1} | \boldsymbol{\beta}, \mathbf{w} &\sim W(\alpha_n, \boldsymbol{\Sigma}_n), \\ \text{where } \mathbf{B}_n &= (\mathbf{B}_0^{-1} + \mathbf{X}^{*\top} \mathbf{X}^*)^{-1}, \quad \boldsymbol{\beta}_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \mathbf{X}^{*\top} \mathbf{w}^*), \quad \boldsymbol{\Omega}^{-1} = \\ &\quad \mathbf{C}^\top \mathbf{C}, \quad \mathbf{X}_i^{*\top} = \mathbf{C}^\top \mathbf{R}_i, \quad \mathbf{w}_i^* = \mathbf{C}^\top \mathbf{w}_i, \quad \mathbf{X}^* = \begin{bmatrix} \mathbf{X}_1^* \\ \mathbf{X}_2^* \\ \vdots \\ \mathbf{X}_N^* \end{bmatrix}, \quad \alpha_n = \alpha_0 + N, \quad \boldsymbol{\Sigma}_n = \\ &\quad (\boldsymbol{\Sigma}_0 + \sum_{i=1}^N (\mathbf{w}_i - \mathbf{R}_i \boldsymbol{\beta})^\top (\mathbf{w}_i - \mathbf{R}_i \boldsymbol{\beta}))^{-1}. \end{aligned}$$

We can collapse the multinomial vector \mathbf{y}_i into the indicator variable $d_i = \sum_{l=1}^{L-1} l \times \mathbb{1}_{\max(\mathbf{w}_i) = w_{il}}$.⁸ Then the distribution of $w_{il} | \boldsymbol{\beta}, \boldsymbol{\Omega}^{-1}, d_i$ is an $L-1$ dimensional Gaussian distribution truncated over the appropriate cone in \mathcal{R}^{L-1} . [149] propose drawing from the univariate conditional distributions $w_{il} | \mathbf{w}_{i,-l}, \boldsymbol{\beta}, \boldsymbol{\Omega}^{-1}, d_i \sim TN_{I_{il}}(m_{il}, \tau_{il}^2)$, where

$$I_{il} = \begin{cases} w_{il} > \max(\mathbf{w}_{i,-l}, 0), & d_i = l \\ w_{il} < \max(\mathbf{w}_{i,-l}, 0), & d_i \neq l \end{cases},$$

and permuting the columns and rows of $\boldsymbol{\Omega}^{-1}$ so that the l -th column and row is the last,

$$\boldsymbol{\Omega}^{-1} = \begin{bmatrix} \boldsymbol{\Omega}_{-l,-l} & \boldsymbol{\omega}_{-l,l} \\ \boldsymbol{\omega}_{l,-l} & \omega_{l,l} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{\Omega}_{-l,-l}^{-1} + \tau_{ll}^{-2} \mathbf{f}_l \mathbf{f}_l^\top & -\mathbf{f}_l \tau_{ll}^{-2} \\ -\tau_{ll}^{-2} \mathbf{f}_l^\top & \tau_{ll}^{-2} \end{bmatrix}$$

where $\mathbf{f}_l = \boldsymbol{\Omega}_{-l,-l}^{-1} \boldsymbol{\omega}_{-l,l}$, $\tau_{ll}^2 = \omega_{ll} - \boldsymbol{\omega}_{l,-l} \boldsymbol{\Omega}_{-l,-l}^{-1} \boldsymbol{\omega}_{-l,l}$, $m_{il} = \mathbf{r}_{il}^\top \boldsymbol{\beta} + \mathbf{f}_l^\top (\mathbf{w}_{i,-l} - \mathbf{R}_{i,-l} \boldsymbol{\beta})$, $\mathbf{w}_{i,-l}$ is an $L-2$ dimensional vector of all components of \mathbf{w}_i excluding w_{il} , \mathbf{r}_{il} is the l -th row of \mathbf{R}_i , $l = 1, 2, \dots, L-1$.

The identified parameters are obtained by normalizing with respect to one of the diagonal elements $\frac{1}{\omega_{1,1}^{0.5}} \boldsymbol{\beta}$ and $\frac{1}{\omega_{1,1}} \boldsymbol{\Omega}$.⁹

A warning is required here! This model is an example where we have to make decisions about setting the model in an identified parameter space or unidentified space. The mixing properties of the posterior draws can be better in the latter case [150], this means less computational burden. However, we should recover the identified space in a final stage. In addition, we should take into account that defining priors in the unidentified space may have unintended consequences on the posterior distributions of the identified space [159]. The multinomial probit model that is presented in this section is set in

⁸Observe that the identification issue in this model is due to scaling w_{il} by a positive constant does not change the value of d_i .

⁹Our GUI is based on the *bayesm* package that takes into account this identification restriction to display the outcomes of the posterior chains.

the unidentified space [149]. A version of the multinomial probit in the identified space is presented by [150].

Example: Choice of fishing mode

We used in this application the dataset *3Fishing.csv* from [26, p. 491]. The dependent variable is mutually exclusive alternatives regarding fishing modes (mode), where beach is equal to 1, pier is equal to 2, private boat is equal to 3, and chartered boat (baseline alternative) is equal to 4. In this model, we have

$$\mathbf{X}_i = \begin{pmatrix} 1 & 0 & 0 & 0 & \text{Income}_i & 0 & 0 & 0 & \text{Price}_{i,1} & \text{Catch rate}_{i,1} \\ 0 & 1 & 0 & 0 & 0 & \text{Income}_i & 0 & 0 & \text{Price}_{i,2} & \text{Catch rate}_{i,2} \\ 0 & 0 & 1 & 0 & 0 & 0 & \text{Income}_i & 0 & \text{Price}_{i,3} & \text{Catch rate}_{i,3} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \text{Income}_i & \text{Price}_{i,4} & \text{Catch rate}_{i,4} \end{pmatrix}.$$

In this example chartered boat is the base category, the number of choice categories is four, there are two alternative-specific regressors (price and catch rate), and one non alternative-specific regressor (income). This setting involves the estimation of eight location parameters (β): three intercepts, three for income, one for price, and one for catch rate. This is the order of the posterior chains in our GUI. Note that the location coefficients are set equal to 0 for the baseline category. For multinomial models, we strongly recommend using the last category as the baseline.

We also get posterior estimates for a 3×3 covariance matrix (four alternatives minus one), where the element (1,1) is equal to 1 due to identification restrictions, and elements 2 and 4 are the same, as well as 3 and 7, and 6 and 8, due to symmetry.¹⁰ Observe that this identification restriction implies *Nan* values in [82] and [97] tests for element (1,1) of the covariance matrix, and just eight dependence factors associated with the remaining elements of the covariance matrix.

Once our GUI is displayed (see beginning of this chapter), we should follow Algorithm A8 to run multinomial probit models in our GUI (see Chapter 5 for details), which in turn uses the command *rmnpGibbs* from the *bayesm* package.

We ran 100,000 MCMC iterations plus 10,000 as burn-in with a thinning parameter equal to 5, where all priors use default values for the hyperparameters in our GUI. We found that the 95% credible intervals of the coefficient associated with income for beach and private boat alternatives are equal to (8.58e-06, 8.88e-05) and (3.36e-05, 1.45e-04). This suggests that the probability of choosing these alternatives increases compared to a chartered boat when income increases. In addition, an increase in the price or a decrease in the catch rate for specific fishing alternatives imply lower probabilities of choosing them as the 95% credible intervals are (-9.91e-03, -3.83e-03) and (1.40e-01, 4.62e-01), respectively. However, the posterior chain diagnostics suggest there are convergence issues with the posterior draws (see exercise 5).

¹⁰This is the order in the pdf, eps and csv files that can be downloaded from our GUI.

Algorithm A8 Multinomial probit models

- 1: Select *Univariate Models* on the top panel
- 2: Select *Multinomial Probit* model using the left radio button
- 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
- 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
- 5: Select dependent and independent variables using the *Formula builder* table
- 6: Select the number of the **Base Alternative**
- 7: Select the **Number of choice categorical alternatives**
- 8: Select the **Number of alternative specific variables**
- 9: Select the **Number of Non-alternative specific variables**
- 10: Click the *Build formula* button to generate the formula in **R** syntax.
- 11: Set the hyperparameters: mean vector, covariance matrix, scale matrix and degrees of freedom. This step is not necessary as by default our GUI uses non-informative priors
- 12: Click the *Go!* button
- 13: Analyze results
- 14: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons

R. code. Choice of fishing mode, results

```

1 Iterations = 10005:110000
2 Thinning interval = 5
3 Number of chains = 1
4 Sample size per chain = 20000
5 Quantiles for each variable:
6          2.5%      25%      50%      75%     97.5%
7 cte_1   -5.83e-01 -4.08e-01 -3.22e-01 -2.37e-01 -7.93e-02
8 cte_2   -1.93e-01 -4.14e-02  2.16e-02  7.93e-02  1.93e-01
9 cte_3   -8.15e-01 -5.43e-01 -4.29e-01 -3.33e-01 -1.70e-01
10 NAS_1_1  8.58e-06  3.61e-05  4.95e-05  6.27e-05  8.88e-05
11 NAS_1_2 -3.24e-05 -7.04e-06  5.52e-06  1.93e-05  5.17e-05
12 NAS_1_3  3.36e-05  6.38e-05  8.08e-05  9.99e-05  1.45e-04
13 AS_1    -9.91e-03 -7.90e-03 -6.86e-03 -5.93e-03 -3.83e-03
14 AS_2    1.40e-01  2.25e-01  2.72e-01  3.28e-01  4.62e-01

```

6.5 The multinomial logit model

The multinomial logit model is used to model mutually exclusive discrete outcomes or qualitative response variables. However, this model assumes the independence of irrelevant alternatives (IIA), meaning that the choice between two alternatives does not depend on a third alternative. We consider the multinomial mixed logit model (not to be confused with the random parameters logit model), which accounts for both alternative-varying regressors (conditional) and alternative-invariant regressors (multinomial) simultaneously.¹¹

In this setting there are L mutually exclusive alternatives, and the dependent variable y_{il} is equal to 1 if the l th alternative is chosen by individual i , and 0 otherwise, $l = \{1, 2, \dots, L\}$. The likelihood function is $p(\beta|y, X) = \prod_{i=1}^N \prod_{l=1}^L p_{il}^{y_{il}}$, where the probability that individual i chooses the alternative l is given by $p_{il} := p(y_i = l|\beta, X) = \frac{\exp\{\mathbf{x}_{il}^\top \beta_l\}}{\sum_{j=1}^L \exp\{\mathbf{x}_{ij}^\top \beta_j\}}$, \mathbf{y} and \mathbf{X} are the vector and matrix of the dependent variable and regressors, and β is the vector containing all the coefficients. Remember that coefficients associated with alternative-invariant regressors are set to 0 for the baseline category, and the coefficients associated with the alternative-varying regressors are the same for all the categories. In addition, we assume $\beta \sim N(\beta_0, B_0)$ as prior distribution. Thus, the posterior distribution is $\pi(\beta|y, X) \propto p(\beta|y, X) \times \pi(\beta)$.

As the multinomial logit model does not have a standard posterior distribution, [189] propose a “tailored” independent Metropolis–Hastings algorithm where the proposal distribution is a multivariate Student’s t distribution with v degrees of freedom (tuning parameter), mean equal to the maximum likelihood estimator, and scale equal to the inverse of the Hessian matrix.

Example: Simulation exercise

Let’s do a simulation exercise to check the performance of the Metropolis–Hastings algorithm to perform inference in the multinomial logit model. Assume a situation where there are three alternatives, one alternative-invariant regressor plus the intercept, and three alternative-varying regressors. The population parameters are $\beta_1 = [1 -2.5 0.5 0.8 -3]^\top$, $\beta_2 = [1 -3.5 0.5 0.8 -3]^\top$ and $\beta_3 = [0 0 0.5 0.8 -3]^\top$, the first two elements of the vectors are associated with the intercept and the alternative-invariant regressor, and the last three elements with the alternative-varying regressors. The sample size is 1000, and all regressors are simulated from standard normal distributions.

We can deploy our GUI using the command line at the beginning of this chapter. We should follow Algorithm A9 to run multinomial logit models in our GUI (see Chapter 5 for details):

The following code in **R** shows how to implement the M-H algorithm from scratch. The first part simulates the dataset, the second part builds the

¹¹The multinomial mixed logit model can be implemented as a conditional logit model.

Algorithm A9 Multinomial logit models

- 1: Select *Univariate Models* on the top panel
 - 2: Select *Multinomial Logit* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Select dependent and independent variables using the *Formula builder* table
 - 6: Select the **Base Alternative**
 - 7: Select the **Number of choice categorical alternatives**
 - 8: Select the **Number of alternative specific variables**
 - 9: Select the **Number of Non-alternative specific variables**
 - 10: Click the *Build formula* button to generate the formula in **R** syntax.
 - 11: Set the hyperparameters: mean vector and covariance matrix. This step is not necessary as by default our GUI uses non-informative priors
 - 12: Select the tuning parameter for the Metropolis-Hastings algorithm, that is, the **Degrees of freedom: Multivariate Student's t distribution**
 - 13: Click the *Go!* button
 - 14: Analyze results
 - 15: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

loglikelihood function, and the third part implements the M-H algorithm. We use vague priors centered on zero, and covariance matrix $1000\mathbf{I}_7$. We observe that the posterior estimates closely match the population parameters, and all 95% credible intervals contain the population parameters.

R. code. Simulation of the multinomial logit model

```

1 remove(list = ls())
2 set.seed(12345)
3 # Simulation of data
4 N<-1000 # Sample Size
5 B<-c(0.5,0.8,-3); B1<-c(-2.5,-3.5,0); B2<-c(1,1,0)
6 # Alternative specific attributes of choice 1, for instance,
  price, quality and duration of choice 1
7 X1<-matrix(cbind(rnorm(N,0,1),rnorm(N,0,1),rnorm(N,0,1)),N,
  length(B))
8 # Alternative specific attributes of choice 2, for instance,
  price, quality and duration of choice 2
9 X2<-matrix(cbind(rnorm(N,0,1),rnorm(N,0,1),rnorm(N,0,1)),N,
  length(B))
10 # Alternative specific attributes of choice 3, for instance,
  price, quality and duration of choice 3
11 X3<-matrix(cbind(rnorm(N,0,1),rnorm(N,0,1),rnorm(N,0,1)),N,
  length(B))
12 X4<-matrix(rnorm(N,1,1),N,1)
13 V1<-B2[1]+X1%*%B+B1[1]*X4; V2<-B2[2]+X2%*%B+B1[2]*X4; V3<-B2
  [3]+X3%*%B+B1[3]*X4
14 suma<-exp(V1)+exp(V2)+exp(V3)
15 p1<-exp(V1)/suma; p2<-exp(V2)/suma; p3<-exp(V3)/suma
16 p<-cbind(p1,p2,p3)
17 y<- apply(p,1, function(x) sample(1:3, 1, prob = x, replace =
  TRUE))
18 y1<-y==1; y2<-y==2; y3<-y==3

```

R. code. Simulation of the multinomial logit model

```

1 # Log likelihood
2 log.L<- function(Beta){
3   V1<-Beta[1]+Beta[3]*X4+X1%*%Beta[5:7]
4   V2<-Beta[2]+Beta[4]*X4+X2%*%Beta[5:7]
5   V3<- X3%*%Beta[5:7]
6   suma<-exp(V1)+exp(V2)+exp(V3)
7   p11<-exp(V1)/suma;  p22<-exp(V2)/suma;  p33<-exp(V3)/suma
8   suma2<-NULL
9   for(i in 1:N){
10     suma1<-y1[i]*log(p11[i])+y2[i]*log(p22[i])+y3[i]*log(p33
11       [i])
12     suma2<-c(suma2,suma1)}
13   logL<-sum(suma2)
14   return(-logL)
15 }
16 # Parameters: Proposal
17 k <- 7
18 res.optim<-optim(rep(0, k), log.L, method="BFGS", hessian=
19   TRUE)
20 MeanT <- res.optim$par
21 ScaleT <- as.matrix(Matrix:::forceSymmetric(solve(res.optim$ 
22   hessian))) # Force this matrix to be symmetric
23 # Hyperparameters: Priors
24 B0 <- 1000*diag(k); b0 <- rep(0, k)
25 MHfunction <- function(iter, tuning){
26   Beta <- rep(0, k); Acept <- NULL
27   BetasPost <- matrix(NA, iter, k)
28   pb <- winProgressBar(title = "progress bar", min = 0, max
29     = iter, width = 300)
30   for(s in 1:iter){
31     LogPostBeta <- -log.L(Beta) + mvtnorm::dmvnorm(Beta,
32       mean = b0, sigma = B0, log = TRUE)
33     BetaC <- c(LaplacesDemon::rmvt(n=1, mu = MeanT, S =
34       ScaleT, df = tuning))
35     LogPostBetaC <- -log.L(BetaC) + mvtnorm::dmvnorm(BetaC,
36       mean = b0, sigma = B0, log = TRUE)
37     alpha <- min(exp((LogPostBetaC-mvtnorm::dmvt(BetaC,
38       delta = MeanT, sigma = ScaleT, df = tuning, log = TRUE))-
39       (LogPostBeta-mvtnorm::dmvt(Beta, delta = MeanT, sigma =
40         ScaleT, df = tuning, log = TRUE))), 1)
41     u <- runif(1)
42     if(u <= alpha){
43       Acepti <- 1; Beta <- BetaC
44     }else{
45       Acepti <- 0; Beta <- Beta
46     }
47     BetasPost[s, ] <- Beta; Acept <- c(Acept, Acepti)
48     setWinProgressBar(pb, s, title=paste( round(s/iter*100,
49       0), "% done"))
50   }
51   close(pb); AcepRate <- mean(Acept)
52   Results <- list(AcepRate = AcepRate, BetasPost = BetasPost
53     )
54   return(Results)
55 }
```

R. code. Simulation of the multinomial logit model

```

1 # MCMC parameters
2 mcmc <- 10000; burnin <- 1000; thin <- 5; iter <- mcmc +
  burnin; keep <- seq(burnin, iter, thin); tuning <- 6 # Degrees of freedom
3 ResultsPost <- MHfunction(iter = iter, tuning = tuning)
4 summary(coda::mcmc(ResultsPost$BetasPost[keep[-1], ]))
5 Iterations = 1:2000
6 Thinning interval = 1
7 Number of chains = 1
8 Sample size per chain = 2000
9 1. Empirical mean and standard deviation for each variable,
10 plus standard error of the mean:
11      Mean     SD Naive SE Time-series SE
12 [1,] 0.9711 0.20162 0.004508      0.004508
13 [2,] 0.9742 0.20934 0.004681      0.004681
14 [3,] -2.4350 0.18950 0.004237      0.004137
15 [4,] -3.4195 0.24656 0.005513      0.005513
16 [5,] 0.5253 0.07396 0.001654      0.001654
17 [6,] 0.8061 0.08007 0.001790      0.001790
18 [7,] -3.0853 0.17689 0.003955      0.003955
19 2. Quantiles for each variable:
20      2.5%    25%    50%    75%   97.5%
21 var1 0.5862 0.8367 0.9650 1.1017 1.3683
22 var2 0.5679 0.8310 0.9681 1.1151 1.3761
23 var3 -2.8239 -2.5607 -2.4291 -2.3050 -2.0812
24 var4 -3.9176 -3.5806 -3.4074 -3.2496 -2.9423
25 var5 0.3840 0.4761 0.5250 0.5759 0.6647
26 var6 0.6555 0.7494 0.8064 0.8616 0.9604
27 var7 -3.4476 -3.1991 -3.0777 -2.9641 -2.7500

```

6.6 Ordered probit model

The ordered probit model is used when there is a natural order in the categorical response variable. In this case, there is a latent variable $y_i^* = \mathbf{x}_i^\top \boldsymbol{\beta} + \mu_i$, $\mu_i \stackrel{i.i.d.}{\sim} N(0, 1)$ such that $y_i = l$ if and only if $\alpha_{l-1} < y_i^* \leq \alpha_l$, $l = \{1, 2, \dots, L\}$, where $\alpha_0 = -\infty$, $\alpha_1 = 0$ and $\alpha_L = \infty$.¹² Then,

¹²Identification issues necessitate setting the variance in this model equal to 1 and $\alpha_1 = 0$. Observe that multiplying y_i^* by a positive constant or adding a constant to all of the cut-offs and subtracting the same constant from the intercept does not affect y_i .

$p(y_i = l) = \Phi(\alpha_l - \mathbf{x}_i^\top \boldsymbol{\beta}) - \Phi(\alpha_{l-1} - \mathbf{x}_i^\top \boldsymbol{\beta})$, and the likelihood function is $p(\boldsymbol{\beta}, \boldsymbol{\alpha} | \mathbf{y}, \mathbf{X}) = \prod_{i=1}^N p(y_i = l | \boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{X})$.

There are independent priors of this model, $\pi(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \pi(\boldsymbol{\beta}) \times \pi(\boldsymbol{\gamma})$, where $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\boldsymbol{\gamma} \sim N(\boldsymbol{\gamma}_0, \boldsymbol{\Gamma}_0)$, $\boldsymbol{\gamma} = [\gamma_2 \ \gamma_3 \ \dots \ \gamma_{L-1}]^\top$, such that $\boldsymbol{\alpha} = [\exp\{\gamma_2\} \ \sum_{l=2}^3 \exp\{\gamma_l\} \ \dots \ \sum_{l=2}^{L-1} \exp\{\gamma_l\}]^\top$. The latter structure imposes the ordinal condition in the cut-offs.

This model does not have a standard conditional posterior distribution for $\boldsymbol{\gamma}$ ($\boldsymbol{\alpha}$), but it does have a standard conditional distribution for $\boldsymbol{\beta}$ once data augmentation is used. Then, we can use a Metropolis-within-Gibbs sampling algorithm. In particular, we use Gibbs sampling algorithms to draw $\boldsymbol{\beta}$ and \mathbf{y}^* ,

$$\boldsymbol{\beta} | \mathbf{y}^*, \boldsymbol{\alpha}, \mathbf{X} \sim N(\boldsymbol{\beta}_n, \mathbf{B}_n),$$

where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n(\mathbf{B}_0^{-1}\boldsymbol{\beta}_0 + \mathbf{X}^\top \mathbf{y}^*)$, and $\mathbf{y}_i^* | \boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{y}, \mathbf{X} \sim TN_{(\alpha_{y_i-1}, \alpha_{y_i})}(\mathbf{x}_i^\top \boldsymbol{\beta}, 1)$.

We use a random-walk Metropolis–Hastings algorithm for $\boldsymbol{\gamma}$ that has as proposal a Gaussian distribution with mean equal to the current value, and covariance matrix $s^2(\boldsymbol{\Gamma}_0^{-1} + \hat{\boldsymbol{\Sigma}}_\gamma^{-1})^{-1}$, where $s > 0$ is a tuning parameter, and $\hat{\boldsymbol{\Sigma}}_\gamma$ is the sample covariance matrix associated with $\boldsymbol{\gamma}$ from the maximum likelihood estimation.

Example: Determinants of preventive health care visits

We used the file named *2HealthMed.csv* in this applications. In particular, the dependent variable is *MedVisPrevOr*, which is an ordered variable equal to 1 if the individual did not visit a physician for preventive reasons, 2 if the individual visited once in that year, and so on, until it is equal to 6 for visiting five or more times. The latter category is 1.6% of the sample. Observe that the dependent variable has six categories.

In this example, the set of regressors is given by *SHI*, which an indicator of being in the subsidized health care system (1 means being in the system), sex (*Female*), age (linear and squared), socioeconomic conditions indicator (*Est2* and *Est3*), the lowest is the baseline category, self perception of health status (*Fair*, *Good* and *Excellent*), where *Bad* is the baseline, and education level, primary (*PriEd*), high school (*HighEd*), vocational (*VocEd*), and university (*UnivEd*), *no education* is the baseline category.

We ran this application with 50,000 MCMC iterations plus 10,000 as burn-in, and thinning parameter equal to 5. This setting means 10,000 effective posterior draws. We set $\boldsymbol{\beta}_0 = \mathbf{0}_{11}$, $\mathbf{B}_0 = 1000\mathbf{I}_{11}$, $\boldsymbol{\gamma}_0 = \mathbf{0}_4$, $\boldsymbol{\Gamma}_0 = \mathbf{I}_4$, and the tuning parameter is 1.

We can run the ordered probit models in our GUI following the steps in the Algorithm A10.

The following **R** code shows how to perform inference in this model using the command *rordprobitGibbs* from the *bayesm* library, which is the command that our GUI uses.

Algorithm A10 Ordered probit models

- 1: Select *Univariate Models* on the top panel
- 2: Select *Ordered Probit* model using the left radio button
- 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
- 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
- 5: Select dependent and independent variables using the *Formula builder* table
- 6: Click the *Build formula* button to generate the formula in **R** syntax. Remember that this formula must have -1 to omit the intercept in the specification.
- 7: Set the hyperparameters: mean vectors and covariance matrices. This step is not necessary as by default our GUI uses non-informative priors
- 8: Select the tuning parameter for the Metropolis-Hastings algorithm
- 9: Click the *Go!* button
- 10: Analyze results
- 11: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons

R. code. Determinants of preventive health care visits

```

1 rm(list = ls())
2 set.seed(010101)
3 Data <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/2HealthMed.
  csv", sep = ",", header = TRUE, quote = "")
4 attach(Data)
5 y <- MedVisPrevOr
6 # MedVisPrevOr: Ordered variable for preventive visits to
  # doctors in one year: 1 (none), 2 (once), ... 6 (five or
  # more)
7 X <- cbind(SHI, Female, Age, Age2, Est2, Est3, Fair, Good,
  Excellent, PriEd, HighEd, VocEd, UnivEd)
8 k <- dim(X)[2]
9 L <- length(table(y))
10 # Hyperparameters
11 b0 <- rep(0, k); c0 <- 1000; B0 <- c0*diag(k)
12 gamma0 <- rep(0, L-2); Gamma0 <- diag(L-2)
13 # MCMC parameters
14 mcmc <- 60000+1; thin <- 5; tuningPar <- 1/(L-2)^0.5
15 DataApp <- list(y = y, X = X, k = L)
16 Prior <- list(betabar = b0, A = solve(B0), dstarbar = gamma0
  , Ad = Gamma0)
17 mcmcpars <- list(R = mcmc, keep = 5, s = tuningPar)
18 PostBeta <- bayesm::rordprobitGibbs(Data = DataApp, Prior =
  Prior, Mcmc = mcmcpars)

```

R. code. Determinants of preventive health care visits, results

```

1 BetasPost <- coda::mcmc(PostBeta[["betadraw"]])
2 colnames(BetasPost) <- c("SHI", "Female", "Age", "Age2", "Est2",
   Est3", "Fair", "Good", "Excellent", "PriEd", "HighEd",
   VocEd", "UnivEd")
3 summary(BetasPost)
4 Iterations = 1:12000
5 Thinning interval = 1
6 Number of chains = 1
7 Sample size per chain = 12000
8 1. Empirical mean and standard deviation for each variable,
9 plus standard error of the mean:
10 Mean           SD    Naive SE Time-series SE
11 SHI            0.0654824 2.281e-02 2.082e-04 3.357e-04
12 Female         -0.0374788 1.908e-02 1.742e-04 1.742e-04
13 Age             0.0190336 1.869e-03 1.706e-05 4.576e-05
14 Age2            -0.0002328 2.438e-05 2.225e-07 6.690e-07
15 Est2            0.0949445 2.226e-02 2.032e-04 4.659e-04
16 Est3            -0.1383965 3.411e-02 3.114e-04 3.459e-04
17 Fair             0.6451828 5.375e-02 4.907e-04 3.924e-03
18 Good             0.7343932 4.955e-02 4.523e-04 4.491e-03
19 Excellent        0.9826531 6.393e-02 5.836e-04 5.261e-03
20 PriEd            0.0309418 2.376e-02 2.169e-04 2.221e-04
21 HighEd           -0.1805753 2.910e-02 2.656e-04 3.456e-04
22 VocEd            0.1395760 9.640e-02 8.800e-04 9.291e-04
23 UnivEd           -0.2218120 1.189e-01 1.086e-03 1.086e-03
24 2. Quantiles for each variable:
25          2.5%      25%      50%      75%     97.5%
26 SHI            0.02090  0.04995  0.06540  0.08085  0.11021
27 Female         -0.07463 -0.05042 -0.03777 -0.02456  0.00023
28 Age             0.01550  0.01781  0.01902  0.02023  0.02268
29 Age2            -0.00028 -0.00024 -0.00023 -0.00021 -0.00018
30 Est2            0.05149  0.08004  0.09482  0.10968  0.13933
31 Est3            -0.20559 -0.16144 -0.13815 -0.11563 -0.07179
32 Fair             0.55799  0.61295  0.64148  0.67268  0.74395
33 Good             0.66690  0.70808  0.73032  0.75406  0.81064
34 Excellent        0.88919  0.94770  0.97836  1.01026  1.08460
35 PriEd            -0.01584  0.01493  0.03101  0.04718  0.07732
36 HighEd           -0.23782 -0.20035 -0.18021 -0.16073 -0.12435
37 VocEd            -0.04911  0.07474  0.13811  0.20414  0.33331
38 UnivEd           -0.45381 -0.30239 -0.22193 -0.14148  0.00863
39 # Convergence diagnostics
40 coda::geweke.diag(BetasPost)
41 coda::raftery.diag(BetasPost, q=0.5, r=0.05, s = 0.95)
42 coda::heidel.diag(BetasPost)
43 # Cut offs
44 Cutoffs <- PostBeta[["cutdraw"]]
45 summary(Cutoffs)
46 coda::geweke.diag(Cutoffs)
47 coda::heidel.diag(Cutoffs)
48 coda::raftery.diag(Cutoffs[,-1], q=0.5, r=0.05, s = 0.95)

```

The results suggest that older individuals (at decreasing rate) in the subsidized health program, characterized in the second socioeconomic status with increasing good self perception of health condition, and not having high school as their highest education degree, have a higher probability of visiting a physician for preventive health aims. Convergence diagnostics look well, except for the self health perception draws.

We also got the posterior estimates of the cutoffs in the ordered probit model. These estimates are necessary to calculate the probability that an individual is in a specific category of visiting physicians. Due to identification restrictions, the first cutoff is set equal to 0. That is why we have *Nan* values in [82] and [97] tests, and we observe only four values in the [173] test, which correspond to the remaining free cutoffs. It seems that these cutoff estimates have some convergence issues when taking as diagnostic tool the [173] test. Their dependence factors are also very high.

6.7 Negative binomial model

The dependent variable in the negative binomial model is a nonnegative integer or count. In contrast to the Poisson model, the negative binomial model takes into account over-dispersion. The Poisson model has equal mean and variance (equi-dispersion).

We assume that $y_i \stackrel{i.n.d.}{\sim} NB(\gamma, \theta_i)$, that is, the density function for individual i is $\frac{\Gamma(y_i + \gamma)}{\Gamma(\gamma)y_i!}(1 - \theta_i)^{y_i}\theta_i^\gamma$, where the success probability is $\theta_i = \frac{\gamma}{\lambda_i + \gamma}$, $\lambda_i = \exp\{\mathbf{x}_i^\top \boldsymbol{\beta}\}$ is the mean, and $\gamma = \exp\{\alpha\}$ is the target for number of successful trials, or dispersion parameter.

We assume independent priors for this model are $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\alpha \sim G(\alpha_0, \delta_0)$.¹³

This model does not have standard conditional posterior distributions, so [189] use a random-walk Metropolis–Hastings algorithm where the proposal distribution for $\boldsymbol{\beta}$ is Gaussian centered at the current stage with covariance matrix $s_{\boldsymbol{\beta}}^2 \hat{\Sigma}_{\boldsymbol{\beta}}$ where $s_{\boldsymbol{\beta}}$ is a tuning parameter and $\hat{\Sigma}_{\boldsymbol{\beta}}$ is the maximum likelihood covariance estimator. In addition, the proposal for α is normal centered at the current value, with variance $s_\alpha^2 \hat{\sigma}_\alpha^2$ where s_α is a tuning parameter and $\hat{\sigma}_\alpha^2$ is the maximum likelihood variance estimator.

Example: Simulation exercise

Let's do a simulation exercise to check the performance of the M-H algorithms in the negative binomial model. There are two regressors, $x_{i1} \sim U(0, 1)$ and $x_{i2} \sim N(0, 1)$, and the intercept. The dispersion parameter is $\gamma = \exp\{1.2\}$, and $\boldsymbol{\beta} = [1 \ 1 \ 1]^\top$. The sample size is 1,000.

¹³ G denotes a gamma density.

We run this simulation using 10,000 MCMC iterations, a burn-in equal to 1,000, and a thinning parameter equal to 5. We set vague priors for the location parameters, particularly, $\beta_0 = \mathbf{0}_3$ and $B_0 = 1000\mathbf{I}_3$, and $\alpha_0 = 0.5$ and $\delta_0 = 0.1$, which are the default values in the *rnegbinRw* command from *bayesm* package in **R**. In addition, the tuning parameters of the Metropolis–Hastings algorithms are $s_\beta = 2.93/k^{1/2}$ and $s_\alpha = 2.93$, which are also the default parameters in *rnegbinRw*, k is the number of location parameters.

We can run the negative binomial models in our GUI following the steps in the Algorithm A11.

Algorithm A11 Negative binomial models

- 1: Select *Univariate Models* on the top panel
 - 2: Select *Negative Binomial (Poisson)* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Select dependent and independent variables using the *Formula builder* table
 - 6: Click the *Build formula* button to generate the formula in **R** syntax. You can modify the formula in the **Main equation** box using valid arguments of the *formula* command structure in **R**
 - 7: Set the hyperparameters: mean vector, covariance matrix, shape and scale parameters. This step is not necessary as by default our GUI uses non-informative priors
 - 8: Select the tuning parameters for the Metropolis-Hastings algorithms
 - 9: Click the *Go!* button
 - 10: Analyze results
 - 11: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

The following **R** code shows how to perform inference in the negative binomial model programming the M-H algorithms from scratch. We ask to estimate this example using the *rnegbinRw* command in exercise 8.

We observe from the results that all 95% credible intervals encompass the population parameters, and the posterior means are very close to the population parameters.

R. code. Simulation of the negative binomial model

```
1 rm(list = ls())
2 set.seed(010101)
3 N <- 2000 # Sample size
4 x1 <- runif(N); x2 <- rnorm(N)
5 X <- cbind(1, x1, x2); k <- dim(X)[2]; B <- rep(1, k)
6 alpha <- 1.2; gamma <- exp(alpha); lambda <- exp(X%*%B)
7 y <- rnbnom(N, mu = lambda, size = gamma)
8 # log likelihood
9 logLik <- function(par){
10   alpha <- par[1]; beta <- par[2:(k+1)]
11   gamma <- exp(alpha)
12   lambda <- exp(X%*%beta)
13   logLikNB <- sum(sapply(1:N, function(i){dnbinom(y[i], size
14     = gamma, mu = lambda[i], log = TRUE)}))
15   return(-logLikNB)
16 }
16 # Parameters: Proposal
17 par0 <- rep(0.5, k+1)
18 res.optim <- optim(par0, logLik, method="BFGS", hessian=TRUE
19 )
20 res.optim$par
20 res.optim$convergence
21 Covar <- solve(res.optim$hessian)
22 CovarBetas <- Covar[2:(k+1),2:(k+1)]
23 VarAlpha <- Covar[1:1]
24 # Hyperparameters: Priors
25 B0 <- 1000*diag(k); b0 <- rep(0, k)
26 alpha0 <- 0.5; delta0 <- 0.1
```

R. code. Simulation of the negative binomial model, M-H algorithm

```

1 # Metropolis-Hastings function
2 MHfunction <- function(iter, sbeta, salpha){
3   Beta <- rep(0, k); Acept1 <- NULL; Acept2 <- NULL
4   BetasPost <- matrix(NA, iter, k); alpha <- 1
5   alphaPost <- rep(NA, iter); par <- c(alpha, Beta)
6   pb <- winProgressBar(title = "progress bar", min = 0, max
7     = iter, width = 300)
8   for(s in 1:iter){
9     LogPostBeta <- -logLik(par) + dgamma(alpha, shape =
10       alpha0, scale = delta0, log = TRUE) + mvtnorm::dmvnorm(
11         Beta, mean = b0, sigma = B0, log = TRUE)
12     BetaC <- c(MASS::mvrnorm(1, mu = Beta, Sigma = sbeta^2 *
13       CovarBetas))
14     parC <- c(alpha, BetaC)
15     LogPostBetaC <- -logLik(parC) + dgamma(alpha, shape =
16       alpha0, scale = delta0, log = TRUE) + mvtnorm::dmvnorm(
17         BetaC, mean = b0, sigma = B0, log = TRUE)
18     alpha1 <- min(exp((LogPostBetaC - mvtnorm::dmvnorm(BetaC
19       , mean = Beta, sigma = sbeta^2*CovarBetas, log = TRUE))-
20       (LogPostBeta - mvtnorm::dmvnorm(Beta, mean = Beta,
21         sigma = sbeta^2*CovarBetas, log = TRUE))),1)
22     u1 <- runif(1)
23     if(u1 <= alpha1){Acept1i <- 1; Beta <- BetaC}else{
24       Acept1i <- 0; Beta <- Beta
25     }
26     par <- c(alpha, Beta)
27     LogPostBeta <- -logLik(par) + dgamma(alpha, shape =
28       alpha0, scale = delta0, log = TRUE) + mvtnorm::dmvnorm(
29         Beta, mean = b0, sigma = B0, log = TRUE)
30     alphaC <- rnorm(1, mean = alpha, sd = salpha*VarAlpha
31       ^0.5)
32     parC <- c(alphaC, Beta)
33     LogPostBetaC <- -logLik(parC) + dgamma(alphaC, shape =
34       alpha0, scale = delta0, log = TRUE) + mvtnorm::dmvnorm(
35         Beta, mean = b0, sigma = B0, log = TRUE)
36     alpha2 <- min(exp((LogPostBetaC - dnorm(alphaC, mean =
37       alpha, sd = salpha*VarAlpha^0.5, log = TRUE))-
38       (LogPostBeta - dnorm(alpha, mean = alpha, sd = salpha*
39         VarAlpha^0.5, log = TRUE))),1)
40     u2 <- runif(1)
41     if(u2 <= alpha2){Acept2i <- 1; alpha <- alphaC}else{
42       Acept2i <- 0; alpha <- alpha
43     }
44     BetasPost[s, ] <- Beta; alphaPost[s] <- alpha
45     Acept1 <- c(Acept1, Acept1i); Acept2 <- c(Acept2,
46     Acept2i)
47     setWinProgressBar(pb, s, title=paste( round(s/iter*100,
48       0), "% done"))
49   }
50   close(pb)
51   AcepRateBeta <- mean(Acept1); AcepRateAlpha <- mean(Acept2
52     )
53   Results <- list(AcepRateBeta = AcepRateBeta, AcepRateAlpha
54     = AcepRateAlpha, BetasPost = BetasPost, alphaPost =
55       alphaPost)
56   return(Results)
57 }
```

R. code. Simulation of the negative binomial model, results

```

1 # MCMC parameters
2 mcmc <- 10000
3 burnin <- 1000
4 thin <- 5
5 iter <- mcmc + burnin
6 keep <- seq(burnin, iter, thin)
7 sbeta <- 2.93/sqrt(k); salpha <- 2.93
8 # Run M-H
9 ResultsPost <- MHfunction(iter = iter, sbeta = sbeta, salpha
10 = salpha)
11 ResultsPost$AcepRateBeta
12 ResultsPost$AcepRateAlpha
13 summary(coda::mcmc(ResultsPost$BetasPost[keep[-1], ]))
14 Iterations = 1:2000
15 Thinning interval = 1
16 Number of chains = 1
17 Sample size per chain = 2000
18 1. Empirical mean and standard deviation for each variable,
19 plus standard error of the mean:
20      Mean        SD   Naive SE Time-series SE
21 [1,] 1.0270 0.04799 0.0010730      0.0014727
22 [2,] 0.9981 0.07752 0.0017333      0.0024262
23 [3,] 0.9677 0.02343 0.0005239      0.0007182
24 2. Quantiles for each variable:
25      2.5%    25%    50%    75% 97.5%
26 var1 0.9343 0.9943 1.0255 1.0592 1.122
27 var2 0.8445 0.9448 0.9980 1.0520 1.144
28 var3 0.9242 0.9512 0.9678 0.9839 1.013
29 summary(coda::mcmc(ResultsPost$alphaPost[keep[-1]]))
30 Iterations = 1:2000
31 Thinning interval = 1
32 Number of chains = 1
33 Sample size per chain = 2000
34 1. Empirical mean and standard deviation for each variable,
35 plus standard error of the mean:
36      Mean        SD   Naive SE Time-series SE
37 1.282664      0.058769 0.001314      0.001427
38 2. Quantiles for each variable:
39      2.5%    25%    50%    75% 97.5%
40 1.173 1.242 1.282 1.320 1.407

```

6.8 Tobit model

The dependent variable is partially observed in Tobit models due to sampling schemes, whereas the regressors are completely observed. In particular,

$$y_i = \begin{cases} L, & y_i^* < L \\ y_i^*, & L \leq y_i^* < U \\ U, & y_i^* \geq U \end{cases},$$

where $y_i^* \stackrel{i.n.d.}{\sim} N(\mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2)$.¹⁴

We use conjugate independent priors $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$, and data augmentation using \mathbf{y}_C^* such that $y_{C_i}^* \stackrel{i.n.d.}{\sim} N(\mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2)$, $y_{C_i} = \{y_{C_i^L}^* \cup y_{C_i^U}^*\}$ are lower and upper censored data. This allows implementing the Gibbs sampling algorithm [37]. Then,

$$\pi(\boldsymbol{\beta}, \sigma^2, \mathbf{y}^* | \mathbf{y}, \mathbf{X}) \propto \prod_{i=1}^N \left[\mathbb{1}_{y_i=L} \mathbb{1}_{y_{C_i^L}^* < L} + \mathbb{1}_{L \leq y_i < U} + \mathbb{1}_{y_i=U} \mathbb{1}_{y_{C_i^U}^* \geq U} \right] \\ \times N(y_i^* | \mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2) \times N(\boldsymbol{\beta} | \boldsymbol{\beta}_0, \mathbf{B}_0) \times IG(\sigma^2 | \alpha_0/2, \delta_0/2)$$

The posterior distributions are

$$y_{C_i}^* | \boldsymbol{\beta}, \sigma^2, \mathbf{y}, \mathbf{X} \sim \begin{cases} TN_{(-\infty, L)}(\mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2), & y_i = L \\ TN_{[U, \infty)}(\mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2), & y_i = U \end{cases},$$

$$\boldsymbol{\beta} | \sigma^2, \mathbf{y}, \mathbf{X} \sim \mathcal{N}(\boldsymbol{\beta}_n, \sigma^2 \mathbf{B}_n),$$

$$\sigma^2 | \boldsymbol{\beta}, \mathbf{y}, \mathbf{X} \sim IG(\alpha_n/2, \delta_n/2),$$

where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \sigma^{-2} \mathbf{X}^\top \mathbf{X})^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \sigma^{-2} \mathbf{X}^\top \mathbf{y}^*)$, $\alpha_n = \alpha_0 + N$ and $\delta_n = \delta_0 + (\mathbf{y}^* - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y}^* - \mathbf{X}\boldsymbol{\beta})$.

Example: The market value of soccer players in Europe continues

We continue the example of the market value of soccer players from Section 6.1. We specify the same equation, but assume the sample is censored from below, and have just information of soccer players whose market value is higher than one million euros. The dependent variable is $\log(ValueCens)$, and the left censoring point is 13.82.

The Algorithm A12 shows how to estimate Tobit models in our GUI. Our GUI uses the command *MCMCTobit* from the package *MCMCpack*.

We run this application using the same hyperparameters that we set in

¹⁴We can set L or U equal to $-\infty$ or ∞ to model data censored in just one side.

Algorithm A12 Tobit models

- 1: Select *Univariate Models* on the top panel
 - 2: Select *Tobit* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Select dependent and independent variables using the *Formula builder* table
 - 6: Click the *Build formula* button to generate the formula in **R** syntax. You can modify the formula in the **Main equation** box using valid arguments of the *formula* command structure in **R**
 - 7: Set the left and right censoring points. To censor above only, specify *-Inf* in the left censoring box, and to censor below only, specify *Inf* in the right censoring box
 - 8: Set the hyperparameters: mean vector, covariance matrix, shape and scale parameters. This step is not necessary as by default our GUI uses non-informative priors
 - 9: Click the *Go!* button
 - 10: Analyze results
 - 11: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

the example of Section 6.1. All results seem similar to those in the example of linear models. In addition, the posterior chains seem to achieve good diagnostics.

R. code. The value of soccer player with left censoring

```

1 rm(list = ls()); set.seed(010101)
2 Data <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/1
  ValueFootballPlayers.csv", sep = ",", header = TRUE,
  quote = "")
3 attach(Data)
4 y <- log(ValueCens)
5 X <- cbind(1, Perf, Age, Age2, NatTeam, Goals, Exp, Exp2)
6 k <- dim(X)[2]
7 N <- dim(X)[1]
8 # Hyperparameters
9 d0 <- 0.001; a0 <- 0.001
10 b0 <- rep(0, k); c0 <- 1000; B0 <- c0*diag(k)
11 B0i <- solve(B0)
12 # MCMC parameters
13 mcmc <- 50000
14 burnin <- 10000
15 tot <- mcmc + burnin
16 thin <- 1
17 # Posterior distributions using packages: MCMCpack sets the
  # model in terms of the precision matrix
18 posterior <- MCMCpack::MCMCtobit(y~X-1, b0=b0, B0 = B0i, c0
  = a0, d0 = d0, burnin = burnin, mcmc = mcmc, thin =
  thin, below = 13.82, above = Inf)
19 summary(coda::mcmc(posterior))
20 Iterations = 1:50000
21 Thinning interval = 1
22 Number of chains = 1
23 Sample size per chain = 50000
24 1. Empirical mean and standard deviation for each variable,
  plus standard error of the mean:
25
26      Mean        SD  Naive SE Time-series SE
27 X       1.045830 2.641038 1.181e-02     1.673e-02
28 XPerf    0.033990 0.004515 2.019e-05     2.247e-05
29 XAge     1.025099 0.213368 9.542e-04     1.335e-03
30 XAge2    -0.021871 0.004004 1.791e-05     2.542e-05
31 XNatTeam 0.847495 0.125924 5.631e-04     6.393e-04
32 XGoals   0.010088 0.001649 7.377e-06     7.691e-06
33 XExp     0.174383 0.069948 3.128e-04     3.846e-04
34 XExp2    -0.005652 0.002970 1.328e-05     1.561e-05
35 sigma2   0.982906 0.095965 4.292e-04     6.727e-04
36 2. Quantiles for each variable:
37      2.5%       25%       50%       75%      97.5%
38 X      -4.174794 -0.725691  1.076420  2.840533  6.1935618
39 XPerf   0.025110  0.030949  0.033980  0.037003  0.0428650
40 XAge    0.608620  0.880648  1.023043  1.168486  1.4480001
41 XAge2   -0.029801 -0.024556 -0.021822 -0.019164 -0.0140990
42 XNatTeam 0.603953  0.762394  0.846461  0.932056  1.0960274
43 XGoals   0.006875  0.008977  0.010091  0.011197  0.0133323
44 XExp     0.038752  0.127167  0.173880  0.221355  0.3122043
45 XExp2    -0.011483 -0.007623 -0.005654 -0.003662  0.0001615
46 sigma2   0.811953  0.915246  0.977257  1.043158  1.1879232

```

R. code. The value of soccer player with left censoring, Gibbs sampler

```

1 # Gibbs sampling functions
2 Xtx <- t(X)%*%X
3 PostBeta <- function(Yl, sig2){
4   Bn <- solve(B0i + sig2^(-1)*Xtx)
5   bn <- Bn%*%(B0i%*%b0 + sig2^(-1)*t(X)%*%Yl)
6   Beta <- MASS::mvrnorm(1, bn, Bn)
7   return(Beta)
8 }
9 PostYl <- function(Beta, L, U, i){
10   Ylmean <- X[i, ]%*%Beta
11   if(y[i] == L){
12     Yli <- truncnorm::rtruncnorm(1, a = -Inf, b = L, mean =
13       Ylmean, sd = sig2^0.5)
14   }else{
15     if(y[i] == U){
16       Yli <- truncnorm::rtruncnorm(1, a = U, b = Inf, mean =
17         Ylmean, sd = sig2^0.5)
18     }else{
19       Yli <- y[i]
20     }
21   }
22   return(Yli)
23 }
24 PostSig2 <- function(Beta, Yl){
25   dn <- d0 + t(Yl - X%*%Beta)%*%(Yl - X%*%Beta)
26   sig2 <- invgamma::rinvgamma(1, shape = an/2, rate = dn/2)
27   return(sig2)
28 }
29 PostBetas <- matrix(0, mcmc+burnin, k); Beta <- rep(0, k)
30 PostSigma2 <- rep(0, mcmc+burnin); sig2 <- 1
31 L <- log(1000000); U <- Inf
32 # create progress bar in case that you want to see
33 # iterations progress
34 pb <- winProgressBar(title = "progress bar", min = 0, max =
35   tot, width = 300)
36 for(s in 1:tot){
37   Yl <- sapply(1:N, function(i){PostYl(Beta = Beta, L = L, U =
38     U, i)})
39   Beta <- PostBeta(Yl = Yl, sig2)
40   sig2 <- PostSig2(Beta = Beta, Yl = Yl)
41   PostBetas[s,] <- Beta; PostSigma2[s] <- sig2
42   setWinProgressBar(pb, s, title=paste( round(s/tot*100, 0),
43     "% done"))
44 }
45 close(pb)
46 keep <- seq((burnin+1), tot, thin)
47 PosteriorBetas <- PostBetas[keep,]
48 colnames(PosteriorBetas) <- c("Intercept", "Perf", "Age", "
49   Age2", "NatTeam", "Goals", "Exp", "Exp2")
50 summary(coda::mcmc(PosteriorBetas))
51 summary(coda::mcmc(PostSigma2[keep]))

```

6.9 Quantile regression

In quantile regression the location parameters vary according to the quantile of the dependent variable. Let $q_\tau(\mathbf{x}_i) = \mathbf{x}_i^\top \boldsymbol{\beta}_\tau$ denote the τ -th ($0 < \tau < 1$) quantile regression function of y_i given \mathbf{x}_i such that $y_i = \mathbf{x}_i^\top \boldsymbol{\beta}_\tau + \mu_i$ where $\int_{-\infty}^0 f_\tau(\mu_i) d\mu_i = \tau$, that is, the τ -th quantile of μ_i is 0.

In particular, [127] propose $f_\tau(\mu_i) = \tau(1-\tau) \exp\{-\mu_i(\tau - \mathbb{1}_{\mu_i < 0})\}$ (asymmetric Laplace distribution), where $\mu_i(\tau - \mathbb{1}_{\mu_i < 0})$ is the check (loss) function. These authors propose the location-scale mixture of normals with a representation given by $\mu_i = \theta e_i + \psi \sqrt{e_i} z_i$ where $\theta = \frac{1-2\tau}{\tau(1-\tau)}$, $\psi^2 = \frac{2}{\tau(1-\tau)}$, $e_i \sim E(1)$ and $z_i \sim N(0, 1)$, $e_i \perp z_i$.¹⁵ As a consequence of this representation and the fact that the sample is i.i.d., $p(\mathbf{y}|\boldsymbol{\beta}_\tau, \mathbf{e}, \mathbf{X}) \propto \left(\prod_{i=1}^n e_i^{-1/2}\right) \exp\left\{-\sum_{i=1}^N \frac{(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}_\tau - \theta e_i)^2}{2\psi^2 e_i}\right\}$.

Taking as prior a normal distribution for $\boldsymbol{\beta}_\tau$, that is, $\boldsymbol{\beta}_\tau \sim N(\boldsymbol{\beta}_{\tau 0}, \mathbf{B}_{\tau 0})$, and using data augmentation for \mathbf{e} , we can implement a Gibbs sampling algorithm in this model. The posterior distributions are

$$\begin{aligned} \boldsymbol{\beta}_\tau | \mathbf{e}, \mathbf{y}, \mathbf{X} &\sim N(\boldsymbol{\beta}_{n\tau}, \mathbf{B}_{n\tau}), \\ e_i | \boldsymbol{\beta}_\tau, \mathbf{y}, \mathbf{X} &\sim GIG(1/2, \alpha_{ni}, \delta_{ni}), \end{aligned} \quad ^{16}$$

where $\mathbf{B}_{n\tau} = \left(\mathbf{B}_{\tau 0}^{-1} + \sum_{i=1}^N \frac{\mathbf{x}_i \mathbf{x}_i^\top}{\psi^2 e_i}\right)^{-1}$, $\boldsymbol{\beta}_{n\tau} = \mathbf{B}_{n\tau} \left(\mathbf{B}_{\tau 0}^{-1} \boldsymbol{\beta}_{\tau 0} + \sum_{i=1}^N \frac{\mathbf{x}_i (y_i - \theta e_i)}{\psi^2 e_i}\right)$, $\alpha_{ni} = (y_i - \mathbf{x}_i^\top \boldsymbol{\beta}_\tau)^2 / \psi^2$ and $\delta_{ni} = 2 + \theta^2 / \psi^2$.

Example: The market value of soccer players in Europe continues

We continue the example of the market value of soccer players from Section 6.1. Now, we want to know if the marginal effect of having been in the national team changes with the quantile of the market value of top soccer players in Europe. Thus, we have same regressors as in the example in the previous section, but analyze the effects in the 0.5-th and 0.9-th quantiles of the *NatTeam*.

The Algorithm A13 shows how to estimate Tobit models in our GUI. Our GUI uses the command *MCMCquantreg* from the package *MCMCpack*. The next are code shows to perform this using this package.

The results show that at the median market value, the 95% credible interval for the coefficient associated with *national team* is (0.34, 1.02), with a posterior mean of 0.69. At the 0.9 quantile, these values are (0.44, 1.59) and 1.03, respectively. It appears that being on the national team increases the market value of more expensive players more significantly on average, although there is some overlap in the credible intervals.

¹⁵ E denotes an exponential density.

¹⁶ GIG denotes a generalized inverse Gaussian density.

Algorithm A13 Quantile regression

- 1: Select *Univariate Models* on the top panel
- 2: Select *Tobit* model using the left radio button
- 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
- 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
- 5: Select dependent and independent variables using the *Formula builder* table
- 6: Click the *Build formula* button to generate the formula in **R** syntax. You can modify the formula in the **Main equation** box using valid arguments of the *formula* command structure in **R**
- 7: Set the left and right censoring points. To censor above only, specify *-Inf* in the left censoring box, and to censor below only, specify *Inf* in the right censoring box
- 8: Set the hyperparameters: mean vector, covariance matrix, shape and scale parameters. This step is not necessary as by default our GUI uses non-informative priors
- 9: Click the *Go!* button
- 10: Analyze results
- 11: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons

R. code. The value of soccer player, quantile regression

```

1 rm(list = ls()); set.seed(010101)
2 Data <- read.csv("https://raw.githubusercontent.com/
    besmarter/BSTApp/refs/heads/master/DataApp/1
    ValueFootballPlayers.csv", sep = ",", header = TRUE,
    quote = "")
3 attach(Data)
4 y <- log(ValueCens)
5 X <- cbind(1, Perf, Age, Age2, NatTeam, Goals, Exp, Exp2)
6 k <- dim(X)[2]; N <- dim(X)[1]
7 # Hyperparameters
8 b0 <- rep(0, k); c0 <- 1000; B0 <- c0*diag(k); B0i <- solve(
    B0)
9 # MCMC parameters
10 mcmc <- 50000; burnin <- 10000
11 tot <- mcmc + burnin; thin <- 1

```

R. code. The value of soccer player, quantile regression

```

1 # Quantile
2 q <- 0.5
3 posterior05 <- MCMCpack::MCMCquantreg(y~X-1, tau = q, b0=b0
   , B0 = B0i, burnin = burnin, mcmc = mcmc, thin = thin,
   below = 13.82, above = Inf)
4 summary(coda::mcmc(posterior05))
5 1. Empirical mean and standard deviation for each variable,
6 plus standard error of the mean:
7          Mean        SD Naive SE Time-series SE
8 X       7.374348 2.916758 1.304e-02      2.291e-02
9 XPerf   0.029325 0.005938 2.655e-05      5.241e-05
10 XAge    0.550633 0.241596 1.080e-03      1.903e-03
11 XAge2   -0.012027 0.004597 2.056e-05      3.643e-05
12 XNatTeam 0.685275 0.170768 7.637e-04      1.587e-03
13 XGoals   0.010608 0.002425 1.085e-05      1.951e-05
14 XExp    0.092561 0.085499 3.824e-04      6.799e-04
15 XExp2   -0.002979 0.003877 1.734e-05      2.941e-05
16 2. Quantiles for each variable:
17          2.5%     25%     50%     75%    97.5%
18 X       1.74594  5.405772  7.351090  9.2994982 13.216024
19 XPerf   0.01753  0.025340  0.029354  0.0333155  0.040906
20 XAge    0.06845  0.390780  0.553187  0.7139430  1.016664
21 XAge2   -0.02087 -0.015141 -0.012095 -0.0089849 -0.002813
22 XNatTeam 0.34645  0.572081  0.686735  0.7996086  1.016189
23 XGoals   0.00578  0.009055  0.010562  0.0121751  0.015403
24 XExp    -0.06761  0.034149  0.089632  0.1482128  0.267536
25 XExp2   -0.01094 -0.005456 -0.002891 -0.0004099  0.004466
26 q <- 0.9
27 posterior09 <- MCMCpack::MCMCquantreg(y~X-1, tau = q, b0=b0
   , B0 = B0i, burnin = burnin, mcmc = mcmc, thin = thin,
   below = 13.82, above = Inf)
28 summary(coda::mcmc(posterior09))
29 1. Empirical mean and standard deviation for each variable,
30 plus standard error of the mean:
31          Mean        SD Naive SE Time-series SE
32 X       8.860043 5.933902 2.654e-02      6.686e-02
33 XPerf   0.019663 0.010241 4.580e-05      1.140e-04
34 XAge    0.532823 0.483213 2.161e-03      5.397e-03
35 XAge2   -0.012328 0.008864 3.964e-05      9.620e-05
36 XNatTeam 1.033384 0.294271 1.316e-03      3.389e-03
37 XGoals   0.008781 0.004340 1.941e-05      4.991e-05
38 XExp    0.132760 0.177677 7.946e-04      2.125e-03
39 XExp2   -0.002713 0.007639 3.416e-05      8.531e-05
40 2. Quantiles for each variable:
41          2.5%     25%     50%     75%    97.5%
42 X      -2.7084122 4.829341  8.821031 12.850002 20.66191
43 XPerf  -0.0001863 0.012782  0.019605  0.026495  0.03991
44 XAge   -0.4180422 0.207000  0.532486  0.858221  1.48632
45 XAge2  -0.0300400 -0.018216 -0.012235 -0.006345  0.00497
46 XNatTeam 0.4384014 0.840123  1.038986  1.234456  1.59482
47 XGoals  0.0019513 0.005661  0.008176  0.011327  0.01881
48 XExp   -0.2320608 0.014760  0.139452  0.256663  0.46053
49 XExp2  -0.0162717 -0.007954 -0.003198  0.002031  0.01385

```

6.10 Bayesian bootstrap regression

We implement the Bayesian bootstrap [191] for linear regression models. In particular, the Bayesian bootstrap simulates the posterior distributions assuming that the sample cumulative distribution function (cdf) is the population cdf (this assumption is also implicit in the frequentist bootstrap [63]).

Given $y_i \stackrel{i.n.d.}{\sim} \mathcal{F}$ where \mathcal{F} does not define a particular parametric family of distributions, $i = 1, 2, \dots, N$, but sets $E(Y_i|\mathbf{x}_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$, such that \mathbf{x}_i is a K dimensional vector of regressors and $\boldsymbol{\beta}$ is a K dimensional vector of parameters, the Bayesian bootstrap generates posterior probabilities for each y_i where the values of Y that are not observed have zero posterior probability.

The algorithm to implement the Bayesian bootstrap is the following:

Algorithm A14 Bayesian bootstrap from scratch in linear regression

- 1: Draw $\mathbf{g} \sim Dir(\alpha_1, \alpha_2, \dots, \alpha_N)$ such that $\alpha_i = 1 \forall i$.
- 2: $\mathbf{g} = (g_1, g_2, \dots, g_N)$ is the vector of probabilities to attach to $(y_1, \mathbf{x}_1^\top), (y_2, \mathbf{x}_2^\top), \dots, (y_n, \mathbf{x}_N^\top)$ for each Bayesian bootstrap replication.
- 3: Sample (y_i, \mathbf{x}_i^\top) N times with replacement and probabilities g_i , $i = 1, 2, \dots, N$.
- 4: Estimate $\boldsymbol{\beta}$ using ordinary least squares in the model $E(\mathbf{Y}|\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$, \mathbf{y} being an S_1 dimensional vector of realizations of \mathbf{Y} , and \mathbf{X} an $S_1 \times k$ matrix from the previous stage.*
- 5: Repeat this process S times.
- 6: The distribution of $\boldsymbol{\beta}^{(s_2)}$ is the Bayesian distribution of $\boldsymbol{\beta}$.

*Ordinary least squares is the posterior mean of $\boldsymbol{\beta}$ using Jeffrey's prior in a linear regression.

Example: Simulation exercise

Let's do a simulation exercise to check the performance of the Algorithm A14 to perform inference using the Bayesian bootstrap. The data generating process is given by two regressors that distribute normal standard. The location vector is $\boldsymbol{\beta} = [1 \ 1]^\top$, and variance $\sigma^2 = 1$, the sample size is 1,000.

Algorithm A15 shows how to use our GUI to run the Bayesian bootstrap. Our GUI is based on the *bayesboot* command from *bayesboot* package in **R**. Exercise 11 asks about using this package to perform inference in this simulation, and compared the results with the ones that we get using our GUI setting $S = 10000$.

The following **R** code shows how to program the Bayesian bootstrap from scratch. We observe from the results that all 95% credible intervals encompass the population parameters, and the posterior means are close to the population parameters.

Algorithm A15 Bayesian bootstrap in linear regression

- 1: Select *Univariate Models* on the top panel
- 2: Select *Bootstrap* model using the left radio button
- 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend. You should see a preview of the dataset
- 4: Select number of bootstrap replications using the *Range sliders*
- 5: Select dependent and independent variables using the *Formula builder* table
- 6: Click the *Build formula* button to generate the formula in **R** syntax. You can modify the formula in the **Main equation** box using valid arguments of the *formula* command structure in **R**
- 7: Click the *Go!* button
- 8: Analyze results
- 9: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons

R. code. Bayesian bootstrap

```

1 rm(list = ls()); set.seed(010101)
2 N <- 1000; x1 <- runif(N); x2 <- rnorm(N)
3 X <- cbind(x1, x2); k <- dim(X)[2]
4 B <- rep(1, k+1); sig2 <- 1
5 u <- rnorm(N, 0, sig2); y <- cbind(1, X)%*%B + u
6 data <- as.data.frame(cbind(y, X))
7 names(data) <- c("y", "x1", "x2")
8 Reg <- function(d){
9   Reg <- lm(y ~ x1 + x2, data = d)
10  Bhat <- Reg$coef
11  return(Bhat)
12 }
13 S <- 10000; alpha <- 1
14 BB <- function(S, df, alpha){
15   Betas <- matrix(NA, S, dim(df)[2])
16   N <- dim(df)[1]
17   pb <- winProgressBar(title = "progress bar", min = 0, max
18   = S, width = 300)
19   for(s in 1:S){
20     g <- LaplacesDemon::rdirichlet(N, alpha)
21     ids <- sample(1:N, size = N, replace = TRUE, prob = g)
22     datas <- df[ids,]
23     names(datas) <- names(df)
24     Bs <- Reg(d = datas)
25     Betas[s, ] <- Bs
26     setWinProgressBar(pb, s, title=paste( round(s/S*100, 0),
27     "% done"))
28   }
29   close(pb)
30   return(Betas)
31 }
32 BBs <- BB(S = S, df = data, alpha = alpha)
33 summary(coda::mcmc(BBs))

```

R. code. Bayesian bootstrap, results

```

1 Iterations = 1:10000
2 Thinning interval = 1
3 Number of chains = 1
4 Sample size per chain = 10000
5 1. Empirical mean and standard deviation for each variable,
6 plus standard error of the mean:
7          Mean        SD  Naive SE Time-series SE
8 [1,] 0.9172 0.06386 0.0006386      0.0006291
9 [2,] 1.1733 0.10888 0.0010888      0.0010201
10 [3,] 1.0137 0.03386 0.0003386      0.0003386
11 2. Quantiles for each variable:
12        2.5%       25%       50%       75%   97.5%
13 var1 0.7926 0.8743 0.9169 0.9599 1.043
14 var2 0.9608 1.0984 1.1739 1.2468 1.389
15 var3 0.9473 0.9910 1.0136 1.0365 1.079

```

6.11 Summary

In this chapter, we present the core univariate regression models and demonstrate how to perform Bayesian inference using Markov Chain Monte Carlo methods. Specifically, we cover a mix of algorithms: Gibbs sampling, Metropolis-Hastings, nested M-H, and M-H-within-Gibbs. These algorithms form the foundation to perform Bayesian inference in more complex settings using cross-sectional data sets.

6.12 Exercises

1. Get the posterior conditional distributions of the Gaussian linear model assuming independent priors $\pi(\beta, \sigma^2) = \pi(\beta) \times \pi(\sigma^2)$, where $\beta \sim N(\beta_0, \mathbf{B}_0)$ and $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$.
2. Given the model $y_i \sim N(\mathbf{x}_i^\top \beta, \sigma^2/\tau_i)$ (Gaussian linear model with heteroskedasticity) with independent priors, $\pi(\beta, \sigma^2, \boldsymbol{\tau}) = \pi(\beta) \times \pi(\sigma^2) \times \prod_{i=1}^N \pi(\tau_i)$, where $\beta \sim N(\beta_0, \mathbf{B}_0)$, $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$ and $\tau_i \sim G(v/2, v/2)$. Show that $\beta | \sigma^2, \boldsymbol{\tau}, \mathbf{y}, \mathbf{X} \sim$

$N(\beta_n, \mathbf{B}_n)$, $\sigma^2 | \beta, \tau, \mathbf{y}, \mathbf{X} \sim IG(\alpha_n/2, \delta_n/2)$ and $\tau_i | \beta, \sigma^2, \mathbf{y}, \mathbf{X} \sim G(v_{1n}/2, v_{2in}/2)$, where $\tau = [\tau_1 \dots \tau_n]^\top$, $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \sigma^{-2} \mathbf{X}^\top \Psi \mathbf{X})^{-1}$, $\beta_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \beta_0 + \sigma^{-2} \mathbf{X}^\top \Psi \mathbf{y})$, $\alpha_n = \alpha_0 + N$, $\delta_n = \delta_0 + (\mathbf{y} - \mathbf{X}\beta)^\top \Psi (\mathbf{y} - \mathbf{X}\beta)$, $v_{1n} = v + 1$, $v_{2in} = v + \sigma^{-2}(y_i - \mathbf{x}_i^\top \beta)^2$, and $\Psi = \text{diagonal } \{\tau_i\}$.

3. The market value of soccer players in Europe continues

Use the setting of the previous exercise to perform inference using a Gibbs sampling algorithm of the the market value of soccer players in Europe setting $v = 5$ and same other hyperparameters as the homoscedastic case. Is there any meaningful difference for the coefficient associated with the national team compared to the application in the homoscedastic case?

4. Example: Determinants of hospitalization continues

Program a Gibbs sampling algorithm in the application of determinants of hospitalization.

5. Choice of the fishing mode continues

- Run the Algorithm A8 of the book to show the results of the Geweke [82], Raftery [173] and Heidelberger [97] tests using our GUI.
- Use the command *rmnpGibbs* to do the example of the choice of the fishing mode.

6. Simulation exercise of the multinomial logit model continues

Perform inference in the simulation of the multinomial logit model using the command *rmnlIndepMetrop* from the *bayesm* package of **R** and using our GUI.

7. Simulation of the ordered probit model

Simulate an ordered probit model where the first regressor distributes $N(6, 5)$ and the second distributes $G(1, 1)$, the location parameter is $\beta = [0.5 \ -0.25 \ 0.5]^\top$, and the cutoffs is the vector $\alpha = [0 \ 1 \ 2.5]^\top$. Program from scratch a Metropolis-within-Gibbs sampling algorithm to perform inference in this simulation.

8. Simulation of the negative binomial model continues

Perform inference in the simulation of the negative binomial model using the *bayesm* package in **R** software.

9. The market value of soccer players in Europe continues

Perform the application of the value of soccer players with left censoring at one million Euros in our GUI using the Algorithm A12, and the hyperparameters of the example.

10. **The market value of soccer players in Europe continues**
Program from scratch the Gibbs sampling algorithm in the example of the market value of soccer players at the 0.75 quantile.
11. Use the *bayesboot* package to perform inference in the simulation exercise of Section 6.10, and compared the results with the ones that we get using our GUI setting $S = 10000$.

7

Multivariate models

We describe how to perform Bayesian inference in multivariate response models: multivariate regression, seemingly unrelated regression, instrumental variables, and multivariate probit model. In particular, we show the posterior distributions of the parameters, and perform some applications and simulations. Again, we show how to perform inference in these models using three levels of programming skills: GUI, packages, and programming from scratch the algorithms. Finally, there are some mathematical and computational exercises.

Remember that we can run our GUI typing

R code. How to display our graphical user interface

```
1 shiny::runGitHub("besmarter/BSTApp", launch.browser = T)
```

in the **R** package console or any **R** code editor. However, users should see Chapter 5 for seeing other options.

7.1 Multivariate regression

A complete presentation of this model is given in Section 3.4. We show here the setting, and the posterior distributions for facility in exposition. In particular, there are M multiply dependent variables which share the same set of regressors, and their stochastic errors are contemporaneously correlated. In particular, $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_M]$ is a $N \times M$ matrix that is generated by $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{U}$ where \mathbf{X} is an $N \times K$ matrix of regressors, $\mathbf{B} = [\boldsymbol{\beta}_1 \boldsymbol{\beta}_2 \dots \boldsymbol{\beta}_M]$ is a $K \times M$ matrix of parameters, and $\mathbf{U} = [\boldsymbol{\mu}_1 \boldsymbol{\mu}_2 \dots \boldsymbol{\mu}_M]$ is a matrix of stochastic random errors such that $\boldsymbol{\mu}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, $i = 1, 2, \dots, N$ is each row of \mathbf{U} .

The prior is given by $\mathbf{B}|\Sigma \sim N(\mathbf{B}_0, \mathbf{V}_0, \Sigma)$ and $\Sigma \sim IW(\Psi_0, \alpha_0)$. Therefore, the conditional posterior distributions are

$$\mathbf{B}|\Sigma, \mathbf{Y}, \mathbf{X} \sim N(\mathbf{B}_n, \mathbf{V}_n, \Sigma),$$

$$\Sigma|\mathbf{Y}, \mathbf{X} \sim IW(\Psi_n, \alpha_n),$$

where $\mathbf{V}_n = (\mathbf{X}^\top \mathbf{X} + \mathbf{V}_0^{-1})^{-1}$, $\mathbf{B}_n = \mathbf{V}_n(\mathbf{V}_0^{-1}\mathbf{B}_0 + \mathbf{X}^\top \mathbf{X}\hat{\mathbf{B}})$, $\hat{\mathbf{B}} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y}$, $\Psi_n = \Psi_0 + \mathbf{S} + \mathbf{B}_0^\top \mathbf{V}_0^{-1}\mathbf{B}_0 + \hat{\mathbf{B}}^\top \mathbf{X}^\top \mathbf{X}\hat{\mathbf{B}} - \mathbf{B}_n^\top \mathbf{V}_n^{-1}\mathbf{B}_n$, and $\alpha_n = \alpha_0 + N$. We can use a Gibbs sampling algorithm in this model since the conditional posterior distributions are standard.

Example: The effect of institutions on per capita gross domestic product

To illustrate multivariate regression models, we used the dataset provided by [1], who analyzed the effect of property rights on economic growth.

Let's assume that the point of departure is the following *simultaneous structural* economic model.¹

$$\log(\text{pcGDP95}_i) = \beta_1 + \beta_2 \text{PAER}_i + \beta_3 \text{Africa} + \beta_4 \text{Asia} + \beta_5 \text{Other} + u_{1i}, \quad (7.1)$$

$$\text{PAER}_i = \alpha_1 + \alpha_2 \log(\text{pcGDP95}_i) + \alpha_3 \log(\text{Mort}_i) + u_{2i}, \quad (7.2)$$

where *pcGDP95*, *PAER* and *Mort* are the per capita gross domestic product (GDP) in 1995, the average index of protection against expropriation between 1985 and 1995, and the settler mortality rate during the time of colonization, respectively. *Africa*, *Asia* and *Other* are dummies for continents, with *America* as the baseline group.

In this model, there is *reverse (simultaneous) causality* due to the contemporaneous effect of *GDP* on *PAER*, and vice versa.² Therefore, estimation of the Equations 7.1 and 7.2 without taking into account this phenomenon generates posterior mean estimates that are *biased* and *inconsistent* from a sampling (frequentist) point of view.³ A potential strategy to tackle this issue is to estimate the *reduced-form* model, that is, a model without *simultaneous causality* where all *endogenous variables* are function of *exogenous variables*. The former variables are determined within the model ($\log(\text{pcGDP95}_i)$ and *PAER* in this example), and the latter are determined outside the model ($\log(\text{Mort}_i)$, *Africa*, *Asia*, and *Other* in this example).

¹This is a model that captures the potential underlying economic relationship between the variables.

²*Simultaneous causality* is the most controversial causation issue from a philosophy of science perspective. The root of the issue is that causation is typically based on the time sequence of cause and effect.

³Observe that $E[u_1 \text{PAER}] \neq 0$, which means failing to meet a necessary requirement to get *unbiased* and *consistent* estimators of β . See Exercise 1.

Replacing Equation 7.2 into Equation 7.1, and solving for $\log(\text{pcGDP95})$,

$$\log(\text{pcGDP95}_i) = \pi_1 + \pi_2 \log(\text{Mort}_i) + \pi_3 \text{Africa} + \pi_4 \text{Asia} + \pi_5 \text{Other} + e_{1i}. \quad (7.3)$$

Then, replacing Equation 7.3 into Equation 7.2, and solving for PAER ,

$$\text{PAER}_i = \gamma_1 + \gamma_2 \log(\text{Mort}_i) + \gamma_3 \text{Africa} + \gamma_4 \text{Asia} + \gamma_5 \text{Other} + e_{2i}, \quad (7.4)$$

where $\pi_2 = \frac{\beta_2 \alpha_3}{1 - \beta_2 \alpha_2}$ and $\gamma_2 = \frac{\alpha_3}{1 - \beta_2 \alpha_2}$ given $\beta_2 \alpha_2 \neq 1$, that is, independent equations (see Exercise 2).

Observe that equations 7.3 and 7.4 have the form of a multivariate regression model where the common set of regressors is $\mathbf{X} = [\log(\text{Mort}) \text{ Africa} \text{ Asia} \text{ Other}]$ and $\mathbf{Y} = [\log(\text{pcGDP95}) \text{ PAER}]$. Thus, we can estimate this model using the setup of this section.

Thus, we estimate in a first stage the parameters from the *reduced-form* model (Equations 7.3 and 7.4), but the main interest is the parameters of the *structural* model (Equations 7.1 and 7.2). Thus, a valid question is if we can recover the *structural* parameters from the *reduced-form* parameters. There are two criteria to respond this question: the order condition, which is necessary, and the rank condition, which is necessary and sufficient.

The order condition

Given a system of equations with M endogenous variables, and K exogenous variables (including the intercept), there are two ways to assess the order condition:

- The parameters of an equation in the system are identified if there are at least $M - 1$ variables excluded from the equation (*exclusion restrictions*). The equation is *exactly identified* if the number of excluded variables is $M - 1$, and is *over identified* if the number of excluded variables is greater than $M - 1$.
- The parameters of equation m in the system are identified if $K - K_m \geq M_m - 1$, where K_m and M_m are the number of exogenous and endogenous variables in equation m , respectively. The m -th equation is *exactly identified* if $K - K_m = M_m - 1$, and *over identified* if $K - K_m > M_m - 1$.

We can see from Equations 7.1 and 7.2 in this example that $K = 5$, $M = 2$, $K_1 = 4$, $K_2 = 2$, $M_1 = 2$ and $M_2 = 2$. This means that $K - K_1 = 1 = M - 1$ and $K - K_2 = 3 > M - 1 = 1$, that is, the order condition says that both equations satisfy the necessary condition of identification, the first equation would be *exactly identified*, and the second equation would be *over identified*. Observe that there is one excluded variable from the first equation, and there are three excluded variables from the second equation.

The rank condition

The rank condition (necessary and sufficient) says that given a *structural* model with M equations (M endogenous variables), an equation is identified if and only if there is at least one determinant different from zero from a $(M - 1) \times (M - 1)$ matrix built using the excluded variables in the analyzed equation, but included in any other equation of the system.

It is useful to build the *identification matrix* to implement the *rank* condition. Table 7.1 shows this matrix in this example.

TABLE 7.1

Identification matrix.

log(pcGDP95)	PAER	Constant	log(Mort)	Africa	Asia	Other
1	$-\beta_2$	$-\beta_1$	0	$-\beta_3$	$-\beta_4$	$-\beta_5$
$-\alpha_2$	1	$-\alpha_1$	$-\alpha_3$	0	0	0

The only excluded variable in the log(pcGDP95) equation is log(Mort). Then, there is just one matrix that can be built using the excluded variables from this equation $[-\alpha_3]$ (see column 4 in Table 7.1). Thus, the determinant of this matrix is $-\alpha_3$, and as far as this coefficient is different to zero, that is, that the mortality rate is relevant in the PAER equation ($\alpha_3 \neq 0$), the coefficients in log(pcGDP95) equation are *exactly identified*. For instance, $\beta_2 = \frac{\pi_2}{\gamma_2}$, which is the effect of property rights on GDP, is exactly identified.

Observe the importance of excluding log(Mort) from the log(pcGDP95) equation, but including log(Mort) in the PAER equation. This is called *exclusion restriction*, and it is the requirement of having an exogenous source of variability in the PAER equation that helps to identify the log(pcGDP95) equation. Having relevant exogenous sources of variability is a very important aspect in identification, estimation and inference of *structural* parameters.

Regarding the identification of the *structural* parameters in the PAER equation, there are three potential matrices that can be constructed: $[-\beta_3]$, $[-\beta_4]$ and $[-\beta_5]$ (see columns 5, 6 and 7 in Table 7.1), as far as any of these parameters are relevant in the log(pcGDP95) equation, we achieve identification of the PAER equation. In this case, this equation is *over identified*, that is, there are many ways to find the parameters in this equations. For instance, $\alpha_2 = \gamma_3/\pi_3 = \gamma_4/\pi_4 = \gamma_5/\pi_5$ (see Exercise 2).

In general, trying to recover the *structural* parameters from the *reduced-form* parameters can be challenging due to the requirement of relevant identification restrictions that can be hard to find in some applications.⁴

We set non-informative priors in this example, $B_0 = [\mathbf{0}_5 \ \mathbf{0}_5]$, $V_0 = 100I_K$, $\Psi_0 = 5I_2$ and $\alpha_0 = 5$.⁵ Once our GUI is displayed (see beginning of this

⁴Good text books at introductory level for identification in linear systems are [93, Chap. 19] and [223, Chap. 16].

⁵Observe that we are setting the priors in the *reduced-form* model; this may have unintended consequences for the posterior distributions of the *structural* parameters, which are ultimately the parameters researchers are interested in. See [126, p. 302] for good references in this topic.

chapter), we should follow Algorithm A16 to run multivariate linear models in our GUI (see Chapter 5 for details, particularly how to set the data set):

Algorithm A16 Multivariate linear model

- 1: Select *Multivariate Models* on the top panel
 - 2: Select *Simple Multivariate* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Select the number of dependent variables in the box **Number of endogenous variables: m**
 - 6: Select the number of independent variables (including the intercept) in the box **Number of exogenous variables: k**
 - 7: Set the hyperparameters: mean vectors, covariance matrix, degrees of freedom, and the scale matrix. This step is not necessary as by default our GUI uses non-informative priors
 - 8: Click the *Go!* button
 - 9: Analyze results
 - 10: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

The following **R** code shows how to perform the Gibss sampling algorithm in this example using the dataset *4Institutions.csv*. We ask to run this example using the *rmultireg* command from the *bayesm* package as an exercise. We find that the posterior mean *structural* effect of property rights on GDP is 0.98, and the 95% credible interval is (0.56, 2.87). This means that there is evidence supporting a positive effect of property rights on gross domestic product.

R code. The effect of institutions on per capita GDP

```

1 rm(list = ls())
2 set.seed(12345)
3 DataInst <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/4Institutions
  .csv", sep = ",", header = TRUE, quote = "")
4 attach(DataInst)
5 Y <- cbind(logpcGDP95, PAER)
6 X <- cbind(1, logMort, Africa, Asia, Other)
7 M <- dim(Y)[2]
8 K <- dim(X)[2]
9 N <- dim(Y)[1]
10 # Hyperparameters
11 B0 <- matrix(0, K, M)
12 c0 <- 100
13 V0 <- c0*diag(K)
14 Psi0 <- 5*diag(M)
15 a0 <- 5
16 # Posterior parameters
17 Bhat <- solve(t(X)%*%X)%*%t(X)%*%Y
18 S <- t(Y - X%*%Bhat)%*%(Y - X%*%Bhat)
19 Vn <- solve(solve(V0) + t(X)%*%X)
20 Bn <- Vn%*%(solve(V0)%*%B0 + t(X)%*%X%*%Bhat)
21 Psin <- Psi0 + S + t(B0)%*%solve(V0)%*%B0 + t(Bhat)%*%t(X)%*
  %X%*%Bhat - t(Bn)%*%solve(Vn)%*%Bn
22 an <- a0 + N
23 #Posterior draws
24 s <- 10000 #Number of posterior draws
25 SIGs <- replicate(s, LaplacesDemon::rinvwishart(an, Psin))
26 BsCond <- sapply(1:s, function(s) {MixMatrix::rmatrixnorm(n
  = 1, mean=Bn, U = Vn, V = SIGs[, , s])})
27 summary(coda::mcmc(t(BsCond)))
28 SIGMs <- t(sapply(1:s, function(l) {gdata::lowerTriangle(
  SIGs[,,l], diag=TRUE, byrow=FALSE)}))
29 summary(coda::mcmc(SIGMs))
30 hdiBs <- HDInterval::hdi(t(BsCond), credMass = 0.95) #
  Highest posterior density credible interval
31 hdiBs
32 hdiSIG <- HDInterval::hdi(SIGMs, credMass = 0.95) # Highest
  posterior density credible interval
33 hdiSIG
34 beta2 <- BsCond[2,]/BsCond[7,]
35 summary(coda::mcmc(beta1)) # Effect of property rights on
  GDP
36 Iterations = 1:10000
37 Thinning interval = 1
38 Number of chains = 1
39 Sample size per chain = 10000
40 1. Empirical mean and standard deviation for each variable,
  plus standard error of the mean:
41 Mean           SD      Naive SE Time-series SE
42 0.9796        16.8430     0.1684       0.1684
43 2. Quantiles for each variable:
44 2.5%    25%    50%    75%   97.5%
45 0.5604  0.7984  0.9677  1.2329  2.8709

```

7.2 Seemingly unrelated regression

In seemingly unrelated regression (SUR) models there are M dependent variables with potentially different regressors such that the stochastic errors are contemporaneously correlated. This is $\mathbf{y}_m = \mathbf{X}_m \boldsymbol{\beta}_m + \boldsymbol{\mu}_m$, where \mathbf{y}_m is a N -dimensional vector, \mathbf{X}_m is a matrix of dimension $N \times K_m$ of regressors, $\boldsymbol{\beta}_m$ is a K_m -dimensional vector of location parameters, and $\boldsymbol{\mu}_m$ is a N -dimensional vector of stochastic errors, $m = 1, 2, \dots, M$.

Setting $\boldsymbol{\mu}_i = [\mu_{i1} \mu_{i2} \dots \mu_{iM}]^\top$ such that $\boldsymbol{\mu}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, and stacking the M equations, we can write $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\mu}$ where $\mathbf{y} = [\mathbf{y}_1^\top \mathbf{y}_2^\top \dots \mathbf{y}_M^\top]^\top$ is a MN -dimensional vector, $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^\top \boldsymbol{\beta}_2^\top \dots \boldsymbol{\beta}_M^\top]^\top$ is a K dimensional vector, $K = \sum_{m=1}^M K_m$, \mathbf{X} is an $MN \times K$ block diagonal matrix composed of \mathbf{X}_m , that is,

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{X}_M \end{bmatrix},$$

and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^\top \boldsymbol{\mu}_2^\top \dots \boldsymbol{\mu}_M^\top]^\top$ is a MN -dimensional vector of stochastic errors such that $\boldsymbol{\mu} \sim N(\mathbf{0}, \boldsymbol{\Sigma} \otimes \mathbf{I}_N)$. Then,

$$p(\boldsymbol{\beta}, \boldsymbol{\Sigma} | \mathbf{y}, \mathbf{X}) \propto |\boldsymbol{\Sigma}|^{-N/2} \exp \left\{ -\frac{1}{2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_N) (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\}.$$

Using independent priors $\pi(\boldsymbol{\beta}) \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\pi(\boldsymbol{\Sigma}^{-1}) \sim W(\alpha_0, \boldsymbol{\Psi}_0)$, the posterior distributions are

$$\boldsymbol{\beta} | \boldsymbol{\Sigma}, \mathbf{y}, \mathbf{X} \sim N(\boldsymbol{\beta}_n, \mathbf{B}_n),$$

$$\boldsymbol{\Sigma}^{-1} | \boldsymbol{\beta}, \mathbf{y}, \mathbf{X} \sim W(\alpha_n, \boldsymbol{\Psi}_n),$$

where $\mathbf{B}_n = (\mathbf{X}^\top (\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_N) \mathbf{X} + \mathbf{B}_0^{-1})^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \mathbf{X}^\top (\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_N) \mathbf{y})$, $\alpha_n = \alpha_0 + N$ and $\boldsymbol{\Psi}_n = (\boldsymbol{\Psi}_0^{-1} + \mathbf{U}^\top \mathbf{U})^{-1}$, where \mathbf{U} is an $N \times M$ matrix whose columns are $\mathbf{y}_m - \mathbf{X}_m \boldsymbol{\beta}_m$.

We can demonstrate, through straightforward yet tedious algebra, that by defining $\mathbf{y}_i = [y_{i1} \ y_{i2} \ \dots \ y_{iM}]$ and

$$\mathbf{X}_i = \begin{bmatrix} x_{1i}^\top & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & x_{2i}^\top & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & x_{Mi}^\top \end{bmatrix},$$

we alternatively have $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \sum_{i=1}^N \mathbf{X}_i^\top \boldsymbol{\Sigma}^{-1} \mathbf{X}_i)^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \sum_{i=1}^N \mathbf{X}_i^\top \boldsymbol{\Sigma}^{-1} \mathbf{y}_i)^{-1}$ and $\boldsymbol{\Psi}_n = (\boldsymbol{\Psi}_0^{-1} + \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i^\top \boldsymbol{\beta})(\mathbf{y}_i - \mathbf{X}_i^\top \boldsymbol{\beta})^\top)^{-1}$.

Observe that we have standard conditional posteriors, thus, we can employ a Gibbs sampling algorithm to get the posterior draws.

Example: Utility demand

Let's use the dataset *Utilities.csv* to estimate a seemingly unrelated regression model for utilities. We use the same setting as in Exercise 14 in Chapter 3 where we ask to estimate a multivariate regression model omitting households with no consumption in any utility. We see in this exercise that not all regressors are relevant for the demand of electricity, water and gas. Thus, we estimate the following model:

$$\begin{aligned}\log(\text{electricity}_i) &= \beta_1 + \beta_2 \log(\text{electricity price}_i) + \beta_3 \log(\text{water price}_i) \\ &\quad + \beta_4 \log(\text{gas price}_i) + \beta_5 \text{IndSocio1}_i + \beta_6 \text{IndSocio2}_i + \beta_7 \text{Altitude}_i \\ &\quad + \beta_8 \text{Nrooms}_i + \beta_9 \text{HouseholdMem}_i + \beta_{10} \log(\text{Income}_i) + \mu_{i1} \\ \log(\text{water}_i) &= \alpha_1 + \alpha_2 \log(\text{electricity price}_i) + \alpha_3 \log(\text{water price}_i) \\ &\quad + \alpha_4 \log(\text{gas price}_i) + \alpha_5 \text{IndSocio1}_i + \alpha_6 \text{IndSocio2}_i \\ &\quad + \alpha_7 \text{Nrooms}_i + \alpha_8 \text{HouseholdMem}_i + \mu_{i2} \\ \log(\text{gas}_i) &= \gamma_1 + \gamma_2 \log(\text{electricity price}_i) + \gamma_3 \log(\text{water price}_i) \\ &\quad + \gamma_4 \log(\text{gas price}_i) + \gamma_5 \text{IndSocio1}_i + \gamma_6 \text{IndSocio2}_i + \gamma_7 \text{Altitude}_i \\ &\quad + \gamma_8 \text{Nrooms}_i + \gamma_9 \text{HouseholdMem}_i + \mu_{i3},\end{aligned}$$

where electricity, water and gas are the monthly consumption of electricity (kWh), water (m^3) and gas (m^3) of Colombian households. There is information of 2103 households regarding average prices of electricity (USD/kWh), water (USD/ m^3) and gas (USD/ m^3), indicators of socioeconomic conditions of the neighborhood where the household is located (IndSocio1 is the lowest and IndSocio3 is the highest), an indicator if the household is located in a municipality that is above 1000 meters above the sea level, the number of rooms in the house, the number of members of the households, and monthly income (USD).

Since there are different sets of regressors in each equation and we suspect correlation between the stochastic errors of the three equations, we should estimate a seemingly unrelated regressions (SUR) model. We expect unobserved correlation in these equations because we are modelling utilities, and in some cases, a single provider handles all three services and issues one bill.

Algorithm A17 shows how to estimate SUR models in our GUI. Our GUI uses the command *rsurGibbs* from the *bayesm* package in R software. See Chapter 5 for details, particularly how to set the data set, and templates in our GitHub repository (<https://github.com/besmarter/BSTAApp>) in the folders **DataApp** and **DataSim**.

The following code shows how to program this application using this package. We use 10000 MCMC iterations, $\beta_0 = \mathbf{0}_{27}$, $B_0 = 100I_{27}$, $\alpha_0 = 5$ and $\Psi = 5I_3$.

We find that the posterior median estimates of the own-price elasticities of

demand of electricity, water and gas are -1.88, -0.36 and -0.62, where there are not 95% credible intervals that encompass 0. This means that a 1% increase in the prices of electricity, water and gas imply a 1.88%, 0.36% and 0.62% decrease in the monthly consumption of these utilities, respectively.⁶ In general, there is evidence supporting the relevance of all regressors in these equations, except a few exceptions, and unobserved correlation in the demand of these services supporting the relevance of a SUR model in this application.

Algorithm A17 Seemingly unrelated regression

- 1: Select *Multivariate Models* on the top panel
 - 2: Select *Seemingly Unrelated Regression* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Select the number of dependent variables in the box **Number of endogenous variables: m**
 - 6: Select the number of independent variables in the box **TOTAL number Exogenous Variables: k**. This is the sum of all exogenous variables over all equations including intercepts. In the example of **Utility demand**, it is equal to 27
 - 7: Set the hyperparameters: mean vectors, covariance matrix, degrees of freedom, and the scale matrix. This step is not necessary as by default our GUI uses non-informative priors
 - 8: Click the *Go!* button
 - 9: Analyze results
 - 10: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

⁶This is an example where there can be concerns regarding *biased* and *inconsistent* posterior mean estimates, for instance, due to *reverse causality* between quantity and demand. These concerns are valid; although, we are using micro-level data, which implies no demand-supply simultaneity. In addition, the utility providers are operating in regulated natural monopoly markets, this implies no endogeneity due to searching provider strategies. Finally, we took prices directly from provider records, this avoids price measurement errors [179].

R code. Utility demand in Colombia

```

1 rm(list = ls())
2 set.seed(010101)
3 library(dplyr)
4 DataUt <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTAApp/refs/heads/master/DataApp/Utilities.csv"
  , sep = ",", header = TRUE, quote = "")
5 DataUtEst <- DataUt %>%
6 filter(Electricity != 0 & Water != 0 & Gas != 0)
7 attach(DataUtEst)
8 y1 <- log(Electricity); y2 <- log(Water); y3 <- log(Gas)
9 X1 <- cbind(1, LnPriceElect, LnPriceWater, LnPriceGas,
  IndSocio1, IndSocio2, Altitude, Nrooms, HouseholdMem,
  Lnincome)
10 X2 <- cbind(1, LnPriceElect, LnPriceWater, LnPriceGas,
  IndSocio1, IndSocio2, Nrooms, HouseholdMem)
11 X3 <- cbind(1, LnPriceElect, LnPriceWater, LnPriceGas,
  IndSocio1, IndSocio2, Altitude, Nrooms, HouseholdMem)
12 regdata <- NULL
13 regdata[[1]] <- list(y = y1, X = X1); regdata[[2]] <- list(y
  = y2, X = X2); regdata[[3]] <- list(y = y3, X = X3)
14 M <- length(regdata); K1 <- dim(X1)[2]; K2 <- dim(X2)[2]; K3
  <- dim(X3)[2]
15 K <- K1 + K2 + K3
16 # Hyperparameters
17 b0 <- rep(0, K); c0 <- 100; B0 <- c0*diag(K); V <- 5*diag(M)
  ; a0 <- M
18 Prior <- list(betabar = b0, A = solve(B0), nu = a0, V = V)
19 #Posterior draws
20 S <- 10000; keep <- 1; Mcmc <- list(R = S, keep = keep)
21 PosteriorDraws <- bayesm::rsurGibbs(Data = list(regdata =
  regdata), Mcmc = Mcmc, Prior = Prior)

```

R code. Utility demand in Colombia, results

```

1 Bs <- PosteriorDraws[["betadraw"]]
2 Names <- c("Const", "LnPriceElect", "LnPriceWATER", "
3     LnPriceGas", "IndSocio1", "IndSocio2",
4 "Altitude", "Nrooms", "HouseholdMem", "Lnincome", "Const",
4 "LnPriceElect", "LnPriceWATER", "LnPriceGas", "IndSocio1", "
5     IndSocio2",
5 "Nrooms", "HouseholdMem", "Const",
6 "LnPriceElect", "LnPriceWATER", "LnPriceGas", "IndSocio1", "
7     IndSocio2",
7 "Altitude", "Nrooms", "HouseholdMem")
8 colnames(Bs) <- Names
9 summary(coda::mcmc(Bs))
10 summary(PosteriorDraws[["Sigmadraw"]])
11 2. Quantiles for each variable:
12          2.5%    25%    50%    75%   97.5%
13 Const      0.44452  1.03120  1.342407  1.65192  2.25376
14 LnPriceElect -2.39679 -2.06328 -1.882706 -1.70369 -1.36996
15 LnPriceWATER -0.44221 -0.38678 -0.356850 -0.32669 -0.26969
16 LnPriceGas   -0.21655 -0.13777 -0.098191 -0.05902  0.01872
17 IndSocio1    -0.87630 -0.78653 -0.737701 -0.68840 -0.59675
18 IndSocio2    -0.24601 -0.18286 -0.151440 -0.11896 -0.05681
19 Altitude     -0.27080 -0.23838 -0.220742 -0.20385 -0.17259
20 Nrooms       0.04596  0.06178  0.070023  0.07835  0.09422
21 HouseholdMem 0.06600  0.07994  0.086857  0.09411  0.10785
22 Lnincome     0.03836  0.05421  0.062957  0.07165  0.08717
23 Const        0.88957  1.73496  2.169638  2.62170  3.47216
24 LnPriceElect -0.81956 -0.31624 -0.054075  0.21132  0.71842
25 LnPriceWATER -0.49559 -0.40995 -0.364248 -0.32026 -0.23639
26 LnPriceGas    0.06075  0.16754  0.226690  0.28570  0.39476
27 IndSocio1    -0.64203 -0.50302 -0.427819 -0.35226 -0.21315
28 IndSocio2    -0.50401 -0.40949 -0.359821 -0.31063 -0.21199
29 Nrooms       0.05688  0.08023  0.093139  0.10555  0.12968
30 HouseholdMem 0.10041  0.12065  0.131506  0.14260  0.16314
31 Const        -2.28569 -1.58566 -1.220078 -0.84612 -0.14787
32 LnPriceElect -2.42484 -2.01228 -1.797269 -1.57889 -1.16396
33 LnPriceWATER -0.10684 -0.03923 -0.004088  0.03153  0.09905
34 LnPriceGas    -0.76526 -0.67445 -0.625899 -0.57734 -0.48125
35 IndSocio1    -0.91381 -0.80243 -0.744909 -0.68577 -0.57341
36 IndSocio2    -0.31791 -0.24388 -0.203300 -0.16415 -0.09012
37 Altitude     0.24896  0.29099  0.311668  0.33256  0.37278
38 Nrooms       0.06050  0.07921  0.089386  0.09943  0.11793
39 HouseholdMem 0.14467  0.16144  0.170024  0.17843  0.19431
40 summary(coda::mcmc(PosteriorDraws[["Sigmadraw"]]))
41 2. Quantiles for each variable:
42          2.5%    25%    50%    75%   97.5%
43 var1 0.19912  0.20822  0.21332  0.21863  0.2290
44 var2 0.08183  0.09284  0.09870  0.10475  0.1160
45 var3 0.05121  0.05973  0.06426  0.06882  0.0781
46 var4 0.08183  0.09284  0.09870  0.10475  0.1160
47 var5 0.47763  0.49934  0.51131  0.52387  0.5493
48 var6 0.07318  0.08653  0.09351  0.10079  0.1145
49 var7 0.05121  0.05973  0.06426  0.06882  0.0781
50 var8 0.07318  0.08653  0.09351  0.10079  0.1145
51 var9 0.29523  0.30900  0.31654  0.32428  0.3397

```

We ask in the Exercise 5 to run this application using our GUI and the information in the dataset *Utilities.csv*. Observe that this file should be modified to agree the structure that requires our GUI (see the dataset *5Institutions.csv* in the folder *DataApp* of our GitHub repository - <https://github.com/besmarter/BSTAApp>- for a template). In addition, we ask to program from scratch the Gibbs sampler algorithm in this application.

7.3 Instrumental variable

This inferential approach is used when there are *endogeneity* issues, that is, the stochastic error is not independent of the regressors, this in turn generates *bias* in posterior mean estimates when we use an inferential approach that does not take this issue into. *Endogeneity* can be caused by *reverse causality*, *omitting relevant correlated variables*, or *measurement error* in the regressors.⁷

Let's specify the dependent variable as a linear function of one endogenous regressor and some exogenous regressors. That is, $y_i = \mathbf{x}_{ei}^\top \boldsymbol{\beta}_1 + \beta_s x_{si} + \mu_i$ where $x_{si} = \mathbf{x}_{ei}^\top \boldsymbol{\gamma}_1 + \mathbf{z}_i^\top \boldsymbol{\gamma}_2 + v_i$, x_s is the variable which generates the endogeneity issues ($\mathbb{E}[\mu|x_s] \neq 0$), \mathbf{x}_e are K_1 exogenous regressors ($\mathbb{E}[\mu|\mathbf{x}_e] = \mathbf{0}$), and \mathbf{z} are K_2 instruments, that is, regressors that drive x_s ($\mathbb{E}[x_s \mathbf{z}] \neq \mathbf{0}$), but do not have a direct effect on y ($\mathbb{E}[yz|x_s] = \mathbf{0}$). The equation of y is called the *structural equation*, and is the equation that the researcher is interested in.

Assuming $(u_i, v_i)^\top \stackrel{i.i.d.}{\sim} N(0, \Sigma)$, $\Sigma = [\sigma_{lm}]$, $l, m = 1, 2$, the likelihood function is

$$p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \Sigma | \mathbf{y}, \mathbf{X}, \mathbf{Z}) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Sigma|^{\frac{N}{2}}} \exp \left\{ -\frac{1}{2} \sum_{i=1}^N (y_i - \mathbf{x}_i^\top \boldsymbol{\beta}, x_{si} - \mathbf{w}_i^\top \boldsymbol{\gamma}) \Sigma^{-1} \begin{pmatrix} y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \\ x_{si} - \mathbf{w}_i^\top \boldsymbol{\gamma} \end{pmatrix} \right\},$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^\top \ \beta_s]^\top$, $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^\top \ \boldsymbol{\gamma}_2^\top]^\top$, $\mathbf{x}_i = [\mathbf{x}_{ei}^\top \ x_{si}]^\top$ and $\mathbf{w}_i = [\mathbf{x}_{ei}^\top \ \mathbf{z}_i^\top]^\top$.

We get standard conditional posterior densities using the following independent priors $\boldsymbol{\gamma} \sim N(\boldsymbol{\gamma}_0, \mathbf{G}_0)$, $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\Sigma^{-1} \sim W(\alpha_0, \Psi_0)$. In particular,

$$\boldsymbol{\beta} | \boldsymbol{\gamma}, \Sigma, \mathbf{y}, \mathbf{X}, \mathbf{Z} \sim N(\boldsymbol{\beta}_n, \mathbf{B}_n)$$

$$\boldsymbol{\gamma} | \boldsymbol{\beta}, \Sigma, \mathbf{y}, \mathbf{X}, \mathbf{Z} \sim N(\boldsymbol{\gamma}_n, \mathbf{G}_n)$$

$$\Sigma^{-1} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{y}, \mathbf{X}, \mathbf{Z} \sim W(\alpha_n, \Psi_n)$$

$$\text{where } \boldsymbol{\beta}_n = \mathbf{B}_n \left(\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \omega_1^{-1} \sum_{i=1}^N \left[\mathbf{x}_i \left(y_i - \frac{\sigma_{12}(x_{si} - \mathbf{w}_i^\top \boldsymbol{\gamma})}{\sigma_{22}} \right) \right] \right), \quad \mathbf{B}_n =$$

⁷See [223, Chap. 15] for an introductory treatment of instrumental variable in the Frequentist inferential approach.

$$(\omega_1^{-1} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top + \mathbf{B}_0^{-1})^{-1}, \omega_1 = \sigma_{11} - \sigma_{12}^2 / \sigma_{22}, \mathbf{G}_n = (\omega_2^{-1} \sum_{i=1}^N \mathbf{w}_i \mathbf{w}_i^\top + \mathbf{G}_0^{-1})^{-1}, \gamma_n = \mathbf{G}_n \left(\mathbf{G}_0^{-1} \gamma_0 + \omega_2^{-1} \sum_{i=1}^N \left[\mathbf{w}_i \left(x_{si} - \frac{\sigma_{12}(y_i - \mathbf{x}_i^\top \beta)}{\sigma_{11}} \right) \right] \right), \omega_2 = \sigma_{22} - \sigma_{12}^2 / \sigma_{11}, \Psi_n = \left[\Psi_0^{-1} + \sum_{i=1}^N \begin{pmatrix} y_i - \mathbf{x}_i^\top \beta \\ x_{si} - \mathbf{w}_i^\top \gamma \end{pmatrix} (y_i - \mathbf{x}_i^\top \beta, x_{si} - \mathbf{w}_i^\top \gamma) \right]^{-1}, \alpha_n = \alpha_0 + N, \text{ and } \sigma_{lj} \text{ are the elements of } \Sigma.$$

We also use a Gibbs sampling algorithm in this model since we have standard conditional posterior distributions.

Example: Simulation exercise

Let's simulate the simple process $y_i = \beta_1 + \beta_2 x_{si} + \mu_i$ and $x_{si} = \gamma_1 + \gamma_2 z_i + v_i$ where $[\mu_i \ v_i]^\top \sim N(\mathbf{0}, \Sigma)$, $\Sigma = [\sigma_{lj}]$ such that $\sigma_{12} \neq 0$, $i = 1, 2, \dots, 100$.

Observe that $\mu|v \sim N\left(\frac{\sigma_{12}}{\sigma_{22}}v, \sigma_{11} - \frac{\sigma_{21}^2}{\sigma_{22}}\right)$, this implies that $\mathbb{E}[\mu|x_s] = \mathbb{E}[\mu|v] = \frac{\sigma_{12}}{\sigma_{22}}v \neq 0$ given $\sigma_{12} \neq 0$ and $\mathbb{E}[\mu|z] = 0$. Let's set all location parameters equal to 1, and $\sigma_{11} = \sigma_{22} = 1$, $\sigma_{12} = 0.8$, and $z \sim N(0, 1)$. We know from the large sampling properties of the posterior mean that this converge to the maximum likelihood estimator (see Section 1.1, and [132, 213]), which in this setting is $\hat{\beta}_2 = \frac{\text{Cov}(x_s, y)}{\text{Var}(x_s)}$ which converges in probability to $\beta_2 + \frac{\sigma_{12}}{\sigma_{22}\text{Var}(x_s)} = \beta_2 + \frac{\sigma_{12}}{\sigma_{22}(\gamma_2^2\text{Var}(z) + \sigma_{22})} = 1.4$, that is, the asymptotic bias when using the posterior mean of a linear regression without taking into account endogeneity is 0.4 in this example.

We assess the sampling performance of Bayesian “estimators” simulating this setting 100 times. The following code shows how to do this using a linear model without taking into account the *endogeneity* issue (see Section 6.1), and implementing the variable instrumental model. We use $\mathbf{B}_0 = 1000\mathbf{I}_2$, $\beta_0 = \mathbf{0}_2$, and the parameters of the inverse gamma distribution equal to 0.0005. In the case of the instrumental variable setting, we set $\gamma_0 = \mathbf{0}_2$, $\mathbf{G}_0 = 1000\mathbf{I}_2$, $\alpha_0 = 3$ and $\Psi_0 = 3\mathbf{I}_2$ in addition.

R code. Simulation exercise, sampling properties ordinary and instrumental models

```

1 rm(list = ls()); set.seed(010101)
2 N <- 100; k <- 2
3 B <- rep(1, k); G <- rep(1, 2); s12 <- 0.8
4 SIGMA <- matrix(c(1, s12, s12, 1), 2, 2)
5 z <- rnorm(N); Z <- cbind(1, z); w <- matrix(1,N,1); S <-
6 100
7 U <- replicate(S, MASS::mvrnorm(n = N, mu = rep(0, 2), SIGMA
8 ))
9 x <- G[1] + G[2]*z + U[,2,]; y <- B[1] + B[2]*x + U[,1,]
# Hyperparameters
10 d0 <- 0.001/2; a0 <- 0.001/2
11 b0 <- rep(0, k); c0 <- 1000; B0 <- c0*diag(k)
12 B0i <- solve(B0); g0 <- rep(0, 2)
13 G0 <- 1000*diag(2); G0i <- solve(G0)
14 nu <- 3; Psi0 <- nu*diag(2)
# MCMC parameters
15 mcmc <- 5000; burnin <- 1000
16 tot <- mcmc + burnin; thin <- 1
# Gibbs sampling
17 Gibbs <- function(x, y){
18   Data <- list(y = y, x = x, w = w, z = Z)
19   Mcmc <- list(R = mcmc, keep = thin, nprint = 0)
20   Prior <- list(md = g0, Ad = G0i, mbg = b0, Abg = B0i, nu =
21     nu, V = Psi0)
22   RestIV <- bayesm::rivGibbs(Data = Data, Mcmc = Mcmc, Prior
23     = Prior)
24   PostBIV <- mean(RestIV[["betadraw"]])
25   ResLM <- MCMCpack::MCMCregress(y ~ x + w - 1, b0 = b0, B0
26     = B0i, c0 = a0, d0 = d0)
27   PostB <- mean(ResLM[,1]); Res <- c(PostB,PostBIV)
28   return(Res)
}
29 PosteriorMeans <- sapply(1:S, function(s) {Gibbs(x = x[,s],
30   y = y[,s])})
31 rowMeans(PosteriorMeans)
32 Model <- c(replicate(S, "Ordinary"), replicate(S, "
33   Instrumental"))
34 postmeans <- c(t(PosteriorMeans))
35 df <- data.frame(postmeans, Model, stringsAsFactors = FALSE)
36 library(ggplot2); library(latex2exp)
37 histExo <- ggplot(df, aes(x = postmeans, fill = Model)) +
38   geom_histogram(bins = 40, position = "identity", color =
39     "black", alpha = 0.5) + labs(title = "Overlaid
40   Histograms", x = "Value", y = "Count") + scale_fill_
41   manual(values = c("blue", "red")) + geom_vline(aes(
42     xintercept = mean(postmeans[1:S])), color = "black",
43     linewidth = 1, linetype = "dashed") + geom_vline(aes(
44     xintercept = mean(postmeans[101:200])), color = "black",
45     linewidth = 1, linetype = "dashed") + geom_vline(aes(
46     xintercept = B[2]), color = "green", linewidth = 1,
47     linetype = "dashed") + xlab(TeX("\$E[\backslash\beta_2\$"]))
48     + ylab("Frequency") + ggtitle("Histogram: Posterior means
49     simulating 100 samples")
50 histExo

```

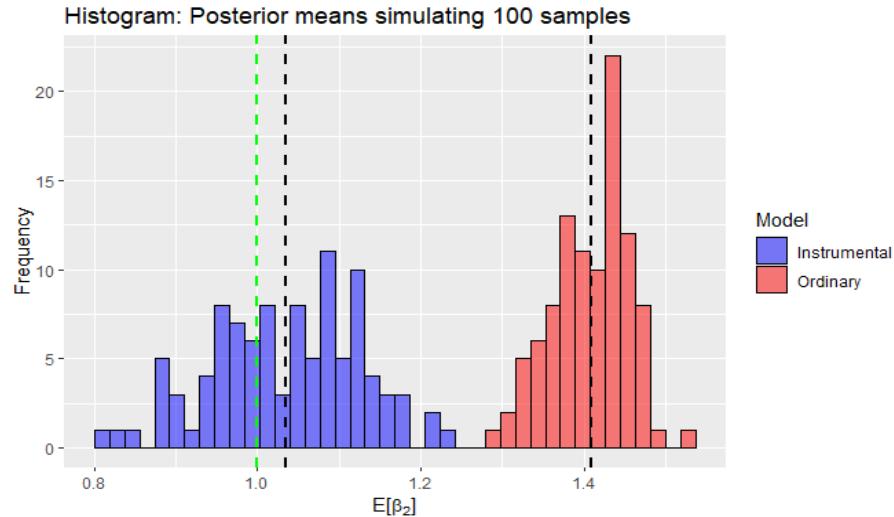


FIGURE 7.1
Histogram of posterior means: Ordinary and instrumental models.

Figure 7.1 displays the histograms of the posterior means of β_2 using the ordinary model without taking endogeneity into account, and the instrumental variable model. In one hand, the mean of the posterior means of the ordinary model is 1.41 (black dashed line in red histogram), this implies a bias equal to 0.41, which is very close to the population bias (0.40). On the other hand, the mean of the posterior means of the instrumental variable model is 1.04 (black dashed line in blue histogram), which is close to the population value of $\beta_2 = 1$ (green dashed line).

We also see that the histogram of the posterior means of the ordinary model is less disperse, that is, this “estimator” is more efficient, which is a well-known result in the Frequentist inferential approach comparing ordinary least squares and two-stage least squares (see [221, Chap. 5]).

Two very relevant aspects in the instrumental variables literature are the *weakness* and *exogeneity* of the instruments. The former refers how strong is the relationship between the instruments and the endogeneous regressors, and the latter refers to the independence of the instruments of the stochastic error in the *structural equation*. We ask in Exercise 6 to use the previous code as a baseline to study this two aspects. Observe the link between the *weakness* and *exogeneity* of the instrument, and the *exclusion restrictions* ($\mathbb{E}[x_s z] \neq \mathbf{0}$ and $\mathbb{E}[yz|x_s] = \mathbf{0}$). This is the point of departure of [47] who propose to assess the plausibility of the *exclusion restrictions* defining *plausible exogeneity* as having prior information that the effect of the instrument in the *structural equation* is near zero, but perhaps not exactly zero.

Algorithm A18 can be used to estimate the instrumental variable model

using our GUI. We ask in Exercise 8 to replicate the example of the effect of institutions on per capita GDP using our GUI.

Algorithm A18 Instrumental variable model

- 1: Select *Multivariate Models* on the top panel
 - 2: Select *Variable instrumental (two equations)* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Write down the formula of the structural equation in the **Main Equation** box. This formula must be written using the syntax of the *formula* command of **R** software. This equation includes intercept by default, do not include it in the equation
 - 6: Write down the formula of the endogenous regressor in the **Instrumental Equation** box. This formula must be written using the syntax of the *formula* command of **R** software. This equation includes intercept by default, do not include it in the equation
 - 7: Set the hyperparameters: mean vectors, covariance matrices, degrees of freedom, and the scale matrix. This step is not necessary as by default our GUI uses non-informative priors
 - 8: Click the *Go!* button
 - 9: Analyze results
 - 10: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

7.4 Multivariate probit model

In the multivariate probit model [62], the response variable $y_{il} = \{0, 1\}$ indicates that individual i makes binary choices regarding no mutually exclusive alternatives $l = 1, 2, \dots, L$, $i = 1, 2, \dots, N$. In particular,

$$y_{il} = \begin{cases} 0, & y_{il}^* \leq 0 \\ 1, & y_{il}^* > 0 \end{cases},$$

where $\mathbf{y}_i^* = \mathbf{X}_i \boldsymbol{\beta} + \boldsymbol{\mu}_i \stackrel{i.i.d.}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma})$, \mathbf{y}_i^* is an unobserved latent L -dimensional vector, $\mathbf{X}_i = \mathbf{x}_i^\top \otimes \mathbf{I}_L$ is an $L \times K$ design matrix of regressors, $K = L \times k$, k is the number of regressors (length of \mathbf{x}_i). In addition, $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^\top \ \boldsymbol{\beta}_2^\top \ \dots \ \boldsymbol{\beta}_k^\top]^\top$, where the $\boldsymbol{\beta}_j$ make up an L -dimensional vector of coefficients, $j = 1, 2, \dots, k$.

The likelihood function in this model is $p(\boldsymbol{\beta}, \boldsymbol{\Sigma} | \mathbf{y}, \mathbf{X}) = \prod_{i=1}^N \prod_{l=1}^L p_{il}^{y_{il}}$ where $p_{il} = p(y_{il}^* \geq 0)$. Observe that $p(y_{il}^* \geq 0) = p(\lambda_{il} y_{il}^* \geq 0)$, $\lambda_{il} > 0$. This generates identification issues because just the correlation matrix can be identified, same case as the univariate probit model where the variance of the model is fixed to 1. We follow the post processing strategy proposed by [62] to get identified parameters, that is, $\tilde{\boldsymbol{\beta}} = \text{vec}\{\boldsymbol{\Lambda}\boldsymbol{B}\}$ and the correlation matrix $\mathbf{R} = \boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda} = \text{diag}\{\sigma_{il}\}^{-1/2}$ and $\boldsymbol{B} = [\beta_1 \ \beta_2 \dots \beta_k]$.⁸

We assume independent priors, $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\boldsymbol{\Sigma}^{-1} \sim W(\alpha_0, \boldsymbol{\Psi}_0)$. We can employ Gibbs sampling in this model because this is a standard Bayesian linear regression model when data augmentation in \mathbf{y}^* is used. The posterior conditional distributions are

$$\begin{aligned} \boldsymbol{\beta} | \boldsymbol{\Sigma}, \mathbf{w} &\sim N(\boldsymbol{\beta}_n, \mathbf{B}_n), \\ \boldsymbol{\Sigma}^{-1} | \boldsymbol{\beta}, \mathbf{w} &\sim W(\alpha_n, \boldsymbol{\Psi}_n), \\ y_{il}^* | \mathbf{y}_{i,-l}^*, \boldsymbol{\beta}, \boldsymbol{\Sigma}^{-1}, \mathbf{y}_i &\sim TN_{I_{il}}(m_{il}, \tau_{il}^2) \end{aligned}$$

where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \mathbf{X}^{*\top} \mathbf{X}^*)^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n(\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \mathbf{X}^{*\top} \mathbf{y}^{**})$, $\boldsymbol{\Sigma}^{-1} = \mathbf{C}^\top \mathbf{C}$, $\mathbf{X}_i^* = \mathbf{C} \mathbf{X}_i$, $\mathbf{y}_i^{**} = \mathbf{C} \mathbf{y}_i^*$, $\alpha_n = \alpha_0 + N$, $\boldsymbol{\Psi}_n = (\boldsymbol{\Psi}_0 + \sum_{i=1}^N (\mathbf{y}_i^* - \mathbf{X}_i \boldsymbol{\beta})^\top (\mathbf{y}_i^* - \mathbf{X}_i \boldsymbol{\beta}))^{-1}$, $m_{il} = \mathbf{x}_{il}^\top \boldsymbol{\beta} + \mathbf{f}_l^\top (\mathbf{y}_{i,-l}^* - \mathbf{X}_{i,-l} \boldsymbol{\beta})$, $\mathbf{y}_{i,-l}^*$ is an $L-1$ dimensional vector of all components of \mathbf{y}_i^* excluding y_{il}^* , \mathbf{x}_{il}^\top is the l -th row of \mathbf{X}_i , $\mathbf{X}_{i,-l}$ is \mathbf{X}_i after deleting the l -th row, $\mathbf{f}_l^\top = \boldsymbol{\omega}_{l,-l}^\top \boldsymbol{\Sigma}_{-l,-l}^{-1}$, $\boldsymbol{\omega}_{l,-l}$ and $\boldsymbol{\Sigma}_{-l,-l}$ are the l -th row of $\boldsymbol{\Sigma}$ extracting the l -th element, and the sub-matrix of $\boldsymbol{\Sigma}$ extracting the l, l element, and $\tau_{il}^2 = \sigma_{l,l} - \boldsymbol{\omega}_{l,-l}^\top \boldsymbol{\Sigma}_{-l,-l}^{-1} \boldsymbol{\omega}_{-l,l}$, and

$$\mathbf{X}^* = \begin{bmatrix} \mathbf{X}_1^* \\ \mathbf{X}_2^* \\ \vdots \\ \mathbf{X}_N^* \end{bmatrix}, \quad I_{il} = \begin{cases} y_{il}^* > 0, & y_{il} = 1 \\ y_{il}^* \leq 0, & y_{il} = 0 \end{cases}, \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\omega}_1^\top \\ \boldsymbol{\omega}_2^\top \\ \vdots \\ \boldsymbol{\omega}_L^\top \end{bmatrix}.$$

The setting in our GUI has same regressors in each binary decision. However, we can see that the multivariate probit model is similar to a SUR model in latent variables. We ask in Exercise 9 to implement a Gibbs sampling algorithm for a multivariate probit model with different regressors in each equation.

Example: Self selection in hospitalization due to a subsidized health care program

We use the dataset *7HealthMed.csv* where the dependent variable is equal to $y = [\text{Hosp SHI}]^\top$ where Hosp is equal to 1 if an individual was hospitalized in the year previous to the survey, 0 otherwise, and SHI is equal to 1 if the individual had subsidized health insurance, and 0 otherwise.

⁸In a Bayesian setting, we can have a non identified model; however, the posterior of the model parameters exists given a proper prior distribution [62].

Recall that our application in binary response models was to uncover the determinants of hospitalization in Medellín (Colombia), where one of the regressors was a binary indicator of being in a subsidized health care program (Section 6.3). We can use a bivariate probit model if we suspect there is a dependence regarding the decisions involving these two variables. We would expect a priori that being in a subsidized health care program would imply a higher probability of being hospitalized *ceteris paribus* due to a reduced cost for the patient. However, if an individual expects to be hospitalized in the future, and the factors that drive this decision are unobserved to the modeller, we would have a feedback effect from being hospitalized on being in a subsidized health care program.

We took into account 7 regressors: a constant, female, age, self perception of health status, fair, good and excellent, taking as reference bad, and the proportion of the individual's age spent living in her/his neighborhood. The last variable tries to take into account the social capital that can affect being in the subsidized health insurance program, as the target population is identified by the local government [180]. We have 12975 individuals who “choose” two options (hospitalization and subsidized regime).

The Algorithm A19 shows how to run a multivariate probit model in our GUI.

Algorithm A19 Multivariate probit model

- 1: Select *Multivariate Models* on the top panel
 - 2: Select *Multivariate Probit* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Write down the number of cross-sectional units in the **Number of individuals: n** box
 - 6: Write down the number of exogenous variables in the **Number of exogenous variables: k** box
 - 7: Write down the number of choices in the **Number of choices: l** box
 - 8: Set the hyperparameters: mean vectors, covariance matrix, degrees of freedom, and the scale matrix. This step is not necessary as by default our GUI uses non-informative priors
 - 9: Click the *Go!* button
 - 10: Analyze results
 - 11: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

R code. Self selection in hospitalization

```

1 rm(list = ls()); set.seed(010101)
2 Data <- read.csv("https://raw.githubusercontent.com/
    besmarter/BSTApp/refs/heads/master/DataApp/7HealthMed.
    csv", sep = ",", header = TRUE, quote = "")
3 attach(Data); str(Data)
4 p <- 2; nd <- 7; N <- length(y)/p; y <- y
5 Xd <- as.matrix(Data[seq(1, p*N, 2),3:9])
6 XcreateMP<-function(p,nxs,nind,Data){
7   pandterm = function(message) {
8     stop(message, call. = FALSE)
9   }
10  if (missing(nxs))
11    pandterm("requires number of regressors: include intercept
        if required")
12  if (missing(nind))
13    pandterm("requires number of units (individuals)")
14  if (missing(Data))
15    pandterm("requires dataset")
16  if (nrow(Data) != nind*2)
17    pandterm("check dataset! number of units times number
        alternatives should be equal to dataset rows")
18  XXDat<-array(0,c(p,1+nxs,nind))
19  XX<-array(0,c(p,nxs*p,nind))
20  YY<-array(0,c(p,1,nind))
21  is<- seq(p,nind*p,p)
22  cis<- seq(nxs,nxs*p+1,nxs)
23  for(i in is){
24    j<-which(i==is)
25    XXDat[,,j]<-as.matrix(Data[c((i-(p-1)):i),-1])
26    YY[,,j]<-XXDat[,1,j]
27    for(l in 1:p){
28      XX[l,((cis[l]-(nxs-1)):cis[l]),j]<-XXDat[l,-1,j]
29    }
30  }
31  return(list(y=YY,X=XX))
32 }
33 Dat <- XcreateMP(p = p, nxs = nd, nind = N, Data = Data)
34 y<-NULL; X<-NULL
35 for(i in 1:dim(Dat$y)[3]){
36  y<-c(y,Dat$y[,i])
37  X<-rbind(X,Dat$X[,i])
38 }
39 DataMP = list(p=p, y=y, X=X)
40 # Hyperparameters
41 k <- dim(X)[2]; b0 <- rep(0, k); c0 <- 1000
42 B0 <- c0*diag(k); B0i <- solve(B0)
43 a0 <- p - 1 + 3; Psi0 <- a0*diag(p)
44 Prior <- list(betabar = b0, A = B0i, nu = a0, V = Psi0)
45 # MCMC parameters
46 mcmc <- 20000; thin <- 5; Mcmc <- list(R = mcmc, keep = thin
    )

```

R code. Self selection in hospitalization, results

```

1 Results <- bayesm::rmvpGibbs(Data = DataMP, Mcmc = Mcmc,
2 Prior = Prior)
3 betatilde1 <- Results$betadraw[,1:7] / sqrt(Results$  

4 sigmadrw[,1])
5 summary(coda::mcmc(beta1))
6 Quantiles for each variable:  

7  

8 2.5%      25%      50%      75%      97.5%
9 var1 -1.2247602 -1.059257 -0.970870 -0.889741 -0.733411
10 var2  0.0269885  0.090098  0.121863  0.155585  0.220662
11 var3  0.0007652  0.002207  0.002925  0.003685  0.005049
12 var4 -0.7477149 -0.598898 -0.522549 -0.445173 -0.296718
13 var5 -1.4520842 -1.309922 -1.234633 -1.160468 -1.018396
14 var6 -1.3503717 -1.182381 -1.092511 -1.005472 -0.837939
15 var7 -0.1791758 -0.103506 -0.064418 -0.024499  0.051849
16 betatilde2 <- Results$betadraw[,8:14] / sqrt(Results$  

17 sigmadrw[,4])
18 summary(coda::mcmc(beta1))
19 Quantiles for each variable:  

20  

21 2.5%      25%      50%      75%      97.5%
22 var1  0.306343  0.477265  0.564698  0.656616  0.82932
23 var2  0.258347  0.289819  0.305902  0.322116  0.35284
24 var3  0.007848  0.008656  0.009124  0.009591  0.01045
25 var4 -0.488810 -0.313532 -0.218459 -0.130373  0.04144
26 var5 -0.677686 -0.511529 -0.418139 -0.332415 -0.17322
27 var6 -0.703355 -0.527642 -0.433378 -0.341355 -0.16989
28 var7  0.164388  0.203623  0.224533  0.245306  0.28513
29 sigmadrw12 <- Results$sigmadrw[,3] / (Results$sigmadrw  

30 [,1]*Results$sigmadrw[,4])^0.5
31 summary(coda::mcmc(sigmadrw12))
32 Quantiles for each variable:  

33 2.5%      25%      50%      75%      97.5%
34 -0.070515 -0.025009 -0.002895  0.018432  0.060986

```

We set 20,000 MCMC iterations using a thinning parameter equal to 5. The hyperparameters are $\beta_0 = \mathbf{0}_{14}$, $B_0 = 100I_{14}$, $\alpha_0 = 4$ and $\Psi_0 = 4I_2$.⁹

The previous R code shows to get the posterior draws using the *rmvpGibbs* command from the *bayesm* package. The results suggest that females, older people whose self perception of health status is bad have a higher probability of being hospitalized. In addition, female, older people whose self perception of health status is bad or regular and have lived a higher proportion of their life in the actual neighborhood have a higher probability of being in the subsidized

⁹Take into account that the order of the location coefficients in our GUI is by equations, not by regressors as the theory setting in this section. This is important to set hyperparameters and read the results of the location parameters.

health care system. However, the results suggest that there is no unobserved correlation between the two equations as the 95% credible interval of the correlation is (-0.07, 0.06).

7.5 Summary

We show the setting and posterior distributions of the most common multivariate models in this chapter. The multivariate setting allows tackling *endogeneity* issues by using the conditional distribution of a multivariate normal vector. In addition, we always get posterior conditional distributions that are standard families (multivariate normal, Wishart and truncated normal); this allows implementing the Gibbs sampling algorithm in all these models.

7.6 Exercises

1. Show that $\mathbb{E}[u_1 \text{PAER}] = \frac{\alpha_1}{1-\beta_1\alpha_1}\sigma_1^2$ assuming that $\mathbb{E}[u_1 u_2] = 0$ where $\text{Var}(u_1) = \sigma_1^2$ in the effect of institutions on per capita GDP.
2. Show that $\beta_1 = \pi_1/\gamma_1$ in the effect of institutions on per capita GDP.
3. **The effect of institutions on per capita gross domestic product continues I**

Use the *rmultireg* command from the *bayesm* package to perform inference in the example of the effect of institutions on per capita GDP.

4. Demand and supply simulation

Given the structural demand-supply model:

$$\begin{aligned} q_i^d &= \beta_1 + \beta_2 p_i + \beta_3 y_i + \beta_4 pc_i + \beta_5 ps_i + u_{i1} \\ q_i^s &= \alpha_1 + \alpha_2 p_i + \alpha_3 er_i + u_{i2}, \end{aligned}$$

where q^d is demand, q^s is supply, p , y , pc , ps and er are price, income, complementary price, substitute price, and exchange rate. Complementary and substitute prices are prices of a complementary and substitute goods of q . Assume that $\boldsymbol{\beta} = [5 - 0.5 0.8 - 0.4 0.7]^\top$, $\boldsymbol{\alpha} = [-2 0.5 - 0.4]^\top$, $u_1 \sim N(0, 0.5^2)$ and $u_2 \sim N(0, 0.5^2)$. In addition, assume that $y \sim N(10, 1)$, $pc \sim N(5, 1)$, $ps \sim N(5, 1)$ and $tc \sim N(15, 1)$.

- Find the *reduce-form* model using that in equilibrium demand and supply are equal, that is, $q^d = q^s$. This condition defines the observable quantity (q).
- Simulate p and q from the *reduce-form* equations.
- Preform inference of the *reduce-form* model using the *rmultireg* command from the *bayesm* package.
- Use the posterior draws of the *reduce-form* parameters to perform inference of the *structural* parameters. Any issue? Hint: Are all *structural* parameters exactly identified?

5. Utility demand continues

- Run the **Utility demand** application using our GUI and the information in the dataset *Utilities.csv*. Hint: This file should be modified to agree the structure that requires our GUI (see the dataset *5Institutions.csv* in the folder *DataApp* of our GitHub repository -<https://github.com/besmarter/BSTAApp>- for a template).
- Program from scratch the Gibbs sampler algorithm in this application.

6. Simulation exercise of instrumental variables continues I

- (a) Use the setting of the simulation exercise of instrumental variables to analyze what happens when the instrument is weak, for instance, setting $\gamma_2 = 0.2$, and compare the performance of the posterior means of the ordinary and instrumental models.
- (b) Perform a simulation that helps to analyze how the degree of exogeneity of the instrument affects the performance of the posterior mean of the instrumental variable model.

7. Simulation exercise of instrumental variables continues II

Program from scratch the Gibbs sampling algorithm of the instrumental model for the simulation exercise of the instrumental variables.

8. The effect of institutions on per capita gross domestic product continues II

Estimate the structural Equation 7.1 using the instrumental variable model where the instrument of PAER is $\log(Mort)$. Compare the effect of property rights on per capita GDP of this model with the effect estimated in the example of the effect of institutions on per capita gross domestic product. Use the file *6Institutions.csv* to do this exercise in our GUI, and set $B_0 = 100I_5$, $\beta_0 = \mathbf{0}_5$, $\gamma_0 = \mathbf{0}_2$, $G_0 = 100I_2$, $\alpha_0 = 3$ and $\Psi_0 = 3I_2$. The MCMC iterations, burn-in and thinning parameters are 50000, 1000 and 5, respectively.

9. Multivariate probit with different regressors

Let's do a simulation exercise where $y_{i1}^* = 0.5 - 1.2x_{i11} + 0.7x_{i12} + 0.8x_{i3} + \mu_{i1}$ and $y_{i2}^* = 1.5 - 0.8x_{i21} + 0.5x_{i22} + \mu_{i2}$, $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$, where all regressors distribute standard normal, and $N = 5000$. Use $\beta_0 = \mathbf{0}$, $B_0 = 1000B$, $\alpha_0 = 4$ and $\Psi_0 = 4I_2$. Set number of iterations 2000 and a thinning parameter equal to 5.

- Perform inference using the setting of Section 7.4, that is, assuming that x_{i3} could have an effect on y_{i2} .
- Program a Gibbs sampling algorithm taking into account that there are different regressors in each binary decision, that is, x_{i3} does not have an effect on y_{i2} .



8

Time series models

In this chapter, we provide a brief introduction to performing inference in time series models using a Bayesian framework. There is a large literature in time series in statistics and econometrics, and it would be impossible to present a good treatment in a few pages of an introductory book. However, there are excellent books in Bayesian inference in time series, see for instance, [218, 163, 165].

A time series is a sequence of observations collected in chronological order, allowing us to track how variables change over time. However, it also introduces technical challenges, as we must account for statistical features such as autocorrelation and stationarity. Since time series data is time-dependent, we adjust our notation. Specifically, we use t and T instead of i and N to explicitly indicate time.

Our starting point in this chapter is the *state-space representation* of time series models. Much of the Bayesian inference literature in time series adopts this approach, as it allows dynamic systems to be modeled in a structured way. This representation provides modularity, flexibility, efficiency, and interpretability in complex models where the state evolves over time. It also enables the use of recursive estimation methods, such as the *Kalman filter* for dynamic Gaussian linear models and the *particle filter* (also known as *sequential Monte Carlo*) for non-Gaussian and nonlinear state-space models. The latter method is especially useful for *online* predictions or when there are data storage limitations. These inferential tools are based on the sequential updating process of Bayes' rule, where the posterior at time t becomes the prior at time $t + 1$ (see Equation 1.14).

Remember that we can run our GUI typing

R code. How to display our graphical user interface

```
1 shiny::runGitHub("besmarter/BSTApp", launch.browser = T)
```

in the **R** package console or any **R** code editor, and once our GUI is

deployed, select *Time series Models*. However, users should see Chapter 5 for details.

8.1 State-space representation

A *state-space model* is composed by of an *unobservable state vector* $\beta_t \in \mathbb{R}^K$, and an *observed measure* $\mathbf{Y}_t \in \mathbb{R}^M$, $t = 1, 2, \dots$ such that (i) β_t is a *Markov process*, this is, $\pi(\beta_t | \beta_{1:t-1}) = \pi(\beta_t | \beta_{t-1})$, all the information regarding β_t based on all its history up to $t-1$ is carried by β_{t-1} , and (ii) \mathbf{Y}_t is independent of \mathbf{Y}_s conditional on β_t , $s < t$ [163, Chap. 2].

These assumptions imply that $\pi(\beta_{0:t}, \mathbf{Y}_{1:t}) = \pi(\beta_0) \prod_{s=1}^t \pi(\beta_s | \beta_{s-1}) \pi(\mathbf{Y}_s | \beta_s)$.¹

There are three key aims in *state-space models*: *filtering*, *smoothing*, and *forecasting*. In *filtering*, we aim to estimate the current state given observations up to time t , specifically obtaining the density $\pi(\beta_s | \mathbf{y}_{1:t})$ for $s = t$. In *smoothing*, we conduct a retrospective analysis of the system, obtaining $\pi(\beta_s | \mathbf{y}_{1:t})$ for $s < t$. In *forecasting*, we forecast future observations by first obtaining $\pi(\beta_s | \mathbf{y}_{1:t})$ as an intermediate step to compute $\pi(\mathbf{Y}_s | \mathbf{y}_{1:t})$ for $s > t$. A valuable feature of these methods is that all these densities can be calculated recursively. [163] show the recursive equations in Propositions 2.1 (filtering), 2.3 (smoothing) and 2.5 (forecasting).

An important class of *state-space models* is the *Gaussian linear state-space model*, also known as, *dynamic linear model*:

$$\begin{aligned} \mathbf{Y}_t &= \mathbf{X}_t \beta_t + \boldsymbol{\mu}_t && \text{(Observation equations)} \\ \beta_t &= \mathbf{G}_t \beta_{t-1} + \mathbf{w}_t && \text{(States equations),} \end{aligned}$$

where $\beta_0 \sim N(\mathbf{b}_0, \mathbf{B}_0)$, $\boldsymbol{\mu}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}_t)$, $\mathbf{w}_t \sim N(\mathbf{0}, \boldsymbol{\Omega}_t)$, β_0 , $\boldsymbol{\mu}_t$ and \mathbf{w}_t are independent, \mathbf{X}_t and \mathbf{G}_t are $M \times K$ and $K \times K$ known matrices. Observe that this assumption implies that $\mathbf{Y}_t | \beta_t \sim N(\mathbf{X}_t \beta_t, \boldsymbol{\Sigma}_t)$, and $\beta_t | \beta_{t-1} \sim N(\mathbf{G}_t \beta_{t-1}, \boldsymbol{\Omega}_t)$.²

Let $\beta_{t-1} | \mathbf{y}_{1:t-1} \sim N(\mathbf{b}_{t-1}, \mathbf{B}_{t-1})$, then, we can get the *Kalman filter* by getting

1. The one-step-ahead predictive distribution of β_t given $\mathbf{y}_{1:t-1}$ is $\beta_t | \mathbf{y}_{1:t-1} \sim N(\mathbf{a}_t, \mathbf{R}_t)$, where $\mathbf{a}_t = \mathbf{G}_t \mathbf{b}_{t-1}$ and $\mathbf{R}_t = \mathbf{G}_t \mathbf{B}_{t-1} \mathbf{G}_t^\top + \boldsymbol{\Omega}_t$.
2. The one-step-ahead predictive distribution of \mathbf{Y}_t given $\mathbf{y}_{1:t-1}$ is $\mathbf{Y}_t | \mathbf{y}_{1:t-1} \sim N(\mathbf{f}_t, \mathbf{Q}_t)$, where $\mathbf{f}_t = \mathbf{X}_t \mathbf{a}_t$ and $\mathbf{Q}_t = \mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \boldsymbol{\Sigma}_t$.

¹A *state-space model* where the states are random variables taking discrete values is called *hidden Markov model*.

²A general *state-space model* is given by $\mathbf{Y}_t = \mathbf{f}_t(\beta_t, \boldsymbol{\mu}_t)$, and $\beta_t = \mathbf{m}_t(\beta_{t-1}, \mathbf{w}_t)$ for arbitrary functions \mathbf{f}_t and \mathbf{m}_t , and distributions for $\boldsymbol{\mu}_t$ and \mathbf{w}_t , and a prior distribution for β_0 .

3. The distribution of the one-step-ahead prediction error $\mathbf{e}_t = \mathbf{Y}_t - \mathbb{E}[\mathbf{Y}_t | \mathbf{y}_{1:t-1}] = \mathbf{Y}_t - \mathbf{f}_t$ is $N(\mathbf{0}, \mathbf{Q}_t)$ [200, Chap. 6].
4. The filtering distribution of β_t given $\mathbf{y}_{1:t}$ is $\beta_t | \mathbf{y}_{1:t} \sim N(\mathbf{b}_t, \mathbf{B}_t)$, where $\mathbf{b}_t = \mathbf{a}_t + \mathbf{K}_t \mathbf{e}_t$, $\mathbf{K}_t = \mathbf{R}_t \mathbf{X}_t^\top \mathbf{Q}_t^{-1}$ is the *Kalman gain*, and $\mathbf{B}_t = \mathbf{R}_t - \mathbf{R}_t \mathbf{X}_t^\top \mathbf{Q}_t^{-1} \mathbf{X}_t \mathbf{R}_t$.

The formal proofs of these results can be found in [163, Chap 2]. Just take into account that the logic of these results follow the results of the Seemingly unrelated regression (SUR) model in Section 7.2 for a particular time period. In addition, we know that the posterior distribution using information up to $t-1$ becomes the prior in t (see Equation 1.14, $\pi(\boldsymbol{\theta} | \mathbf{y}_{1:t}) \propto p(y_t | \mathbf{y}_{1:t-1}, \boldsymbol{\theta}) \times \pi(\boldsymbol{\theta} | \mathbf{y}_{1:t-1})$). This is the updating process from $\beta_t | \mathbf{y}_{1:t-1} \sim N(\mathbf{a}_t, \mathbf{R}_t)$ to $\beta_t | \mathbf{y}_{1:t} \sim N(\mathbf{b}_t, \mathbf{B}_t)$. Moreover, the posterior mean and variance of the SUR model with independent conjugate priors for a particular time period can be written as $\mathbf{a}_t + \mathbf{R}_t \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \Sigma_t)^{-1} (\mathbf{y}_t - \mathbf{X}_t \mathbf{a}_t)$ and $\mathbf{R}_t - \mathbf{R}_t \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \Sigma_t)^{-1} \mathbf{X}_t \mathbf{R}_t^\top$, respectively. Let's see this, we know from Section 7.2 that $\mathbf{B}_t = (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1}$ and $\beta_t = \mathbf{B}_t (\mathbf{R}_t^{-1} \mathbf{a}_t + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{y}_t)$. Thus, let's show that both conditional posterior distributions are the same. In particular, the posterior mean in the *state-space representation* is $[\mathbf{I}_K - \mathbf{R}_t \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \Sigma_t)^{-1} \mathbf{X}_t] \mathbf{a}_t + \mathbf{R}_t \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \Sigma_t)^{-1} \mathbf{y}_t$, where

$$\begin{aligned} \mathbf{R}_t \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \Sigma_t)^{-1} &= \mathbf{R}_t \mathbf{X}_t^\top [\Sigma_t^{-1} - \Sigma_t^{-1} \mathbf{X}_t (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} \mathbf{X}_t^\top \Sigma_t^{-1}] \\ &= \mathbf{R}_t [\mathbf{I}_K - \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1}] \mathbf{X}_t^\top \Sigma_t^{-1} \\ &= \mathbf{R}_t (\mathbf{I}_K - [\mathbf{I}_K - \mathbf{R}_t^{-1} (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1}]) \mathbf{X}_t^\top \Sigma_t^{-1} \\ &= (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} \mathbf{X}_t^\top \Sigma_t^{-1}, \end{aligned}$$

where the first equality uses the Woodbury matrix identity (matrix inversion lemma), and the third equality uses $\mathbf{D}(\mathbf{D} + \mathbf{E})^{-1} = \mathbf{I} - \mathbf{E}(\mathbf{D} + \mathbf{E})^{-1}$.

Thus, $[\mathbf{I}_K - \mathbf{R}_t \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \Sigma_t)^{-1} \mathbf{X}_t] \mathbf{a}_t + \mathbf{R}_t \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \Sigma_t)^{-1} \mathbf{y}_t = [\mathbf{I}_K - (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t] \mathbf{a}_t + (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{y}_t = (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} \mathbf{R}_t^{-1} \mathbf{a}_t + (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{y}_t = (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} (\mathbf{R}_t^{-1} \mathbf{a}_t + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{y}_t) = (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} (\mathbf{R}_t^{-1} \mathbf{a}_t + \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t \hat{\beta}_t)$. The second equality uses $\mathbf{I} - (\mathbf{D} + \mathbf{E})^{-1} \mathbf{D} = (\mathbf{D} + \mathbf{E})^{-1} \mathbf{E}$, and $\hat{\beta}_t = (\mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{X}_t)^{-1} \mathbf{X}_t^\top \Sigma_t^{-1} \mathbf{y}_t$. This means that the posterior mean is a weighted average of the prior mean, and the maximum likelihood estimator (generalized least squares estimator).

The weights are linked to the signal-to-noise ratio, that is, the proportion of the total variability ($\Omega_t + \Sigma_t$) due to the signal (Ω_t) versus the noise (Σ_t). Note that in the simplest case where $M = K = 1$, and $\mathbf{X}_t = \mathbf{G}_t = 1$, then $\mathbf{K}_t = \mathbf{R}_t \mathbf{Q}_t^{-1} = (B_{t-1} + \Omega_t) / (B_{t-1} + \Omega_t + \Sigma_t)$. Thus, the weight associated with the observations is equal to 1 if $\Sigma_t = 0$, that is, the posterior mean is equal to the actual observation. On the other hand, if Σ_t increases compare to Ω_t , there is more weight to the prior information, and consequently, the posterior mean is smoother as it heavily depends on the history. We ask in

Exercise 1 to perform simulations with different signal-to-noise ratios to see the effects on the system.

The equality of variances of both approaches is as follows:

$$\begin{aligned}
Var[\beta_t | \mathbf{y}_{1:t}] &= \mathbf{R}_t - \mathbf{R}_t \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{R}_t \mathbf{X}_t^\top + \boldsymbol{\Sigma}_t)^{-1} \mathbf{X}_t \mathbf{R}_t \\
&= \mathbf{R}_t - \mathbf{R}_t \mathbf{X}_t^\top (\boldsymbol{\Sigma}_t^{-1} - \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t)^{-1} \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1}) \mathbf{X}_t \mathbf{R}_t \\
&= \mathbf{R}_t - \mathbf{R}_t \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t \mathbf{R}_t + \mathbf{R}_t \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t)^{-1} \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t \mathbf{R}_t \\
&= \mathbf{R}_t - \mathbf{R}_t \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t \mathbf{R}_t + \mathbf{R}_t \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t [\mathbf{I}_K - (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t)^{-1} \mathbf{R}_t^{-1}] \mathbf{R}_t \\
&= \mathbf{R}_t - \mathbf{R}_t \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t)^{-1} \\
&= \mathbf{R}_t [\mathbf{I}_K - \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t)^{-1}] \\
&= \mathbf{R}_t [\mathbf{I}_K - (\mathbf{I}_K - \mathbf{R}_t^{-1} (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t)^{-1})] \\
&= (\mathbf{R}_t^{-1} + \mathbf{X}_t^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{X}_t)^{-1},
\end{aligned}$$

where the second equality uses the Woodbury matrix identity, the fourth equality uses $(\mathbf{D} + \mathbf{E})^{-1} \mathbf{D} = \mathbf{I} - (\mathbf{D} + \mathbf{E})^{-1} \mathbf{E}$, and the seventh equality uses $\mathbf{D}(\mathbf{D} + \mathbf{E})^{-1} = \mathbf{I} - \mathbf{E}(\mathbf{D} + \mathbf{E})^{-1}$.

The *Kalman filter* allows calculating recursively in a forward way $\pi(\beta_t | \mathbf{y}_{1:t})$ from $\pi(\beta_{t-1} | \mathbf{y}_{1:t-1})$ starting from $\pi(\beta_0)$.

Let $\beta_{t+1} | \mathbf{y}_{1:T} \sim N(\mathbf{s}_{t+1}, \mathbf{S}_{t+1})$, then we can get the *Kalman smoother* by $\beta_t | \mathbf{y}_{1:T} \sim N(\mathbf{s}_t, \mathbf{S}_t)$, where $\mathbf{s}_t = \mathbf{b}_t + \mathbf{B}_t \mathbf{G}_{t+1}^\top \mathbf{R}_{t+1}^{-1} (\mathbf{s}_{t+1} - \mathbf{a}_{t+1})$ and $\mathbf{S}_t = \mathbf{B}_t - \mathbf{B}_t \mathbf{G}_{t+1}^\top \mathbf{R}_{t+1}^{-1} (\mathbf{R}_{t+1} - \mathbf{S}_{t+1}) \mathbf{R}_{t+1}^{-1} \mathbf{G}_{t+1} \mathbf{B}_t$. The proof can be found in [163, Chap 2].

Thus, we can calculate the *Kalman smoother* starting from $t = T - 1$, that is, $\beta_T | \mathbf{y}_{1:T} \sim N(\mathbf{s}_T, \mathbf{S}_T)$. However, this is the filtering distribution at T , which means $\mathbf{s}_T = \mathbf{b}_T$ and $\mathbf{S}_T = \mathbf{B}_T$, and then, we should proceed recursively in a backward way.

Finally, the forecasting recursion in the *dynamic linear model*, given $\mathbf{a}_t(0) = \mathbf{b}_t$ and $\mathbf{R}_t(0) = \mathbf{B}_t$, $h \geq 1$, is given by

1. The forecasting distribution of $\beta_{t+h} | \mathbf{y}_{1:t}$ is $N(\mathbf{a}_t(h), \mathbf{R}_t(h))$, where $\mathbf{a}_t(h) = \mathbf{G}_{t+h} \mathbf{a}_t(h-1)$ and $\mathbf{R}_t(h) = \mathbf{G}_{t+h} \mathbf{R}_t(h-1) \mathbf{G}_{t+h}^\top + \boldsymbol{\Omega}_{t+h}$.
2. The forecasting distribution $\mathbf{Y}_{t+h} | \mathbf{y}_{1:t}$ is $N(\mathbf{f}_t(h), \mathbf{Q}_t(h))$, where $\mathbf{f}_t(h) = \mathbf{X}_{t+h} \mathbf{a}_t(h)$ and $\mathbf{Q}_t(h) = \mathbf{X}_{t+h} \mathbf{R}_t(h) \mathbf{X}_{t+h}^\top + \boldsymbol{\Sigma}_{t+h}$.

The proof can be found in [163, Chap 2].

These recursive equations allows to perform probabilistic forecasting h -steps-ahead for the state and observation equations.

These results show how to use these recursive equations to perform filtering, smoothing and forecasting in *dynamic linear models (Gaussian linear state-space models)*. Despite that these algorithms look simple, they suffer from numerical instability that lead to non-symmetric and negative definite calculated covariance matrices. Thus, special care should be put when working with them.

In addition, this set up assumes that the Σ_t and Ω_t are known. However, this is no the case in most situations. Thus, we should estimate them. One option is to perform maximum likelihood estimation. However, this approach does not take into account the uncertainty associated with the fact that Σ_t and Ω_t are unknown when their estimates are *plug in the state space* recursions. On the other hand, we can use a Bayesian approach, and perform the recursions associated with each posterior draw of the unknown parameters. Thus, we take into account their uncertainty.

The point of departure is the posterior distribution, such that

$$\pi(\boldsymbol{\theta}, \beta_0, \dots, \beta_T | \mathbf{y}, \mathbf{X}, \mathbf{G}) \propto \pi(\beta_0 | \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) \prod_{t=1}^T \pi(\beta_t | \beta_{t-1}, \boldsymbol{\theta}) \pi(\mathbf{y}_t | \beta_t, \boldsymbol{\theta}),$$

where $\boldsymbol{\theta}$ is the vector of unknown parameters.

We can compute $\pi(\beta_s, \boldsymbol{\theta} | \mathbf{y}_{1:t}) = \pi(\beta_s | \mathbf{y}_{1:t}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta} | \mathbf{y}_{1:t})$, for $s = t$ (*filtering*), $s < t$ (*smoothing*), and $s > t$ (*forecasting*). The marginal posterior distribution of the states is $\pi(\beta_s | \mathbf{y}_{1:t}) = \int_{\boldsymbol{\Theta}} \pi(\beta_s | \mathbf{y}_{1:t}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta} | \mathbf{y}_{1:t}) d\boldsymbol{\theta}$.

We can use the Gibbs sampling algorithm to get the posterior draws in the *dynamic linear model* assuming conjugate families. In particular, let's see the univariate case with *random walk states*,

$$Y_t = \mathbf{x}_t^\top \boldsymbol{\beta}_t + \mu_t \quad (8.1)$$

$$\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} + \mathbf{w}_t, \quad (8.2)$$

where $\mu_t \sim N(0, \sigma^2)$ and $\mathbf{w}_t \sim N(\mathbf{0}, \text{diag}\{\omega_1^2, \dots, \omega_K^2\})$. We assume that $\pi(\sigma^2, \omega_1^2, \dots, \omega_K^2, \boldsymbol{\beta}_0) = \pi(\sigma^2) \pi(\omega_1^2), \dots, \pi(\omega_K^2) \pi(\boldsymbol{\beta}_0)$ where $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$, $\omega_k^2 \sim IG(\alpha_{k0}/2, \delta_{k0}/2)$, $k = 1, \dots, K$, and $\boldsymbol{\beta}_0 \sim N(\mathbf{b}_0, \mathbf{B}_0)$. Thus, the conditional posterior distributions are $\sigma^2 | \mathbf{y}, \mathbf{X}, \boldsymbol{\beta}_{1:T} \sim IG(\alpha_n/2, \delta_n/2)$, where $\alpha_n = T + \alpha_0$ and $\delta_n = \sum_{t=1}^T (y_t - \mathbf{x}_t^\top \boldsymbol{\beta}_t)^2 + \delta_0$, and $\omega_k^2 | \mathbf{y}, \mathbf{X}, \boldsymbol{\beta}_{0:T} \sim IG(\alpha_{kn}/2, \delta_{kn}/2)$, where $\alpha_{kn} = T + \alpha_{k0}$ and $\delta_{kn} = \sum_{t=1}^T (\boldsymbol{\beta}_{t,k} - \boldsymbol{\beta}_{t-1,k})^2 + \delta_{k0}$. The vector of the dependent variable is \mathbf{y} , and all regressors are in \mathbf{X} .

We also need to sample the states from $\pi(\boldsymbol{\beta}_{1:T} | \mathbf{y}, \mathbf{X}, \sigma^2, \omega_1^2, \dots, \omega_K^2)$. This can be done using the forward filtering backward sampling (FFBS) algorithm [31, 68, 199]. This algorithm is basically a simulation version of the *smoothing* recursion, which allows getting draws of the states, even if we do not have analytical solutions, for instance, in non-linear settings. See below and [163, Chap. 3] for details. A word of caution here, users should be careful to set non-informative priors in this setting, and in general, settings where there are a large number of parameters (see [126, Chap. 8] for details). Thus, it is useful to use empirical Bayes methods focusing on relevant hyperparameters, for instance, the hyperparameters of the inverse-gamma distributions which define the signal-to-noise ratio.

We use the command *dlmGibbsDIG* from the *dlm* package in our GUI to perform Bayesian inference in the univariate *dynamic linear model* with

random walk states. This function uses the FFBS algorithm, and assumes independent gamma priors for the precision (inverse of variance) parameters. In addition, this package uses the singular value decomposition to calculate the covariance matrices to avoid numerical instability.

Algorithm A20 shows how to perform inference in univariate *dynamic linear model* with random walk states in our GUI. See also Chapter 5 for details regarding the dataset structure.

Algorithm A20 Dynamic linear models

- 1: Select *Time series Model* on the top panel
 - 2: Select *Dynamic linear model* using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Set the hyperparameters of the *precision of the observation equation*: prior mean and variance.
 - 6: Set the hyperparameters of the *precision of the state equations*: just one set of prior mean and variance parameters.
 - 7: Click the *Go!* button
 - 8: Analyze results
 - 9: Download posterior chains of variances of observation and state equations, and posterior chains of states using the *Download Posterior Chains of Variances* and *Download Posterior Chains of States* buttons
-

Example: Simulation exercise of the dynamic linear model

We simulate the process $y_t = \beta_{t1} + x_t\beta_{t2} + \mu_t$ and $\beta_t = \beta_{t-1} + w_t$, $t = 1, 2, \dots, 200$, where $\beta_t = [\beta_{t1} \ \beta_{t2}]^\top$, $\mu_t \sim N(0, 0.5^2)$, $w_t \sim N(\mathbf{0}, \text{diag}\{0.2, 0.1\})$, $x_t \sim N(1, 1)$, β_0 and B_0 are the OLS estimates and variance of the recursive OLS estimates (see below), respectively.

The following algorithm shows how to perform inference using *dlmGibbsDIG*, and compares the results to the maximum likelihood estimator. The latter is based on *dlmMLE* function. We also use the *dlmSvd2var* function, that is based on the singular value decomposition, to calculate the variance of the smoothing states.

Users can observe that we employ a straightforward strategy for setting the hyperparameters. First, we recursively estimate the model using ordinary least squares (OLS), progressively increasing the sample size, and save the location parameters. Next, we compute the covariance matrix of this sequence and use it to set the priors: the prior mean of the precision of the state vector is set equal to the inverse of the maximum element of the main diagonal of this covariance matrix (*a.theta*), and the prior variance is set equal to ten times this value (*b.theta*). For the observation equation, the prior mean of the precision is set equal to the inverse the OLS variance estimate (*a.y*), and

the prior variance is set equal to ten times this value ($b.y$). We perform some sensitivity analysis of the results regarding the hyperparameters, and it seems that the results are robust. However, we encourage to give more consideration to empirical Bayes methods for setting hyperparameters in *state-space models*.

R code. Simulation: Dynamic linear model

```

1 rm(list = ls()); set.seed(010101)
2 T <- 200; sig2 <- 0.5^2
3 x <- rnorm(T, mean = 1, sd = 1)
4 X <- cbind(1, x); B0 <- c(1, 0.5)
5 K <- length(B0)
6 e <- rnorm(T, mean = 0, sd = sig2^0.5)
7 Omega <- diag(c(0.2, 0.1))
8 w <- MASS::mvrnorm(T, c(0, 0), Omega)
9 Bt <- matrix(NA, T, K); Bt[1,] <- B0
10 yt <- rep(NA, T)
11 yt[1] <- X[1,] %*% B0 + e[1]
12 for(t in 1:T){
13   if(t == 1){
14     Bt[t,] <- w[t,]
15   }else{
16     Bt[t,] <- Bt[t-1,] + w[t,]
17   }
18   yt[t] <- X[t,] %*% Bt[t,] + e[t]
19 }
20 RegLS <- lm(yt ~ x)
21 SumRegLS <- summary(RegLS)
22 SumRegLS; SumRegLS$sigma^2
23 Bp <- matrix(RegLS$coefficients, T, K, byrow = TRUE)
24 S <- 20
25 for(t in S:T){
26   RegLSt <- lm(yt[1:t] ~ x[1:t])
27   Bp[t,] <- RegLSt$coefficients
28 }
29 # plot(Bp[S:T,2], type = "l")
30 VarBp <- var(Bp)
31 # State space model
32 ModelReg <- function(par){
33   Mod <- dlm::dlmModReg(x, dV = exp(par[1]), dW = exp(par[2:3]),
34                           m0 = RegLS$coefficients, C0 = VarBp)
35   return(Mod)
36 }
37 outMLEReg <- dlm::dlmMLE(yt, parm = rep(0, K+1), ModelReg)
38 exp(outMLEReg$par)
39 RegFilter <- dlm::dlmFilter(yt, ModelReg(outMLEReg$par))
40 RegSmooth <- dlm::dlmSmooth(yt, ModelReg(outMLEReg$par))
41 SmoothB2 <- RegSmooth$s[-1,2]
42 VarSmooth <- dlm::dlmSvd2var(u = RegSmooth[["U.S"]], RegSmooth[["D.S"]])
43 SDVarSmoothB2 <- sapply(2:(T+1), function(t){VarSmooth[[t]][K,K]^0.5})
44 LimInfB2 <- SmoothB2 - qnorm(0.975)*SDVarSmoothB2
45 LimSupB2 <- SmoothB2 + qnorm(0.975)*SDVarSmoothB2
46 # Gibbs
47 MCMC <- 2000; burnin <- 1000
48 a.y <- (SumRegLS$sigma^2)^(-1); b.y <- 10*a.y; a.theta <- (
49   max(diag(VarBp)))^(-1); b.theta <- 10*a.theta
50 gibbsOut <- dlm::dlmGibbsDIG(yt, mod = dlm::dlmModReg(x), a.y = a.y,
51                                 b.y = b.y, a.theta = a.theta, b.theta = b.theta,
52                                 n.sample = MCMC, thin = 5, save.states = TRUE)

```

R code. Simulation: Dynamic linear model

```

1 B2t <- matrix(0, MCMC - burnin, T + 1)
2 for(t in 1:(T+1)){
3   B2t[,t] <- gibbsOut[["theta"]][t,2,-c(1:burnin)]
4 }
5 Lims <- apply(B2t, 2, function(x){quantile(x, c(0.025,
6   0.975))})
7 summary(coda::mcmc(gibbsOut[["dV"]]))
8 summary(coda::mcmc(gibbsOut[["dW"]]))
9 # Figure
10 require(latex2exp) # LaTeX equations in figures
11 xx <- c(1:(T+1), (T+1):1)
12 yy <- c(Lims[1,], rev(Lims[2,]))
13 plot(xx, yy, type = "n", xlab = "Time", ylab = TeX("$\\beta_{t2}$"))
14 polygon(xx, yy, col = "lightblue", border = "lightblue")
15 xxML <- c(1:T, T:1)
16 yyML <- c(LimInfB2, rev(LimSupB2))
17 polygon(xxML, yyML, col = "blue", border = "blue")
18 lines(colMeans(B2t), col = "red", lw = 2)
19 lines(Bt[,2], col = "black", lw = 2)
20 lines(SmoothB2, col = "green", lw = 2)
21 title("State vector: Slope parameter")

```

Figure 8.1 shows the comparison between maximum likelihood (ML) and Bayesian inference. The light blue (Bayesian) and dark blue (maximum likelihood) shadows show the credible and confidence intervals at 95% for the state slope parameter (β_{t2}). We see that the Bayesian interval encompass the ML interval. This is a reflection of the extra uncertainty of the unknown variances. The black line is the actual trajectory of β_{t2} , the green and red lines are the *smoothing* recursions using the ML and Bayesian estimates (posterior mean), respectively.

Example: Effects of inflation on interest rate I

We use the dataset *16INTDEF.csv* provided by [223, Chaps. 10] to study the effects of inflation on interest rate. The specification is $\Delta i_t = \beta_{t1} + \beta_{t2}\Delta inf_t + \beta_{t3}\Delta def_t + \mu_t$ and $\beta_t = \beta_{t-1} + \mathbf{w}_t$, where $\Delta z_t = z_t - z_{t-1}$ is the difference operator, i_t is the three-month T-bill rate, inf_t is the annual inflation rate based on the consumer price index (CPI), and def_t is the federal budget deficit as percentage of gross domestic product (GDP) from 1948 to 2003 in the USA. In addition, $\mu_t \sim N(0, \sigma^2)$, $\mathbf{w}_t \sim N(\mathbf{0}, \text{diag}\{\omega_1^2, \omega_1^2\})$. We assume inverse-gamma distributions for the priors of scale parameters, and set 12000 MCMC iterations, 2000 as burn-in, and 10 the thinning parameter.

The following code shows how to perform this application. We use the

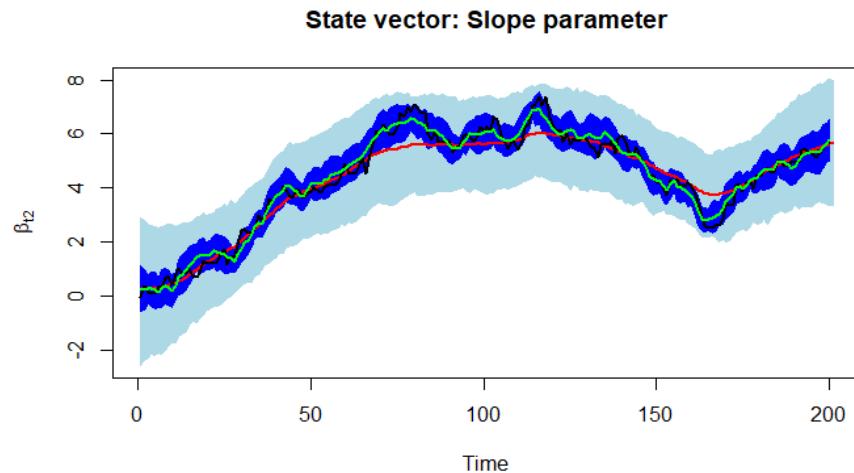


FIGURE 8.1
Simulation: Dynamic linear model.

variance of the recursive estimation of the OLS to set the hyperparameters of the inverse-gamma distribution for the variance of w_t , and the OLS estimate of the variance of the model to set the hyperparameters of the distribution of σ^2 . Note that as we are using the function *dlmGibbsDIG* from the package *dlm*, the hyperparameters are set in terms of precision parameters.

Figure 8.2 shows the posterior results of the effect of the inflation on the interest rate. This is a fan chart indicating deciles from 10% to 90%, the red shaded area shows around the median value, and the black line is the mean value of the state associated with the annual change in inflation. We see that the annual changes in interest rate are weakly positive related to annual changes in inflation.

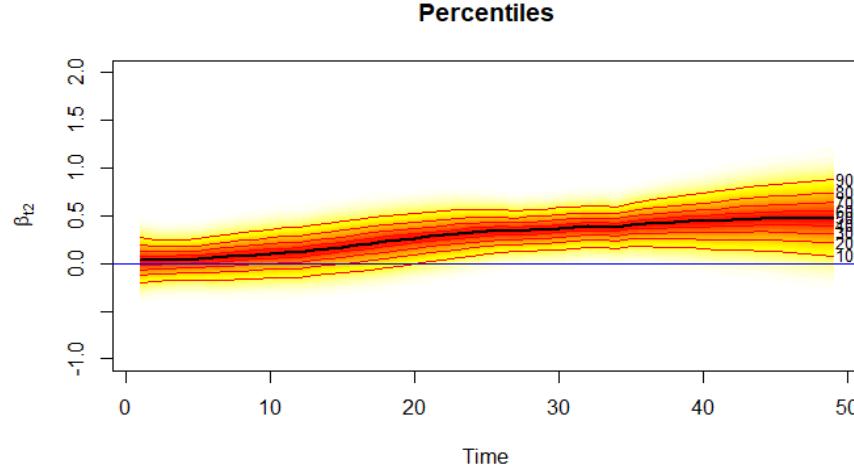
R code. Dynamic linear model: Effects of inflation on interest rate

```

1 rm(list = ls()); set.seed(010101)
2 DataIntRate <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/16INTDEF.csv"
  , sep = ",", header = TRUE, quote = "")
3 attach(DataIntRate); Xt <- cbind(diff(inf), diff(def))
4 K <- dim(Xt)[2] + 1; yt <- diff(i3)
5 T <- length(yt); RegLS <- lm(yt ~ Xt)
6 SumRegLS <- summary(RegLS); SumRegLS; SumRegLS$sigma^2
7 # Recursive OLS
8 Bp <- matrix(RegLS$coefficients, T, K, byrow = TRUE)
9 S <- 20
10 for(t in S:T){
11   RegLSt <- lm(yt[1:t] ~ Xt[1:t,])
12   Bp[t,] <- RegLSt$coefficients
13 }
14 VarBp <- var(Bp)
15 # State space model
16 ModelReg <- function(par){
17   Mod <- dlm::dlmModReg(Xt, dV = exp(par[1]), dW = exp(par
  [2:(K+1)]), m0 = RegLS$coefficients,
18   CO = diag(VarBp))
19   return(Mod)
20 }
21 MCMC <- 12000; burnin <- 2000; thin <- 10
22 a.y <- (SumRegLS$sigma^2)^(-1); b.y <- 10*a.y; a.theta <- (
  max(diag(VarBp)))^(-1); b.theta <- 10*a.theta
23 gibbsOut <- dlm::dlmGibbsDIG(yt, mod = dlm::dlmModReg(Xt), a
  .y = a.y, b.y = b.y, a.theta = a.theta, b.theta = b.
  theta, n.sample = MCMC, thin = 5, save.states = TRUE)
24 B2t <- matrix(0, MCMC - burnin, T + 1)
25 for(t in 1:(T+1)){
26   B2t[,t] <- gibbsOut[["theta"]][t,2,-c(1:burnin)]
27 }
28 dV <- coda::mcmc(gibbsOut[["dV"]][-c(1:burnin)])
29 dW <- coda::mcmc(gibbsOut[["dW"]][-c(1:burnin),])
30 summary(dV); summary(dW)
31 plot(dV); plot(dW)
32 library(fanplot); library(latex2exp)
33 df <- as.data.frame(B2t)
34 plot(NULL, main="Percentiles", xlim = c(1, T+1), ylim = c
  (-1, 2), xlab = "Time", ylab = TeX("$\\beta_{t1}$"))
35 fan(data = df); lines(colMeans(B2t), col = "black", lw = 2)
36 abline(h=0, col = "blue")

```

We can extend the *dynamic linear model* with *random walk states* to take into account time invariant location parameters. In particular, we follow [51],

**FIGURE 8.2**

Effects of inflation on interest rate: Dynamic linear model.

who propose the *simulation smoother*. This algorithm overcomes some shortcomings of the FFBS algorithm, such as slow convergence and computational overhead. We focus on the case $M = 1$,

$$Y_t = \mathbf{z}_t^\top \boldsymbol{\alpha} + \mathbf{x}_t^\top \boldsymbol{\beta}_t + \mathbf{h}_t^\top \boldsymbol{\epsilon}_t, \quad t = 1, 2, \dots, T. \quad (\text{Observation equation}) \quad (8.3)$$

$$\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} + \mathbf{H}_t \boldsymbol{\epsilon}_t, \quad t = 1, 2, \dots, T. \quad (\text{States equations}), \quad (8.4)$$

where \mathbf{z}_t and \mathbf{x}_t are L -dimensional and K -dimensional vectors of regressors associated with time-invariant and time-varying parameters, respectively, \mathbf{h}_t is a vector of dimension $1+K$, \mathbf{H}_t is a matrix of dimension $K \times 1+K$, $\boldsymbol{\beta}_0 = \mathbf{0}$ and $\boldsymbol{\epsilon}_t \sim N(\mathbf{0}_{1+K}, \sigma^2 \mathbf{I}_{1+K})$.

Observe that this specification encompasses Equations 8.1 and 8.2 setting $\boldsymbol{\epsilon}_t = [\mu_t \ \mathbf{w}_t^\top]^\top$, $\mathbf{h}_t = [1 \ 0 \ \dots \ 0]$, $\mathbf{H}_t = [\mathbf{0}_K \ \mathbf{U}_{K \times K}]$ such that $\text{diag}\{\omega_1^2 \ \dots \ \omega_K^2\} = \sigma^2 \mathbf{U} \mathbf{U}^\top$, $\boldsymbol{\alpha} = \mathbf{0}$, and $\mathbf{h}_t \mathbf{H}_t^\top = \mathbf{0}_K$.

The nice idea of [51] was to propose an efficient algorithm to get draws from $\boldsymbol{\eta}_t = \mathbf{F}_t \boldsymbol{\epsilon}_t$, where the most common choice is $\mathbf{F}_t = \mathbf{H}_t$, which means drawing samples from the perturbations of the states, and then, recovering the states from Equation 8.4 and $\boldsymbol{\beta}_0 = \mathbf{0}$. [51] present a more general version of the *state space model* than the one presented here.

Using the system given by Equations 8.3 and 8.4, $\mathbf{F}_t = \mathbf{H}_t$ and $\mathbf{h}_t \mathbf{H}_t^\top = \mathbf{0}_K$, the *filtering* recursions are given by $e_t = Y_t - \mathbf{z}_t^\top \boldsymbol{\alpha} - \mathbf{x}_t^\top \mathbf{b}_{t-1}$, $q_t = \mathbf{x}_t^\top \mathbf{B}_{t-1} \mathbf{x}_t + \mathbf{h}_t^\top \mathbf{h}_t$, $\mathbf{K}_t = \mathbf{B}_{t-1} \mathbf{x}_t q_t^{-1}$, $\mathbf{b}_t = \mathbf{b}_{t-1} + \mathbf{K}_t e_t$, and $\mathbf{B}_t = \mathbf{B}_{t-1} - \mathbf{B}_{t-1} \mathbf{x}_t \mathbf{K}_t^\top + \mathbf{H}_t \mathbf{H}_t^\top$, where $\mathbf{b}_0 = \mathbf{0}$ and $\mathbf{B}_0 = \mathbf{H}_0 \mathbf{H}_0^\top$. See system 2 in [51] for a more general case. We should save e_t (innovation vector), q_t (scale innovation variance) and \mathbf{K}_t (*Kalman gain*) from this recursion.

Then, setting $\mathbf{r}_T = \mathbf{0}$ and $\mathbf{M}_T = \mathbf{0}_K$, we run backwards from $t = T-1, T-2, \dots, 1$, the following recursions: $\boldsymbol{\Lambda}_{t+1} = \mathbf{H}_{t+1}\mathbf{H}_{t+1}^\top$, $\mathbf{C}_{t+1} = \boldsymbol{\Lambda}_{t+1} - \boldsymbol{\Lambda}_{t+1}\mathbf{M}_{t+1}\boldsymbol{\Lambda}_{t+1}^\top$, $\boldsymbol{\xi}_{t+1} \sim N(\mathbf{0}_K, \sigma^2\mathbf{C}_{t+1})$, $\mathbf{L}_{t+1} = \mathbf{I}_K - \mathbf{K}_{t+1}\mathbf{x}_{t+1}^\top$, $\mathbf{V}_{t+1} = \boldsymbol{\Lambda}_{t+1}\mathbf{M}_{t+1}\mathbf{L}_{t+1}$, $\mathbf{r}_t = \mathbf{x}_{t+1}\mathbf{e}_{t+1}/q_{t+1} + \mathbf{L}_{t+1}^\top\mathbf{r}_{t+1} - \mathbf{V}_{t+1}^\top\mathbf{C}_{t+1}^{-1}\boldsymbol{\xi}_{t+1}$, $\mathbf{M}_t = \mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top/q_{t+1} + \mathbf{L}_{t+1}^\top\mathbf{M}_{t+1}\mathbf{L}_{t+1} + \mathbf{V}_{t+1}^\top\mathbf{C}_{t+1}^{-1}\mathbf{V}_{t+1}$, and $\boldsymbol{\eta}_{t+1} = \boldsymbol{\Lambda}_{t+1}\mathbf{r}_{t+1} + \boldsymbol{\xi}_{t+1}$. [51] show that $\boldsymbol{\eta} = [\boldsymbol{\eta}_1^\top \dots \boldsymbol{\eta}_T^\top]^\top$ is drawn from $p(\mathbf{H}_t\boldsymbol{\epsilon}_t|y_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{h}_t, \mathbf{H}_t, \boldsymbol{\alpha}, \sigma^2, t = 1, 2, \dots, T)$. Thus, we can recover $\boldsymbol{\beta}_t$ using 8.4 and $\boldsymbol{\beta}_0 = \mathbf{0}_K$.

We assume in the model given by Equations 8.3 and 8.4 that $\mathbf{h}_t = [1 \ 0 \ \dots \ 0]^\top$ and $\mathbf{H}_t = [\mathbf{0}_K \ \text{diag}\{1/\tau_1 \dots 1/\tau_K\}]$, and then perform Bayesian inference assuming independent priors, that is, $\pi(\boldsymbol{\beta}_0, \boldsymbol{\alpha}, \sigma^2, \boldsymbol{\tau}) = \pi(\boldsymbol{\beta}_0)\pi(\boldsymbol{\alpha})\pi(\sigma^2)\prod_{k=1}^K \pi(\tau_k^2)$ where $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$, $\tau_k^2 \sim G(v_0/2, v_0/2)$, $k = 1, \dots, K$, $\boldsymbol{\alpha} \sim N(\mathbf{a}_0, \mathbf{A}_0)$ and $\boldsymbol{\beta}_0 \sim N(\mathbf{b}_0, \mathbf{B}_0)$. The conditional posterior distributions are $\sigma^2|\mathbf{y}, \mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}_{0:T}, \boldsymbol{\alpha}, \boldsymbol{\tau} \sim IG(\alpha_n/2, \delta_n/2)$, where $\delta_n = \sum_{t=1}^T [(\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1})^\top \boldsymbol{\Psi} (\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}) + (y_t - \mathbf{z}_t^\top \boldsymbol{\alpha} - \mathbf{x}_t^\top \boldsymbol{\beta}_t)^\top (y_t - \mathbf{z}_t^\top \boldsymbol{\alpha} - \mathbf{x}_t^\top \boldsymbol{\beta}_t)] + \delta_0$ and $\alpha_n = T(K+1) + \alpha_0$, $\boldsymbol{\tau} = [\tau_1 \ \dots \ \tau_K]$, $\boldsymbol{\Psi} = \text{diag}\{\tau_1^2, \dots, \tau_K^2\}$, and $\tau_k^2|\mathbf{y}, \mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}_{0:T}, \sigma^2 \sim G(v_{1n}/2, v_{2kn}/2)$, where $v_{1n} = T + v_0$ and $v_{2kn} = \sigma^{-2} \sum_{t=1}^T (\boldsymbol{\beta}_{t,k} - \boldsymbol{\beta}_{t-1,k})^2 + v_0$, and $\boldsymbol{\alpha}|\mathbf{y}, \mathbf{X}, \mathbf{Z}, \sigma^2, \boldsymbol{\beta}_{1:T}, \boldsymbol{\tau} \sim N(\mathbf{a}_n, \mathbf{A}_n)$, where $\mathbf{A}_n = (\mathbf{A}_0^{-1} + \sigma^{-2} \sum_{t=1}^T \mathbf{z}_t \mathbf{z}_t^\top)^{-1}$ and $\mathbf{a}_n = \mathbf{A}_n(\mathbf{A}_0^{-1}\mathbf{a}_0 + \sigma^{-2} \sum_{t=1}^T \mathbf{z}_t(y_t - \mathbf{x}_t^\top \boldsymbol{\beta}_t))$. The vector of the dependent variable is \mathbf{y} , and all regressors are in \mathbf{X} and \mathbf{Z} .

We can see that all the previous posterior distributions are conditional on the state vector $\boldsymbol{\beta}_{0:T}$, which can be sampled using the *simulation smoother* algorithm conditional on draws of the time-invariant parameters. Thus, the *state space model* provides an excellent illustration of the modular nature of the Bayesian framework where performing inference of more complex models very often simply involves adding new blocks to a MCMC algorithm. This means we can break down a complex inferential problem into smaller, more manageable parts, this is, a “divide and conquer” approach. This is possible due to the structure of conditional posterior distributions. Exercise 3 asks to perform a simulation of the model given by Equations 8.3 and 8.4, and program the MCMC algorithm including the *simulation smoother*.

8.2 ARMA processes

Since the seminal work of [23], autoregressive moving average (ARMA) models have become ubiquitous in time series analysis. Thus, we present a brief introduction to these models in this section.

Let's start with the linear Gaussian model with autoregressive errors,

$$Y_t = \mathbf{x}_t^\top \boldsymbol{\beta} + \mu_t \quad (8.5)$$

$$\phi(L)\mu_t = \epsilon_t, \quad (8.6)$$

where $\epsilon_t \stackrel{iid}{\sim} N(0, \sigma^2)$, $\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p$ is a polynomial in the lag operator (L) , where $Lz_t = z_{t-1}$, and in general, $L^r z_t = z_{t-r}$.

Thus, we see that stochastic error μ_t follows an *autoregressive process of order p*, that is, $\mu_t \sim AR(p)$. It is standard practice to assume that μ_t is second-order stationary, this implies that the mean, variance and autocovariance of μ_t are finite and independent of t and s , although $\mathbb{E}[\mu_t \mu_s]$ may depend on $|t-s|$. Then, all roots of $\phi(L)$ lie outside the unit circle, for instance, given an $AR(1)$, then, $1 - \phi_1 L = 0$, implies, $L = 1/\phi_1$ such that $|\phi_1| < 0$ for the process being second-order stationary.

The likelihood function conditional on the first p observations is

$$\begin{aligned} p(Y_{p+1}, \dots, Y_T | y_p, \dots, y_1, \boldsymbol{\theta}) &= \prod_{t=p+1}^T p(Y_t | H_{t-1}, \boldsymbol{\theta}) \\ &\propto \sigma^{-(T-p)} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{t=p+1}^T (Y_t - \hat{Y}_{t|t-1, \boldsymbol{\theta}})^2 \right\}, \end{aligned}$$

where H_{t-1} is the past history, $\boldsymbol{\theta}$ collects all parameters $(\boldsymbol{\beta}, \phi_1, \dots, \phi_p, \sigma^2)$, and $\hat{Y}_{t|t-1, \boldsymbol{\theta}} = (1 - \phi(L))Y_t + \phi(L)\mathbf{x}_t^\top \boldsymbol{\beta}$.

We can see that multiplying the first expression in Equation 8.5 by $\phi(L)$, we can express the model as

$$Y_t^* = \mathbf{x}_t^{*\top} \boldsymbol{\beta} + \epsilon_t \quad (8.7)$$

where $Y_t^* = \phi(L)Y_t$ and $\mathbf{x}_t^* = \phi(L)\mathbf{x}_t$.

Thus, collecting all observations $t = p+1, p+2, \dots, T$, we have $\mathbf{y}^* = \mathbf{X}^* \boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_{T-p})$, \mathbf{y}^* is a $T-p$ dimensional vector, and \mathbf{X}^* is a $(T-p) \times K$ dimensional matrix.

Assuming that $\boldsymbol{\beta} | \sigma \sim N(\boldsymbol{\beta}_0, \sigma^2 \mathbf{B}_0)$, $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$ and $\boldsymbol{\phi} \sim N(\boldsymbol{\phi}_0, \boldsymbol{\Phi}_0) \mathbb{1}[\boldsymbol{\phi} \in S_\phi]$, where S_ϕ is the stationary region of $\boldsymbol{\phi} = [\phi_1 \dots \phi_p]^\top$. Then, Equation 8.7 implies that $\boldsymbol{\beta} | \sigma^2, \boldsymbol{\phi}, \mathbf{y}, \mathbf{X} \sim N(\boldsymbol{\beta}_n, \sigma^2 \mathbf{B}_n)$, where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \mathbf{X}^{*\top} \mathbf{X}^*)^{-1}$ and $\boldsymbol{\beta}_n = \mathbf{B}_n(\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \mathbf{X}^{*\top} \mathbf{y}^*)$. In addition, $\sigma^2 | \boldsymbol{\beta}, \boldsymbol{\phi}, \mathbf{y}, \mathbf{X} \sim IG(\alpha_n/2, \delta_n/2)$ where $\alpha_n = \alpha_0 + T - p$ and $\delta_n = \delta_0 + (\mathbf{y}^* - \mathbf{X}^* \boldsymbol{\beta})^\top (\mathbf{y}^* - \mathbf{X}^* \boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \mathbf{B}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)$. Thus, the previous conditional posterior distributions imply that we can use a Gibbs sampling algorithm to perform inference of these parameters [39].

We know from Equation 8.5 that $\mu_t = Y_t - \mathbf{x}_t^\top \boldsymbol{\beta}$, from Equation 8.6 that $\mu_t = \phi_1 \mu_{t-1} + \dots + \phi_p \mu_{t-p} + \epsilon_t$, $t = p+1, \dots, T$. In matrix notation $\boldsymbol{\mu} = \mathbf{U} \boldsymbol{\phi} + \boldsymbol{\epsilon}$, where $\boldsymbol{\mu}$ is a $T-p$ dimensional vector, \mathbf{U} is a $(T-p) \times p$ matrix whose t -th row is $[\mu_{t-1} \dots \mu_{t-p}]$. Thus, the posterior distribution of $\boldsymbol{\phi} | \boldsymbol{\beta}, \sigma^2, \mathbf{y}, \mathbf{X}$ is

$N(\phi_n, \Phi_n) \mathbb{1}[\phi \in S_\phi]$, where $\Phi_n = (\Phi_0^{-1} + \sigma^{-2} \mathbf{U}^\top \mathbf{U})$ and $\phi_n = \Phi_n(\Phi_0^{-1} \phi_0 + \sigma^{-2} \mathbf{U}^\top \boldsymbol{\mu})$ (see Exercise 4).

Drawing from the model restricted to stationarity is straightforward: we simply sample from the multivariate normal distribution and discard draws that do not meet the stationarity condition. The proportion of draws that satisfy this restriction represents the conditional probability that the process is stationary.

Example: Effects of inflation on interest rate II

We specified a *dynamic linear model* in the example of the effects of inflation on interest rate to take into account a potential dynamic relationship. However, we can consider dynamics in this example assuming $\Delta i_t = \beta_1 + \beta_2 \Delta \text{inf}_t + \beta_3 \Delta \text{def}_t + \mu_t$ where $\mu_t = \phi \mu_{t-1} + \epsilon_t$, which implies $\Delta i_t = \beta_1(1 - \phi_1) + \phi_1 \Delta i_{t-1} + \beta_2(\Delta \text{inf}_t - \phi_1 \Delta \text{inf}_{t-1}) + \beta_3(\Delta \text{def}_t - \phi_1 \Delta \text{def}_{t-1}) + \epsilon_t$. Thus, we use again the dataset *16INTDEF.csv* provided by [223, Chaps. 10] to provide an illustration of linear regressions with *AR*(1) errors.

The following code shows how to perform this application using vague priors assuming $\alpha_0 = \delta_0 = 0.01$, $\boldsymbol{\beta}_0 = \mathbf{0}$, $\mathbf{B}_0 = \mathbf{I}$, $\boldsymbol{\phi}_0 = \mathbf{0}$ and $\Phi_0 = \mathbf{I}$. We use 15000 MCMC iterations plus a burn-in equal 5000, and thin equal to 5.

R code. AR(1) model: Effects of inflation on interest rate

```

1 rm(list = ls())
2 set.seed(010101)
3 DataIntRate <- read.csv("https://raw.githubusercontent.com/
   besmarter/BSTApp/refs/heads/master/DataApp/16INTDEF.csv"
   , sep = ",", header = TRUE, quote = "")
4 attach(DataIntRate)
5 yt <- diff(i3); ytlag <- dplyr::lag(yt, n = 1)
6 T <- length(yt)
7 Xt <- cbind(diff(inf), diff(def)); Xtag <- dplyr::lag(Xt, n
   = 1)
8 K <- dim(Xt)[2] + 1
9 Reg <- lm(yt ~ ytlag + I(Xt[,-1] - Xtag))
10 SumReg <- summary(Reg); SumReg
11 PostSig2 <- function(Beta, Phi){
12   Xstar<- matrix(NA, T-1, K - 1)
13   ystar <- matrix(NA, T-1, 1)
14   for(t in 2:T){
15     Xstar[t-1,] <- Xt[t,] - Phi*Xt[t-1,]
16     ystar[t-1,] <- yt[t] - Phi*yt[t-1]
17   }
18   Xstar <- cbind(1, Xstar)
19   an <- T - 1 + a0
20   dn <- d0 + t(ystar - Xstar%*%Beta)%*%(ystar - Xstar%*%Beta
      ) + t(Beta - b0)%*%B0i%*%(Beta - b0)
21   sig2 <- rvgamma::rinvgamma(1, shape = an/2, rate = dn/2)
22   return(sig2)
23 }
24 PostBeta <- function(sig2, Phi){
25   Xstar<- matrix(NA, T-1, K - 1)
26   ystar <- matrix(NA, T-1, 1)
27   for(t in 2:T){
28     Xstar[t-1,] <- Xt[t,] - Phi*Xt[t-1,]
29     ystar[t-1,] <- yt[t] - Phi*yt[t-1]
30   }
31   Xstar <- cbind(1, Xstar)
32   Xtxstar <- t(Xstar)%*%Xstar
33   Xtystar <- t(Xstar)%*%ystar
34   Bn <- solve(B0i + Xtxstar)
35   bn <- Bn%*%(B0i%*%b0 + Xtystar)
36   Beta <- MASS::mvrnorm(1, bn, sig2*Bn)
37   return(Beta)
38 }
39 PostPhi <- function(sig2, Beta){
40   u <- yt - cbind(1,Xt)%*%Beta
41   U <- u[-T]
42   ustар <- u[-1]
43   UtU <- t(U)%*%U
44   Utu <- t(U)%*%ustар
45   Phin <- solve(Phi0i + sig2^(-1)*UtU)
46   phin <- Phin%*%(Phi0i%*%phi0 + sig2^(-1)*Utu)
47   Phi <- trunchnorm::rtrunchnorm(1, a = -1, b = 1, mean = phin
      , sd = Phin^0.5)
48   return(Phi)
49 }

```

R code. AR(1) model: Effects of inflation on interest rate

```

1 # Hyperparameters
2 d0 <- 0.01; a0 <- 0.01
3 b0 <- rep(0, K); c0 <- 1;
4 B0 <- c0*diag(K); B0i <- solve(B0)
5 phi0 <- 0; Phi0 <- 1; Phi0i <- 1/Phi0
6 # MCMC parameters
7 mcmc <- 15000
8 burnin <- 5000
9 tot <- mcmc + burnin
10 thin <- 1
11 PostBetas <- matrix(0, mcmc+burnin, K)
12 PostSigma2s <- rep(0, mcmc+burnin)
13 PostPhis <- rep(0, mcmc+burnin)
14 Beta <- rep(0, K); Phi <- 0
15 sig2 <- SumReg$sigma^2; Phi <- SumReg$coefficients[2,1]
16 Beta <- SumReg$coefficients[c(1,3,4),1]
17 pb <- winProgressBar(title = "progress bar", min = 0, max =
   tot, width = 300)
18 for(s in 1:tot){
19   sig2 <- PostSig2(Beta = Beta, Phi = Phi)
20   PostSigma2s[s] <- sig2
21   Beta <- PostBeta(sig2 = sig2, Phi = Phi)
22   PostBetas[s,] <- Beta
23   Phi <- PostPhi(sig2 = sig2, Beta = Beta)
24   PostPhis[s] <- Phi
25   setWinProgressBar(pb, s, title=paste( round(s/tot*100, 0),
   "% done"))
26 }
27 close(pb)
28 keep <- seq((burnin+1), tot, thin)
29 PosteriorBetas <- coda::mcmc(PostBetas[keep ,])
30 summary(PosteriorBetas)
31 PosteriorSigma2 <- coda::mcmc(PostSigma2s[keep])
32 summary(PosteriorSigma2)
33 PosteriorPhi <- coda::mcmc(PostPhis[keep])
34 summary(PosteriorPhi)
35 dfBinf <- as.data.frame(PosteriorBetas[,2])
36 # Basic density
37 p <- ggplot(dfBinf, aes(x=var1)) +
38   geom_density(color="darkblue", fill="lightblue") +
39   geom_vline(aes(xintercept=mean(var1)), color="blue",
   linetype="dashed", linewidth=1) +
40   geom_vline(aes(xintercept=quantile(var1, 0.025)), color="red
   ", linetype="dashed", linewidth=1) +
41   geom_vline(aes(xintercept=quantile(var1, 0.975)), color="red
   ", linetype="dashed", linewidth=1) +
42   labs(title="Density effect of inflation on interest rate", x
   ="Effect of inflation", y = "Density")

```

Figure 8.3 shows the posterior density plot of the effects of inflation rate on interest rate. The posterior mean of this coefficient is approximately 0.25, and the credible interval at 95% is (0, 0.46), which indicates again that the annual changes in interest rate are weakly positive related to annual changes in inflation (see Figure 8.2 as reference).

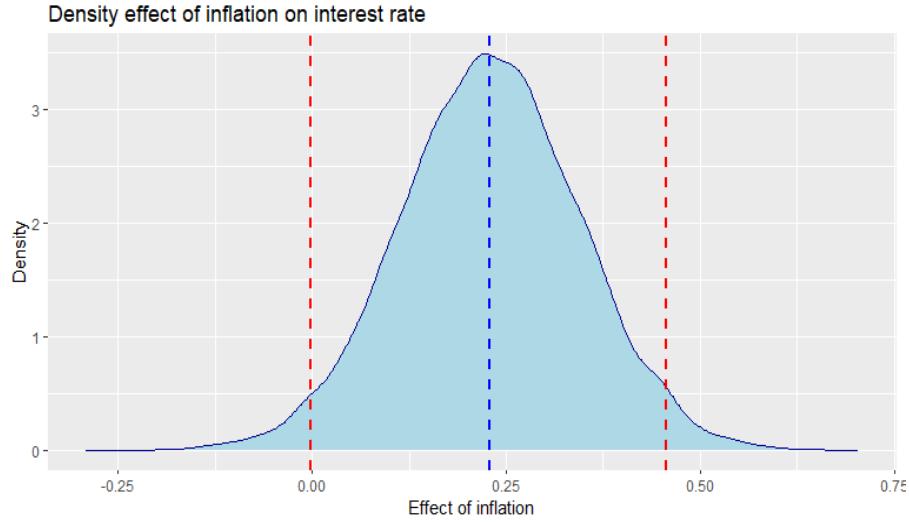


FIGURE 8.3

Density: Effects of inflation on interest rate.

Observe that the previous setting encompasses the particular relevant case $Y_t \sim AR(p)$, it is just omitting the covariates such that $Y_t = \mu_t$. [41] extend the Bayesian inference of linear regression with $AR(p)$ errors to $ARMA(p, q)$ errors using a *state-space* representation.

Setting $Y_t = \mu_t$ such that $Y_t = \sum_{s=1}^p \phi_j Y_{t-s} + \sum_{s=1}^q \theta_s \epsilon_{t-s} + \epsilon_t$, letting $r = \max\{p, q+1\}$, $\phi_s = 0$ for $s > p$ and $\theta_s = 0$ for $s > q$, and defining the matrices $\mathbf{x}^\top = [1 \ 0 \ \dots \ 0]$, $\mathbf{H} = [1 \ \psi_1 \ \dots \ \psi_{r-1}]^\top$, both are r -dimensional vectors,

$$\mathbf{G} = \begin{bmatrix} \phi_1 & 1 & 0 & \dots & 0 \\ \phi_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & & \\ \phi_{r-1} & 0 & 0 & \dots & 1 \\ \phi_r & 0 & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} \phi_1 & \vdots \\ \phi_2 & \vdots \\ \vdots & \vdots \\ \phi_r & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_r & 0 & 0 & \dots & 0 \end{bmatrix} \mathbf{I}_{r-1},$$

which is a $r \times r$ dimensional matrix, and give the *state* vector $\beta_t =$

$[\beta_{1,t} \ \beta_{2,t} \ \dots \ \beta_{r,t}]^\top$, the ARMA model has the following representation:

$$\begin{aligned} Y_t &= \mathbf{x}^\top \boldsymbol{\beta}_t \\ \boldsymbol{\beta}_t &= \mathbf{G}\boldsymbol{\beta}_{t-1} + \mathbf{H}\epsilon_t. \end{aligned}$$

This is a *dynamic linear model* where $\boldsymbol{\Sigma}_t = 0$, and $\boldsymbol{\Omega}_t = \sigma^2 \mathbf{H} \mathbf{H}^\top$ (see [163, 41]).

A nice advantage of the *state-space* representation of the ARMA model is that the evaluation of the likelihood can be performed efficiently using the recursive laws. Extensions to autoregressive integrated moving average ARIMA(p, d, q) models can be seen in [163, Chap. 3]. In ARIMA(p, d, q) models, d refers to the level of integration (difference) that is required to eliminate the stochastic trend in a time series (see [65, Chap. 4] for details).

Example: AR(2) process

Let's see the *state-space* representation of a stationary AR(2) process with intercept, that is, $Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \epsilon_t$, where $\epsilon_t \sim N(0, \sigma^2)$. Thus, $\mathbb{E}[Y_t] = \frac{\mu}{1-\phi_1-\phi_2}$, and variance $Var[Y_t] = \frac{\sigma^2(1-\phi_2)}{1-\phi_2-\phi_1^2-\phi_1^2\phi_2-\phi_2^2+\phi_2^3}$.

In addition, we can proof that setting $z_t = Y_t - \bar{\mu}$, we have $z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + \epsilon_t$ where $\mathbb{E}[z_t] = 0$, and these are equivalent representations (see Exercise 5). Then, setting $\mathbf{x}^\top = [1 \ 0]$, $\mathbf{H} = [1 \ 0]^\top$, $\mathbf{G} = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix}$, $\boldsymbol{\beta}_t = [\beta_{t1} \ \beta_{t2}]^\top$, $\boldsymbol{\Sigma}_t = 0$ and $\boldsymbol{\Omega}_t = \sigma^2$ we have

$$\begin{aligned} z_t &= \mathbf{x}^\top \boldsymbol{\beta}_t && \text{(Observation equations)} \\ \boldsymbol{\beta}_t &= \mathbf{G}\boldsymbol{\beta}_{t-1} + \mathbf{H}\epsilon_t && \text{(States equations).} \end{aligned}$$

We use the function *stan_sarima* from the package *bayesforecast* to perform Bayesian inference in ARMA models in our GUI. The following code shows how to simulate an AR(2) process, and perform Bayesian inference using this function.

R code. Simulation and inference: AR(2) model

```

1 rm(list = ls()); set.seed(010101)
2 T <- 200; mu <- 0.5
3 phi1 <- 0.5; phi2 <- 0.3; sig <- 0.5
4 Ey <- mu/(1-phi1-phi2); Sigy <- sig*((1-phi2)/(1-phi1-
   ^2-phi2*phi1^2-phi2^2+phi2^3))^0.5
5 y <- rnorm(T, mean = Ey, sd = Sigy)
6 e <- rnorm(T, mean = 0, sd = sig)
7 for(t in 3:T){
8   y[t] <- mu + phi1*y[t-1] + phi2*y[t-2] + e[t]
9 }
10 mean(y); sd(y)
11 y <- ts(y, start=c(1820, 1), frequency=1)
12 plot(y)
13 iter <- 10000; burnin <- 5000; thin <- 1; tot <- iter +
   burnin
14 library(bayesforecast)
15 sf1 <- bayesforecast::stan_sarima(y, order = c(2, 0, 0),
   prior_mu0 = normal(0, 1),
16 prior_ar = normal(0, 1), prior_sigma0 = inverse.gamma(0.01/
   2, 0.01/2),
17 seasonal = c(0, 0, 0), iter = tot, warmup = burnin, chains =
   1)
18 keep <- seq(burnin+1, tot, thin)
19 Postmu <- sf1[["stanfit"]]\$sim[["samples"]][[1]][["mu0"]][
   keep]
20 Postsig <- sf1[["stanfit"]]\$sim[["samples"]][[1]][["sigma0"]]
   ][keep]
21 Postphi1 <- sf1[["stanfit"]]\$sim[["samples"]][[1]][["ar0[1]"]]
   ][keep]
22 Postphi2 <- sf1[["stanfit"]]\$sim[["samples"]][[1]][["ar0[2]"]]
   ][keep]
23 Postdraws <- coda::mcmc(cbind(Postmu, Postsig, Postphi1,
   Postphi2))
24 summary(Postdraws)
25 Quantiles for each variable:
26          2.5%    25%    50%    75%   97.5%
27 Postmu  0.39914 0.5732 0.6625 0.7518 0.9346
28 Postsig  0.47696 0.5071 0.5248 0.5439 0.5829
29 Postphi1 0.42384 0.5159 0.5634 0.6089 0.6979
30 Postphi2 0.06034 0.1456 0.1920 0.2361 0.3286
31 plot(Postdraws)

```

We perform 10000 MCMC iterations plus a burn-in equal 5000 assuming $\sigma^2 \sim IG(0.01/2, 0.01/2)$, $\mu \sim N(0, 1)$ and $\phi_k \sim N(0, 1)$, $k = 1, 2$. The trace plots look well, and all 95% credible intervals encompass the population values.

Algorithm A21 shows how to perform inference in $ARMA(p, q)$ models using our GUI. See also Chapter 5 for details regarding the dataset structure.

Algorithm A21 Autoregressive Moving Average (ARMA) models

- 1: Select *Time series Model* on the top panel
- 2: Select *ARMA* using the left radio button
- 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
- 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
- 5: Set the order of the *ARMA* model, p and q parameters
- 6: Set the frequency: annual (1), quarterly (4), monthly (12), etc
- 7: Set the location and scale hyperparameters of the *intercept*, autoregressive (*AR*), moving average (*MA*) and standard deviation. Take into account that there is just one set of hyperparameters for *AR* and *MA* coefficients. This step is not necessary as by default our GUI uses non-informative priors
- 8: Click the *Go!* button
- 9: Analyze results
- 10: Download posterior chains using the *Download Posterior Chains* button

The function *stan_sarima* uses software *Stan* [206], which in turn uses *Hamiltonian Monte Carlo* (HMC). The following code shows how to perform Bayesian inference in the *AR(2)* model programming the HMC from scratch. We should clarify that this is only an illustration as HMC is less efficient than the Gibbs sampler in this example. However, HMC can outperform traditional MCMC algorithms in more complex models, especially when dealing with high-dimensional probability distributions or when MCMC struggles with poor mixing due to posterior correlation.

We perform the simulation in the first block setting $\mu = 0.5$, $\phi_1 = 0.5$, $\phi_2 = 0.3$ and $\sigma = 0.25$, and the sample size equal 200. Then, we set the hyperparameters, and the function to calculate the logarithm of the posterior distribution. We parametrize the model using $\tau = \log(\sigma^2)$ such that $\sigma^2 = \exp\{\tau\}$, consequently, avoiding issues due to the restriction of non-negativity of σ^2 . Thus, we must take into account the Jacobian due to the transformation, that is $d\sigma^2/d\tau = \exp(\tau)$. Then, we have the function to calculate the gradient vector of the log posterior distribution. We should calculate the gradient vector analytically because using finite difference can be computationally expensive. However, it can be a good idea to check the analytical calculations evaluation the function at the maximum posterior estimate, where this function should return values near 0, or compare results with finite differences in a few evaluation points.

The posterior distribution is given by³

$$\begin{aligned}\pi(\mu, \phi_1, \phi_2, \tau | \mathbf{y}) &\propto \prod_{t=3}^T (\exp(\tau))^{-1/2} \exp \left\{ -\frac{1}{2 \exp(\tau)} (y_t - \mu - \phi_1 y_{t-1} - \phi_2 y_{t-2})^2 \right\} \\ &\quad \times \exp \left\{ -\frac{1}{2\sigma_\mu^2} (\mu - \mu_0)^2 \right\} \times \exp \left\{ -\frac{1}{2\sigma_{\phi_1}^2} (\phi_1 - \phi_{10})^2 \right\} \\ &\quad \times \exp \left\{ -\frac{1}{2\sigma_{\phi_2}^2} (\phi_2 - \phi_{20})^2 \right\} \times \exp \{-(\alpha_0/2 + 1)\tau\} \exp \{-\delta_0/(2 \exp(\tau))\} \exp(\tau).\end{aligned}$$

The components of the gradient vector of the log posterior distribution are given by

$$\begin{aligned}\frac{\partial \log(\pi(\mu, \phi_1, \phi_2, \tau | \mathbf{y}))}{\partial \mu} &= \frac{\sum_{t=3}^T (y_t - \mu - \phi_1 y_{t-1} - \phi_2 y_{t-2})}{\exp(\tau)} - \frac{1}{\sigma_\mu^2} (\mu - \mu_0) \\ \frac{\partial \log(\pi(\mu, \phi_1, \phi_2, \tau | \mathbf{y}))}{\partial \phi_1} &= \frac{\sum_{t=3}^T (y_t - \mu - \phi_1 y_{t-1} - \phi_2 y_{t-2}) y_{t-1}}{\exp(\tau)} - \frac{1}{\sigma_{\phi_1}^2} (\phi_1 - \phi_{10}) \\ \frac{\partial \log(\pi(\mu, \phi_1, \phi_2, \tau | \mathbf{y}))}{\partial \phi_2} &= \frac{\sum_{t=3}^T (y_t - \mu - \phi_1 y_{t-1} - \phi_2 y_{t-2}) y_{t-2}}{\exp(\tau)} - \frac{1}{\sigma_{\phi_2}^2} (\phi_2 - \phi_{20}) \\ \frac{\partial \log(\pi(\mu, \phi_1, \phi_2, \tau | \mathbf{y}))}{\partial \tau} &= -\frac{(T-2)}{2} + \frac{\sum_{t=3}^T (y_t - \mu - \phi_1 y_{t-1} - \phi_2 y_{t-2})^2}{2 \exp(\tau)} \\ &\quad - (\alpha_0/2 + 1) + \delta_0/(2 \exp(\tau)) + 1.\end{aligned}$$

Then, we have the code for the Hamiltonian Monte Carlo as given in Chapter 4. We set the initial values equal to $\mu = \bar{y} = 1/(T-2) \sum_{t=3}^T y_t$, $\phi_1 = \phi_2 = 0$ and $\exp((1/(T-2)) \sum_{t=3}^T (y_t - \bar{y})^2)$, and M equal to the inverse covariance matrix of the posterior distribution evaluated at the maximum a posterior estimate. In addition, ϵ is randomly draw from a uniform distribution between 0 and $2\epsilon_0$, and L is the highest integer near $1/\epsilon$, this to approximately satisfy $L\epsilon = 1$.

We can check that all 95% credible intervals encompass the population values, and the posterior means are near the population values. The acceptance rate is higher than 65% on average, thus we should increase the base step (ϵ_0). In addition, we do not impose the stationary conditions on ϕ_1 and ϕ_2 . Exercise 6 asks to program a HMC taking into account these requirements.

³Take into account that we do not consider the first two observations when present the likelihood, this is no an issue when there is a large sample size.

R code. Simulation and inference: AR(2) model using Hamiltonian Monte Carlo

```

1 # Simulation AR(2)
2 rm(list = ls()); set.seed(010101); T <- 1000; K <- 4
3 mu <- 0.5; phi1 <- 0.5; phi2 <- 0.3; sig <- 0.5
4 Ey <- mu/(1-phi1-phi2); Sigy <- sig*((1-phi2)/(1-phi2-phi1
  ^2-phi2*phi1^2-phi2^2+phi2^3))^0.5
5 y <- rnorm(T, mean = Ey, sd = Sigy); e <- rnorm(T, mean = 0,
  sd = sig)
6 for(t in 3:T){
7   y[t] <- mu + phi1*y[t-1] + phi2*y[t-2] + e[t]
8 }
9 # Hyperparameters
10 d0 <- 0.01; a0 <- 0.01; mu0 <- 0; MU0 <- 1
11 phi0 <- c(0, 0); Phi0 <- diag(2)
12 # Log posterior multiply by -1 to use optim
13 LogPost <- function(theta, y){
14   mu <- theta[1]; phi1 <- theta[2]; phi2 <- theta[3]
15   tau <- theta[4]; sig2 <- exp(tau); logLik <- NULL
16   for(t in 3:T){
17     logLikt <- dnorm(y[t], mean = mu + phi1*y[t-1] + phi2*y[
      t-2], sd = sig2^0.5, log = TRUE)
18     logLik <- c(logLik, logLikt)
19   }
20   logLik <- sum(logLik)
21   logPrior <- dnorm(mu, mean = mu0, sd = MU0^0.5, log = TRUE
    ) + dnorm(phi1, mean = phi0[1], sd = Phi0[1,1]^0.5, log
    = TRUE) + dnorm(phi2, mean = phi0[2], sd = Phi0
    [2,2]^0.5, log = TRUE) + invgamma::dinvgamma(sig2, shape
    = a0/2, rate = d0/2, log = TRUE)
22   logPosterior <- logLik + logPrior + tau
23   return(-logPosterior) # Multiply by -1 to minimize using
    optim
24 }
25 theta0 <- c(mean(y), 0, 0, var(y))
26 Opt <- optim(theta0, LogPost, y = y, hessian = TRUE)
27 theta0 <- Opt$par; VarPost <- solve(Opt$hessian)
28 # Gradient log posterior
29 GradientTheta <- function(theta, y){
30   mu <- theta[1]; phi1 <- theta[2]; phi2 <- theta[3]
31   tau <- theta[4]; sig2 <- exp(tau); SumLik <- matrix(0, 3,
    1)
32   SumLik2 <- NULL
33   for(t in 3:T){
34     xt <- matrix(c(1, y[t-1], y[t-2]), 3, 1)
35     SumLikt <- (y[t] - (mu + phi1*y[t-1] + phi2*y[t-2]))*xt
36     SumLik2t <- (y[t] - (mu + phi1*y[t-1] + phi2*y[t-2]))^2
37     SumLik <- rowSums(cbind(SumLik, SumLikt))
38     SumLik2 <- sum(SumLik2, SumLik2t)
39   }
40   Grad_mu <- SumLik[1]/sig2 - (1/MU0)*(mu - mu0)
41   Grad_phi1 <- SumLik[2]/exp(tau) - 1/Phi0[1,1]*(phi1 - phi0
    [1])
42   Grad_phi2 <- SumLik[3]/exp(tau) - 1/Phi0[2,2]*(phi2 - phi0
    [2])
43   Grad_tau <- -(T-2)/2 + SumLik2/(2*exp(tau)) - (a0/2 + 1) +
    d0/(2*exp(tau)) + 1
44   Grad <- c(Grad_mu, Grad_phi1, Grad_phi2, Grad_tau)
45   return(Grad)
46 }
```

R code. Simulation and inference: AR(2) model using Hamiltonian Monte Carlo

```

1 # Hamiltonian Monte Carlo function
2 HMC <- function(theta, y, epsilon, M){
3   L <- ceiling(1/epsilon)
4   Minv <- solve(M); thetata <- theta
5   K <- length(thetata)
6   mom <- t(mvtnorm::rmvnorm(1, rep(0, K), M))
7   logPost_Mom_t <- -LogPost(thetata, y) + mvtnorm::dmvnorm(t
      (mom), rep(0, K), M, log = TRUE)
8   for(l in 1:L){
9     if(l == 1 | l == L){
10       mom <- mom + 0.5*epsilon*GradientTheta(theta, y)
11       theta <- theta + epsilon*Minv%*%mom
12     }else{
13       mom <- mom + epsilon*GradientTheta(theta, y)
14       theta <- theta + epsilon*Minv%*%mom
15     }
16   }
17   logPost_Mom_star <- -LogPost(theta, y) + mvtnorm::dmvnorm
      (t(mom), rep(0, K), M, log = TRUE)
18   alpha <- min(1, exp(logPost_Mom_star-logPost_Mom_t))
19   u <- runif(1)
20   if(u <= alpha){
21     thetaNew <- c(theta)
22   }else{
23     thetaNew <- thetata
24   }
25   rest <- list(theta = thetaNew, Prob = alpha)
26   return(rest)
27 }
28 # Posterior draws
29 S <- 1000; burnin <- 1000; thin <- 2; tot <- S + burnin
30 thetaPost <- matrix(NA, tot, K)
31 ProbAccept <- rep(NA, tot)
32 theta0 <- c(mean(y), 0, 0, exp(var(y)))
33 M <- solve(VarPost); epsilon0 <- 0.1
34 pb <- winProgressBar(title = "progress bar", min = 0, max =
      tot, width = 300)
35 for(s in 1:tot){
36   epsilon <- runif(1, 0, 2*epsilon0)
37   L <- ceiling(1/epsilon)
38   HMCs <- HMC(theta = theta0, y, epsilon, M)
39   theta0 <- HMCs$theta
40   thetaPost[s,] <- HMCs$theta
41   ProbAccept[s] <- HMCs$Prob
42   setWinProgressBar(pb, s, title=paste( round(s/tot*100, 0),
      "% done"))
43 }
44 close(pb)
45 keep <- seq((burnin+1), tot, thin)
46 thetaF <- coda::mcmc(thetaPost[keep,])
47 summary(thetaF)
48 summary(exp(thetaF[,K]))
49 ProbAcceptF <- coda::mcmc(ProbAccept[keep])
50 summary(ProbAcceptF)

```

8.3 Stochastic volatility models

A notable example of non-linear and non-Gaussian *state-space models* is stochastic volatility models (SVMs), which are widely used to model the volatility of financial returns. SVMs have gained significant attention due to their flexibility, ability to capture complex dynamics such as asymmetries, and ease of generalization to simultaneously model multiple returns, making them advantageous over generalized autoregressive conditional heteroskedasticity (GARCH) models proposed by [21]. However, estimating SVMs is more challenging than estimating GARCH models. This is because GARCH models determine variance in a deterministic manner, whereas SVMs do so stochastically. Consequently, GARCH models are typically estimated using maximum likelihood methods, while SVMs require Bayesian approaches, adding complexity to the estimation process.

The specification of the stochastic volatility model is given by

$$y_t = \mathbf{x}_t^\top \boldsymbol{\beta} + \exp\{0.5h_t\} \mu_t \quad (\text{Observation equation}) \quad (8.8)$$

$$h_t = \mu + \phi(h_{t-1} - \mu) + \sigma w_t \quad (\text{State equation}), \quad (8.9)$$

where y_t are the log-returns, \mathbf{x}_t are controls, $\boldsymbol{\beta}$ are time-invariant location parameters, $\mu_t \sim N(0, 1)$, $w_t \sim N(0, 1)$, $\mu_t \perp w_t$, the initial log-variance process $h_0 \sim N(\mu, \sigma^2/(1 - \phi^2))$, μ , ϕ and σ are the level, persistence and standard deviation of the log-variance, respectively.

Given the specification in Equations 8.8 and 8.9, we can write down the observation equation as $\log\{(y_t - \mathbf{x}_t^\top \boldsymbol{\beta})^2\} = h_t + \log(\mu_t^2)$, thus there is now a linear, but non-Gaussian *state-space model*. [121] approximate the distribution of $\log(\mu_t^2)$ by a mixture of normal distributions, that is, $\log(\mu_t^2)|l_t \sim N(m_{l_t}, s_{l_t}^2)$, $l_t \in \{1, 2, \dots, 10\}$ defines the mixture component indicator at time t . Thus, $\log\{(y_t - \mathbf{x}_t^\top \boldsymbol{\beta})^2\} = h_t + \log(\mu_t^2)$ and $h_t = \mu + \phi(h_{t-1} - \mu) + \sigma w_t$ can be written as a linear and conditionally Gaussian *state-space model*, where $\log\{(y_t - \mathbf{x}_t^\top \boldsymbol{\beta})^2\} = m_{l_t} + h_t + \mu_t^2$, $\mu_t \sim N(0, s_{l_t}^2)$.

We use the *stochvol* package in our GUI to perform MCMC inference in the SVMs [100]; this package in turn is based on the MCMC algorithms proposed by [121]. The default prior distributions in the *stochvol* package are $\boldsymbol{\beta} \sim N(\mathbf{b}_0, \mathbf{B}_0)$, $\mu \sim N(\mu_0, \sigma_{\mu_0}^2)$, $(\phi + 1)/2 \sim B(\alpha_0, \beta_0)$, and $\sigma^2 \sim G(1/2, 1/(2\sigma_{\sigma^2}^2))$. The prior distribution of ϕ is set to achieve stationarity of the process ($\phi \in (-1, 1)$). Most of applications find $\phi \approx 1$, thus authors of the package recommend to set $\alpha_0 \gtrsim 5$ and $\beta_0 \approx 1.5$. The prior distribution of σ^2 is equivalent to $\sigma \sim |N(0, \sigma_{\sigma^2}^2)|$ (half-normal distribution); this is recommended by the authors as the convenient conjugate inverse-gamma distribution does not work well due to bounding σ away from 0, which is no a good feature when modeling the log-variance of log-returns.

Algorithm A22 shows how to perform inference in stochastic volatility

models using our GUI. See also Chapter 5 for details regarding the dataset structure.

Algorithm A22 Stochastic volatility models

- 1: Select *Time series Model* on the top panel
 - 2: Select *Stochastic volatility* using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Set the hyperparameters: the mean and standard deviation of the Gaussian prior for the regression parameters, mean and standard deviation for the Gaussian prior distribution of the level of the log-volatility, shape parameters for the Beta prior distribution of the transformed persistence parameter, and the positive real number, which stands for the scaling of the transformed volatility of log-volatility. This step is not necessary as by default our GUI uses default values in *stochvol* package
 - 6: Click the *Go!* button
 - 7: Analyze results
 - 8: Download posterior chains of the fixed coefficients, and the states using the *Download Posterior Chains Fixed* and *Download Posterior States* buttons
-

Example: Simulation exercise of the stochastic volatility model

The following code shows how to simulate and perform Bayesian inference in the stochastic volatility model using the function *svsample* from the *stochvol* package. We set the stochastic volatility parameters $\mu = -10$, $\phi = 0.95$ and $\sigma = 0.3$. We assume two regressors that distribute standard normal, $\beta = [0.5 \ 0.3]^\top$, and the sample size is 1250, which is approximately having daily returns during 5 years. We use the default hyperparameters, 10000 MCMC iterations, a burn-in equal 5000 and a thin parameter equal 5.

The summary statistics of the posterior draws show that all 95% credible intervals encompass the population parameters, and posterior chains seem to achieve convergence. Figure 8.4 displays the posterior results of the volatility (h_t). The posterior mean (blue) follow the “observed” series (black), and the 95% credible intervals (light blue) encompass most of the time the “observed” series.

R code. Simulation and inference: Stochastic volatility model

```

1 rm(list = ls()); set.seed(010101)
2 T <- 1250; K <- 2
3 X <- matrix(rnorm(T*K), T, K)
4 B <- c(0.5, 0.3); mu <- -10; phi <- 0.95; sigma <- 0.3
5 h <- numeric(T); y <- numeric(T)
6 h[1] <- rnorm(1, mu, sigma / sqrt(1 - phi^2)) # Initial
    state
7 y[1] <- X[,1]*%*%B + rnorm(1, 0, exp(h[1] / 2))           #
    Initial observation
8 for (t in 2:T) {
9   h[t] <- mu + phi*(h[t-1]-mu) + rnorm(1, 0, sigma)
10  y[t] <- X[,t]*%*%B + rnorm(1, 0, sd = exp(0.5*h[t]))
11 }
12 df <- as.data.frame(cbind(y, X))
13 colnames(df) <- c("y", "x1", "x2")
14 MCMC <- 10000; burnin <- 10000; thin <- 5
15 res <- stochvol::svsample(y, designmatrix = X, draws = MCMC,
    burnin = burnin, thin = thin, priormu = c(0, 100),
    priorsigma = c(1), priorphi = c(5, 1.5), priorbeta = c
    (0, 10000))
16 summary(res[["para"]][[1]][,-c(4,5)])
17 summary(res[["beta"]])
18 ht <- res[["latent"]][[1]]
19 library(dplyr)
20 library(ggplot2)
21 library(latex2exp)
22 ggplot2::theme_set(theme_bw())
23 x_means <- colMeans(ht)
24 x_quantiles <- apply(ht, 2, function(x) quantile(x, probs =
    c(0.025, 0.975)))
25 df <- tibble(t = seq(1, T), mean = x_means, lower = x_
    quantiles[1, ], upper = x_quantiles[2, ], x_true = h,
    observations = y)
26 plot_filtering_estimates <- function(df) {
27   p <- ggplot(data = df, aes(x = t)) + geom_ribbon(aes(ymin =
      lower, ymax = upper), alpha = 1, fill = "lightblue") +
      geom_line(aes(y = x_true), colour = "black", alpha = 1,
      linewidth = 0.5) + geom_line(aes(y = mean), colour =
      "blue", linewidth = 0.5) + ylab(TeX("\$h_{\$t\$}")) + xlab("Time")
28   print(p)
29 }
30 plot_filtering_estimates(df)

```

So far, we have used MCMC algorithms to perform inference in *state-space models*. These algorithms require all observations to estimate the unknown

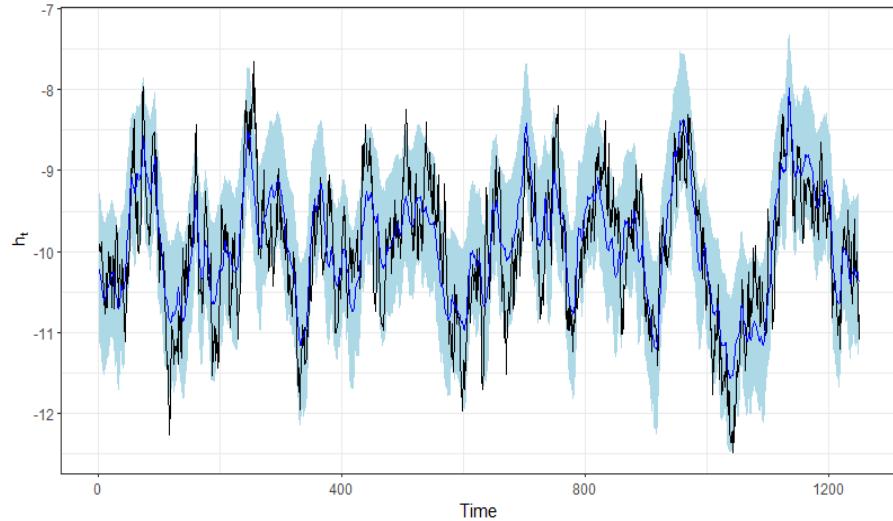


FIGURE 8.4
Stochastic volatility model.

parameters, a process referred to as offline or batch inference. However, this approach has limitations when online inference is needed, as every new observation requires simulating a new posterior chain. This is because MCMC algorithms do not naturally adapt to sequential updates. In contrast, particle filter algorithms, which are a subset of sequential Monte Carlo (SMC) methods, are specifically designed for sequential use, making them suitable for online inference.

Remember from Chapter 4 that particle filters (sequential Monte Carlo) are algorithms that allow computing a numerical approximation to the filtering distribution $\pi(\theta_{1:t} | \mathbf{y}_{1:t})$ sequentially in time. This is particularly relevant in non-linear and non-Gaussian models where there is no analytical solution for the filtering distribution.

The following code shows how to perform particle filtering in the vanilla stochastic volatility model assuming that the proposal distribution is the conditional prior distribution, that is, $q(h_t | h_{t-1}, y_t) = \pi(h_t | h_{t-1})$, which is normal with mean $\mu + \phi(h_{t-1} - \mu)$ and variance σ^2 . This choice implies that the incremental importance weights are equal to $p(y_t | h_t)$, which is $N(0, \exp(h_t))$. Therefore, the weights are proportional to the likelihood function. We perform multinomial resampling every time period in the code, and start the algorithm in the stationary distribution of h_t . Remember that there are other resampling approaches that are more efficient, for instance, residual resampling (see Section 4.3). We ask in Exercise 7 to modify this code to perform resampling when the effective sample size is lower than 50% of the number of

particles. In addition, we ask to program a sequential importance sampling, and check why is important to perform resampling in this simple example.

Figure 8.5 illustrates the filtering recursion using SMC with uneven weights (blue line), even weights (purple line), bands corresponding to plus/minus two standard deviations (light blue shaded area), and the true state (black line).⁴ The results indicate that SMC performs well even in a simple implementation, with no significant differences between using even and uneven weights (see Chapter 4).

In this example we use the population parameters to perform the filtering recursion. However, this is no the case in practice, that is, we have to estimate the time invariant parameters. Therefore, there are more elaborate algorithms to achieve this (see Chapter 4). For instance, [6] propose particle Markov chain Monte Carlo, this is a family of methods that combines MCMC and SMC. See [48] for a tutorial of particle Metropolis-Hastings in **R**. A potential practical solution, for applications that require a sequential updating of a posterior distribution over an unbounded time horizon, is to estimate offline the time invariant parameters using MCMC algorithms up to a specific time period, and then update sequentially online the state vector during subsequent time periods, and iterate this process. This is not optimal, but it can be practical.

⁴This standard deviation estimates the conditional posterior's standard deviation derived from the particles, not the estimator's standard deviation. The latter requires several independent particle runs on the same data.

R code. Simulation and inference: Stochastic volatility model programming sequential Monte Carlo from scratch

```

1 rm(list = ls()); set.seed(010101)
2 T <- 1250; mu <- -10; phi <- 0.95; sigma <- 0.3
3 h <- numeric(T); y <- numeric(T)
4 h[1] <- rnorm(1, mu, sigma / sqrt(1 - phi^2))
5 y[1] <- rnorm(1, 0, exp(h[1] / 2))
6 for (t in 2:T) {
7   h[t] <- mu + phi*(h[t-1]-mu) + rnorm(1, 0, sigma)
8   y[t] <- rnorm(1, 0, sd = exp(0.5*h[t]))
9 }
10 N <- 10000
11 log_Weights <- matrix(NA, N, T) # Log weights
12 Weights <- matrix(NA, N, T) # Weights
13 WeightsST <- matrix(NA, N, T) # Normalized weights
14 WeightsSTT <- matrix(1/N, N, T) # Normalized weights bar
15 particles <- matrix(NA, N, T) # Particles
16 particlesT <- matrix(NA, N, T) # Particles bar
17 logalphas <- matrix(NA, N, T) # Incremental importance
18 particles[, 1] <- rnorm(N, mu, sigma / sqrt(1 - phi^2)) #
  Stationary prior
19 log_Weights[, 1] <- dnorm(y[1], 0, sd = exp(0.5*particles
  [,1]), log = TRUE) # Likelihood
20 Weights[, 1] <- exp(log_Weights[, 1])
21 WeightsST[, 1] <- Weights[, 1] / sum(Weights[, 1])
22 ESS[1] <- (sum(WeightsST[, 1]^2))^{(-1)}
23 ind <- sample(1:N, size = N, replace = TRUE, prob =
  WeightsST[, 1]) # Resample
24 particles[, 1] <- particles[ind, 1] # Resampled particles
25 particlesT[, 1] <- particles[, 1] # Resampled particles
26 WeightsST[, 1] <- rep(1/N, N) # Resampled weights
27 pb <- winProgressBar(title = "progress bar", min = 0, max =
  T, width = 300)
28 for (t in 2:T) {
29   particles[, t] <- rnorm(N, mu + phi*(particles[, t - 1] -
    mu), sigma) # Sample from proposal
30   logalphas[, t] <- dnorm(y[t], 0, sd = exp(0.5*particles[, t]),
    log = TRUE)
31   Weights[, t] <- exp(logalphas[, t])
32   WeightsST[, t] <- Weights[, t] / sum(Weights[, t])
33   if(t < T){
34     ind <- sample(1:N, size = N, replace = TRUE, prob =
      WeightsST[, t])
35     particles[, 1:t] <- particles[ind, 1:t]
36   }else{
37     ind <- sample(1:N, size = N, replace = TRUE, prob =
      WeightsST[, t])
38     particlesT[, 1:t] <- particles[ind, 1:t]
39   }
40   setWinProgressBar(pb, t, title=paste( round(t/T*100, 0), "
    % done"))
41 }
42 close(pb)

```

R code. Simulation and inference: Stochastic volatility model programming sequential Monte Carlo from scratch

```
1 FilterDist <- colSums(particles * WeightsST)
2 SDFilterDist <- (colSums(particles^2 * WeightsST) -
3   FilterDist^2)^0.5
4 FilterDistT <- colSums(particlesT * WeightsSTT)
5 SDFilterDistT <- (colSums(particlesT^2 * WeightsSTT) -
6   FilterDistT^2)^0.5
7 MargLik <- colMeans(Weights)
8 plot(MargLik, type = "l")
9 library(dplyr)
10 library(ggplot2)
11 require(latex2exp)
12 ggplot2::theme_set(theme_bw())
13 Tfig <- 250
14 keepFig <- 1:Tfig
15 df <- tibble(t = keepFig,
16   mean = FilterDist[keepFig],
17   lower = FilterDist[keepFig] - 2*SDFilterDist[keepFig],
18   upper = FilterDist[keepFig] + 2*SDFilterDist[keepFig],
19   meanT = FilterDistT[keepFig],
20   lowerT = FilterDistT[keepFig] - 2*SDFilterDistT[keepFig],
21   upperT = FilterDistT[keepFig] + 2*SDFilterDistT[keepFig],
22   x_true = h[keepFig])
23 plot_filtering_estimates <- function(df) {
24   p <- ggplot(data = df, aes(x = t)) +
25     geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 1,
26     fill = "lightblue") +
27     geom_line(aes(y = x_true), colour = "black", alpha = 1,
28     linewidth = 0.5) +
29     geom_line(aes(y = mean), colour = "blue", linewidth = 0.5)
30   +
31   geom_line(aes(y = meanT), colour = "purple", linewidth =
32   0.5) +
33   ylab(TeX("$h_{\{t\}}$")) + xlab("Time")
34   print(p)
35 }
36 plot_filtering_estimates(df)
```

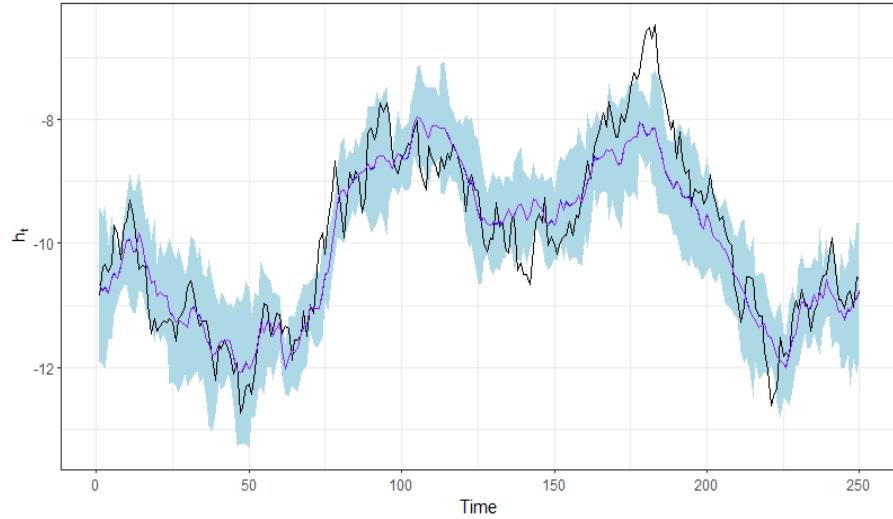


FIGURE 8.5
Stochastic volatility model: Sequential Monte Carlo (SMC).

8.4 Vector Autoregressive models

Another widely used methodological approach in time series analysis is the vector autoregressive (VAR) model, which extends AR(p) models to the multivariate case. Since the seminal work by Sims (1980) [202], these models have become a cornerstone of macroeconomic research to perform forecasts, and impulse-response (structural) analysis. This chapter provides an introduction to Bayesian inference in VAR models, with detailed discussions available in [125, 53, 224, 34].

The *reduced-form* VAR(p) model can be written as

$$\mathbf{Y}_t = \boldsymbol{\nu} + \sum_{j=1}^p \mathbf{A}_j \mathbf{Y}_{t-j} + \boldsymbol{\mu}_t, \quad (8.10)$$

where \mathbf{Y}_t is a M -dimensional vector having information of M time series variables, $\boldsymbol{\nu}$ is a M -dimensional vector of intercepts, \mathbf{A}_j are $M \times M$ matrices of coefficients, and $\boldsymbol{\mu}_t \stackrel{iid}{\sim} N_M(\mathbf{0}, \boldsymbol{\Sigma})$ are stochastic errors, $t = 1, 2, \dots, T$ and $j = 1, 2, \dots, p$. Other deterministic terms and exogenous variables can be added to the specification without main difficulty, we do not do this to keep simply the notation. In addition, we assume that the stability condition is satisfied such that the stochastic process is stationary (see [98, Chap. 2] for details), and we have available p presample values for each variable.

Following the matrix-form notation of the multivariate regression model (see sections 3.4 and 7.1), we can set $\mathbf{Y} = [\mathbf{Y}_1 \ \mathbf{Y}_2 \ \dots \ \mathbf{Y}_M]$, which is an $T \times M$ matrix, $\mathbf{x}_t = [1 \ \mathbf{Y}_{t-1}^\top \ \dots \ \mathbf{Y}_{t-p}^\top]$ is a $(1+Mp)$ -dimensional row vector, we define $K = 1 + Mp$ to facilitate notation, and set

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_T \end{bmatrix},$$

which is a $T \times K$ matrix, $\mathbf{B} = [\mathbf{v} \ \mathbf{A}_1 \ \mathbf{A}_2 \ \dots \ \mathbf{A}_P]^\top$ is a $K \times M$ matrix of parameters, and $\mathbf{U} = [\boldsymbol{\mu}_1 \ \boldsymbol{\mu}_2 \ \dots \ \boldsymbol{\mu}_M]$ is a $T \times M$ -dimensional matrix of stochastic random errors such that $\mathbf{U} \sim N_{T \times M}(\mathbf{0}_{T \times M}, \boldsymbol{\Sigma} \otimes \mathbf{I}_T)$. Thus, we can express the VAR(p) model in the form of a multivariate regression model,

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{U}.$$

We can assume conjugate priors to facilitate computation, that is, $\pi(\mathbf{B}, \boldsymbol{\Sigma}) = \pi(\mathbf{B}|\boldsymbol{\Sigma})\pi(\boldsymbol{\Sigma})$ where $\mathbf{B}|\boldsymbol{\Sigma} \sim N_{K \times M}(\mathbf{B}_0, \mathbf{V}_0, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma} \sim IW(\boldsymbol{\Psi}_0, \alpha_0)$. Thus, $\pi(\mathbf{B}, \boldsymbol{\Sigma}|\mathbf{Y}, \mathbf{X}) = \pi(\mathbf{B}|\boldsymbol{\Sigma}, \mathbf{Y}, \mathbf{X})\pi(\boldsymbol{\Sigma}|\mathbf{Y}, \mathbf{X})$ where $\mathbf{B}|\boldsymbol{\Sigma}, \mathbf{Y}, \mathbf{X} \sim N_{K \times M}(\mathbf{B}_n, \mathbf{V}_n, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma}|\mathbf{Y}, \mathbf{X} \sim IW(\boldsymbol{\Psi}_n, \alpha_n)$, $\mathbf{B}_n = (\mathbf{V}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}(\mathbf{V}_0^{-1}\mathbf{B}_0 + \mathbf{X}^\top \mathbf{X}\hat{\mathbf{B}})$, $\mathbf{V}_n = (\mathbf{V}_0^{-1} + \mathbf{X}^\top \mathbf{X})^{-1}$, $\boldsymbol{\Psi}_n = \boldsymbol{\Psi}_0 + \mathbf{S} + \mathbf{B}_0^\top \mathbf{V}_0^{-1} \mathbf{B}_0 + \hat{\mathbf{B}}^\top \mathbf{X}^\top \mathbf{X}\hat{\mathbf{B}} - \mathbf{B}_n^\top \mathbf{V}_n^{-1} \mathbf{B}_n$, $\mathbf{S} = (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})^\top (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})$, $\hat{\mathbf{B}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$, and $\alpha_n = T + \alpha_0$.

We also know from Section 3.4 that the marginal posterior distribution of \mathbf{B} is matrix variate t distribution, $T_{K \times M}(\mathbf{B}_n, \mathbf{V}_n, \boldsymbol{\Psi}_n)$ with $\alpha_n + 1 - M$ degrees of freedom, and the predictive density of \mathbf{Y}_{T+1} given \mathbf{Y} , $\pi(\mathbf{Y}_{T+1}|\mathbf{Y})$ is a matrix (multivariate) t distribution $T_{1,M}(\alpha_n - M + 1, \mathbf{x}_{T+1}\mathbf{B}_n, 1 + \mathbf{x}_{T+1}\mathbf{V}_n\mathbf{x}_{T+1}^\top, \boldsymbol{\Psi}_n)$.

Thus, we see that once we write a VAR(p) model in the right way, we can perform Bayesian inference as we did in the multivariate regression model. However, assuming conjugate priors has some limitations. First, VAR(p) models have many parameters, for instance, 4 lags and 6 variables, implies 150 $((1 + 6 \times 4) \times 6)$ location parameters plus 21 $(6 \times (6 + 1)/2)$ scale parameters of the covariance matrix, this implies loss of precision, particularly using macroeconomic data due to no having large sample sizes regularly. Thus, it is desirable to impose prior restrictions in the specification of the model. This cannot be done using conjugate priors. Second, natural conjugate priors do not allow for flexible extensions such as having different regressors in different equations. Third, the prior structure implies that the prior covariance of the coefficients in any two equations must be proportional each other, this is because the prior covariance form is $\boldsymbol{\Sigma} \otimes \mathbf{V}_0$. However, this does not make sense in some applications, for instance, imposing prior zero restrictions in some coefficients would imply that the prior variance of these coefficients should be near zero. However, this has not to be the case for all coefficients in the model.

Thus, we can tackle the first previous issue by thinking about the VAR(p)

specification as we did in the seemingly unrelated equations (SUR) model, where we have different regressors in different equations, and account for unobserved dependence. Thus, we can impose zero restrictions in the VAR(p) model improving its parsimony. Following the setting of Section 7.2, we have $\mathbf{Y}_m = \mathbf{Z}_m \boldsymbol{\beta}_m + \boldsymbol{\mu}_m$, where \mathbf{Y}_m is a T -dimensional vector corresponding to time series variable m -th, \mathbf{Z}_m is a matrix of dimension $T \times K_m$ of regressors, $\boldsymbol{\beta}_m$ is a K_m -dimensional vector of location parameters, and $\boldsymbol{\mu}_m$ is a T -dimensional vector of stochastic errors, $m = 1, 2, \dots, M$.

Stacking the M equations, we can write $\mathbf{y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\mu}$ where $\mathbf{y} = [\mathbf{Y}_1^\top \mathbf{Y}_2^\top \dots \mathbf{Y}_M^\top]^\top$ is a MT -dimensional vector, $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^\top \boldsymbol{\beta}_2^\top \dots \boldsymbol{\beta}_M^\top]^\top$ is a K dimensional vector, $K = \sum_{m=1}^M K_m$, \mathbf{Z} is an $MT \times K$ block diagonal matrix composed of \mathbf{Z}_m , that is,

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{Z}_M \end{bmatrix},$$

and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^\top \boldsymbol{\mu}_2^\top \dots \boldsymbol{\mu}_M^\top]^\top$ is a MT -dimensional vector of stochastic errors such that $\boldsymbol{\mu} \sim N(\mathbf{0}, \boldsymbol{\Sigma} \otimes \mathbf{I}_T)$.

We can use independent priors in this model to overcome the limitations of the conjugate prior, that is, $\pi(\boldsymbol{\beta}) \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\pi(\boldsymbol{\Sigma}^{-1}) \sim W(\alpha_0, \boldsymbol{\Psi}_0)$. Thus, we know from Section 7.2 that the posterior distributions are

$$\boldsymbol{\beta} | \boldsymbol{\Sigma}, \mathbf{y}, \mathbf{Z} \sim N(\boldsymbol{\beta}_n, \mathbf{B}_n),$$

$$\boldsymbol{\Sigma}^{-1} | \boldsymbol{\beta}, \mathbf{y}, \mathbf{Z} \sim W(\alpha_n, \boldsymbol{\Psi}_n),$$

where $\mathbf{B}_n = (\mathbf{Z}^\top (\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_T) \mathbf{Z} + \mathbf{B}_0^{-1})^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \mathbf{Z}^\top (\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_T) \mathbf{y})$, $\alpha_n = \alpha_0 + T$ and $\boldsymbol{\Psi}_n = (\boldsymbol{\Psi}_0^{-1} + \mathbf{U}^\top \mathbf{U})^{-1}$, where \mathbf{U} is an $T \times M$ matrix whose columns are $\mathbf{Y}_m - \mathbf{Z}_m \boldsymbol{\beta}_m$.⁵

Observe that we have standard conditional posteriors, thus, we can employ a Gibbs sampling algorithm to get the posterior draws. We can calculate the prediction $\mathbf{y}_{T+1} = [Y_{1T+1} \ Y_{2T+1} \ \dots \ Y_{MT+1}]^\top$ knowing that $\mathbf{y}_{T+1} \sim N(\mathbf{Z}_T \boldsymbol{\beta}, \boldsymbol{\Sigma})$, where

$$\mathbf{Z}_T = \begin{bmatrix} \mathbf{z}_{1T}^\top & 0 & \dots & 0 \\ 0 & \mathbf{z}_{2T}^\top & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{z}_{MT}^\top \end{bmatrix},$$

and using the posterior draws of $\boldsymbol{\beta}^{(s)}$ and $\boldsymbol{\Sigma}^{(s)}$, $s = 1, 2, \dots, S$. We can also perform inference of functions of the parameters that are of main interest when using VAR models.

⁵We can also use the alternative representation presented in Section 7.2.

Note that the independent priors allow more flexibility regarding prior information. For instance, we can set $\Psi_0 = \mathbf{S}^{-1}$, $\alpha_0 = T$, $\beta_0 = \mathbf{0}$ and \mathbf{B}_0 as a diagonal matrix, where the variance of the components associated with the coefficients in the m -th equation are such that the prior variance of the coefficients of the own-lags are a_1/l^2 , variances on lag 1 of variable $m \neq j$ are $a_2 s_m^2/(l^2 s_j^2)$, and variance of the intercepts are set $a_3 s_m^2$, $l = 1, 2, \dots, p$, where s_m is the estimated standard error of the residuals in an unrestricted univariate autoregression of variable m versus a constant and its p lags [139, 125]. Note that setting $a_1 > a_2$ implies that own lags are more important to be good predictors than lags of other variables, and dividing by l^2 implies that recent lags are more relevant than further past lags. The specific choice of a_1 , a_2 and a_3 ($a_k > 0$, $k = 1, 2, 3$) depends on each application, but it is easier to elicit these parameters rather than the $K(K + 1)/2$ different components of \mathbf{B}_0 .⁶ This setting is named the *Minnesota prior*, as is based on the seminal proposals of Bayesian VAR models by researchers at the University of Minnesota and the Federal Reserve Bank of Minneapolis [58, 139].⁷

An important non-linear function of parameters when performing VAR analysis is the *impulse response* function, which is, the response of one variable to an impulse in another variable in the model. The *impulse response* function can be deduced using the *MA* representation of the VAR model. In particular, we can write Equation 8.10 using the lag operator (see Section 8.2),

$$\mathbf{Y}_t = \mathbf{v} + (\mathbf{A}_1 L + \mathbf{A}_2 L^2 + \cdots + \mathbf{A}_p L^p) \mathbf{Y}_t + \boldsymbol{\mu}_t, \quad (8.11)$$

thus $\mathbf{A}(L)\mathbf{Y}_t = \mathbf{v} + \boldsymbol{\mu}_t$, where $\mathbf{A}(L) = \mathbf{I}_M - \mathbf{A}_1 L - \mathbf{A}_2 L^2 - \cdots - \mathbf{A}_p L^p$. Let $\Phi(L) := \sum_{s=0}^{\infty} \Phi_s L^s$ an operator such that $\Phi(L)\mathbf{A}(L) = \mathbf{I}_M$. Thus, we have that $\Phi(L)\mathbf{A}(L)\mathbf{Y}_t = (\sum_{s=0}^{\infty} \Phi_s L^s) \mathbf{v} + (\sum_{s=0}^{\infty} \Phi_s L^s) \boldsymbol{\mu}_t = \boldsymbol{\mu} + \sum_{s=0}^{\infty} \Phi_s \boldsymbol{\mu}_{t-s}$. Note that $L^s \mathbf{v} = \mathbf{v}$ because \mathbf{v} is constant, thus we set $\sum_{s=0}^{\infty} \Phi_s L^s \mathbf{v} = \sum_{s=0}^{\infty} \Phi_s \mathbf{v} = \Phi(1) \mathbf{v} = (\mathbf{I}_M - \mathbf{A}_1 - \mathbf{A}_2 - \cdots - \mathbf{A}_p)^{-1} \mathbf{v} := \boldsymbol{\mu}$, which is the mean of the process [98, Chap. 2]. Therefore, the MA representation of the VAR is

$$\mathbf{Y}_t = \boldsymbol{\mu} + \sum_{s=0}^{\infty} \Phi_s \boldsymbol{\mu}_{t-s}, \quad (8.12)$$

where $\Phi_0 = \mathbf{I}_M$, and we can get the coefficients in Φ_s by the recursion $\Phi_s = \sum_{l=1}^s \Phi_{s-l} \mathbf{A}_l$, $\mathbf{A}_l = \mathbf{0}$, $l > p$ and $s = 1, 2, \dots$ [98, Chap. 2]. This impulse response function is called *forecast error impulse response function*.

The MA coefficients contain the impulse responses of the system. In particular, $\phi_{mj,s}$, which is the mj -th element of the matrix Φ_s , represents the

⁶We use in our GUI the *bvar tools* package, where this package has a slightly different notation such $a_2 = a_1 \kappa_2$ and $a_3 = a_1 \kappa_3$. Thus, we should set a_1 , κ_2 and κ_3 .

⁷In the case that the variables are not stationary, which is more probable when using variables in levels, like gross domestic product, $\beta_0 = \mathbf{0}$, except in the elements associated with the first own lags of the dependent variables of each equation, where the prior mean equals 1. In addition, the original proposal of the Minnesota prior set $\Sigma = \mathbf{S}/T$, thus it did not take into account uncertainty regarding Σ .

response of the m -th variable to a unit shock of the variable j in the system, s periods ago, provided that the effect is not contaminated by other shocks in the system. The long-term effects (total multipliers) are given by $\Psi_\infty := \sum_{s=1}^{\infty} \Phi_s = (\mathbf{I}_M - \mathbf{A}_1 - \mathbf{A}_2 - \cdots - \mathbf{A}_p)^{-1}$.

An assumption in these *impulse response* functions is that a shock occurs in only one variable at a time. This can be questionable as different shocks may be correlated, consequently, occurring simultaneously. Thus, the *impulse response* analysis can be performed based on the alternative MA representation, $\mathbf{Y}_t = \boldsymbol{\mu} + \sum_{s=0}^{\infty} \Phi_s \mathbf{P} \mathbf{P}^{-1} \boldsymbol{\mu}_{t-s} = \boldsymbol{\mu} + \sum_{s=0}^{\infty} \Theta_s \mathbf{w}_{t-s}$, where $\Theta_s = \Phi_s \mathbf{P}$ and $\mathbf{w}_t = \mathbf{P}^{-1} \boldsymbol{\mu}_t$, \mathbf{P} is a lower triangular matrix such that $\Sigma = \mathbf{P} \mathbf{P}^\top$ (Cholesky factorization/decomposition). Note that the covariance matrix of \mathbf{w}_t is \mathbf{I}_M due to $\mathbb{E}[\mathbf{w}_t \mathbf{w}_t^\top] = \mathbb{E}[\mathbf{P}^{-1} \boldsymbol{\mu}_t \boldsymbol{\mu}_t^\top (\mathbf{P}^{-1})^\top] = \mathbf{P}^{-1} \Sigma (\mathbf{P}^{-1})^\top = \mathbf{P}^{-1} \mathbf{P} \mathbf{P}^\top (\mathbf{P}^{-1})^\top = \mathbf{I}_M$.

In this representation is sensible to assume that each shock occurs independently due to the covariance matrix of \mathbf{w}_t being an identity. In addition, a unit shock is a shock of size one standard deviation due the result of the covariance matrix. This is named the *ortogonalized impulse response*, where $\theta_{mj,s}$, which is the mj -th element of the matrix Θ_s , represents the response of the m -th variable to a standard deviation shock of the variable j in the system, s periods ago. The critical point with the orthogonalized impulse responses is that the order of the variables in the VAR is really important because implicitly establishes a recursive model, that is, the m -th equation in the system may contain $Y_{1t}, Y_{2t}, \dots, Y_{m-1t}$, but not $Y_{mt}, Y_{m+1t}, \dots, Y_{Mt}$ on the hand-right side of its equation. Thus, Y_{mt} cannot have an instantaneous impact on Y_{jt} for $j < m$ [98, Chap. 2].

Beyond the fascinating macroeconomic implications embedded in the specification of VAR models, the key point for this section is that we can infer impulse response functions using the posterior draws.

Example: US fiscal system

Let's use the dataset provided by [225] of the US fiscal system, where ttr is the quarterly total tax revenue, gs is the quarterly total government spending, and gdp is the quarterly gross domestic product, all expressed in log, real, per person terms, and the period is 1948q1 to 2024q2. This dataset is the *18USAfiscal.csv* file. [152] analyze the US fiscal policy shocks using these variables.

Let's estimate a VAR model where $\mathbf{y}_t = [\Delta(ttr_t) \Delta(gs_t) \Delta(gdp_t)]^\top$, that is, we work with the log differences (variation rates), and we set $p = 1$. We use the package *bvarools* to estimate the *forecast error* and *orthogonalized* impulse response functions. We use vague independent priors setting $\beta_0 = \mathbf{0}$, $\mathbf{B}_0 = 100\mathbf{I}$, $\mathbf{V}_0 = 5^{-1}\mathbf{I}$ and $\alpha_0 = 3$, and the Minnesota prior setting $a_1 = 2$, $\kappa_2 = 0.5$ and $\kappa_3 = 5$ (default values).⁸

⁸The *bvarools* package uses the inverse Wishart distribution as prior for Σ , where the hyperparameters are the degrees of freedom of the error term, and the prior error variance of endogenous variables.

The following code shows how to do this, take into account that we use the first 301 observations to estimate the model, and keep the last 4 observations to check the forecasting performance. Figures 8.6 and 8.7 show the impulse response functions of g_s with respect to g_s , the *forecast error impulse response* using vague independent priors, and the *orthogonalized impulse response* using the Minnesota prior, respectively. We see that the effect of the Minnesota prior is to decrease uncertainty. In addition, the forecasting exercise results indicate that these assumptions have same effects in this example. In particular, Figure 8.8 shows that the mean forecasts using the vague prior (green line) and the Minnesota prior (red line) are indistinguishable from the true observations (black line). However, the Minnesota prior enhances forecast precision, as its 95% predictive interval (blue shaded area) is narrower and fully contained within the 95% predictive interval obtained using vague priors (light blue shaded area). This improvement is attributable to the shrinkage properties of the Minnesota prior.

R code. VAR model: US fiscal shocks

```

1 rm(list = ls()); set.seed(010101)
2 DataUSfilcal <- read.csv("https://raw.githubusercontent.com/
   besmarter/BSTApp/refs/heads/master/DataApp/18USAfiscal.
   csv", sep = ",", header = TRUE, quote = "")
3 attach(DataUSfilcal) # upload data
4 Y <- cbind(diff(as.matrix(DataUSfilcal[,-c(1:2)])))
5 T <- dim(Y)[1]-1; K <- dim(Y)[2]
6 Ynew <- Y[-c((T-2):(T+1)), ] # Use 4 last observations to
   check forecast
7 y1 <- Ynew[-1, 1]; y2 <- Ynew[-1, 2]; y3 <- Ynew[-1, 3]
8 X1 <- cbind(1, lag(Ynew)); X1 <- X1[-1,]
9 X2 <- cbind(1, lag(Ynew)); X2 <- X2[-1,]
10 X3 <- cbind(1, lag(Ynew)); X3 <- X3[-1,]
11 M <- dim(Y)[2]; K1 <- dim(X1)[2]; K2 <- dim(X2)[2]; K3 <-
   dim(X3)[2]
12 K <- K1 + K2 + K3
13 # Hyperparameters
14 b0 <- 0; c0 <- 100; V0 <- 5^(-1); a0 <- M
15 #Posterior draws
16 MCMC <- 10000; burnin <- 1000; H <- 10; YnewPack <- ts(Ynew)
17 model <- bvartools::gen_var(YnewPack, p = 1, deterministic =
   "const", iterations = MCMC, burnin = burnin) # Create
   model
18 model <- bvartools::add_priors(model, coef = list(v_i =
   ^-1, v_i_det = c0^-1, const = b0), sigma = list(df = a0,
   scale = V0/a0), coint_var = FALSE) # Add priors
19 object <- bvartools::draw_posterior(model) # Posterior draws
20 ir <- bvartools::irf.bvar(object, impulse = "gs", response =
   "gs", n.ahead = H, type = "feir", cumulative = FALSE) #
   Calculate IR
21 # Plot IR
22 plot_IR <- function(df) {
23   p <- ggplot(data = df, aes(x = t)) + geom_ribbon(aes(ymin =
   lower, ymax = upper), alpha = 1, fill = "lightblue") +
   geom_line(aes(y = mean), colour = "blue", linewidth =
   0.5) + ylab("Impulse response") + xlab("Time") + xlim(0,
   H)
24   print(p)
25 }
26 dfNew <- tibble(t = 0:H, mean = as.numeric(ir[,2]), lower =
   as.numeric(ir[,1]), upper = as.numeric(ir[,3]))
27 FigNew <- plot_IR(dfNew)
28 # Using Minnesota prior
29 modelMin <- bvartools::gen_var(YnewPack, p = 1,
   deterministic = "const", iterations = MCMC, burnin =
   burnin)
30 modelMin <- bvartools::add_priors(modelMin, minnesota = list
   (kappa0 = 2, kappa1 = 0.5, kappa3 = 5), coint_var =
   FALSE) # Minnesota prior
31 objectMin <- bvartools::draw_posterior(modelMin) # Posterior
   draws
32 irMin <- bvartools::irf.bvar(objectMin, impulse = "gs",
   response = "gs", n.ahead = H, type = "feir", cumulative =
   FALSE) # Calculate IR
33 dfNewMin <- tibble(t = 0:H, mean = as.numeric(irMin[,2]),
   lower = as.numeric(irMin[,1]), upper = as.numeric(irMin
   [,3]))
34 FigNewMin <- plot_IR(dfNewMin)

```

R code. VAR model: US fiscal shocks

```

1  ### Forecasting
2 bvar_pred <- predict(object, n.ahead = 4, new_d = rep(1, 4))
3 bvar_predOR <- predict(objectMin, n.ahead = 4, new_d = rep
(1, 4))
4 dfFore <- tibble(t = c((T-2):(T+1)), mean = as.numeric(bvar_
pred[["fcst"]][["gs"]][,2]), lower = as.numeric(bvar_
pred[["fcst"]][["gs"]][,1]), upper = as.numeric(bvar_
pred[["fcst"]][["gs"]][,3]), mean1 = as.numeric(bvar_
predOR[["fcst"]][["gs"]][,2]), lower1 = as.numeric(bvar_
predOR[["fcst"]][["gs"]][,1]), upper1 = as.numeric(bvar_
predOR[["fcst"]][["gs"]][,3]), true = as.numeric(Y[c((T-
2):(T+1)),2]))
5 plot_FORE <- function(df) {
6   p <- ggplot(data = dfFore, aes(x = t)) + geom_ribbon(aes(
ymin = lower, ymax = upper), alpha = 1, fill =
"lightblue") + geom_ribbon(aes(ymin = lower1, ymax =
upper1), alpha = 1, fill = "blue") + geom_line(aes(y =
mean), colour = "green", linewidth = 0.5) + geom_line(
aes(y = mean1), colour = "red", linewidth = 0.5) +
geom_line(aes(y = true), colour = "black", linewidth = 0.5) +
ylab("Forecast") + xlab("Time") + xlim(c((T-2),(T+1)))
7   print(p)
8 }
9 FigFore <- plot_FORE(dfFore)

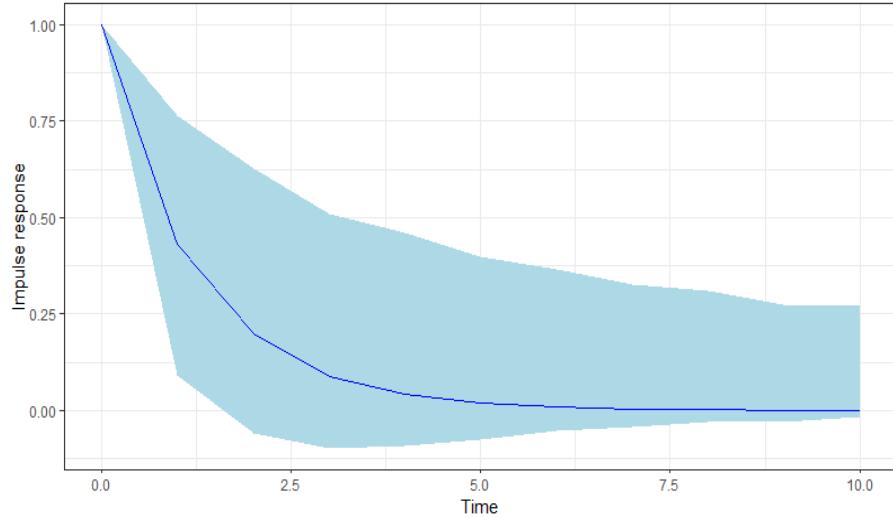
```

Algorithm A23 shows how to do perform inference in VAR models using our GUI. See also Chapter 5 for details regarding the dataset structure.

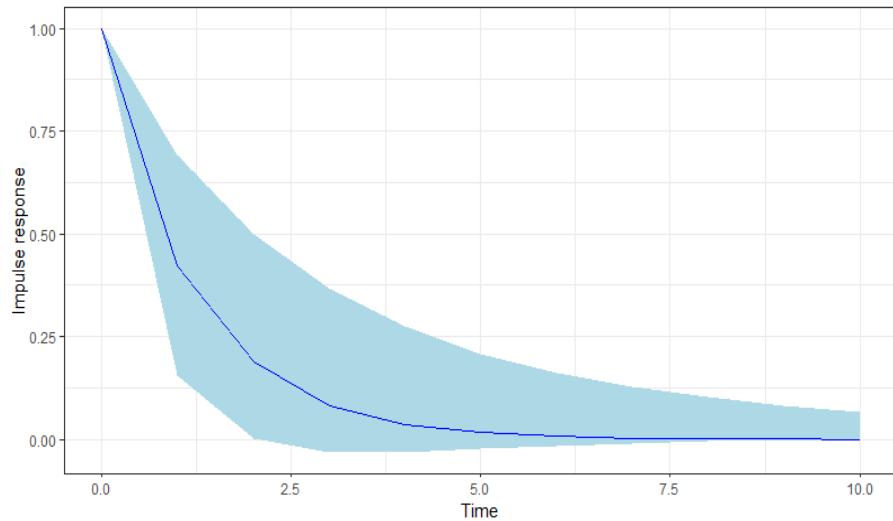
There are other good packages in **R** to perform Bayesian inference in VAR models. For instance, *bayesianVARs* package implements inference of reduced-form VARs with stochastic volatility [141], *BVAR* package performs inference using hierarchical priors [158], *bvarsrv* implements time-varying parameters models [128], *bsvars* performs estimation of structural VAR models [225], and *bsvarSIGNs* to estimating structural VAR models with sign restrictions [226].

8.5 Summary

We present a brief review of Bayesian inference in time series models. In particular, we introduce the *state-space* representation and demonstrate how to perform inferential analysis for these models, focusing on the dynamic linear model and the stochastic volatility model. Additionally, we show that

**FIGURE 8.6**

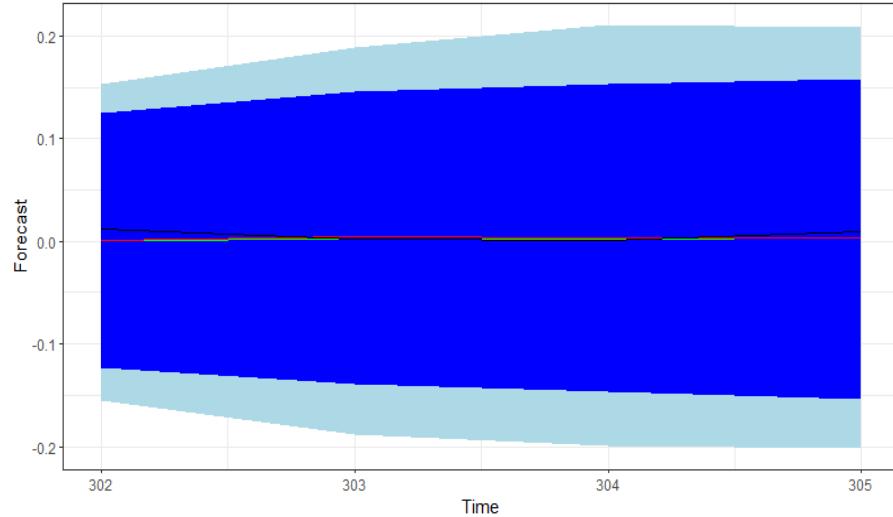
Forecasting error impulse response: gs with respect to gs .

**FIGURE 8.7**

Ortogonalized impulse response: gs with respect to gs .

$ARMA(p, q)$ processes can be expressed in *state-space* form and provide methods for estimating such models.

We include code for implementing computational inference algorithms such

**FIGURE 8.8**Forecast performance: *gs*.**Algorithm A23** Vector Autoregressive models

- 1: Select *Time series Model* on the top panel
- 2: Select *VAR models* using the left radio button
- 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
- 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
- 5: Set the number of lags (*p*)
- 6: Set the hyperparameters for the Minnesota prior: a_1 , κ_2 and κ_2 . This step is not necessary as by default our GUI uses default values in *bvar tools* package
- 7: Select the type of *impulse response functions*: forecast error or orthogonalized, and ordinary or cumulative.
- 8: Set the time horizon for the impulse response functions and the forecasts
- 9: Click the *Go!* button
- 10: Analyze results
- 11: Download impulse responses and forecasts using the *Download Impulse Responses*, and *Forecast* buttons

as sequential Monte Carlo (SMC), Hamiltonian Monte Carlo (HMC), and various Markov chain Monte Carlo (MCMC) methods. Finally, we introduce

VAR(p) models, detailing how to perform impulse-response analysis and forecasting within this framework.

Time series analysis is a very active research area with amazing methodological developments and applications. Interested readers can see excellent material in chapters 7 and 9 in [86], chapters 17 to 20 in [34], and references in there.

8.6 Exercises

1. Simulate the *dynamic linear model* assuming $X_t \sim N(1, 0.1\sigma^2)$, $w_t \sim N(0, 0.5\sigma^2)$, $\mu_t \sim N(0, \sigma^2)$, $\beta_0 = 1$, $B_0 = 0.5\sigma^2$, $\sigma^2 = 0.25$, and $G_t = 1$, $t = 1, \dots, 100$. Then, perform the filtering recursion fixing $\Sigma = 25 \times 0.25$, $\Omega_1 = 0.5\Sigma$ (high signal-to-noise ratio) and $\Omega_2 = 0.1\Sigma$ (low signal-to-noise ratio). Plot and compare the results.
2. Simulate the *dynamic linear model* $y_t = \beta_t x_t + \mu_t$, $\beta_t = \beta_{t-1} + w_t$, where $x_t \sim N(1, 0.1\sigma^2)$, $w_t \sim N(0, 0.5\sigma^2)$, $\mu_t \sim N(0, \sigma^2)$, $\beta_0 = 0$, $B_0 = 0.5\sigma^2$, and $\sigma^2 = 1$, $t = 1, \dots, 100$. Perform the filtering and smoothing recursions from scratch.
3. Simulate the process $y_t = \alpha z_t + \beta_t x_t + \mathbf{h}^\top \boldsymbol{\epsilon}_t$, $\beta_t = \beta_{t-1} + \mathbf{H}^\top \boldsymbol{\epsilon}_t$, where $\mathbf{h}^\top = [1 \ 0]$, $\mathbf{H}^\top = [0 \ 1/\tau]$, $\mathbf{v}_t \sim N(\mathbf{0}_2, \sigma^2 \mathbf{I}_2)$, $x_t \sim N(1, 2\sigma^2)$, $z_t \sim N(0, 2\sigma^2)$, $\alpha = 2$, $\tau^2 = 5$ and $\sigma^2 = 0.1$, $t = 1, \dots, 200$. Assume $\pi(\beta_0, \alpha, \sigma^2, \tau) = \pi(\beta_0)\pi(\alpha)\pi(\sigma^2)\pi(\tau^2)$ where $\sigma^2 \sim IG(\alpha_0/2, \delta_0/2)$, $\tau^2 \sim G(v_0/2, v_0/2)$, $\alpha \sim N(a_0, A_0)$ and $\beta_0 \sim N(b_0, B_0)$ such that $\alpha_0 = \delta_0 = 1$, $v_0 = 5$, $a_0 = 0$, $A_0 = 1$, $\beta_0 = 0$, $B_0 = \sigma^2/\tau^2$. Program the MCMC algorithm including the *simulation smoother*.
4. Show that the posterior distribution of $\phi|\beta, \sigma^2, \mathbf{y}, \mathbf{X}$ in the model $Y_t = \mathbf{x}_t^\top \boldsymbol{\beta} + \mu_t$ where $\phi(L)\mu_t = \boldsymbol{\epsilon}_t$ and $\boldsymbol{\epsilon}_t \stackrel{iid}{\sim} N(0, \sigma^2)$ is $N(\phi_n, \Phi_n) \mathbb{1}[\phi \in S_\phi]$, where $\Phi_n = (\Phi_0^{-1} + \sigma^{-2} \mathbf{U}^\top \mathbf{U})$, $\phi_n = \Phi_n(\Phi_0^{-1} \phi_0 + \sigma^{-2} \mathbf{U}^\top \boldsymbol{\mu})$, and S_ϕ is the stationary region of ϕ .
5. Show that in the AR(2) stationary process, $Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \epsilon_t$, where $\epsilon_t \sim N(0, \sigma^2)$, $\mathbb{E}[Y_t] = \frac{\mu}{1-\phi_1-\phi_2}$, and $Var[Y_t] = \frac{\sigma^2(1-\phi_2)}{1-\phi_2-\phi_1^2-\phi_1^2\phi_2-\phi_2^2+\phi_2^3}$.
6. Program a Hamiltonian Monte Carlo taking into account the stationary restrictions on ϕ_1 and ϕ_2 , and ϵ_0 such that the acceptance rate is near 65%.
7. **Stochastic volatility model**
 - Program a sequential importance sampling (SIS) from scratch in the vanilla stochastic volatility model setting $\mu = -10$,

$\phi = 0.95$, $\sigma = 0.3$ and $T = 250$. Check what happen with its performance.

- Modify the sequential Monte Carlo (SMC) to perform multinomial resampling when the effective sample size is lower than 50% the number of particles.
8. Estimate the vanilla stochastic volatility model using the dataset *17ExcRate.csv* provided by [178] of the exchange rate log daily returns from USD/EUR, USD/GBP and GBP/EUR one year before and after the WHO declared the COVID-19 pandemic on 11 March 2020.
 9. Simulate the VAR(1) process

$$\begin{bmatrix} y_{1t} \\ y_{2t} \\ y_{3t} \end{bmatrix} = \begin{bmatrix} 2.8 \\ 2.2 \\ 1.3 \end{bmatrix} + \begin{bmatrix} 0.5 & 0 & 0 \\ 0.1 & 0.1 & 0.3 \\ 0 & 0.2 & 0.3 \end{bmatrix} \begin{bmatrix} y_{1t-1} \\ y_{2t-1} \\ y_{3t-1} \end{bmatrix} + \begin{bmatrix} \mu_{1t} \\ \mu_{2t} \\ \mu_{3t} \end{bmatrix},$$

where $\Sigma = \begin{bmatrix} 2.25 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 0.74 \end{bmatrix}$.

- Use vague independent priors setting $\beta_0 = \mathbf{0}$, $B_0 = 100I$, $V_0 = 5I$ and $\alpha_0 = 3$, and estimate a VAR(1) model using the *rsurGibbs* function from the package *bayesm*. Then, program from scratch algorithms to perform inference of the *forecast error* and *orthogonalized* impulse response functions, and compare with the population impulse response functions, that is, using the population parameters.
- Using the previous setting perform inference of the *forecast error* and the *orthogonalized* impulse responses using the package *bvarTools*.



9

Longitudinal/Panel data models

We describe how to perform inference in panel/longitudinal models using a Bayesian framework. In this context, multiple cross-sectional units are observed repeatedly over time, a structure referred to as panel data by econometricians and longitudinal data by statisticians. Specifically, we present models for continuous (normal), binary (logit), and count (Poisson) responses. Applications and exercises illustrate the potential of these models.

In panel/longitudinal data sets, we have y_{it} where $i = 1, 2, \dots, N$ and $t = 1, 2, \dots, T_i$. If $T_i = T$ for all i , the dataset is *balanced*; otherwise, it is *unbalanced*. Longitudinal data typically involves by far more cross-sectional units than time periods, this is called typically a *short panel*. It assumes that cross-sectional units are independent, though serial correlation exists within each unit over time, and unobserved heterogeneity for each unit must be accounted for. We can treat this unobserved heterogeneity as random variables, assuming it is either independent or dependent on control variables. Econometricians refer to these cases as *random effects* and *fixed effects*, respectively. The Bayesian literature takes a different approach, modeling the panel structure hierarchically, where the unobserved heterogeneity may or may not depend on other controls.¹

Remember that the easiest way to run our GUI is typing

R code. How to display our graphical user interface

```
1 shiny::runGitHub("besmarter/BSTApp", launch.browser = T)
```

in the **R** package console or any **R** code editor, and once our GUI is deployed, select *Hierarchical Longitudinal Models*. However, users should see Chapter 5 for details.

¹See [183] for a nice comparison of Frequentist and Bayesian treatments of panel data models.

9.1 Normal model

The panel/longitudinal normal model establishes $\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{W}_i\mathbf{b}_i + \boldsymbol{\mu}_i$ where \mathbf{y}_i are T_i -dimensional vectors corresponding to units $i = 1, 2, \dots, N$, \mathbf{X}_i and \mathbf{W}_i are $T_i \times K_1$ and $T_i \times K_2$ matrices, respectively. In the statistical literature, $\boldsymbol{\beta}$ is a K_1 -dimensional vector of *fixed effects*, and \mathbf{b}_i is a K_2 -dimensional vector of unit-specific *random effects* that allow unit-specific means, and enable to capture marginal dependence among the observations on the cross-sectional units. We assume normal stochastic errors, $\boldsymbol{\mu}_i \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_{T_i})$, which means that the likelihood function is

$$\begin{aligned} p(\boldsymbol{\beta}, \mathbf{b}, \sigma^2 | \mathbf{y}, \mathbf{X}, \mathbf{W}) &\propto \prod_{i=1}^N |\sigma^2 \mathbf{I}_{T_i}|^{-1/2} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta} - \mathbf{W}_i\mathbf{b}_i)^\top (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta} - \mathbf{W}_i\mathbf{b}_i) \right\} \\ &= (\sigma^2)^{-\frac{\sum_{i=1}^N T_i}{2}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta} - \mathbf{W}_i\mathbf{b}_i)^\top (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta} - \mathbf{W}_i\mathbf{b}_i) \right\}, \end{aligned}$$

where $\mathbf{b} = [\mathbf{b}_1^\top, \mathbf{b}_2^\top, \dots, \mathbf{b}_N^\top]^\top$.

Panel data modeling in the Bayesian approach assumes a hierarchical structure in the *random effects*. Following [38], there is a first stage where $\mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$, \mathbf{D} allows serial correlation within each cross-sectional unit i , and then, there is a second stage where $\mathbf{D} \sim IW(d_0, d_0 \mathbf{D}_0)$. Thus, we can see that there is an additional layer of priors as there is a prior on the hyperparameter \mathbf{D} .

In addition, we have standard conjugate prior distributions for $\boldsymbol{\beta}$ and σ^2 , $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{B}_0)$ and $\sigma^2 \sim IG(\alpha_0, \delta_0)$.

[38] propose a blocking algorithm to perform inference in longitudinal hierarchical models by considering the distribution of \mathbf{y}_i marginalized over the random effects. Given that $\mathbf{y}_i | \boldsymbol{\beta}, \mathbf{b}_i, \sigma^2, \mathbf{X}_i, \mathbf{W}_i \sim N(\mathbf{X}_i\boldsymbol{\beta} + \mathbf{W}_i\mathbf{b}_i, \sigma^2 \mathbf{I}_{T_i})$, we can see that $\mathbf{y}_i | \boldsymbol{\beta}, \mathbf{D}, \sigma^2, \mathbf{X}_i, \mathbf{W}_i \sim N(\mathbf{X}_i\boldsymbol{\beta}, \mathbf{V}_i)$, where $\mathbf{V}_i = \sigma^2 \mathbf{I}_{T_i} + \mathbf{W}_i \mathbf{D} \mathbf{W}_i^\top$ given that $\mathbb{E}[\mathbf{b}_i] = \mathbf{0}$ and $Var[\mathbf{b}_i] = \mathbf{D}$. If we have just random intercepts, then $\mathbf{W}_i = \mathbf{i}_{T_i}$, where \mathbf{i}_{T_i} is a T_i -dimensional vector of ones. Thus, $\mathbf{V}_i = \sigma^2 \mathbf{I}_{T_i} + \sigma_b^2 \mathbf{i}_{T_i} \mathbf{i}_{T_i}^\top$, the variance is $\sigma^2 + \sigma_b^2$ and the covariance is σ_b^2 within each cross-sectional unit through time.

We can deduce the posterior distribution of $\boldsymbol{\beta}$ given σ^2 and \mathbf{D} ,

$$\begin{aligned} \pi(\boldsymbol{\beta} | \sigma^2, \mathbf{D}, \mathbf{y}, \mathbf{X}, \mathbf{W}) &\propto \exp \left\{ -\frac{1}{2} \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^\top \mathbf{V}_i^{-1} (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}) \right\} \\ &\quad \times \exp \left\{ -\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{B}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right\}. \end{aligned}$$

This implies that (see Exercise 1)

$$\boldsymbol{\beta} | \sigma^2, \mathbf{D}, \mathbf{y}, \mathbf{X}, \mathbf{W} \sim N(\boldsymbol{\beta}_n, \mathbf{B}_n),$$

where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \sum_{i=1}^N \mathbf{X}_i^\top \mathbf{V}_i^{-1} \mathbf{X}_i)^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n(\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \sum_{i=1}^N \mathbf{X}_i^\top \mathbf{V}_i^{-1} \mathbf{y}_i)$.

We can use the likelihood $p(\boldsymbol{\beta}, \mathbf{b}_i, \sigma^2 | \mathbf{y}, \mathbf{X}, \mathbf{W})$ to get the posterior distributions of \mathbf{b}_i , σ^2 and \mathbf{D} . In particular,

$$\begin{aligned} \pi(\mathbf{b}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{D}, \mathbf{y}, \mathbf{X}, \mathbf{W}) &\propto \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i)^\top (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i) \right\} \\ &\quad \times \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \mathbf{b}_i^\top \mathbf{D}^{-1} \mathbf{b}_i \right\} \\ &\propto \exp \left\{ -\frac{1}{2} \sum_{i=1}^N (-2\mathbf{b}_i^\top (\sigma^{-2} \mathbf{W}_i^\top (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta})) + \mathbf{b}_i^\top (\sigma^{-2} \mathbf{W}_i^\top \mathbf{W}_i + \mathbf{D}^{-1}) \mathbf{b}_i) \right\} \\ &\propto \exp \left\{ -\frac{1}{2} (-2\mathbf{b}_i^\top \mathbf{B}_{ni}^{-1} \mathbf{B}_{ni} (\sigma^{-2} \mathbf{W}_i^\top (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta})) + \mathbf{b}_i^\top \mathbf{B}_{ni}^{-1} \mathbf{b}_i) \right\} \\ &= \exp \left\{ -\frac{1}{2} (-2\mathbf{b}_i^\top \mathbf{B}_{ni}^{-1} \mathbf{b}_{ni} + \mathbf{b}_i^\top \mathbf{B}_{ni}^{-1} \mathbf{b}_i) \right\}, \end{aligned}$$

where $\mathbf{B}_{ni} = (\sigma^{-2} \mathbf{W}_i^\top \mathbf{W}_i + \mathbf{D}^{-1})^{-1}$ and $\mathbf{b}_{ni} = \mathbf{B}_{ni} (\sigma^{-2} \mathbf{W}_i^\top (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta}))$.

We can complete the square in this expression by adding and subtracting $\mathbf{b}_{ni}^\top \mathbf{B}_{ni}^{-1} \mathbf{b}_{ni}$. Thus,

$$\begin{aligned} \pi(\mathbf{b}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{D}, \mathbf{y}, \mathbf{X}, \mathbf{W}) &\propto \exp \left\{ -\frac{1}{2} (-2\mathbf{b}_i^\top \mathbf{B}_{ni}^{-1} \mathbf{b}_{ni} + \mathbf{b}_i^\top \mathbf{B}_{ni}^{-1} \mathbf{b}_i + \mathbf{b}_{ni}^\top \mathbf{B}_{ni}^{-1} \mathbf{b}_{ni} - \mathbf{b}_{ni}^\top \mathbf{B}_{ni}^{-1} \mathbf{b}_{ni}) \right\} \\ &\propto \exp \{ (\mathbf{b}_i - \mathbf{b}_{ni})^\top \mathbf{B}_{ni}^{-1} (\mathbf{b}_i - \mathbf{b}_{ni}) \}. \end{aligned}$$

This is the kernel of a multivariate normal distribution with mean \mathbf{b}_{ni} and variance \mathbf{B}_{ni} . Thus,

$$\mathbf{b}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{D}, \mathbf{y}, \mathbf{X}, \mathbf{W} \sim N(\mathbf{b}_{ni}, \mathbf{B}_{ni}),$$

Let's see the posterior distribution of σ^2 ,

$$\begin{aligned} \pi(\sigma^2 | \boldsymbol{\beta}, \mathbf{b}, \mathbf{y}, \mathbf{X}, \mathbf{W}) &\propto (\sigma^2)^{-\frac{\sum_{i=1}^N T_i}{2}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i)^\top (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i) \right\} \\ &\quad \times (\sigma^2)^{-\alpha_0 - 1} \exp \left\{ -\frac{\delta_0}{\sigma^2} \right\} \\ &= (\sigma^2)^{-\frac{\sum_{i=1}^N T_i}{2} - \alpha_0 - 1} \\ &\quad \times \exp \left\{ -\frac{1}{\sigma^2} \left(\delta_0 + \sum_{i=1}^N \frac{(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i)^\top (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i)}{2} \right) \right\}. \end{aligned}$$

Thus,

$$\sigma^2 | \boldsymbol{\beta}, \mathbf{b}, \mathbf{y}, \mathbf{X}, \mathbf{W} \sim IG(\alpha_n, \delta_n),$$

where $\alpha_n = \alpha_0 + \frac{1}{2} \sum_{i=1}^N T_i$ and $\delta_n = \delta_0 + \frac{1}{2} \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i)^\top (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i)$.

The posterior distribution of \mathbf{D} is the following,

$$\begin{aligned}\pi(\mathbf{D}|\mathbf{b}) &\propto |\mathbf{D}|^{-N/2} \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \mathbf{b}_i^\top \mathbf{D}^{-1} \mathbf{b}_i \right\} \\ &\quad \times |\mathbf{D}|^{-(d_0+K_2+1)/2} \exp \left\{ -\frac{1}{2} \text{tr}(d_0 \mathbf{D}_0 \mathbf{D}^{-1}) \right\} \\ &= |\mathbf{D}|^{-(d_0+N+K_2+1)/2} \exp \left\{ -\frac{1}{2} \text{tr} \left(\left(d_0 \mathbf{D}_0 + \sum_{i=1}^N \mathbf{b}_i \mathbf{b}_i^\top \right) \mathbf{D}^{-1} \right) \right\}.\end{aligned}$$

This is the kernel of an inverse Wishart distribution with degrees of freedom $d_n = d_0 + N$ and scale matrix $\mathbf{D}_n = d_0 \mathbf{D}_0 + \sum_{i=1}^N \mathbf{b}_i \mathbf{b}_i^\top$. Thus,

$$\mathbf{D}|\mathbf{b} \sim IW(d_n, \mathbf{D}_n).$$

Observe that the posterior distribution of \mathbf{D} depends just on \mathbf{b} .

All the posterior conditional distributions belong to standard families, this implies that we can use a Gibbs sampling algorithm to perform inference in these hierarchical normal models.

Example: The relation between productivity and public investment

We used the dataset named *8PublicCap.csv* used by [174] to analyze the relation between public investment and gross state product in the setting of a spatial panel dataset consisting of 48 US states from 1970 to 1986. In particular, we perform inference based on the following equation

$$\log(gsp_{it}) = b_i + \beta_1 + \beta_2 \log(pcap_{it}) + \beta_3 \log(pc_{it}) + \beta_4 \log(emp_{it}) + \beta_5 \text{unemp}_{it} + \mu_{it},$$

where gsp is the gross state product, pcap is public capital, and pc is private capital all in USD, emp is employment (people), and unemp is the unemployment rate in percentage.

Algorithm A24 shows how to perform inference in hierarchical longitudinal normal models in our GUI. See also Chapter 5 for details regarding the dataset structure.

We ask in Exercise 2 to run this application in our GUI using 10000 MCMC iterations plus a burn-in equal to 5000 iterations, and a thinning parameter equal to 1. We also used the default values for the hyperparameters of the prior distributions, that is, $\boldsymbol{\beta}_0 = \mathbf{0}_5$, $\mathbf{B}_0 = \mathbf{I}_5$, $\alpha_0 = \delta_0 = 0.001$, $d_0 = 5$ and $\mathbf{D}_0 = \mathbf{I}_1$. It seems that all posterior draws come from stationary distributions, as suggested by the diagnostics and posterior plots (see Exercise 2).

The following code uses the command *MCMChregress* from the package *MCMCpack* to run this application. This command is also used by our GUI to perform inference in hierarchical longitudinal normal models.

Algorithm A24 Hierarchical longitudinal normal models

- 1: Select *Hierarchical Longitudinal Model* on the top panel
 - 2: Select *Normal* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Write down the formula of the *fixed effects* equation in the **Main Equation: Fixed Effects** box. This formula must be written using the syntax of the *formula* command of **R** software. This equation includes intercept by default, do not include it in the equation
 - 6: Write down the formula of the *random effects* equation in the **Main Equation: Random Effects** box without writing the dependent variable, that is, starting the equation with the *tilde* symbol. This formula must be written using the syntax of the *formula* command of **R** software. This equation includes intercept by default, do not include it in the equation. If there are just random intercepts do not write anything in this box
 - 7: Write down the name of the grouping variable, that is, the variable that indicates the cross-sectional units
 - 8: Set the hyperparameters of the *fixed effects*: mean vector, covariance matrix, shape and scale parameters. This step is not necessary as by default our GUI uses non-informative priors
 - 9: Set the hyperparameters of the *random effects*: degrees of freedom and scale matrix of the inverse Wishart distribution. This step is not necessary as by default our GUI uses non-informative priors
 - 10: Click the *Go!* button
 - 11: Analyze results
 - 12: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

We can see that the 95% symmetric credible intervals for public capital, private capital, employment, and unemployment are (-2.54e-02, -2.06e-02), (2.92e-01, 2.96e-01), (7.62e-01, 7.67e-01) and (-5.47e-03, -5.31e-03), respectively. The posterior mean elasticity estimate of public capital to gsp is -0.023, that is, an increase by 1% in public capital means a 0.023% decrease in gross state product. The posterior mean estimates of private capital and employment elasticities are 0.294 and 0.765, respectively. In addition, 1 percentage point increase in the unemployment rate means a decrease of 0.54% in gsp. It seems that all these variables are statistically relevant. In addition, the posterior mean estimates of the variance associated with the unobserved heterogeneity and stochastic errors are 1.06e-01 and 1.45e-03. We obtained the posterior chain of the proportion of the variance associated with the unob-

served heterogeneity. The 95% symmetric credible interval is (0.98, 0.99) for this proportion, that is, unobserved heterogeneity is very important to explain the total variability.

R code. The relationship between productivity and public investment, MCMChregress command

```

1 rm(list = ls())
2 set.seed(12345)
3 DataGSP <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/8PublicCap.
  csv", sep = ",", header = TRUE, quote = "")
4 attach(DataGSP)
5 K1 <- 5; K2 <- 1
6 b0 <- rep(0, K1); B0 <- diag(K1)
7 r0 <- 5; R0 <- diag(K2)
8 a0 <- 0.001; d0 <- 0.001
9 Resultshreg <- MCMCpack::MCMChregress(fixed = log(gsp)^log(
  pcap)+log(pc)+log(emp)+unemp, random = ~1, group = "id",
  data = DataGSP, burnin = 5000, mcmc = 10000, thin = 1,
  r = r0, R = R0, nu = a0, delta = d0)
10 Betas <- Resultshreg[["mcmc"]][,1:K1]
11 Sigma2RanEff <- Resultshreg[["mcmc"]][,54]
12 Sigma2 <- Resultshreg[["mcmc"]][,55]
13 summary(Betas)
14 Quantiles for each variable:
15          2.5%    25%    50%    75%   97.5%
16 beta.(Intercept) 2.3145 2.3246 2.3301 2.335 2.3455
17 beta.log(pcap) -0.0254 -0.0239 -0.0231 -0.022 -0.0206
18 beta.log(pc)    0.2917 0.2930 0.2937 0.294 0.2957
19 beta.log(emp)   0.7619 0.7637 0.7646 0.765 0.7672
20 beta.unemp     -0.0054 -0.0054 -0.0053 -0.005 -0.0053
21 summary(Sigma2RanEff)
22 Quantiles for each variable:
23          2.5%    25%    50%    75%   97.5%
24 0.07208 0.09086 0.10331 0.11751 0.15600
25 summary(Sigma2)
26 Quantiles for each variable:
27          2.5%    25%    50%    75%   97.5%
28 0.001316 0.001403 0.001451 0.001501 0.001606
29 summary(Sigma2RanEff/(Sigma2RanEff+Sigma2))
30 Quantiles for each variable:
31          2.5%    25%    50%    75%   97.5%
32 0.9799 0.9842 0.9861 0.9879 0.9909

```

There are many extensions of this model, for instance, [38] propose to introduce heteroskedasticity in this model by assuming $\mu_{it}|\tau_{it} \sim N(0, \sigma^2/\tau_{it})$, $\tau_{it} \sim G(v/2, v/2)$. We ask in Exercise 2 to perform inference in the relation between productivity and public investment example using this setting.

Another potential extension is to allow dependence between \mathbf{b}_i and some controls, let's say \mathbf{z}_i , a K_3 -dimensional vector, and assume $\mathbf{b}_i \sim N(\mathbf{Z}_i\boldsymbol{\gamma}, \mathbf{D})$ where $\mathbf{Z}_i = \mathbf{I}_{K_2} \otimes \mathbf{z}_i^\top$, and complete the model using a prior for $\boldsymbol{\gamma}$, $\boldsymbol{\gamma} \sim N(\boldsymbol{\gamma}_0, \boldsymbol{\Gamma}_0)$. We ask to perform a simulation using this setting in Exercise 3.

Example: Simulation exercise of the longitudinal normal model with heteroskedasticity

Let's perform a simulation exercise to assess some potential extensions of the longitudinal hierarchical normal model. The point of departure is to assume that $y_{it} = \beta_0 + \beta_1 x_{it1} + \beta_2 x_{it2} + \beta_3 x_{it3} + b_i + w_{it1} b_{i1} + \mu_{it}$ where $x_{itk} \sim N(0, 1)$, $k = 1, 2, 3$, $w_{it1} \sim N(0, 1)$, $b_i \sim N(0, 0.7^{1/2})$, $b_{i1} \sim N(0, 0.6^{1/2})$, $\mu_{it} \sim N(0, 0.1^{1/2})$, and $\boldsymbol{\beta} = [0.5 \ 0.4 \ 0.6 \ -0.6]^\top$, $i = 1, 2, \dots, 50$. The sample size is 2000 in an *unbalanced panel structure*.

Following same stages as in this section and Exercise 1, the posterior conditional distributions assuming that $\mu_{it} | \tau_{it} \sim N(0, \sigma^2 / \tau_{it})$, $\tau_{it} \sim G(v/2, v/2)$ are given by

$$\boldsymbol{\beta} | \sigma^2, \boldsymbol{\tau}, \mathbf{D}, \mathbf{y}, \mathbf{X}, \mathbf{W} \sim N(\boldsymbol{\beta}_n, \mathbf{B}_n),$$

where $\boldsymbol{\tau} = [\tau_{it}]^\top$, $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \sum_{i=1}^N \mathbf{X}_i^\top \mathbf{V}_i^{-1} \mathbf{X}_i)^{-1}$, $\boldsymbol{\beta}_n = \mathbf{B}_n (\mathbf{B}_0^{-1} \boldsymbol{\beta}_0 + \sum_{i=1}^N \mathbf{X}_i^\top \mathbf{V}_i^{-1} \mathbf{y}_i)$, $\mathbf{V}_i = \sigma^2 \boldsymbol{\Psi}_i + \sigma_b^2 \mathbf{i}_{T_i} \mathbf{i}_{T_i}^\top$ and $\boldsymbol{\Psi}_i = \text{diag}\{\tau_{it}^{-1}\}$.

$$\mathbf{b}_i | \boldsymbol{\beta}, \sigma^2, \boldsymbol{\tau}, \mathbf{D}, \mathbf{y}, \mathbf{X}, \mathbf{W} \sim N(\mathbf{b}_{ni}, \mathbf{B}_{ni}),$$

where $\mathbf{B}_{ni} = (\sigma^{-2} \mathbf{W}_i^\top \boldsymbol{\Psi}_i^{-1} \mathbf{W}_i + \mathbf{D}^{-1})^{-1}$ and $\mathbf{b}_{ni} = \mathbf{B}_{ni} (\sigma^{-2} \mathbf{W}_i^\top \boldsymbol{\Psi}_i^{-1} (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta}))$.

$$\sigma^2 | \boldsymbol{\beta}, \mathbf{b}, \boldsymbol{\tau}, \mathbf{y}, \mathbf{X}, \mathbf{W} \sim IG(\alpha_n, \delta_n),$$

where $\alpha_n = \alpha_0 + \frac{1}{2} \sum_{i=1}^N T_i$ and $\delta_n = \delta_0 + \frac{1}{2} \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i)^\top \boldsymbol{\Psi}_i^{-1} (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta} - \mathbf{W}_i \mathbf{b}_i)$.

$$\mathbf{D} | \mathbf{b} \sim IW(d_n, \mathbf{D}_n),$$

where $d_n = d_0 + N$ and $\mathbf{D}_n = d_0 \mathbf{D}_0 + \sum_{i=1}^N \mathbf{b}_i \mathbf{b}_i^\top$. And

$$\tau_{it} | \sigma^2, \boldsymbol{\beta}, \mathbf{b}, \boldsymbol{\tau}, \mathbf{y}, \mathbf{X}, \mathbf{W} \sim G(v_{1n}/2, v_{2ni}/2),$$

where $v_{1n} = v + 1$ and $v_{2ni} = v + \sigma^{-2} (y_{it} - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i)^2$.

The following code implements this simulation, and gets draws of the posterior distributions. We set MCMC iterations, burn-in and thinning parameters equal to 5000, 1000 and 1, respectively. In addition, $\boldsymbol{\beta}_0 = \mathbf{0}_5$, $\mathbf{B}_0 = \mathbf{I}_5$, $\alpha_0 = \delta_0 = 0.001$, $d_0 = 2$, $\mathbf{D}_0 = \mathbf{I}_2$ and $v = 5$.

R code. Simulation exercise: Longitudinal normal model with heteroskedasticity from scratch

```

1 rm(list = ls()); set.seed(010101)
2 NT <- 2000; N <- 50
3 id <- c(1:N, sample(1:N, NT - N, replace=TRUE))
4 table(id)
5 x1 <- rnorm(NT); x2 <- rnorm(NT); x3 <- rnorm(NT)
6 X <- cbind(1, x1, x2, x3); K1 <- dim(X)[2]
7 w1 <- rnorm(NT); W <- cbind(1, w1)
8 K2 <- dim(W)[2]; B <- c(0.5, 0.4, 0.6, -0.6)
9 D <- c(0.7, 0.6)
10 b1 <- rnorm(N, 0, sd = D[1]^0.5)
11 b2 <- rnorm(N, 0, sd = D[2]^0.5)
12 b <- cbind(b1, b2)
13 v <- 5; tau <- rgamma(NT, shape = v/2, rate = v/2)
14 sig2 <- 0.1; u <- rnorm(NT, 0, sd = (sig2/tau)^0.5)
15 y <- NULL
16 for(i in 1:NT){
17   yi <- X[i,] %*% B + W[i,] %*% b[id[i],] + u[i]
18   y <- c(y, yi)
19 }
20 Data <- as.data.frame(cbind(y, x1, x2, x3, w1, id))
21 mcmc <- 5000; burnin <- 1000; thin <- 1; tot <- mcmc +
  burnin
22 b0 <- rep(0, K1); B0 <- diag(K1); B0i <- solve(B0)
23 r0 <- K2; R0 <- diag(K2); a0 <- 0.001; d0 <- 0.001
24 PostBeta <- function(sig2, D, tau){
25   XVX <- matrix(0, K1, K1)
26   XVy <- matrix(0, K1, 1)
27   for(i in 1:N){
28     ids <- which(id == i)
29     Ti <- length(ids)
30     Wi <- W[ids, ]
31     tauui <- tau[ids]
32     Vi <- sig2*solve(diag(1/tauui)) + Wi %*% D %*% t(Wi)
33     ViInv <- solve(Vi)
34     Xi <- X[ids, ]
35     XVXi <- t(Xi) %*% ViInv %*% Xi
36     XVX <- XVX + XVXi
37     yi <- y[ids]
38     XVyi <- t(Xi) %*% ViInv %*% yi
39     XVy <- XVy + XVyi
40   }
41   Bn <- solve(B0i + XVX)
42   bn <- Bn %*% (B0i %*% b0 + XVy)
43   Beta <- MASS::mvrnorm(1, bn, Bn)
44   return(Beta)
45 }
```

R code. Simulation exercise: Longitudinal normal model with heteroskedasticity from scratch

```

1 Postb <- function(Beta, sig2, D, tau){
2   Di <- solve(D); bis <- matrix(0, N, K2)
3   for(i in 1:N){
4     ids <- which(id == i)
5     Wi <- W[ids, ]; Xi <- X[ids, ]
6     yi <- y[ids]; taui <- tau[ids]
7     Taui <- solve(diag(1/taui))
8     Wtei <- sig2^(-1)*t(Wi)%%Taui%*%(yi - Xi%*%Beta)
9     Bni <- solve(sig2^(-1)*t(Wi)%%Taui%*%Wi + Di)
10    bni <- Bni%*%Wtei
11    bi <- MASS::mvrnorm(1, bni, Bni)
12    bis[i, ] <- bi
13  }
14  return(bis)
15 }
16 PostSig2 <- function(Beta, bs, tau){
17   an <- a0 + 0.5*NT
18   ete <- 0
19   for(i in 1:N){
20     ids <- which(id == i)
21     Xi <- X[ids, ]; yi <- y[ids]
22     Wi <- W[ids, ]; taui <- tau[ids]
23     Taui <- solve(diag(1/taui))
24     ei <- yi - Xi%*%Beta - Wi%*%bs[i, ]
25     etei <- t(ei)%*%Taui%*%ei
26     ete <- ete + etei
27   }
28   dn <- d0 + 0.5*ete
29   sig2 <- MCMCpack::rinvgamma(1, shape = an, scale = dn)
30   return(sig2)
31 }
32 PostD <- function(bs){
33   rn <- r0 + N
34   btb <- matrix(0, K2, K2)
35   for(i in 1:N){
36     bsi <- bs[i, ]
37     btbi <- bsi%*%t(bsi)
38     btb <- btb + btbi
39   }
40   Rn <- d0*R0 + btb
41   Sigma <- MCMCpack::riwish(v = rn, S = Rn)
42   return(Sigma)
43 }
44 PostTau <- function(sig2, Beta, bs){
45   v1n <- v + 1
46   v2n <- NULL
47   for(i in 1:NT){
48     Xi <- X[i, ]; yi <- y[i]
49     Wi <- W[i, ]; bi <- bs[id[i], ]
50     v2ni <- v + sig2^(-1)*(yi - Xi%*%Beta - Wi%*%bi)^2
51     v2n <- c(v2n, v2ni)
52   }
53   tau <- rgamma(NT, shape = rep(v1n/2, NT), rate = v2n/2)
54   return(tau)
55 }
```

R code. Simulation exercise: Longitudinal normal model with heteroskedasticity from scratch

```

1 PostBetas <- matrix(0, tot, K1); PostDs <- matrix(0, tot, K2
  *(K2+1)/2)
2 PostSig2s <- rep(0, tot); Postbs <- array(0, c(N, K2, tot))
3 PostTaus <- matrix(0, tot, NT); RegLS <- lm(y ~ X - 1)
4 SumLS <- summary(RegLS)
5 Beta <- SumLS[["coefficients"]][,1]
6 sig2 <- SumLS[["sigma"]]^2; D <- diag(K2)
7 tau <- rgamma(NT, shape = v/2, rate = v/2)
8 pb <- winProgressBar(title = "progress bar", min = 0, max =
  tot, width = 300)
9 for(s in 1:tot){
10   bs <- Postb(Beta = Beta, sig2 = sig2, D = D, tau = tau)
11   D <- PostD(bs = bs)
12   Beta <- PostBeta(sig2 = sig2, D = D, tau = tau)
13   sig2 <- PostSig2(Beta = Beta, bs = bs, tau = tau)
14   tau <- PostTau(sig2 = sig2, Beta = Beta, bs = bs)
15   PostBetas[s,] <- Beta
16   PostDs[s,] <- matrixcalc::vech(D)
17   PostSig2s[s] <- sig2
18   Postbs[, , s] <- bs
19   PostTaus[s,] <- tau
20   setWinProgressBar(pb, s, title=paste( round(s/tot*100, 0),
    "% done"))
21 }
22 close(pb)
23 keep <- seq((burnin+1), tot, thin)
24 Bs <- PostBetas[keep,]; Ds <- PostDs[keep,]
25 bs <- Postbs[, , keep]; sig2s <- PostSig2s[keep]
26 taus <- PostTaus[keep,]
27 summary(coda::mcmc(Bs))
28 Quantiles for each variable:
29          2.5%     25%     50%     75%   97.5%
30 var1  0.07833  0.2412  0.3232  0.4022  0.5619
31 var2  0.34101  0.3598  0.3697  0.3793  0.3988
32 var3  0.59596  0.6150  0.6251  0.6351  0.6574
33 var4 -0.63722 -0.6165 -0.6067 -0.5966 -0.5785
34 summary(coda::mcmc(Ds))
35 Quantiles for each variable:
36          2.5%     25%     50%     75%   97.5%
37 var1  0.4720  0.5995  0.68858  0.79206  1.05285
38 var2 -0.2721 -0.1405 -0.08185 -0.02482  0.09186
39 var3  0.3689  0.4644  0.52978  0.60946  0.81999
40 summary(coda::mcmc(sig2s))
41 Quantiles for each variable:
42          2.5%     25%     50%     75%   97.5%
43 0.1022  0.1157  0.1324  0.1683  0.3217

```

We can see that all the 95% credible intervals encompass the population parameters, except for the second *fixed effect* and the variance of the model, but both for a tiny margin.

9.2 Logit model

We can use the framework of Section 9.1 to perform inference in models where we have longitudinal/panel data of binary response variables. In particular, let $y_{it} \sim B(\pi_{it})$ where $\text{logit}(\pi_{it}) = \log\left(\frac{\pi_{it}}{1-\pi_{it}}\right) \equiv y_{it}^*$ such that $y_{it}^* \sim N(\mathbf{x}_{it}^\top \boldsymbol{\beta} + \mathbf{w}_{it}^\top \mathbf{b}_i, \sigma^2)$. Thus, we can *augment* the model with the latent variable y_{it}^* , and perform inference using a Metropolis-within-Gibbs sampling algorithm based on the posterior conditional distributions of the previous section.

We can implement a Gibbs sampling algorithm to sample draws from the posterior conditional distributions of $\boldsymbol{\beta}$, σ^2 , \mathbf{b}_i and \mathbf{D} using the equations in Section 9.1 conditional on \mathbf{y}_i^* . Then, we can use a random walk Metropolis-Hastings algorithm to sample y_{it}^* where the proposal distribution is Gaussian with mean y_{ij}^* and variance equal to v^2 , that is, $y_{ij}^{*c} = y_{it}^* + \epsilon_{it}$ where $\epsilon_{it} \sim \mathcal{N}(0, v^2)$, v is a tuning parameter to get good acceptance rates. We should take into account for doing predictions that $\mathbb{E}[\pi_{it}] = 1/(1 + \exp\left\{(x_{it}^\top \boldsymbol{\beta} + w_{it}^\top \mathbf{b}_i)/\sqrt{1 + (16\sqrt{3}/(15\pi))^2\sigma^2}\right\})$ [57, pag. 136].

The posterior distribution of this model is

$$\begin{aligned} \pi(\boldsymbol{\beta}, \sigma^2, \mathbf{b}_i, \mathbf{D}, \mathbf{y}^* | \mathbf{y}, \mathbf{X}, \mathbf{W}) &\propto \prod_{i=1}^N \prod_{t=1}^{T_i} \left\{ \pi_{it}^{y_{it}^*} (1 - \pi_{it})^{1-y_{it}^*} \right. \\ &\quad \times (\sigma^2)^{-1} \exp\left\{ -\frac{1}{2\sigma^2} (y_{it}^* - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i)^\top (y_{it}^* - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i) \right\} \Big\} \\ &\quad \times \exp\left\{ -\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{B}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right\} \\ &\quad \times \exp\left\{ -\frac{1}{2} \sum_{i=1}^N \mathbf{b}_i^\top \mathbf{D}^{-1} \mathbf{b}_i \right\} \\ &\quad \times (\sigma^2)^{-\alpha_0-1} \exp\left\{ -\frac{\delta_0}{\sigma^2} \right\} \\ &\quad \times |\mathbf{D}|^{-(d_0+K_2+1)/2} \exp\left\{ -\frac{1}{2} \text{tr}(d_0 \mathbf{D}_0 \mathbf{D}^{-1}) \right\}. \end{aligned}$$

We can get samples of y_{it}^* from a normal distribution with mean equal to $\mathbf{x}_{it}^\top \boldsymbol{\beta} + \mathbf{w}_{it}^\top \mathbf{b}_i$ and variance σ^2 , and use these samples to get $\pi_{it} = \frac{1}{1+e^{-y_{it}^*}}$, $y_{it}^{*c} = y_{it}^* + \epsilon_{it}$ and $\pi_{it}^c = \frac{1}{1+e^{-y_{it}^{*c}}}$, and calculate the acceptance rate of the

Metropolis-Hastings algorithm,

$$\alpha = \min \left(1, \frac{\pi_{it}^{cy_{it}} (1 - \pi_{it}^c)^{(1-y_{it})} \times \exp \left\{ -\frac{1}{2\sigma^2} (y_{it}^{c*} - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i)^\top (y_{it}^{c*} - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i) \right\}}{\pi_{it}^{y_{it}} (1 - \pi_{it})^{(1-y_{it})} \times \exp \left\{ -\frac{1}{2\sigma^2} (y_{it}^* - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i)^\top (y_{it}^* - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i) \right\}} \right).$$

Example: Doctor visits in Germany

We used the dataset *9VisitDoc.csv* provided by [219].² We analyze the determinants of a binary variable (DocVis) which is equal to 1 if an individual visited a physician in the last three months, and 0 otherwise. The dataset contains 32837 observations of 9197 individuals in an *unbalanced longitudinal/panel* dataset over the years 1995–1999 from the German Socioeconomic Panel Data.

The specification is given by

$$\begin{aligned} \text{logit}(\pi_{it}) &= \beta_1 + \beta_2 \text{Age} + \beta_3 \text{Male} + \beta_4 \text{Sport} + \beta_5 \text{LogInc} \\ &\quad + \beta_6 \text{GoodHealth} + \beta_7 \text{BadHealth} + b_i + b_{i1} \text{Sozh}, \end{aligned}$$

where $\pi_{it} = p(\text{DocVis}_{it} = 1)$.

This specification controls for age, a gender indicator (with 1 representing male), whether the individual practices any sport (with 1 for sport), the logarithm of monthly gross income, and self-perception of health status, where “good” and “bad” are compared to a baseline of “regular.” Additionally, we assume that unobserved heterogeneity is linked to whether the individual receives welfare payments (with Sozh equal to 1 for receiving welfare).

We set 10000 MCMC iterations plus 1000 burn-in, and a thinning parameter equal to 10. In addition, $\boldsymbol{\beta}_0 = \mathbf{0}_7$, $\mathbf{B}_0 = \mathbf{I}_7$, $\alpha_0 = \delta_0 = 0.001$, $d_0 = 5$ and $\mathbf{D}_0 = \mathbf{I}_2$.

The Algorithm A25 shows how to perform inference of the hierarchical longitudinal logit model using our GUI. We show in the following code how to perform inference of this example using the command *MCMChlogit* from the *MCMCpack* package. We fixed the variance for over-dispersion (σ^2) setting *FixOD* = 1 in this example. Our GUI does not fix this value, that is, it sets *FixOD* = 0, which is the default value in the command *MCMChlogit*. We ask to replicate this example using our GUI in Exercise 4. The command *MCMChlogit* uses an adaptive algorithm to tune v based on an optimal acceptance rate equal to 0.44.

²See <http://qed.econ.queensu.ca/jae/2004-v19.4/winkelmann/> for details

R code. Doctor visits in Germany

```

1 rm(list = ls())
2 set.seed(12345)
3 Data <- read.csv("https://raw.githubusercontent.com/
  besmarter/BSTApp/refs/heads/master/DataApp/9VisitDoc.csv"
  , sep = ",", header = TRUE, quote = "")
4 attach(Data)
5 K1 <- 7; K2 <- 2; N <- 9197
6 b0 <- rep(0, K1); B0 <- diag(K1)
7 r0 <- 5; R0 <- diag(K2)
8 a0 <- 0.001; d0 <- 0.001
9 RegLogit <- glm(DocVis ~ Age + Male + Sport + LogInc +
  GoodHealth + BadHealth, family = binomial(link = "logit"
  ))
10 SumLogit <- summary(RegLogit)
11 Beta0 <- SumLogit[["coefficients"]][,1]
12 mcmc <- 10000; burnin <- 1000; thin <- 10
13 # MCMChlogit
14 Resultshlogit <- MCMCpack::MCMChlogit(fixed = DocVis ~ Age +
  Male + Sport + LogInc + GoodHealth + BadHealth, random
  = "Sozh", group = "id", data = Data, burnin = burnin, mcmc
  = mcmc, thin = thin, mubeta = b0, Vbeta = B0, r = r0, R
  = R0, nu = a0, delta = d0, beta.start = Beta0, FixOD =
  1)
15 Betas <- Resultshlogit[["mcmc"]][,1:K1]
16 Sigma2RanEff <- Resultshlogit[["mcmc"]][,c(K2*N+K1+1, 2*N+K1
  +K2^2)]
17 summary(Betas)
18 Quantiles for each variable:
19          2.5%      25%      50%      75%     97.5%
20 beta.(Intercept) -1.1085 -0.6428 -0.4169 -0.166  0.280
21 beta.Age         0.0051  0.0078  0.0095  0.010  0.013
22 beta.Male        -1.1914 -1.1325 -1.0981 -1.065 -1.008
23 beta.Sport        0.2256  0.2846  0.3159  0.348  0.401
24 beta.LogInc       0.1782  0.2357  0.2661  0.299  0.367
25 beta.GoodHealth   -1.1648 -1.1046 -1.0701 -1.040 -0.983
26 beta.BadHealth     1.2233  1.3242  1.3716  1.426  1.533
27 summary(Sigma2RanEff)
28 Quantiles for each variable:
29          2.5%      25%      50%      75%     97.5%
30 VCV.(Intercept).(Intercept) 2.0749  2.1709  2.238  2.303  2.422
31 VCV.Sozh.Sozh        0.3536  0.4875  0.626  0.906  1.271

```

The results suggest that age, sports, income and a bad perception of health status increase the probability of visiting the physician, the posterior estimates have 95% symmetric credible intervals equal to (5.1e-03, 1.3e-02), (0.23, 0.40), (0.18, 0.37) and (1.22, 1.53), whereas men have a lower probability of visiting a physician, the 95% credible interval is (-1.19, -1.01), and individuals who

have a good perception of their health status also have a lower probability of visiting the doctor, the 95% credible interval is (-1.16, -0.98). The 95% credible interval of the variances of the unobserved heterogeneity associated with the welfare program is (0.35, 1.27).

Algorithm A25 Hierarchical longitudinal logit models

- 1: Select *Hierarchical Longitudinal Model* on the top panel
 - 2: Select *Logit* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Write down the formula of the *fixed effects* equation in the **Main Equation: Fixed Effects** box. This formula must be written using the syntax of the *formula* command of **R** software. This equation includes intercept by default, do not include it in the equation
 - 6: Write down the formula of the *random effects* equation in the **Main Equation: Random Effects** box without writing the dependent variable, that is, starting the equation with the *tilde* symbol. This formula must be written using the syntax of the *formula* command of **R** software. This equation includes intercept by default, do not include it in the equation. If there are just random intercepts do not write anything in this box
 - 7: Write down the name of the grouping variable, that is, the variable that indicates the cross-sectional units
 - 8: Set the hyperparameters of the *fixed effects*: mean vector, covariance matrix, shape and scale parameters. This step is not necessary as by default our GUI uses non-informative priors
 - 9: Set the hyperparameters of the *random effects*: degrees of freedom and scale matrix of the inverse Wishart distribution. This step is not necessary as by default our GUI uses non-informative priors
 - 10: Click the *Go!* button
 - 11: Analyze results
 - 12: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

9.3 Poisson model

We can use same ideas as in Section 9.2 to perform inference in longitudinal/panel datasets where the dependent variable takes non-negative integers. Let's assume that $y_{it} \sim P(\lambda_{it})$ where $\log(\lambda_{it}) = y_{it}^*$ such that $y_{it}^* \sim N(\mathbf{x}_{it}^\top \boldsymbol{\beta} + \mathbf{w}_{it}^\top \mathbf{b}_i, \sigma^2)$. We can *augment* the model with the latent variable y_{it}^* , and again use a Metropolis-within-Gibbs algorithm to perform inference in this model.

The posterior distribution of this model is

$$\begin{aligned} \pi(\boldsymbol{\beta}, \sigma^2, \mathbf{b}_i, \mathbf{D}, \mathbf{y}^* | \mathbf{y}, \mathbf{X}, \mathbf{W}) &\propto \prod_{i=1}^N \prod_{t=1}^{T_i} \left\{ \lambda_{it}^{y_{it}} \exp\{-\lambda_{it}\} \right. \\ &\quad \times (\sigma^2)^{-1} \exp \left\{ -\frac{1}{2\sigma^2} (y_{it}^* - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i)^\top (y_{it}^* - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i) \right\} \Big\} \\ &\quad \times \exp \left\{ -\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{B}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right\} \\ &\quad \times \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \mathbf{b}_i^\top \mathbf{D}^{-1} \mathbf{b}_i \right\} \\ &\quad \times (\sigma^2)^{-\alpha_0-1} \exp \left\{ -\frac{\delta_0}{\sigma^2} \right\} \\ &\quad \times |\mathbf{D}|^{-(d_0+K_2+1)/2} \exp \left\{ -\frac{1}{2} \text{tr}(d_0 \mathbf{D}_0 \mathbf{D}^{-1}) \right\}. \end{aligned}$$

We can get samples of y_{it}^* from a normal distribution with mean equal to $\mathbf{x}_{it}^\top \boldsymbol{\beta} + \mathbf{w}_{it}^\top \mathbf{b}_i$ and variance σ^2 , and use these samples to get $\lambda_{it} = \exp(y_{it}^*)$, $y_{it}^{*c} = y_{it}^* + \epsilon_{it}$, where $\epsilon_{it} \sim \mathcal{N}(0, v^2)$, v is a tuning parameter to get good acceptance rates, and $\lambda_{it}^c = \exp(y_{it}^{*c})$. The acceptance rate of the Metropolis-Hastings algorithm is

$$\alpha = \min \left(1, \frac{\lambda_{it}^{cy_{it}} \exp(-\lambda_{it}^c) \times \exp \left\{ -\frac{1}{2\sigma^2} (y_{it}^{*c} - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i)^\top (y_{it}^{*c} - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i) \right\}}{\lambda_{it}^{y_{it}} \exp(-\lambda_{it}) \times \exp \left\{ -\frac{1}{2\sigma^2} (y_{it}^* - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i)^\top (y_{it}^* - \mathbf{x}_{it}^\top \boldsymbol{\beta} - \mathbf{w}_{it}^\top \mathbf{b}_i) \right\}} \right).$$

In addition, we should use the posterior conditional distributions from Section 9.1 to complete the algorithm getting samples of $\boldsymbol{\beta}$, σ^2 , \mathbf{b}_i and \mathbf{D} replacing y_{it} by y_{it}^* .

We should take into account for doing predictions that $\mathbb{E}[\lambda_{it}] = \exp \{ \mathbf{x}_{it}^\top \boldsymbol{\beta} + \mathbf{w}_{it}^\top \mathbf{b}_i + 0.5\sigma^2 \}$ [57, pag. 137].

Example: Simulation exercise

Let's perform a simulation exercise to assess the performance of the hierarchical longitudinal Poisson model. The point of departure is to assume that $y_{it}^* = \beta_0 + \beta_1 x_{it1} + \beta_2 x_{it2} + \beta_3 x_{it3} + b_i + w_{it1} b_{i1}$ where $x_{itk} \sim N(0, 1)$, $k = 1, 2, 3$,

$w_{it1} \sim N(0, 1)$, $b_i \sim N(0, 0.7^{1/2})$, $b_{i1} \sim N(0, 0.6^{1/2})$, $\beta = [0.5 \ 0.4 \ 0.6 \ -0.6]^\top$, $i = 1, 2, \dots, 50$, and $y_{it} \sim P(\lambda_{it})$, where $\lambda_{it} = \exp(y_{it}^*)$. The sample size is 1000 in an *unbalanced panel structure*.

Let's set $\beta_0 = \mathbf{0}_4$, $B_0 = \mathbf{I}_4$, $\alpha_0 = \delta_0 = 0.001$, $d_0 = 2$ and $D_0 = \mathbf{I}_2$. The number of MCMC iterations, burn-in and thinning parameters are 15000, 5000 and 10, respectively.

We can perform inference of the hierarchical longitudinal Poisson model in our GUI using Algorithm A26. Our GUI is based on the command *MCMChpoisson* from the *MCMCpack* package.

Algorithm A26 Hierarchical longitudinal Poisson models

- 1: Select *Hierarchical Longitudinal Model* on the top panel
 - 2: Select *Poisson* model using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Select MCMC iterations, burn-in and thinning parameters using the *Range sliders*
 - 5: Write down the formula of the *fixed effects* equation in the **Main Equation: Fixed Effects** box. This formula must be written using the syntax of the *formula* command of **R** software. This equation includes intercept by default, do not include it in the equation
 - 6: Write down the formula of the *random effects* equation in the **Main Equation: Random Effects** box without writing the dependent variable, that is, starting the equation with the *tilde* symbol. This formula must be written using the syntax of the *formula* command of **R** software. This equation includes intercept by default, do not include it in the equation. If there are just random intercepts do not write anything in this box
 - 7: Write down the name of the grouping variable, that is, the variable that indicates the cross-sectional units
 - 8: Set the hyperparameters of the *fixed effects*: mean vector, covariance matrix, shape and scale parameters. This step is not necessary as by default our GUI uses non-informative priors
 - 9: Set the hyperparameters of the *random effects*: degrees of freedom and scale matrix of the inverse Wishart distribution. This step is not necessary as by default our GUI uses non-informative priors
 - 10: Click the *Go!* button
 - 11: Analyze results
 - 12: Download posterior chains and diagnostic plots using the *Download Posterior Chains* and *Download Posterior Graphs* buttons
-

The following code shows how to perform inference in the hierarchical longitudinal Poisson model programming the Metropolis-within-Gibbs sampler.

R code. Simulation exercise: Hierarchical longitudinal Poisson model

```

1 rm(list = ls()); set.seed(010101)
2 NT <- 1000; N <- 50
3 id <- c(1:N, sample(1:N, NT - N, replace=TRUE))
4 x1 <- rnorm(NT); x2 <- rnorm(NT); x3 <- rnorm(NT)
5 X <- cbind(1, x1, x2, x3)
6 K1 <- dim(X)[2]; w1 <- rnorm(NT)
7 W <- cbind(1, w1); K2 <- dim(W)[2]
8 B <- c(0.5, 0.4, 0.6, -0.6)
9 D <- c(0.7, 0.6); sig2 <- 0.1
10 b1 <- rnorm(N, 0, sd = D[1]^0.5)
11 b2 <- rnorm(N, 0, sd = D[2]^0.5)
12 b <- cbind(b1, b2)
13 yl <- NULL
14 for(i in 1:NT){
15   ylmeani <- X[i,] %*% B + W[i,] %*% b[id[i],]
16   yli <- rnorm(1, ylmeani, sig2^0.5)
17   yl <- c(yl, yli)
18 }
19 lambdait <- exp(yl); y <- rpois(NT, lambdait)
20 Data <- as.data.frame(cbind(y, x1, x2, x3, w1, id))
21 mcmc <- 15000; burnin <- 5000; thin <- 10; tot <- mcmc +
  burnin
22 b0 <- rep(0, K1); B0 <- diag(K1); B0i <- solve(B0)
23 r0 <- K2; R0 <- diag(K2); a0 <- 0.001; d0 <- 0.001
24 LatentMHV1 <- function(tuning, Beta, bs, sig2){
25   ylhat <- rep(0, NT)
26   accept <- NULL
27   for(i in 1:NT){
28     ids <- which(id == i)
29     yi <- y[i]
30     ylhatmeani <- X[i,] %*% Beta + W[i,] %*% bs[id[i],]
31     ylhati <- rnorm(1, ylhatmeani, sd = sig2^0.5)
32     lambdahati <- exp(ylhati)
33     ei <- rnorm(1, 0, sd = tuning)
34     ylpropri <- ylhati + ei
35     lambdapropi <- exp(ylpropri)
36     logPosthati <- sum(dpois(yi, lambdahati, log = TRUE) +
      dnorm(ylhati, ylhatmeani, sig2^0.5, log = TRUE))
37     logPostpropri <- sum(dpois(yi, lambdapropi, log = TRUE) +
      dnorm(ylpropri, ylhatmeani, sig2^0.5, log = TRUE))
38     alphai <- min(1, exp(logPostpropri - logPosthati))
39     ui <- runif(1)
40     if(ui <= alphai){
41       ylhati <- ylpropri; accepti <- 1
42     }else{
43       ylhati <- ylhati; accepti <- 0
44     }
45     ylhat[i] <- ylhati
46     accept <- c(accept, accepti)
47   }
48   res <- list(ylhat = ylhat, accept = mean(accept))
49   return(res)
50 }
```

R code. Simulation exercise: Hierarchical longitudinal Poisson model

```

1 PostBeta <- function(D, ylhat, sig2){
2   XVX <- matrix(0, K1, K1); XVy <- matrix(0, K1, 1)
3   for(i in 1:N){
4     ids <- which(id == i); Ti <- length(ids)
5     Wi <- W[ids, ]
6     Vi <- diag(Ti)*sig2 + Wi%*%D%*%t(Wi)
7     ViInv <- solve(Vi); Xi <- X[ids, ]
8     XVXi <- t(Xi)%*%ViInv%*%Xi
9     XVX <- XVX + XVXi
10    yi <- ylhat[ids]
11    XVy <- t(Xi)%*%ViInv%*%yi
12    XVy <- XVy + XVy
13  }
14  Bn <- solve(B0i + XVX); bn <- Bn%*%(B0i%*%b0 + XVy)
15  Beta <- MASS::mvrnorm(1, bn, Bn)
16  return(Beta)
17 }
18 Postb <- function(Beta, D, ylhat, sig2){
19   Di <- solve(D); bis <- matrix(0, N, K2)
20   for(i in 1:N){
21     ids <- which(id == i)
22     Wi <- W[ids, ]; Xi <- X[ids, ]
23     yi <- ylhat[ids]
24     Wtei <- sig2^(-1)*t(Wi)%*%(yi - Xi%*%Beta)
25     Bni <- solve(sig2^(-1)*t(Wi)%*%Wi + Di)
26     bni <- Bni%*%Wtei
27     bi <- MASS::mvrnorm(1, bni, Bni)
28     bis[i, ] <- bi
29   }
30   return(bis)
31 }
32 PostD <- function(bs){
33   rn <- r0 + N; btb <- matrix(0, K2, K2)
34   for(i in 1:N){
35     bsi <- bs[i, ]; btbi <- bsi%*%t(bsi)
36     btb <- btb + btbi
37   }
38   Rn <- d0*R0 + btb
39   Sigma <- MCMCpack::riwish(v = rn, S = Rn)
40   return(Sigma)
41 }
42 PostSig2 <- function(Beta, bs, ylhat){
43   an <- a0 + 0.5*NT; ete <- 0
44   for(i in 1:N){
45     ids <- which(id == i)
46     Xi <- X[ids, ]
47     yi <- ylhat[ids]
48     Wi <- W[ids, ]
49     ei <- yi - Xi%*%Beta - Wi%*%bs[i, ]
50     etei <- t(ei)%*%ei
51     ete <- ete + etei
52   }
53   dn <- d0 + 0.5*ete
54   sig2 <- MCMCpack::rinvgamma(1, shape = an, scale = dn)
55   return(sig2)
56 }
```

R code. Simulation exercise: Hierarchical longitudinal Poisson model

```

1 PostBetas <- matrix(0, tot, K1); PostDs <- matrix(0, tot, K2
  *(K2+1)/2)
2 Postbs <- array(0, c(N, K2, tot)); PostSig2s <- rep(0, tot)
3 Accepts <- rep(NULL, tot)
4 RegPois <- glm(y ~ X - 1, family = poisson(link = "log"))
5 SumPois <- summary(RegPois)
6 Beta <- SumPois[["coefficients"]][,1]
7 sig2 <- sum(SumPois[["deviance.resid"]])^2/SumPois[["df.
  residual"]]
8 D <- diag(K2); bs1 <- rnorm(N, 0, sd = D[1,1]^0.5)
9 bs2 <- rnorm(N, 0, sd = D[2,2]^0.5); bs <- cbind(bs1, bs2)
10 tuning <- 0.1; ropt <- 0.44
11 tunepariter <- seq(round(tot/10, 0), tot, round(tot/10, 0));
    l <- 1
12 pb <- winProgressBar(title = "progress bar", min = 0, max =
  tot, width = 300)
13 for(s in 1:tot){
14   LatY <- LatentMHV1(tuning = tuning, Beta = Beta, bs = bs,
    sig2 = sig2)
15   ylhat <- LatY[["ylhat"]]
16   bs <- Postb(Beta = Beta, D = D, ylhat=ylhat, sig2 = sig2)
17   D <- PostD(bs = bs)
18   Beta <- PostBeta(D = D, ylhat = ylhat, sig2 = sig2)
19   sig2 <- PostSig2(Beta = Beta, bs = bs, ylhat = ylhat)
20   PostBetas[s,] <- Beta
21   PostDs[s,] <- matrixcalc::vech(D)
22   Postbs[, , s] <- bs; PostSig2s[s] <- sig2
23   AcceptRate <- LatY[["accept"]]
24   Accepts[s] <- AcceptRate
25   if(AcceptRate > ropt){
26     tuning = tuning*(2-(1-AcceptRate)/(1-ropt))
27   }else{
28     tuning = tuning/(2-AcceptRate/ropt)
29   }
30   if(s == tunepariter[1]){
31     print(AcceptRate); l <- l + 1
32   }
33   setWinProgressBar(pb, s, title=paste( round(s/tot*100, 0),
    "% done"))
34 }
35 close(pb)
36 keep <- seq((burnin+1), tot, thin)
37 Bs <- PostBetas[keep,]; Ds <- PostDs[keep,]
38 bs <- Postbs[, , keep]; sig2s <- PostSig2s[keep]
39 summary(coda::mcmc(Bs))
40 Quantiles for each variable:
41      2.5%     25%     50%     75%   97.5%
42 var1  0.1038  0.3803  0.5199  0.6534  0.9259
43 var2  0.2432  0.3166  0.3608  0.4003  0.4796
44 var3  0.4213  0.5017  0.5453  0.5885  0.6682
45 var4 -0.7038 -0.6149 -0.5729 -0.5269 -0.4459
46 summary(coda::mcmc(Ds))
47 Quantiles for each variable:
48      2.5%     25%     50%     75%   97.5%
49 var1  0.3331  0.4788  0.5732  0.69316 0.99135
50 var2 -0.3354 -0.1926 -0.1277 -0.06692 0.03674
51 var3  0.1252  0.2182  0.2780  0.34731 0.51055

```

We can see that all 95% credible intervals encompass the population parameters of the *fixed effects*, the posterior medians are relatively near the population values. However, we do not get good posterior estimates of the covariance matrix of the *random effects* as the 95% credible intervals do not encompass the second element of the diagonal of this matrix. In addition, the posterior draws of this algorithm over-estimates the over-dispersion parameter.

9.4 Summary

We present in this chapter how to perform inference in longitudinal/panel data models from a Bayesian perspective. In particular, the Bayesian approach uses a hierarchical structure where the *random effects* have priors depending on hyperparameters which in turn have also priors. We show the three most common cases: continuous, binary and count dependent variables. These basic models that we show in this chapter can be easily extended to more flexible cases given the hierarchical structure.

9.5 Exercises

1. Show that the posterior distribution of $\beta|\sigma^2, \mathbf{D}$ is $N(\beta_n, \mathbf{B}_n)$, where $\mathbf{B}_n = (\mathbf{B}_0^{-1} + \sum_{i=1}^N \mathbf{X}_i^\top \mathbf{V}_i^{-1} \mathbf{X}_i)^{-1}$, $\beta_n = \mathbf{B}_n(\mathbf{B}_0^{-1}\beta_0 + \sum_{i=1}^N \mathbf{X}_i^\top \mathbf{V}_i^{-1} \mathbf{y}_i)$.
2. **The relation between productivity and public investment example continues**

- Perform inference of this example using our GUI.
- Program from scratch a Gibbs sampling algorithm to perform this application. Set $\beta_0 = \mathbf{0}_5$, $\mathbf{B}_0 = \mathbf{I}_5$, $\alpha_0 = \delta_0 = 0.001$, $d_0 = 5$ and $\mathbf{D}_0 = \mathbf{I}_1$.
- Perform inference in this example assuming that $\mu_{it}|\tau_{it} \sim N(0, \sigma^2/\tau_{it})$ and $\tau_{it} \sim G(v/2, v/2)$ setting $v = 5$.

3. **Simulation exercise of the longitudinal normal model continues**

Assume that $y_{it} = \beta_0 + \beta_1 x_{it1} + \beta_2 x_{it2} + \beta_3 x_{it3} + \beta_4 z_{i1} + b_i + w_{it1} b_{i1} + \mu_{it}$ where $x_{itk} \sim N(0, 1)$, $k = 1, 2, 3$, $z_{i1} \sim B(0.5)$, $w_{it1} \sim N(0, 1)$, $b_i \sim N(0, 0.7^{1/2})$, $b_{i1} \sim N(0, 0.6^{1/2})$, $\mu_{it} \sim N(0, 0.1^{1/2})$

$\beta = [0.5 \ 0.4 \ 0.6 \ -0.6 \ 0.7]^\top$, $i = 1, 2, \dots, 50$, and the sample size is 2000 in an *unbalanced panel structure*. In addition, we assume that b_i dependents on $z_i = [1 \ z_{i1}]^\top$ such that $b_i \sim N(Z_i\gamma, D)$ where $Z_i = I_{K_2} \otimes z_i^\top$, where $\gamma = [1 \ 1 \ 1 \ 1]$. The prior for γ is $N(\gamma_0, \Gamma_0)$ where we set $\gamma_0 = \mathbf{0}_4$ and $\Gamma_0 = I_4$.

- Perform inference in this model without taking into account the dependence between b_i and z_{i1} , and compare the posterior estimates with the population parameters.
- Perform inference in this model taking into account the dependence between b_i and z_{i1} , and compare the posterior estimates with the population parameters.

4. Doctor visits in Germany continues I

Replicate this example using our GUI, which by default does not fix the over-dispersion parameter (σ^2), and compare the results with the results of this example in Section 9.2.

5. Simulation exercise of the longitudinal logit model

Perform a simulation exercise to assess the performance of the hierarchical longitudinal logit model. The point of departure is to assume that $y_{it}^* = \beta_0 + \beta_1 x_{it1} + \beta_2 x_{it2} + \beta_3 x_{it3} + b_i + w_{it1} b_{i1}$ where $x_{itk} \sim N(0, 1)$, $k = 1, 2, 3$, $w_{it1} \sim N(0, 1)$, $b_i \sim N(0, 0.7^{1/2})$, $b_{i1} \sim N(0, 0.6^{1/2})$, $\beta = [0.5 \ 0.4 \ 0.6 \ -0.6]^\top$, $i = 1, 2, \dots, 50$, and $y_{it} \sim B(\pi_{it}$, where $\pi_{it} = 1/(1 + \exp(y_{it}^*))$). The sample size is 1000 in an *unbalanced panel structure*.

- Perform inference using the command *MCMChlogit* fixing the over-dispersion parameter, and using $\beta_0 = \mathbf{0}_4$, $B_0 = I_4$, $\alpha_0 = \delta_0 = 0.001$, $d_0 = 2$ and $D_0 = I_2$.
- Program from scratch a Metropolis-within-Gibbs algorithm to perform inference in this simulation.

6. Doctor visits in Germany continues II

Take a sub-sample of the first 500 individuals of the datatset *9VisitDoc.csv* to perform inference in the number of visits to doctors (*DocNum*) with the same specification of the example of **Doctor visits in Germany** of Section 9.2.



10

Bayesian model average

We outline in this chapter a framework for addressing model uncertainty and averaging across different models in a probabilistically consistent manner. The discussion tackles two major computational challenges in Bayesian model averaging: the vast space of possible models and the absence of analytical solutions for the marginal likelihood.

We begin by illustrating the approach within the Gaussian linear model, assuming exogeneity of the regressors, and extend the analysis to cases with endogenous regressors. Additionally, we adapt the framework to generalized linear models, including the logit, gamma, and Poisson families.

For dynamic models, we demonstrate how Bayesian model averaging can be implemented in contexts requiring online predictions. Lastly, we explore alternative methods for computing marginal likelihoods, especially when the Bayesian information criterion's asymptotic approximation proves inadequate.

Remember that we can run our GUI typing

R code. How to display our graphical user interface

```
1 shiny::runGitHub("besmarter/BSTApp", launch.browser = T)
```

in the **R** package console or any **R** code editor, and once our GUI is deployed, select *Bayesian Model Averaging*. However, users should see Chapter 5 for other options and details.

10.1 Foundation

Remember from Chapter 1 that Bayesian model averaging (BMA) is an approach which takes into account model uncertainty. In particular, we consider uncertainty in the regressors (variable selection) in a regression framework

where there are K possible explanatory variables.¹ This implies 2^K potential models indexed by parameters $\boldsymbol{\theta}_m$, $m = 1, 2, \dots, 2^K$.

Following [201], the posterior model probability is

$$\pi(\mathcal{M}_j|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{M}_j)\pi(\mathcal{M}_j)}{\sum_{m=1}^{2^K} p(\mathbf{y}|\mathcal{M}_m)\pi(\mathcal{M}_m)},$$

where $\pi(\mathcal{M}_j)$ is the prior model probability,²

$$p(\mathbf{y}|\mathcal{M}_j) = \int_{\Theta_j} p(\mathbf{y}|\boldsymbol{\theta}_j, \mathcal{M}_j)\pi(\boldsymbol{\theta}_j|\mathcal{M}_j)d\boldsymbol{\theta}_j$$

is the marginal likelihood, and $\pi(\boldsymbol{\theta}_j|\mathcal{M}_j)$ is the prior distribution of $\boldsymbol{\theta}_j$ conditional on model \mathcal{M}_j .

Following [170], the posterior distribution of $\boldsymbol{\theta}$ is

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \sum_{m=1}^{2^K} \pi(\boldsymbol{\theta}_m|\mathbf{y}, \mathcal{M}_m)\pi(\mathcal{M}_m|\mathbf{y})$$

where $\pi(\boldsymbol{\theta}_m|\mathbf{y}, \mathcal{M}_m)$ is the posterior distribution of $\boldsymbol{\theta}$ under model \mathcal{M}_m , $\mathbb{E}[\boldsymbol{\theta}|\mathbf{y}] = \sum_{m=1}^{2^K} \hat{\boldsymbol{\theta}}_m \pi(\mathcal{M}_m|\mathbf{y})$, $Var(\theta_{km}|\mathbf{y}) = \sum_{m=1}^{2^K} \pi(\mathcal{M}_m|\mathbf{y}) \widehat{Var}(\theta_{km}|\mathbf{y}, \mathcal{M}_m) + \sum_{m=1}^{2^K} \pi(\mathcal{M}_m|\mathbf{y}) (\hat{\theta}_{km} - \mathbb{E}[\theta_{km}|\mathbf{y}])^2$, $\hat{\boldsymbol{\theta}}_m$ is the posterior mean and $\widehat{Var}(\theta_{km}|\mathbf{y}, \mathcal{M}_m)$ is the posterior variance of the element k -th of $\boldsymbol{\theta}$ under model \mathcal{M}_m .

The posterior variance highlights how the BMA method takes into account model uncertainty. The first term is the weighted variance of each model, averaged over all potential models, and the second term indicates how stable the estimates are across models. The more the estimates differ between models, the greater is the posterior variance.

The posterior predictive distribution is

$$\pi(\mathbf{Y}_0|\mathbf{y}) = \sum_{m=1}^{2^K} p_m(\mathbf{Y}_0|\mathbf{y}, \mathcal{M}_m)\pi(\mathcal{M}_m|\mathbf{y})$$

where $p_m(\mathbf{Y}_0|\mathbf{y}, \mathcal{M}_m) = \int_{\Theta_m} p(\mathbf{Y}_0|\mathbf{y}, \boldsymbol{\theta}_m, \mathcal{M}_m)\pi(\boldsymbol{\theta}_m|\mathbf{y}, \mathcal{M}_m)d\boldsymbol{\theta}_m$ is the posterior predictive distribution under model \mathcal{M}_m .

Another important statistic in BMA is the posterior inclusion probability associated with variable \mathbf{x}_k , $k = 1, 2, \dots, K$, which is

$$PIP(\mathbf{x}_k) = \sum_{m=1}^{2^K} \pi(\mathcal{M}_m|\mathbf{y}) \times \mathbb{1}_{k,m},$$

¹Take into account that K can increase when interaction terms and/or polynomial terms of the original control variables are included.

²We attach equal prior probabilities to each model in our GUI. However, this choice gives more prior probability to the set of models of medium size (think about the k -th row of Pascal's triangle). An interesting alternative is to use the Beta-Binomial prior proposed by [135].

$$\text{where } \mathbb{1}_{k,m} = \begin{cases} 1 & \text{if } \mathbf{x}_k \in \mathcal{M}_m \\ 0 & \text{if } \mathbf{x}_k \notin \mathcal{M}_m \end{cases}.$$

[120] suggest that posterior inclusion probabilities (PIP) less than 0.5 are evidence against the regressor, $0.5 \leq PIP < 0.75$ is weak evidence, $0.75 \leq PIP < 0.95$ is positive evidence, $0.95 \leq PIP < 0.99$ is strong evidence, and $PIP \geq 0.99$ is very strong evidence.

There are two main computational issues in implementing BMA based on variable selection. First, the number of models in the model space is 2^K , which sometimes can be enormous. For instance, three regressors imply just eight models, see Table 10.1, but 40 regressors implies approximately $1.1e+12$ models. Take into account that models always include the intercept, and all regressors should be standardized to avoid scale issues.³ The second computational issue is calculating the marginal likelihood $p(\mathbf{y}|\mathcal{M}_j) = \int_{\Theta_j} p(\mathbf{y}|\boldsymbol{\theta}_j, \mathcal{M}_j) \pi(\boldsymbol{\theta}_j|\mathcal{M}_j) d\boldsymbol{\theta}_j$, which most of the time does not have an analytic solution.

TABLE 10.1
Space of models: Three regressors.

Regressor	Inclusion						
x_1	1	1	1	1	0	0	0
x_2	1	1	0	0	1	1	0
x_3	1	0	1	0	1	0	1

Notes: “1” indicates inclusion of the regressor, and “0” indicates no inclusion. The space of models is composed by 8 models. The model always includes intercept.

The first computational issue is basically a problem of ranking models. This can be tackled using different approaches, such as Occam’s window criterion [142, 172], reversible jump Markov chain Monte Carlo computation [91], Markov chain Monte Carlo model composition [143], and multiple testing using intrinsic priors [32] or nonlocal prior densities [109]. We focus on Occam’s window and Markov chain Monte Carlo model composition in our GUI.⁴

In Occam’s window, a model is discarded if its predictive performance is much worse than that of the best model [142, 172]. Thus, models not belonging to $\mathcal{M}' = \left\{ \mathcal{M}_j : \frac{\max_m \pi(\mathcal{M}_m|\mathbf{y})}{\pi(\mathcal{M}_j|\mathbf{y})} \leq c \right\}$ should be discarded, where c is chosen by the user ([142] propose $c = 20$). In addition, complicated models than are less supported by the data than simpler models are also discarded, that is, $\mathcal{M}'' = \left\{ \mathcal{M}_j : \exists \mathcal{M}_m \in \mathcal{M}', \mathcal{M}_m \subset \mathcal{M}_j, \frac{\pi(\mathcal{M}_m|\mathbf{y})}{\pi(\mathcal{M}_j|\mathbf{y})} > 1 \right\}$. Then, the set of

³Scaling variables is always an important step in variable selection.

⁴Variable selection (model selection or regularization) is a topic related to model uncertainty. Approaches such as stochastic search variable selection (spike and slab) [80, 81] and Bayesian Lasso [160] are good examples of how to tackle this issue. See Chapter 12.

models used in BMA is $\mathcal{M}^* = \mathcal{M}' \cap \mathcal{M}''^c \in \mathcal{M}$. [172] find that the number of models in \mathcal{M}^* is normally less than 25.

However, the previous theoretical framework requires finding the model with the maximum a posteriori model probability ($\max_m \pi(\mathcal{M}_m | \mathbf{y})$), which implies calculating all possible models in \mathcal{M} . This is computationally burdensome. Hence, a heuristic approach is proposed by [169] based on ideas of [142]. The search strategy is based on a series of nested comparisons of ratios of posterior model probabilities. Let \mathcal{M}_0 be a model with one regressor less than model \mathcal{M}_1 , then:

- If $\log(\pi(\mathcal{M}_0 | \mathbf{y}) / \pi(\mathcal{M}_1 | \mathbf{y})) > \log(O_R)$, then \mathcal{M}_1 is rejected and \mathcal{M}_0 is considered.
- If $\log(\pi(\mathcal{M}_0 | \mathbf{y}) / \pi(\mathcal{M}_1 | \mathbf{y})) \leq -\log(O_L)$, then \mathcal{M}_0 is rejected, and \mathcal{M}_1 is considered.
- If $\log(O_L) < \log(\pi(\mathcal{M}_0 | \mathbf{y}) / \pi(\mathcal{M}_1 | \mathbf{y})) \leq \log(O_R)$, \mathcal{M}_0 and \mathcal{M}_1 are considered.

Here O_R is a number specifying the maximum ratio for excluding models in Occam's window, and $O_L = 1/O_R^2$ is defined by default in [169]. The search strategy can be “up,” adding one regressor, or “down,” dropping one regressor (see [142] for details about the down and up algorithms). The leaps and bounds algorithm [69] is implemented to improve the computational efficiency of this search strategy [169]. Once the set of potentially acceptable models is defined, we discard all the models that are not in \mathcal{M}' , and the models that are in \mathcal{M}'' where 1 is replaced by $\exp\{O_R\}$ due to the leaps and bounds algorithm giving an approximation to BIC, so as to ensure that no good models are discarded.

The second approach that we consider in our GUI to tackle the model space size issue is Markov chain Monte Carlo model composition (MC3) [144]. In particular, given the space of models \mathcal{M}_m , we simulate a chain of \mathcal{M}_s models, $s = 1, 2, \dots, S \ll 2^K$, where the algorithm randomly extracts a candidate model \mathcal{M}_c from a neighborhood of models ($nbd(\mathcal{M}_m)$) that consists of the actual model itself and the set of models with either one variable more or one variable less [172]. Therefore, there is a transition kernel in the space of models $q(\mathcal{M}_m \rightarrow \mathcal{M}_c)$, such that $q(\mathcal{M}_m \rightarrow \mathcal{M}_c) = 0 \forall \mathcal{M}_c \notin nbd(\mathcal{M}_m)$ and $q(\mathcal{M}_m \rightarrow \mathcal{M}_c) = \frac{1}{|nbd(\mathcal{M}_m)|} \forall \mathcal{M}_m \in nbd(\mathcal{M}_m)$, $|nbd(\mathcal{M}_m)|$ being the number of neighbors of \mathcal{M}_m . This candidate model is accepted with probability

$$\alpha(\mathcal{M}_{s-1}, \mathcal{M}_c) = \min \left\{ \frac{|nbd(\mathcal{M}_m)| p(\mathbf{y} | \mathcal{M}_c) \pi(\mathcal{M}_c)}{|nbd(\mathcal{M}^c)| p(\mathbf{y} | \mathcal{M}_{(s-1)}) \pi(\mathcal{M}_{(s-1)})}, 1 \right\}.$$

Observe that by construction $|nbd(\mathcal{M}_m)| = |nbd(\mathcal{M}_c)| = k$, except in extreme cases where a model has only one regressor or has all regressors.

The Bayesian information criterion is a possible solution for the second computational issue in BMA, that is, calculating the marginal likelihood when

there is no an analytic solution. Defining $h(\boldsymbol{\theta}|\mathcal{M}_j) = -\frac{\log(p(\mathbf{y}|\boldsymbol{\theta}_j, \mathcal{M}_j)\pi(\boldsymbol{\theta}_j|\mathcal{M}_j))}{N}$, then $p(\mathbf{y}|\mathcal{M}_j) = \int_{\boldsymbol{\Theta}_j} \exp\{-Nh(\boldsymbol{\theta}|\mathcal{M}_j)\} d\boldsymbol{\theta}_j$. If N is sufficiently large (technically $N \rightarrow \infty$), we can make the following assumptions [99]:

- We can use the Laplace method for approximating integrals [211].
- The posterior mode is reached at the same point as the maximum likelihood estimator (MLE), denoted by $\hat{\boldsymbol{\theta}}_j^{MLE}$.

We get the following results under these assumptions:

$$p(\mathbf{y}|\mathcal{M}_j) \approx \left(\frac{2\pi}{N}\right)^{K_j/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-Nh(\hat{\boldsymbol{\theta}}_j^{MLE}|\mathcal{M}_j)\right\}, \quad N \rightarrow \infty,$$

where $\boldsymbol{\Sigma}$ is the Hessian matrix of $h(\hat{\boldsymbol{\theta}}_j^{MLE}|\mathcal{M}_j)$, and $K_j = \dim\{\boldsymbol{\theta}_j\}$.

This implies

$$\begin{aligned} \log(p(\mathbf{y}|\mathcal{M}_j)) &\approx \frac{K_j}{2} \log(2\pi) - \frac{K_j}{2} \log(N) - \frac{1}{2} \log(|\boldsymbol{\Sigma}|) + \log(p(\mathbf{y}|\hat{\boldsymbol{\theta}}_j^{MLE}, \mathcal{M}_j)) \\ &\quad + \log(\pi(\hat{\boldsymbol{\theta}}_j^{MLE}|\mathcal{M}_j)), \quad N \rightarrow \infty. \end{aligned}$$

Since $\frac{K_j}{2} \log(2\pi)$ and $\log(\pi(\hat{\boldsymbol{\theta}}_j^{MLE}|\mathcal{M}_j))$ are constants as functions of \mathbf{y} , and $|\boldsymbol{\Sigma}|$ is bounded by a finite constant, we have

$$\log(p(\mathbf{y}|\mathcal{M}_j)) \approx -\frac{K_j}{2} \log(N) + \log(p(\mathbf{y}|\hat{\boldsymbol{\theta}}_j^{MLE}, \mathcal{M}_j)) = -\frac{BIC}{2}, \quad N \rightarrow \infty.$$

The marginal likelihood thus asymptotically converges to a linear transformation of the Bayesian Information Criterion (BIC), significantly simplifying its calculation. In addition, the BIC is consistent, that is, the probability of uncovering the population statistical model converges to one as the sample size converges to infinity given a \mathcal{M} -closed view [18, Chap. 6], that is, one of the models in consideration is the population statistical model (data generating process) [194, 25]. In case that there is an \mathcal{M} -completed view of nature, that is, there is a true data generating process, but the space of models that we are comparing does not include it, the BIC asymptotically selects the model that minimizes the Kullback-Leiber (KL) divergence to the true (population) model [45, Chap. 4].

10.2 The Gaussian linear model

The Gaussian linear model specifies $\mathbf{y} = \alpha \mathbf{i}_N + \mathbf{X}_m \boldsymbol{\beta}_m + \boldsymbol{\mu}_m$ such that $\boldsymbol{\mu}_m \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n)$, and \mathbf{X}_m does not have the column of ones. Following [126], the conjugate prior for the location parameters is $\boldsymbol{\beta}_m | \sigma^2 \sim N(\boldsymbol{\beta}_{m0}, \sigma^2 \mathbf{B}_{m0})$, and

the priors for σ^2 and α can be improper, as these parameters are common to all models \mathcal{M}_m . Particularly, $\pi(\sigma^2) \propto 1/\sigma^2$ (Jeffreys' prior for the linear Gaussian model, see [101]) and $\pi(\alpha) \propto 1$.

The selection of the hyperparameters of β_m is more critical, as these parameters are not common to all models. A very common prior for the location parameters in the BMA literature is the Zellner's prior [228], where $\beta_{m0} = \mathbf{0}_m$ and $\mathbf{B}_{m0} = (g_m \mathbf{X}_m^\top \mathbf{X}_m)^{-1}$. Observe that this covariance matrix is similar to the covariance matrix of ordinary least squares estimator of the location parameters. This suggests that there is compatibility between the prior information and the sample information, and the only parameter to elicit is $g_m \geq 0$, which facilitates the elicitation process, as eliciting covariance matrices is a very hard endeavor.

Following same steps as in Section 3.3, the posterior conditional distribution of β_m has covariance matrix $\sigma^2 \mathbf{B}_{mn}$, where $\mathbf{B}_{mn} = ((1 + g_m) \mathbf{X}_m^\top \mathbf{X}_m)^{-1}$ (Exercise 1), which means that $g_m = 0$ implies a non-informative prior, whereas $g_m = 1$ implies that prior and data information have same weights. We follow [66], who recommend

$$g_m = \begin{cases} 1/K^2, & N \leq K^2 \\ 1/N, & N > K^2 \end{cases}.$$

Given the likelihood function,

$$p(\beta_m, \sigma^2 | \mathbf{y}, \mathbf{X}_m) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \alpha \mathbf{i}_N - \mathbf{X}_m \beta_m)^\top (\mathbf{y} - \alpha \mathbf{i}_N - \mathbf{X}_m \beta_m) \right\},$$

the marginal likelihood associated with model \mathcal{M}_m is proportional to (Exercise 1)

$$p(\mathbf{y} | \mathcal{M}_m) \propto \left(\frac{g_m}{1 + g_m} \right)^{k_m/2} \left[(\mathbf{y} - \bar{y} \mathbf{i}_N)^\top (\mathbf{y} - \bar{y} \mathbf{i}_N) - \frac{1}{1 + g_m} (\mathbf{y}^\top \mathbf{P}_{X_m} \mathbf{y}) \right]^{-(N-1)/2},$$

where all parameter are indexed to model \mathcal{M}_m , $\mathbf{P}_{X_m} = \mathbf{X}_m (\mathbf{X}_m^\top \mathbf{X}_m)^{-1} \mathbf{X}_m$ is the projection matrix on the space generated by the columns of \mathbf{X}_m , and \bar{y} is the sample mean of \mathbf{y} .

We implement in our GUI three approaches to perform BMA in the Gaussian linear model: the BIC approximation using the Occam's window approach, the MC3 algorithm using the analytical expression for calculating the marginal likelihood, and an instrumental variable approach based on conditional likelihoods.

Example: Simulation exercise

Let's perform a simulation exercise to assess the performance of the BIC approximation using the Occam's window, and the Markov chain Monte Carlo model composition approaches. Let's set a model where the computational burden is low and we know the data generating process (population statistical

model). In particular, we set 10 regressors such that $x_k \sim N(1, 1)$, $k = 1, \dots, 6$, and $x_k \sim B(0.5)$, $k = 7, \dots, 10$. We set $\beta = [1 \ 0 \ 0 \ 0 \ 0.5 \ 0, 0, 0, 0, -0.7]^\top$ such that just x_1 , x_5 and x_{10} are relevant to drive $y_i = 1 + \mathbf{x}^\top \beta + \mu_i$, $\mu_i \sim N(0, 0.5^2)$. Observe that we just have $2^{10} = 1024$ models in this setting, thus, we can calculate the posterior model probability for each model.

Our GUI uses the commands *bicreg* and *MC3.REG* from the package *BMA* to perform Bayesian model average in the linear regression model using the BIC approximation and MC3, respectively. These commands in turn are based on [168] and [172]. The following code shows how to perform the simulation and get the posterior mean and standard deviation using these commands with the default values of hyperparameters and tuning parameters.

R code. Simulation exercise: Bayesian model average, small setting

```

1 rm(list = ls()); set.seed(010101)
2 N <- 1000
3 K1 <- 6; K2 <- 4; K <- K1 + K2
4 X1 <- matrix(rnorm(N*K1, 1, 1), N, K1)
5 X2 <- matrix(rbinom(N*K2, 1, 0.5), N, K2)
6 X <- cbind(X1, X2); e <- rnorm(N, 0, 0.5)
7 B <- c(1, 0, 0, 0, 0.5, 0, 0, 0, 0, -0.7)
8 y <- 1 + X %*% B + e
9 BMAglm <- BMA::bicreg(X, y, strict = FALSE, OR = 50)
10 summary(BMAglm)

```

We can see from the results that the BIC approximation with the Occam's window, and the MC3 algorithm perform a good job finding the relevant regressors, and their posterior BMA means are very close to the population values. We also see that the BMA results are very similar in the two approaches.

R code. Simulation exercise: Bayesian model average, small setting

```

1 BMAREG <- BMA::MC3.REG(y, X, num.its=500)
2 Models <- unique(BMAREG[["variables"]])
3 nModels <- dim(Models)[1]
4 nVistModels <- dim(BMAREG[["variables"]])[1]
5 PMP <- NULL
6 for(m in 1:nModels){
7   idModm <- NULL
8   for(j in 1:nVistModels){
9     if(sum(Models[m,] == BMAREG[["variables"]][j,]) == K){
10       idModm <- c(idModm, j)
11     }else{
12       idModm <- idModm
13     }
14   }
15   PMPm <- sum(BMAREG[["post.prob"]][idModm])
16   PMP <- c(PMP, PMPm)
17 }
18 PIP <- NULL
19 for(k in 1:K){
20   PIPk <- sum(PMP[which(Models[,k] == 1)])
21   PIP <- c(PIP, PIPk)
22 }
23 plot(PIP)
24 Means <- matrix(0, nModels, K)
25 Vars <- matrix(0, nModels, K)
26 for(m in 1:nModels){
27   idXs <- which(Models[m,] == 1)
28   if(length(idXs) == 0){
29     Regm <- lm(y ~ 1)
30   }else{
31     Xm <- X[, idXs]
32     Regm <- lm(y ~ Xm)
33     SumRegm <- summary(Regm)
34     Means[m, idXs] <- SumRegm[["coefficients"]][-1,1]
35     Vars[m, idXs] <- SumRegm[["coefficients"]][-1,2]^2
36   }
37 }
38 BMAMEANS <- colSums(Means*PMP)
39 BMASD <- (colSums(PMP*Vars) + colSums(PMP*(Means-matrix(rep
  (BMAMEANS, each = nModels), nModels, K))^2))^0.5
40 BMAMEANS
41 [1] 1.001771e+00 -5.322016e-05 6.635422e-06 3.721457e-07
42 [6] 4.976335e-01
42 [6] -1.271339e-04 1.000932e-08 2.107441e-05 6.578654e-06
42 [6] -7.035557e-01
43 BMASD
44 [1] 1.527261e-02 1.353624e-03 5.936816e-04 1.163947e-04
44 [1] 1.566698e-02 1.987360e-03
45 [7] 2.778896e-05 1.270579e-03 6.997305e-04 3.093389e-02
46 BMAMEANS/BMASD

```

We can perform Bayesian model averaging in our GUI for linear Gaussian models using the BIC approximation and MC3 using Algorithms A27 and A28, respectively. We ask in Exercise 2 to perform BMA using the dataset *10ExportDiversificationHHI.csv* from [108].

Algorithm A27 Bayesian model average in linear Gaussian models using the Bayesian information criterion

- 1: Select *Bayesian Model Averaging* on the top panel
 - 2: Select *Normal data* model using the left radio button
 - 3: Select *BIC* using the right radio button under **Which type do you want to perform?**
 - 4: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 5: Type the *OR* number of the Occam's window in the box under **OR: Number between 5 and 50**, this is not necessary as by default there is 50
 - 6: Click the *Go!* button
 - 7: Analyze results: After a few seconds or minutes, a table appears showing, for each regressor in the dataset, the PIP (posterior inclusion probability, **p!=0**), the BMA posterior mean (**EV**), the BMA standard deviation (**SD**), and the posterior mean for models with the highest PMP. At the bottom of the table, for the models with the largest PMP, the number of variables (**nVar**), the coefficient of determination (**r2**), the BIC, and the PMP (**post prob**) are displayed
 - 8: Download posterior results using the *Download results using BIC*. There are two files, the first has the best models by row according to the PMP (last column) indicating with a 1 inclusion of the variable (0 indicates no inclusion), and the second file has the PIP, the BMA expected value and standard deviation for each variable in the dataset
-

We show in the following code how to program a MC3 algorithm from scratch to perform BMA using the setting from Section 10.2. The first part of the code is the function to calculate the log marginal likelihood. This is a small simulation setting, thus we can calculate the marginal likelihood for all 1024 models, and then calculate the posterior model probability standardizing using the model with the largest log marginal likelihood. We see from the results that this model is the data generating process (population statistical model). We also find that the posterior inclusion probabilities for x_1 , x_5 and x_{10} are 1, whereas the PIP for the other variables are less than 0.05. Although BMA allows incorporating model uncertainty in a regression framework, sometimes it is desirable to select just one model. Two compelling alternatives are the model with the largest posterior model probability, and the median probability model. The latter is the model which includes every predictor that has posterior inclusion probability higher than 0.5. The first model is the best

Algorithm A28 Bayesian model average in linear Gaussian models using Markov chain Monte Carlo model composition

- 1: Select *Bayesian Model Averaging* on the top panel
 - 2: Select *Normal data* model using the left radio button
 - 3: Select *MC3* using the right radio button under **Which type do you want to perform?**
 - 4: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 5: Select MC3 iterations using the *Range slider* under the label **MC3 iterations:**
 - 6: Click the *Go!* button
 - 7: Analyze results: After a few seconds or minutes, a table appears showing, for each regressor in the dataset, the PIP (posterior inclusion probability, **p!=0**), the BMA posterior mean (**EV**), the BMA standard deviation (**SD**), and the posterior mean for models with the highest PMP. At the bottom of the table, for the models with the largest PMP, the number of variables (**nVar**), the coefficient of determination (**r2**), the BIC, and the PMP (**post prob**) are displayed
 - 8: Download posterior results using the *Download results using BIC*. There are two files, the first has the best models by row according to the PMP (last column) indicating with a 1 inclusion of the variable (0 indicates no inclusion), and the second file has the PIP, the BMA expected value and standard deviation for each variable in the dataset
-

alternative for prediction in the case of a 0–1 loss function [46], whereas the second is the best alternative when there is a quadratic loss function in prediction [8]. In this simulation, the two criteria indicate selection of the data generating process.

We also show how to estimate the posterior mean and standard deviation based on BMA. We see that the posterior means are very close to the population parameters.

R code. Simulation exercise: Bayesian model average, small setting from scratch

```

1 LogMLfun <- function(Model){
2   indr <- Model == 1
3   kr <- sum(indr)
4   if(kr > 0){
5     gr <- ifelse(N > kr^2, 1/N, kr^(-2))
6     Xr <- matrix(Xnew[, indr], ncol = kr)
7     PX <- Xr%*%solve(t(Xr)%*%Xr)%*%t(Xr)
8     s2pos <- c((t(y - mean(y))%*%(y - mean(y))) - t(y)%*%PX%
9      *y/(1 + gr))
10    mllMod <- (kr/2)*log(gr/(1+gr))-(N-1)/2*log(s2pos)
11  }else{
12    gr <- ifelse(N > kr^2, 1/N, kr^(-2))
13    s2pos <- c((t(y - mean(y))%*%(y - mean(y))))
14    mllMod <- (kr/2)*log(gr/(1+gr))-(N-1)/2*log(s2pos)
15  }
16  return(mllMod)
17}
18 combs <- expand.grid(c(0,1), c(0,1), c(0,1), c(0,1), c(0,1),
19   c(0,1), c(0,1), c(0,1), c(0,1), c(0,1))
20 Xnew <- apply(X, 2, scale)
21 mll <- sapply(1:2^K, function(s){LogMLfun(matrix(combs[s,],
22   1, K))})
23 MaxPMP <- which.max(mll); StMarLik <- exp(mll-max(mll))
24 PMP <- StMarLik/sum(StMarLik)
25 PMP[MaxPMP]
26 combs[MaxPMP,]
27 Var1 Var2 Var3 Var4 Var5 Var6 Var7 Var8 Var9 Var10
28 530 1 0 0 0 1 0 0 0 0 1
29 PIP <- NULL
30 for(k in 1:K){
31   PIPk <- sum(PMP[which(combs[,k] == 1)]); PIP <- c(PIP,
32   PIPk)
33 }
34 PIP
35 [1] 1.00000000 0.03617574 0.03208369 0.03516743 1.00000000
36      0.04795509 0.03457102 0.03468819 0.03510209 1.00000000
37 nModels <- dim(combs)[1]; Means <- matrix(0, nModels, K)
38 Vars <- matrix(0, nModels, K)
39 for(m in 1:nModels){
40   idXs <- which(combs[m,] == 1)
41   if(length(idXs) == 0){
42     Regm <- lm(y ~ 1)
43   }else{
44     Xm <- X[, idXs]; Regm <- lm(y ~ Xm)
45     SumRegm <- summary(Regm)
46     Means[m, idXs] <- SumRegm[["coefficients"]][,-1]
47     Vars[m, idXs] <- SumRegm[["coefficients"]][,-2]^2
48   }
49 }
50 BMAMeans <- colSums(Means*PMP)
51 1.0018105888 -0.0003196423 0.0001489711 0.0002853524
52      0.4976225353 -0.0007229563 0.0005342718 0.0005441905
53      0.0005758708 -0.7035206822
54 BMASd <- (colSums(PMP*Vars) + colSums(PMP*(Means-matrix(rep(
55   (BMAMeans, each = nModels), nModels, K))^2)))^0.5
56 0.015274980 0.003304115 0.002814491 0.003214722 0.015668278
57      0.004694003 0.006400541 0.006435695 0.006528471
58      0.030940753

```

The following part of the code shows how to perform the MC3 algorithm. This algorithm is not necessary in this case due to being a small dimensional problem, but it helps as a pedagogical exercise. The point of departure is to set $S = 100$ random models, and order their log marginal likelihoods. Thus, the logic of the algorithm is to pick the worse model among the S models, and propose a candidate model to compete against it. We repeat this MC3 iterations (1000 in the code). Observe that 1000 iterations is less than the number of potential models (1024). This is the idea of the MC3 algorithm, that is, performing less iterations than the number of elements of the space of models.

In our algorithm, we analyze all model scenarios using different conditionals and reasonably assume the same prior model probability for all models and the same cardinality for both the actual and candidate models. We can calculate the posterior model probability (PMP) in different ways. One way is to recover the unique models from the final set of S models, calculate the log marginal likelihood for these models, and standardize using the best model among them. Another way is to calculate the PMP using the complete set of S final models, accounting for the fact that the same model can appear multiple times in this set, thus requiring us to sum the PMPs of repeated models. An additional way is to calculate the PMP using the relative frequency with which a model appears in the final set of S models. These three methods can yield different PMP, particularly when the number of MC3 iterations is small. In our setting using 1000 MC3 iterations, the data generating process got the largest PMP in the three ways to calculate the PMP.

A remarkable point in this algorithm is that we can get just one model after substantially increasing the number of iterations (try this code using 10000 iterations). This can be a good feature if we require just one model. However, this neglects model uncertainty, which can be a desirable characteristic. We ask to program an algorithm where we end up with S different models after finishing the MC3 iterations (Exercise 3).

R code. Simulation exercise: Bayesian model average, small setting from scratch

```

1 M <- 100
2 Models <- matrix(rbinom(K*M, 1, p = 0.5), ncol=K, nrow = M)
3 mllnew <- sapply(1:M, function(s){LogMLfunt(matrix(Models[s
   ], 1, K))})
4 oind <- order(mllnew, decreasing = TRUE)
5 mllnew <- mllnew[oind]; Models <- Models[oind, ]; iter <-
   1000
6 pb <- winProgressBar(title = "progress bar", min = 0, max =
   iter, width = 300); s <- 1
7 while(s <= iter){
8   ActModel <- Models[M,]; idK <- which(ActModel == 1)
9   Kact <- length(idK)
10  if(Kact < K & Kact > 1){
11    CardMol <- K; opt <- sample(1:3, 1)
12    if(opt == 1){ # Same
13      CandModel <- ActModel
14    }else{
15      if(opt == 2){ # Add
16        All <- 1:K; NewX <- sample(All[-idK], 1)
17        CandModel <- ActModel; CandModel[NewX] <- 1
18      }else{ # Subtract
19        LessX <- sample(idK, 1); CandModel <- ActModel
20        CandModel[LessX] <- 0
21      }
22    }
23  }else{
24    CardMol <- K + 1
25    if(Kact == K){
26      opt <- sample(1:2, 1)
27      if(opt == 1){ # Same
28        CandModel <- ActModel
29      }else{ # Subtract
30        LessX <- sample(1:K, 1); CandModel <- ActModel
31        CandModel[LessX] <- 0
32      }
33    }else{
34      if(K == 1){
35        opt <- sample(1:3, 1)
36        if(opt == 1){ # Same
37          CandModel <- ActModel
38        }else{
39          if(opt == 2){ # Add
40            All <- 1:K; NewX <- sample(All[-idK], 1)
41            CandModel <- ActModel; CandModel[NewX] <- 1
42          }else{ # Subtract
43            LessX <- sample(idK, 1); CandModel <- ActModel
44            CandModel[LessX] <- 0
45          }
46        }
47      }else{ # Add
48        NewX <- sample(1:K, 1); CandModel <- ActModel
49        CandModel[NewX] <- 1
50      }
51    }
52  }

```

R code. Simulation exercise: Bayesian model average, small setting from scratch

```

1  LogMLact <- LogMLfunt(matrix(ActModel, 1, K))
2  LogMLcand <- LogMLfunt(matrix(CandModel, 1, K))
3  alpha <- min(1, exp(LogMLcand-LogMLact))
4  u <- runif(1)
5  if(u <= alpha){
6      mllnew[M] <- LogMLcand; Models[M, ] <- CandModel
7      oind <- order(mllnew, decreasing = TRUE)
8      mllnew <- mllnew[oind]; Models <- Models[oind, ]
9  }else{
10     mllnew <- mllnew; Models <- Models
11 }
12 s <- s + 1
13 setWinProgressBar(pb, s, title=paste( round(s/iter*100, 0)
14 , "% done"))
15 }
16 close(pb)
17 ModelsUni <- unique(Models)
18 mllnewUni <- sapply(1:dim(ModelsUni)[1], function(s){
19     LogMLfunt(matrix(ModelsUni[s,], 1, K)))}
20 StMarLik <- exp(mllnewUni-mllnewUni[1])
21 PMP <- StMarLik/sum(StMarLik) # PMP based on unique selected
22 models
23 nModels <- dim(ModelsUni)[1]
24 StMarLik <- exp(mllnew-mllnew[1])
25 PMPold <- StMarLik/sum(StMarLik) # PMP all selected models
26 PMPot <- NULL
27 PMPap <- NULL
28 FreqMod <- NULL
29 for(m in 1:nModels){
30     idModm <- NULL
31     for(j in 1:M){
32         if(sum(ModelsUni[m, ] == Models[j,]) == K){
33             idModm <- c(idModm, j)
34         }else{
35             idModm <- idModm
36         }
37     }
38     PMPm <- sum(PMPold[idModm]) # PMP unique models using sum
39     of all selected models
40     PMPot <- c(PMPot, PMPm)
41     PMPapm <- length(idModm)/M # PMP using relative frequency
42     in all selected models
43     PMPap <- c(PMPap, PMPapm)
44     FreqMod <- c(FreqMod, length(idModm))
45 }

```

An important issue to account for regressors (model) uncertainty in the identification of causal effects, rather than finding good predictors (association relationships), is endogeneity. Thus, we also implement the instrumental variable approach of Section 7.3 to tackle this issue in BMA. We assume that $\gamma \sim N(\mathbf{0}, \mathbf{I})$, $\beta \sim N(\mathbf{0}, \mathbf{I})$, and $\Sigma^{-1} \sim W(3, \mathbf{I})$ [118].

[134] propose an algorithm based on conditional Bayes factors [55] that allows embedding MC3 within a Gibbs sampling algorithm. Given the candidate (M_c^{2nd}) and actual (M_{s-1}^{2nd}) models for the iteration s in the second stage, the conditional Bayes factor is

$$CBF^{2nd} = \frac{p(\mathbf{y}|M_c^{2nd}, \gamma, \Sigma)}{p(\mathbf{y}|M_{s-1}^{2nd}, \gamma, \Sigma)},$$

where

$$p(\mathbf{y}|M_c^{2nd}, \gamma, \Sigma) = \int_{\mathcal{M}^{2nd}} p(\mathbf{y}|\beta, \gamma, \Sigma) \pi(\beta|M_c^{2nd}) d\beta \propto |\mathbf{B}_n|^{1/2} \exp\left\{\frac{1}{2} \beta_n^\top \mathbf{B}_n^{-1} \beta_n\right\}.$$

In the first stage,

$$CBF^{1st} = \frac{p(\mathbf{y}|M_c^{1st}, \beta, \Sigma)}{p(\mathbf{y}|M_{s-1}^{1st}, \beta, \Sigma)},$$

where

$$p(\mathbf{y}|M_c^{1st}, \beta, \Sigma) = \int_{\mathcal{M}^{1st}} p(\mathbf{y}|\gamma, \beta, \Sigma) \pi(\gamma|M_c^{1st}) d\gamma \propto |\mathbf{G}_n|^{1/2} \exp\left\{\frac{1}{2} \gamma_n^\top \mathbf{G}_n^{-1} \gamma_n\right\}.$$

These conditional Bayes factors assume $\pi(M^{1st}, M^{2nd}) \propto 1$. See [134] for more details of the instrumental variable BMA algorithm.⁵

We perform instrumental variable BMA in our GUI using the package *iwbma*. The Algorithm A29 shows how to perform this in our GUI.

Let's perform a simulation exercise to assess the performance of the instrumental variable BMA to uncover the data generating process in presence of endogeneity.

Example: Simulation exercise

Let's assume that $y_i = 2 + 0.5x_{i1} - x_{i2} + x_{i3} + \mu_i$ where $x_{i1} = 4z_{i1} - z_{i2} + 2z_{i3} + \epsilon_{i1}$ and $x_{i2} = -2z_{i1} + 3z_{i2} - z_{i3} + \epsilon_{i2}$ such that $[\epsilon_{i1} \ \epsilon_{i2} \ \mu_i]^\top \sim N(\mathbf{0}, \Sigma)$

where $\Sigma = \begin{bmatrix} 1 & 0 & 0.8 \\ 0 & 1 & 0.5 \\ 0.8 & 0.5 & 1 \end{bmatrix}$, $i = 1, 2, \dots, 1000$. The endogeneity is due to

the correlation between μ_i and x_{i1} and x_{i2} through the stochastic errors. In addition, there are three instruments, $z_{il} \sim U(0, 1)$, $l = 1, 2, 3$, and another 18 regressors believed to influence y_i , which are distributed according to a standard normal distribution.

⁵[124] and [133] propose other frameworks for BMA taking into account endogeneity.

Algorithm A29 Instrumental variable Bayesian model average in linear Gaussian models

- 1: Select *Bayesian Model Averaging* on the top panel
 - 2: Select *Normal data* model using the left radio button
 - 3: Select *Instrumental variable* using the right radio button under **Which type do you want to perform?**
 - 4: Upload the dataset containing the dependent variable, endogenous regressors, and exogenous regressors including the constant (see Section 5.6 for details). User should select first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 5: Upload the dataset containing the instruments (see Section 5.6 for details). User should select first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File (Instruments)** legend
 - 6: Write down the number of endogenous regressors in the box labeled **Number of Endogenous variables**
 - 7: Select MCMC iterations and burn-in using the *Range slider* under the labels **MCMC iterations:** and **Burn-in Sample:**
 - 8: Click the *Go!* button
 - 9: Analyze results: After a few seconds or minutes, two tables appear showing, for each regressor in the dataset, the PIP (posterior inclusion probability, **p!=0**), and the BMA posterior mean (**EV**). The top table shows the results of the second stage (main equation), and the bottom table shows the results of the first stage (auxiliary equations)
 - 10: Download posterior results using the *Download results using IV*. There are three files, the first file has the posterior inclusion probabilities of each variable, and the BMA posterior means of the coefficients in the first stage equations, the second file shows these results for the second stage (main equation), and the third file has the posteriors chains of all parameters by iteration.
-

The following code shows how to perform IV BMA using the *ivbma* package. We see from the results that the PIP of x_{i1} , x_{i2} , intercept and x_{i3} are equal to 1, whereas the remaining PIP are close to 0. In addition, the BMA means are also close to the population values. The PIP of the first stage equations, as well as their BMA posterior means, are very close to the populations values. The same happens with the covariance matrix.

We ask in Exercise 4 to perform BMA based on the BIC approximation and MC3 in this simulation setting. In addition, we ask in Exercise 5 to use the datasets *11ExportDiversificationHHI.csv* and *12ExportDiversificationHHIInstr.csv* to perform IV BMA assuming that the log of per capita gross domestic product is endogenous (*avgldpcap*). See [108] for details.

R code. Simulation exercise: Instrumental variable Bayesian model average

```

1 rm(list = ls())
2 set.seed(010101)
3 simIV <- function(delta1,delta2,beta0,betas1,betas2,beta2,
4 Sigma,n,z) {
5   eps <- matrix(rnorm(3*n),ncol=3) %*% chol(Sigma)
6   xs1 <- z%*%delta1 + eps[,1]
7   xs2 <- z%*%delta2 + eps[,2]
8   x2 <- rnorm(dim(z)[1])
9   y <- beta0+betas1*xs1+betas2*xs2+beta2*x2 + eps[,3]
10  X <- as.matrix(cbind(xs1,xs2,1,x2))
11  colnames(X) <- c("x1en","x2en","cte","xex")
12  y <- matrix(y,dim(z)[1],1)
13  colnames(y) <- c("y")
14  list(X=X,y=y)
15 }
16 n <- 1000 ; p <- 3
17 z <- matrix(runif(n*p),ncol=p)
18 rho31 <- 0.8; rho32 <- 0.5;
19 Sigma <- matrix(c(1,0,rho31,0,1,rho32,rho31,rho32,1),ncol=3)
20 delta1 <- c(4,-1,2); delta2 <- c(-2,3,-1); betas1 <- .5;
21       betas2 <- -1; beta2 <- 1; beta0 <- 2
22 simiv <- simIV(delta1,delta2,beta0,betas1,betas2,beta2,Sigma
23 ,n,z)
24 nW <- 18
25 W <- matrix(rnorm(nW*dim(z)[1]),dim(z)[1],nW)
26 YXW<-cbind(simiv$y, simiv$X, W)
27 y <- YXW[,1]; X <- YXW[,2:3]; W <- YXW[,-c(1:3)]
28 S <- 10000; burnin <- 1000
29 regivBMA <- ivbma::ivbma(Y = y, X = X, Z = z, W = W, s = S+
30   burnin, b = burnin, odens = S, print.every = round(S/10)
31   , run.diagnostics = FALSE)
32 PIPmain <- regivBMA[["L.bar"]] # PIP outcome
33 PIPmain
34 1.0000 1.0000 1.0000 1.0000 0.0125 0.0382 0.0145 0.0148
35 0.0136 0.0102 0.0070 0.0527 0.0014 0.0077 0.0211 0.0081
36 0.0047 0.0141 0.0028 0.0063 0.0072 0.0220
37 EVmain <- regivBMA[["rho.bar"]] # Posterior mean outcome
38 EVmain
39 5.105361e-01 -9.828459e-01 1.996885e+00 1.005497e+00
40 -1.700857e-04 9.946613e-04 1.086717e-04 -1.448951e-04
41 1.532812e-04 1.356334e-04 -6.027285e-05 9.119699e-04
42 -1.581408e-05 1.050517e-04 2.488002e-04 -6.229493e-05
43 4.292825e-05 3.371366e-05 5.345760e-06 5.933764e-05
44 5.066236e-05 1.516718e-04
45 PIPaux <- regivBMA[["M.bar"]] # PIP auxiliary
46 EVaux <- regivBMA[["lambda.bar"]] # Posterior mean auxiliary
47 plot(EVaux[,1])
48 plot(EVaux[,2])
49 EVsigma <- regivBMA[["Sigma.bar"]] # Posterior mean variance
50 matrix

```

10.3 Generalized linear models

Generalized linear models (GLM) were introduced by [156], and extend the concept of linear regressions to a more general setting. These models are characterized by: i) a dependent variable y_i whose probability distribution function belongs to the exponential family (see Section 3.1), ii) a linear predictor $\eta = \mathbf{x}^\top \boldsymbol{\beta}$, and iii) a link function such that $\mathbb{E}[Y|\mathbf{x}] = g^{-1}(\mathbf{x}^\top \boldsymbol{\beta})$, which implies that $g(\mathbb{E}[Y|\mathbf{x}]) = \mathbf{x}^\top \boldsymbol{\beta}$. GLM can be extended to overdispersed exponential family [148].

We know from Section 3.1 that the Poisson distribution belongs to the exponential family such that $p(y|\lambda) = \frac{\exp(-\lambda)\exp(y\log(\lambda))}{y!}$ or in the canonical form $p(y|\eta) = \frac{\exp(\eta y - \exp(\eta))}{y!}$, where $\eta = \log(\lambda)$, which means that $\mathbf{x}^\top \boldsymbol{\beta} = \log(\lambda)$, and consequently, $\mathbb{E}[Y|\mathbf{x}] = \nabla(\exp(\eta)) = \exp(\eta) = \lambda = \exp(\mathbf{x}^\top \boldsymbol{\beta})$. Then, the link function in the Poisson case is the *log* function. We ask in Exercise 6 to show that the link function in Bernoulli case is the *logit* function. Another examples are the identity function in the case of the Gaussian distribution, and the negative inverse in the case of the gamma density.

We can use the setting of the GLM to perform BMA using the BIC approximation following [168]. In particular, $BIC = k_m \log(N) - 2 \log(p(\hat{\boldsymbol{\theta}}_m|\mathbf{y}))$, where $\hat{\boldsymbol{\theta}}_m$ is the maximum likelihood estimator. Thus, we just need to calculate the likelihood function at the maximum likelihood estimator.

Example: Simulation exercises

Let's perform some simulation exercises to assess the performance of the BIC approximation using the Occam's window in GLMs. There are 27 regressors, where x_{i1} and x_{i2} are just the relevant regressors in all exercises, $i = 1, 2, \dots, 1000$.

- Logit: $x_k \sim N(0, 1)$, $k = 1, \dots, 27$, and $P(Y_i = 1|\mathbf{x}_i) = \exp(0.5 + 0.8x_{i1} - 1.2x_{i2}) / (1 + \exp(0.5 + 0.8x_{i1} - 1.2x_{i2}))$.
- Gamma: $x_k \sim N(0, 0.5^2)$, $k = 1, \dots, 27$, and $y_i \sim G(\alpha, \delta)$ where $\alpha = -(0.5 + 0.2x_{i1}0.1x_{i2})^{-1}$ and $\delta = 1$.
- Poisson: $x_k \sim N(0, 1)$, $k = 1, \dots, 27$, and $\mathbb{E}[Y_i|\mathbf{x}_i] = \lambda_i = \exp(0.5 + 1.1x_{i1} + 0.7x_{i2})$.

Our GUI uses the command *bic.glm* from the *BMA* package to perform BMA using the BIC approximation with the Occam's window in GLMs. The Algorithm A30 shows how to do this in our GUI, and the following code shows how to perform BMA in logit models using the simulation setting.

The results show that the PIPs of x_{i1} and x_{i2} are equal 1 in all three settings, the data generating process gets the highest PMP, and the BMA posterior means are close to the population values in each simulation setting.

Algorithm A30 Bayesian model average in generalized linear models using the Bayesian information criterion

- 1: Select *Bayesian Model Averaging* on the top panel
 - 2: Select the generalized linear model using the left radio button. Options: *Binomial data (Logit)*, *Real positive data (Gamma)* and *Count data (Poisson)*
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Type the *OR* number of the Occam's window in the box under **OR: Number between 5 and 50**, this is not necessary as by default there is 50
 - 5: Type the *OL* number of the Occam's window in the box under **OL: Number between 0.0001 and 1**, this is not necessary as by default there is 0.0025
 - 6: Click the *Go!* button
 - 7: Analyze results: After a few seconds or minutes, a table appears showing, for each regressor in the dataset, the PIP (posterior inclusion probability, **p!=0**), the BMA posterior mean (**EV**), the BMA standard deviation (**SD**), and the posterior mean for models with the highest PMP. At the bottom of the table, for the models with the largest PMP, the number of variables (**nVar**), the BIC, and the PMP (**post prob**) are displayed
 - 8: Download posterior results using the *Download results using BIC*. There are two files, the first has the best models by row according to the PMP (last column) indicating with a 1 inclusion of the variable (0 indicates no inclusion), and the second file has the PIP, the BMA expected value and standard deviation for each variable in the dataset
-

The other variables get PIPs close to 0, except a few exceptions, and the BMA posterior means are also close to 0. This suggests that the BIC approximation does a good job finding the data generating process in generalized linear models.

R code. Simulation exercise: BMA for generalized linear models

```

1 ##### Logit #####
2 rm(list = ls()); set.seed(010101)
3 n<-1000; B<-c(0.5,0.8,-1.2)
4 X<-matrix(cbind(rep(1,n),rnorm(n,0,1),rnorm(n,0,1)),n,length(B))
5 p <- exp(X%*%B)/(1+exp(X%*%B)); y <- rbinom(n, 1, p)
6 nXgar<-25; Xgar<-matrix(rnorm(nXgar*n),n,nXgar)
7 df<-as.data.frame(cbind(y,X[,-1],Xgar))
8 colnames(df) <- c("y", "x1", "x2", "x3", "x4", "x5", "x6", "x7",
9 "x8", "x9", "x10", "x11", "x12", "x13", "x14", "x15", "x16", "x17",
10 "x18", "x19", "x20", "x21", "x22", "x23", "x24", "x25", "x26", "x27")
11 BMAglmLogit <- BMA::bic.glm(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+
12 x10+x11+x12+x13+x14+x15+x16+x17+x18+x19+x20+x21+x22+x23+
13 x24+x25+x26+x27, data = df, glm.family = binomial(link =
14 "logit"), strict = FALSE, OR = 50)
15 summary(BMAglmLogit)
16 ##### Gamma #####
17 rm(list = ls()); set.seed(010101)
18 n<-1000; B<- c(0.5, 0.2, 0.1)
19 X<-matrix(cbind(rep(1,n),rnorm(n,0,0.5),rnorm(n,0,0.5)),n,
20 length(B))
21 y1 <- (X%*%B)^(-1)
22 y <- rgamma(n,y1,scale=1)
23 nXgar<-25; Xgar<-matrix(rnorm(nXgar*n),n,nXgar)
24 df<-as.data.frame(cbind(y,X[,-1],Xgar))
25 colnames(df) <- c("y", "x1", "x2", "x3", "x4", "x5", "x6", "x7",
26 "x8", "x9", "x10", "x11", "x12", "x13", "x14", "x15",
27 "x16", "x17", "x18", "x19", "x20", "x21", "x22", "x23",
28 "x24", "x25", "x26", "x27")
29 BMAglmGamma <- BMA::bic.glm(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+
30 x10+x11+x12+x13+x14+x15+x16+x17+x18+x19+x20+x21+x22+x23+
31 x24+x25+x26+x27, data = df, glm.family = Gamma(link =
32 "inverse"), strict = FALSE, OR = 50)
33 summary(BMAglmGamma)
34 ##### Poisson #####
35 rm(list = ls()); set.seed(010101)
36 n<-1000; B<-c(2,1.1,0.7)
37 X<-matrix(cbind(rep(1,n),rnorm(n,0,1),rnorm(n,0,1)),n,length(B))
38 y1<-exp(X%*%B); y<-rpois(n,y1)
39 nXgar<-25; Xgar<-matrix(rnorm(nXgar*n),n,nXgar)
40 df<-as.data.frame(cbind(y,X[,-1],Xgar))
41 colnames(df) <- c("y", "x1", "x2", "x3", "x4", "x5", "x6", "x7",
42 "x8", "x9", "x10", "x11", "x12", "x13", "x14", "x15",
43 "x16", "x17", "x18", "x19", "x20", "x21", "x22", "x23",
44 "x24", "x25", "x26", "x27")
45 BMAglmPoisson <- BMA::bic.glm(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+
46 x10+x11+x12+x13+x14+x15+x16+x17+x18+x19+x20+x21+x22+x23+
47 x24+x25+x26+x27, data = df, glm.family = poisson(link =
48 "log"), strict = FALSE, OR = 50)
49 summary(BMAglmPoisson)

```

We can take advantage of the *glm* function in **R** to perform BMA programming a MC3 algorithm. The following code shows how to do this. First, we simulate the data, second we have a function to get the log marginal likelihood approximation using the results from the *glm* function. Then, we have the initial models to begin the MC3 algorithm. After this, we have the MC3 algorithm using small modifications of the code that we use to perform MC3 in Gaussian linear models. We can calculate the PMPs, PIPs, BMA means and standard deviations as we previously did.

The simulation setting implies 2^{27} models, which implies approximately 135 million models in the model space. We run our MC3 algorithm using the BIC approximation with 50000 iterations. This takes by far more time than the BIC approximation from the *BMA* package, but it seems to do a good job finding the data generating process as the PMP of this model is equal 1, the posterior inclusion probabilities are equal 1 for x_{i1} and x_{i2} , the posterior means are 1.1 and 0.7, that is, equal to the population values, and the t-ratios are by far higher than 2. However, running 50000 iterations implies mass concentration in one model, in this case the data generating process. If we run 25000 MC3 iterations, the highest PMP is 0.8, but it is not associated with the data generating process. Although, the PIP is equal 1 for x_{i1} and x_{i2} , but there are other regressors that get high PIPs. The BMA means are equal to the population values for x_{i1} and x_{i2} , and the PIPs for the other regressors are equal 0. The t-ratios of the regressors in the population statistical model are larger than 2 by far, whereas the t-ratios of the other regressors are equal to 0. This exercise shows that 25000 iterations were not enough to uncover the data generating process. However, this exercise also shows an important point, we need to analyze all the relevant results from the BMA analysis, no just the PMPs and/or PIPs.

We ask in Exercise 10 to use this approach to perform a BMA algorithm in the logit regression using the simulation setting of logit models of this section.

R code. Simulation exercise: BMA for generalized linear models using MC3 from scratch

```

1 rm(list = ls()); set.seed(010101)
2 n<-1000; B<-c(2,1.1,0.7)
3 X<-matrix(cbind(rep(1,n), rnorm(n,0,1), rnorm(n,0,1)),n,length(B))
4 y1<-exp(X%*%B); y<-rpois(n,y1)
5 nXgar<-25; Xgar<-matrix(rnorm(nXgar*n),n,nXgar)
6 df<-as.data.frame(cbind(y,X[,-1],Xgar))
7 colnames(df) <- c("y", "x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10", "x11", "x12", "x13", "x14", "x15", "x16", "x17", "x18", "x19", "x20", "x21", "x22", "x23", "x24", "x25", "x26", "x27")
8 Xnew <- apply(df[,-1], 2, scale)
9 BICfunt <- function(Model){
10   indr <- Model == 1; kr <- sum(indr)
11   if(kr > 0){
12     Xr <- as.matrix(Xnew[, indr])
13     model <- glm(y ~ Xr, family = poisson(link = "log"))
14     model_bic <- BIC(model)
15     mllMod <- -model_bic/2
16   }else{
17     model <- glm(y ~ 1, family = poisson(link = "log"))
18     model_bic <- BIC(model); mllMod <- -model_bic/2
19   }
20   return(mllMod)
21 }
22 M <- 500; K <- dim(df)[2] - 1
23 Models <- matrix(rbinom(K*M, 1, p = 0.5), ncol = K, nrow = M)
24 mllnew <- sapply(1:M, function(s){BICfunt(matrix(Models[s,], 1, K))})
25 oind <- order(mllnew, decreasing = TRUE)
26 mllnew <- mllnew[oind]; Models <- Models[oind, ]
27 # Hyperparameters MC3
28 iter <- 25000
29 pb <- winProgressBar(title = "progress bar", min = 0, max =
  iter, width = 300)
30 s <- 1
31 while(s <= iter){
32   ActModel <- Models[M,]
33   idK <- which(ActModel == 1)
34   Kact <- length(idK)
35   if(Kact < K & Kact > 1){
36     CardMol <- K
37     opt <- sample(1:3, 1)
38     if(opt == 1){ # Same
39       CandModel <- ActModel
40     }else{
41       if(opt == 2){ # Add
42         All <- 1:K
43         NewX <- sample(All[-idK], 1)
44         CandModel <- ActModel
45         CandModel[NewX] <- 1
46     }else{ # Subtract
47       LessX <- sample(idK, 1)
48       CandModel <- ActModel
49       CandModel[LessX] <- 0
50     }
51   }

```

R code. Simulation exercise: BMA for generalized linear models using MC3 from scratch

```

1  }else{
2    CardMol <- K + 1
3    if(Kact == K){
4      opt <- sample(1:2, 1)
5      if(opt == 1){ # Same
6        CandModel <- ActModel
7      }else{ # Subtract
8        LessX <- sample(1:K, 1)
9        CandModel <- ActModel
10       CandModel[LessX] <- 0
11     }
12   }else{
13     if(K == 1){
14       opt <- sample(1:3, 1)
15       if(opt == 1){ # Same
16         CandModel <- ActModel
17       }else{
18         if(opt == 2){ # Add
19           All <- 1:K
20           NewX <- sample(All[-idK], 1)
21           CandModel <- ActModel
22           CandModel[NewX] <- 1
23         }else{ # Subtract
24           LessX <- sample(idK, 1)
25           CandModel <- ActModel
26           CandModel[LessX] <- 0
27         }
28       }
29     }else{ # Add
30       NewX <- sample(1:K, 1)
31       CandModel <- ActModel
32       CandModel[NewX] <- 1
33     }
34   }
35 }
36 LogMLact <- BICfunct(matrix(ActModel, 1, K))
37 LogMLcand <- BICfunct(matrix(CandModel, 1, K))
38 alpha <- min(1, exp(LogMLcand-LogMLact))
39 u <- runif(1)
40 if(u <= alpha){
41   mllnew[M] <- LogMLcand
42   Models[M, ] <- CandModel
43   oind <- order(mllnew, decreasing = TRUE)
44   mllnew <- mllnew[oind]
45   Models <- Models[oind, ]
46 }else{
47   mllnew <- mllnew
48   Models <- Models
49 }
50 s <- s + 1
51 setWinProgressBar(pb, s, title=paste( round(s/iter*100, 0)
52 , "% done"))
53 close(pb)

```

R code. Simulation exercise: BMA for generalized linear models using MC3 from scratch

```

1 ModelsUni <- unique(Models)
2 mllnewUni <- sapply(1:dim(ModelsUni)[1], function(s){BICfunct
  (matrix(ModelsUni[s,], 1, K))})
3 StMarLik <- exp(mllnewUni-mllnewUni[1])
4 PMP <- StMarLik/sum(StMarLik) # PMP based on unique selected
  models
5 plot(PMP)
6 ModelsUni[,]
7 PIP <- NULL
8 for(k in 1:K){
9   PIPk <- sum(PMP[which(ModelsUni[,k] == 1)])
10  PIP <- c(PIP, PIPk)
11 }
12 plot(PIP)
13 Xnew <- df[,-1]
14 nModels <- dim(ModelsUni)[1]
15 Means <- matrix(0, nModels, K)
16 Vars <- matrix(0, nModels, K)
17 for(m in 1:nModels){
18   idXs <- which(ModelsUni[m,] == 1)
19   if(length(idXs) == 0){
20     Regm <- glm(y ~ 1, family = poisson(link = "log"))
21   }else{
22     Xm <- as.matrix(Xnew[, idXs])
23     Regm <- glm(y ~ Xm, family = poisson(link = "log"))
24     SumRegm <- summary(Regm)
25     Means[m, idXs] <- SumRegm[["coefficients"]][-1,1]
26     Vars[m, idXs] <- SumRegm[["coefficients"]][-1,2]^2
27   }
28 }
29 BMAmmeans <- colSums(Means*PMP)
30 BMAsd <- (colSums(PMP*Vars) + colSums(PMP*(Means-matrix(rep
  (BMAmmeans, each = nModels), nModels, K))^2))^0.5
31 plot(BMAmmeans)
32 plot(BMAsd)
33 plot(BMAmmeans/BMAsd)

```

10.4 Dynamic model averaging

In this section we show how to perform Bayesian model average in state-space models. The point of departure is the univariate random walk state-space

model (see Equations 8.1 and 8.1 in Chapter 8) conditional on model \mathcal{M}_m , $m = 1, 2, \dots, M$.

$$Y_t = \mathbf{x}_{mt}^\top \boldsymbol{\beta}_{mt} + \mu_{mt} \quad (10.1)$$

$$\boldsymbol{\beta}_{mt} = \boldsymbol{\beta}_{mt-1} + \mathbf{w}_{mt}, \quad (10.2)$$

where $\mu_{mt} \sim N(0, \sigma^2)$ and $\mathbf{w}_{mt} \sim N(\mathbf{0}, \boldsymbol{\Omega}_{mt})$.

Given $\boldsymbol{\beta}_{mt-1} | \mathbf{y}_{1:t-1} \sim N(\mathbf{b}_{mt-1}, \mathbf{B}_{mt-1})$, then, we know from Chapter 8 that $\boldsymbol{\beta}_{mt} | \mathbf{y}_{1:t-1} \sim N(\mathbf{b}_{mt-1}, \mathbf{R}_{mt})$, $\mathbf{R}_{mt} = \mathbf{B}_{mt-1} + \boldsymbol{\Omega}_{mt}$.

Specification of $\boldsymbol{\Omega}_t$ can be highly demanding. Thus, a common approach is to express $\boldsymbol{\Omega}_{mt} = \frac{1-\lambda}{\lambda} \mathbf{B}_{mt-1}$, where λ is called the *forgetting parameter* or *discount factor*, because it discounts the matrix \mathbf{B}_{mt-1} that we would have with a deterministic state evolution into the matrix \mathbf{R}_{mt} [163, Chap. 4]. This parameter is typically slightly below 1, and implies that $\mathbf{R}_{mt} = \lambda^{-1} \mathbf{B}_{mt-1}$. ($\lambda^{-1} > 1$).

[171] assume that the model changes infrequently, and its evolution is given by the transition matrix $\mathbf{T} = [t_{ml}]$, where $t_{ml} = P(\mathcal{M}_t = \mathcal{M}_m | \mathcal{M}_{t-1} = \mathcal{M}_l)$.

Then, the aim is to calculate the filtering distribution $p(\boldsymbol{\beta}_{mt}, \mathcal{M}_t | y_t) = \sum_{m=1}^M p(\boldsymbol{\beta}_{mt} | \mathcal{M}_t = \mathcal{M}_m, y_t) p(\mathcal{M}_t = \mathcal{M}_m | y_t)$. Thus, given the conditional distribution of the state at time $t - 1$, $p(\boldsymbol{\beta}_{mt-1}, \mathcal{M}_{t-1} | y_{t-1}) = \sum_{m=1}^M p(\boldsymbol{\beta}_{mt-1} | \mathcal{M}_{t-1} = \mathcal{M}_m, y_{t-1}) p(\mathcal{M}_{t-1} = \mathcal{M}_m | y_{t-1})$, where the conditional distribution of $\boldsymbol{\beta}_{mt-1}$ is approximated by a Gaussian distribution, $\boldsymbol{\beta}_{mt-1} | \mathcal{M}_{t-1} = \mathcal{M}_m, y_{t-1} \sim N(\mathbf{b}_{mt-1}, \mathbf{B}_{mt-1})$, then the first step to get the one-step-ahead predictive distribution is getting the prediction of the model indicator,

$$\begin{aligned} p(\mathcal{M}_t = \mathcal{M}_l | y_{t-1}) &= \sum_{m=1}^M p(\mathcal{M}_{t-1} = \mathcal{M}_m | y_{t-1}) \times t_{lm} \\ &\approx \frac{p(\mathcal{M}_{t-1} = \mathcal{M}_l | y_{t-1})^\delta + c}{\sum_{m=1}^M p(\mathcal{M}_{t-1} = \mathcal{M}_m | y_{t-1})^\delta + c}, \end{aligned}$$

where the second equality is used to avoid dealing with the M^2 elements of the transition matrix \mathbf{T} such that the forgetting parameter δ is used, this parameter is slightly less than 1, and $c = 0.001/M$ is introduced to handle a model probability being brought to computational zero by outliers.

Then, we get the one-step-ahead predictive distribution of the state vector, $\boldsymbol{\beta}_{mt} | \mathcal{M}_t = \mathcal{M}_m, y_{t-1} \sim N(\mathbf{b}_{mt-1}, \lambda^{-1} \mathbf{B}_{mt-1})$

Now, we consider the filtering stage, where the model filtering equation is

$$p(\mathcal{M}_t = \mathcal{M}_l | y_t) = \frac{p(\mathcal{M}_t = \mathcal{M}_l | y_{t-1}) p_l(y_t | y_{t-1})}{\sum_{m=1}^M p(\mathcal{M}_t = \mathcal{M}_m | y_{t-1}) p_m(y_t | y_{t-1})},$$

where $p_m(y_t | y_{t-1})$ is the one-step-ahead predictive distribution of $Y_t | y_{t-1}$, which is $N(f_t, Q_t)$, where $f_t = \mathbf{x}_{mt}^\top \mathbf{b}_{t-1}$ and $Q_t = \mathbf{x}_{mt}^\top \lambda^{-1} \mathbf{B}_{mt-1} \mathbf{x}_{mt} + \sigma^2$ (see Chapter 8).

The states filtering equation is $\beta_{mt} | \mathcal{M}_t = \mathcal{M}_m, y_t \sim N(\mathbf{b}_{mt}, \mathbf{B}_{mt})$ where \mathbf{b}_{mt} and \mathbf{B}_{mt} are given in the Kalman filtering recursion of Chapter 8.

[171] initiate their algorithm assuming equal prior model probabilities, and σ^2 is estimated using a recursive method of moments estimator.

We implement dynamic Bayesian model averaging in our GUI using the function *dma* from the package *dma*. Algorithm A31 shows how to perform inference using our GUI.

Algorithm A31 Dynamic Bayesian model average

- 1: Select *Bayesian Model Averaging* on the top panel
 - 2: Select *Dynamic Bayesian model average* using the left radio button
 - 3: Upload the dataset selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 4: Upload the matrix of models selecting first if there is header in the file, and the kind of separator in the *csv* file of the dataset (comma, semicolon, or tab). Then, use the *Browse* button under the **Choose File** legend
 - 5: Type the *forgetting parameters* in the boxes under **Lambda: Number slightly below 1** and **Delta: Number slightly below 1**, this is not necessary as by default there is 0.99 in both cases
 - 6: Click the *Go!* button
 - 7: Analyze results: After a few seconds or minutes, a table appears showing, for each regressor in the dataset, the dynamic Bayesian average filtering recursions for each state (**Mean** and **Standard deviation**), the posterior model probability (**PMP**), and the Bayesian model average prediction (**Prediction**)
 - 8: Download posterior results using the *Download results DBMA*. There are two files, the first has the dynamic Bayesian average filtering recursions for each state, and the second file has the PMP of each model, and the dynamic Bayesian model average prediction
-

Example: Dynamic Bayesian model average

We perform a simulation exercises where there are 8 (2^3) competing models originated from 3 regressors, $x_{tk} \sim N(0.5, 0.8^2)$, $k = 2, 3, 4$, and $\beta_1 = 0.5$, β_{2t} is a sequence from 1 to 2 in steps given by $1/T$, $\beta_{3t} = \begin{cases} -1, & 1 < t \leq 0.75T \\ 0, & 0.75T < t \leq T \end{cases}$, and $\beta_4 = 1.2$. Then, $y_t = \beta_1 + \beta_{2t}x_{2t} + \beta_{3t}x_{3t} + \beta_4x_{4t} + \mu_t$, where $\mu_t \sim N(0, 1)$, $t = 1, 2, \dots, 500$. This setting implies that during the first 75% of the period the model including all 3 regressors is the data generating process, and after this, the model with regressors 2 and 4 is the data generating process.

The following code shows the simulation exercise, and the results of the dynamic Bayesian model average setting $\lambda = \delta = 0.99$.

R code. Simulation exercise: Dynamic Bayesian model average

```

1 rm(list = ls()); set.seed(010101)
2 T <- 500; K <- 3
3 X <- matrix(rnorm(T*K, mean = 0.5, sd = 0.8), T, K)
4 combs <- expand.grid(c(0,1), c(0,1), c(0,1))
5 B1 <- 0.5; B2t <- seq(1, 2, length.out=T )
6 a <- 0.75; B3t <- c(rep(-1,round(a*T)), rep(0,round((1-a)*T)
    ))
7 B4 <- 1.2; sigma <- 1; mu <- rnorm(T, 0, sigma)
8 y <- B1 + X[,1]*B2t + X[,2]*B3t + X[,3]*B4 + mu
9 T0 <- 50
10 dma.test <- dma::dma(X, y, combs, lambda=.99, gamma=.99,
    initialperiod = T0)
11 plot(dma.test[["pmp"]][-c(1:T0),8], type = "l", col = "green"
    , main = "Posterior model probability: Model all
    regressors vs model regressors 1 and 3", xlab = "Time",
    ylab = "PMP")
12 lines(dma.test[["pmp"]][-c(1:T0),6], col = "red")
13 legend(x = 0, y = 1, legend = c("Model: All regressors", "
    Model: Regressors 1 and 3"), col = c("green", "red"),
    lty=1:1, cex=0.8)
14 require(latex2exp)
15 plot(dma.test[["thetahat.ma"]][-c(1:T0),1], type = "l", col
    = "green", main = "Bayesian model average filtering
    recursion", xlab = "Time", ylab = TeX("$\backslash\beta_{1}$"))
16 abline(h = B1, col = "red")
17 legend(x = 0, y = 0.4, legend = c("State filtering", "State
    population"), col = c("green", "red"), lty=1:1, cex=0.8)
18 plot(dma.test[["thetahat.ma"]][-c(1:T0),2], type = "l", col
    = "green", main = "Bayesian model average filtering
    recursion", xlab = "Time", ylab = TeX("$\backslash\beta_{2t}$"),
    ylim = c(0.5,2))
19 lines(B2t[-c(1:T0)], col = "red")
20 legend(x = 0, y = 0.8, legend = c("State filtering", "State
    population"), col = c("green", "red"), lty=1:1, cex=0.8)
21 plot(dma.test[["thetahat.ma"]][-c(1:T0),3], type = "l", col
    = "green", main = "Bayesian model average filtering
    recursion", xlab = "Time", ylab = TeX("$\backslash\beta_{3t}$"))
22 lines(B3t[-c(1:T0)], col = "red")
23 legend(x = 0, y = -0.4, legend = c("State filtering", "State
    population"), col = c("green", "red"), lty=1:1, cex
    =0.8)
24 plot(dma.test[["thetahat.ma"]][-c(1:T0),4], type = "l", col
    = "green", main = "Bayesian model average filtering
    recursion", xlab = "Time", ylab = TeX("$\backslash\beta_{4t}$"))
25 abline(h = B4, col = "red")
26 legend(x = 0, y = 1.3, legend = c("State filtering", "State
    population"), col = c("green", "red"), lty=1:1, cex=0.8)

```

Figure 10.1 shows the posterior model probabilities of the model with all the regressors (green line), and the model with regressors 2 and 4 (red line). In one hand, we see that the model with all regressors, which is the data generating process in the first period ($t \leq 0.75T$), gets a PMP near 1, and then its PMP decreases. On the other hand, the model with regressors 2 and 4 gets a PMP near 0 in the first part of the period, and then, its PMP gets values higher than 60% on average, when this model becomes the data generating process. These results suggest than in this particular simulation exercise the dynamic Bayesian model average works relatively well calculating the PMPs.

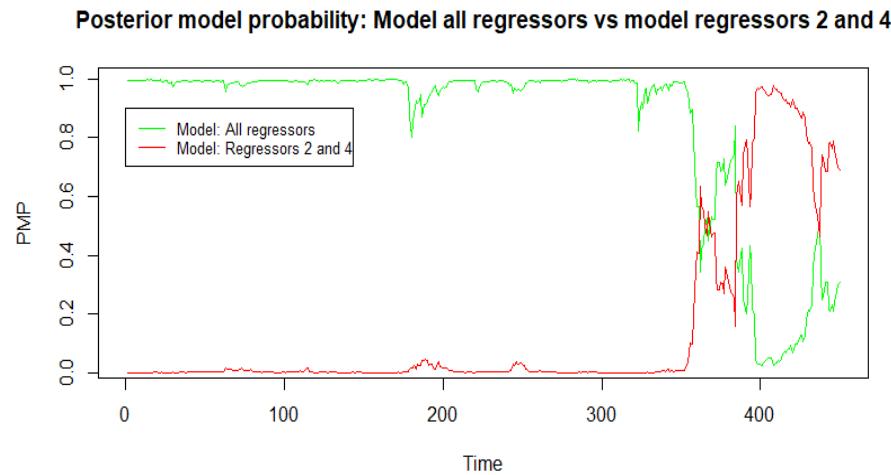
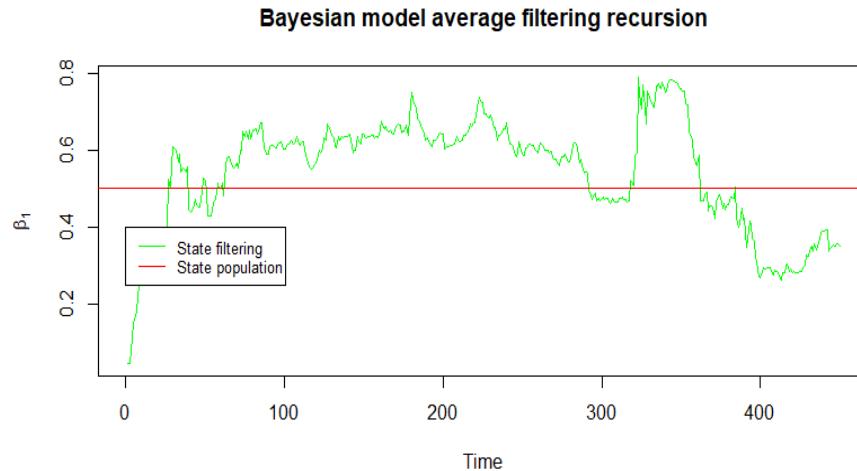


FIGURE 10.1

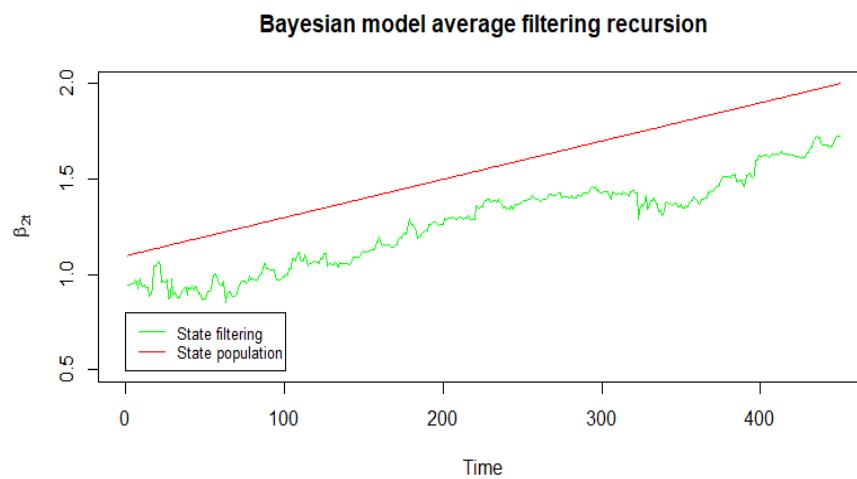
Posterior model probability: Dynamic Bayesian model average.

Figures 10.2, 10.3, 10.4 and 10.5 show a comparison between the Bayesian model average filtering recursion of the states (green lines), and their population values (red lines). We see that the filtering recursions follow the pattern of the population values. However, the values are far from being perfect. This is due to the PMPs of the models matching the data generating process being not equal to 1, this in turn affects the performance of the filtering recursions.

Dynamic Bayesian model average was extended to logit models by [147]. We ask in Exercise 12 to perform a simulation of this model, and perform BMA using the function *logistic.dma* from the *dma* package.

**FIGURE 10.2**

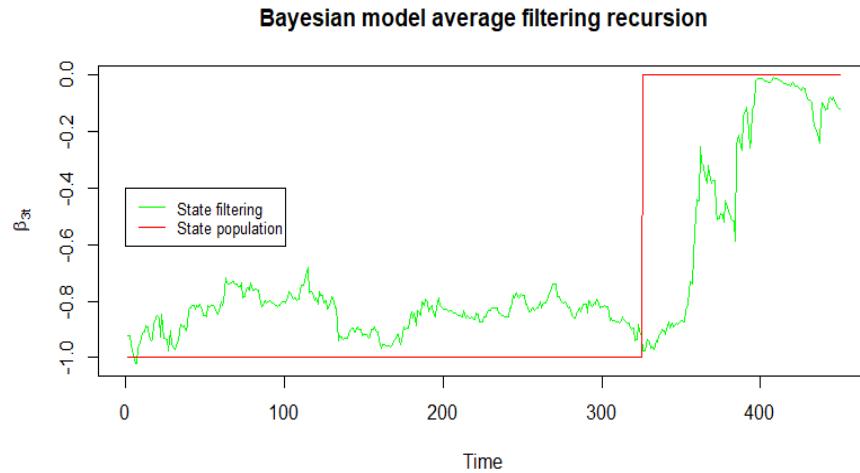
State β_1 : Population versus dynamic Bayesian model average of the filtering recursion.

**FIGURE 10.3**

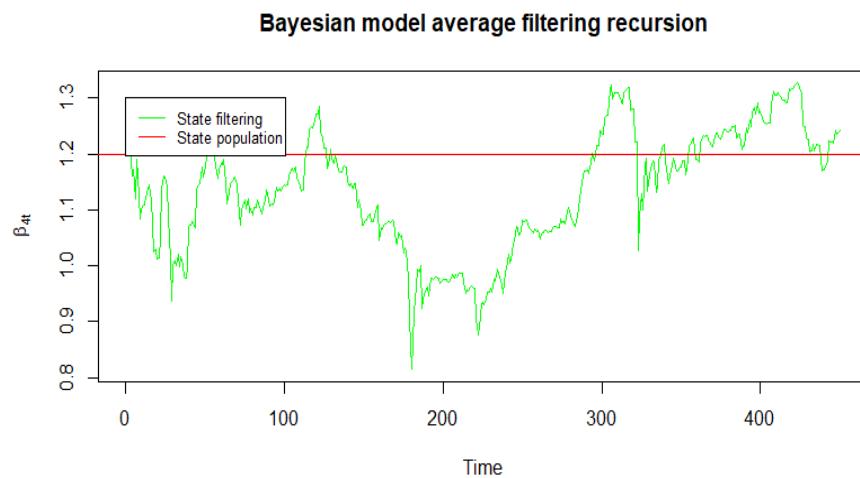
State β_{2t} : Population versus dynamic Bayesian model average of the filtering recursion.

10.5 Calculating the marginal likelihood

The BIC is an asymptotic shortcut to approximate the marginal likelihood, and consequently, obtain the Bayes factors. However, this has limitations in

**FIGURE 10.4**

State β_{3t} : Population versus dynamic Bayesian model average of the filtering recursion.

**FIGURE 10.5**

State β_4 : Population versus dynamic Bayesian model average of the filtering recursion.

moderate and small sample size applications [72]. Thus, there are other meth-

ods to calculate the Bayes factors when there is no an analytical solution of the marginal likelihood.

Observe that calculating the Bayes factor with respect to a reference model (\mathcal{M}_0) help to obtain the posterior model probabilities,

$$\begin{aligned}\pi(\mathcal{M}_j|\mathbf{y}) &= \frac{p(\mathbf{y}|\mathcal{M}_j)\pi(\mathcal{M}_j)}{\sum_{m=1}^M p(\mathbf{y}|\mathcal{M}_m)\pi(\mathcal{M}_m)} \\ &= \frac{p(\mathbf{y}|\mathcal{M}_j)\pi(\mathcal{M}_j)/p(\mathbf{y}|\mathcal{M}_0)}{\sum_{m=1}^M p(\mathbf{y}|\mathcal{M}_m)\pi(\mathcal{M}_m)/p(\mathbf{y}|\mathcal{M}_0)} \\ &= \frac{BF_{j0} \times \pi(\mathcal{M}_j)}{\sum_{m=1}^M BF_{l0} \times \pi(\mathcal{M}_l)}.\end{aligned}$$

Thus, $\pi(\mathcal{M}_j|\mathbf{y}) = \frac{BF_{j0}}{\sum_{m=1}^M BF_{l0}}$ assuming equal prior model probabilities.

In addition, it has been established in many settings that the Bayes factor is consistent, that is, the probability of uncovering the true data generating process converges to 1 when the sample size converges to infinity, or, it asymptotically identifies the model that minimizes the Kullback-Leibler divergence with respect to the data generating process when this is no part of the models into consideration [44, 216, 215].⁶

10.5.1 Savage-Dickey density ratio

The Savage-Dickey density ratio is a way to calculate the Bayes factors when we compare nested models with particular priors [56, 214]. In particular, given the parameter space $\boldsymbol{\theta} = (\boldsymbol{\omega}^\top, \boldsymbol{\psi}^\top)^\top \in \Theta = \Omega \times \Psi$, where we wish to test the null hypothesis $H_0 : \boldsymbol{\omega} = \boldsymbol{\omega}_0$ (model \mathcal{M}_1) versus $H_1 : \boldsymbol{\omega} \neq \boldsymbol{\omega}_0$ (model \mathcal{M}_2), if $\pi(\boldsymbol{\psi}|\boldsymbol{\omega}_0, \mathcal{M}_2) = \pi(\boldsymbol{\psi}|\mathcal{M}_1)$,⁷ then the Bayes factor comparing \mathcal{M}_1 versus \mathcal{M}_2 is

$$BF_{12} = \frac{\pi(\boldsymbol{\omega} = \boldsymbol{\omega}_0|\mathbf{y}, \mathcal{M}_2)}{\pi(\boldsymbol{\omega} = \boldsymbol{\omega}_0|\mathcal{M}_2)}, \quad (10.3)$$

where $\pi(\boldsymbol{\omega} = \boldsymbol{\omega}_0|\mathbf{y}, \mathcal{M}_2)$ and $\pi(\boldsymbol{\omega} = \boldsymbol{\omega}_0|\mathcal{M}_2)$ are the posterior and prior densities of $\boldsymbol{\omega}$ under \mathcal{M}_2 evaluated at $\boldsymbol{\omega}_0$ (see [214]).

Equation 10.3 is called the Savage-Dickey density ratio. A nice feature is that just requires estimation of model \mathcal{M}_2 , and evaluation of the prior and posterior densities. This means no evaluation of the marginal likelihood [126, Chap. 4].

⁶[109] highlight the important difference between pairwise consistency, and model selection consistency. The latter requires consistency of a sequence of pairwise nested comparisons.

⁷Note that a sufficient condition for this assumption is to assume the same prior for the parameters that are the same in each model. [214] incorporate a correction factor when this assumption is not satisfied.

10.5.2 Chib's methods

Another popular method to calculate the marginal likelihood is given by [40] and [43]. The former is an algorithm to calculate the marginal likelihood from the posterior draws of the Gibbs sampling algorithm, and the latter calculates the marginal likelihood from the posterior draws of the Metropolis-Hastings algorithm.

The point of departure in [40] is the identity

$$\pi(\boldsymbol{\theta}^*|\mathbf{y}, \mathcal{M}_m) = \frac{p(\mathbf{y}|\boldsymbol{\theta}^*, \mathcal{M}_m) \times \pi(\boldsymbol{\theta}^*|\mathcal{M}_m)}{p(\mathbf{y}|\mathcal{M}_m)},$$

where $\boldsymbol{\theta}^*$ is a particular value of $\boldsymbol{\theta}$ of high probability, for instance, the mode. This implies that

$$p(\mathbf{y}|\mathcal{M}_m) = \frac{p(\mathbf{y}|\boldsymbol{\theta}^*, \mathcal{M}_m) \times \pi(\boldsymbol{\theta}^*|\mathcal{M}_m)}{\pi(\boldsymbol{\theta}^*|\mathbf{y}, \mathcal{M}_m)}.$$

We can easily calculate the numerator of this expression. However, the critical point in this expression is to calculate the denominator as we know $\pi(\boldsymbol{\theta}^*|\mathbf{y}, \mathcal{M}_m)$ up to a normalizing constant. We can calculate this from the posterior draws. Assume that $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top \ \boldsymbol{\theta}_2^\top]^\top$, then $\pi(\boldsymbol{\theta}^*|\mathbf{y}, \mathcal{M}_m) = \pi(\boldsymbol{\theta}_1^*|\boldsymbol{\theta}_2^*, \mathbf{y}, \mathcal{M}_m) \times \pi(\boldsymbol{\theta}_2^*|\mathbf{y}, \mathcal{M}_m)$. We have the first term because in the Gibbs sampling algorithm the posterior conditional distributions are available. The second is

$$\begin{aligned} \pi(\boldsymbol{\theta}_2^*|\mathbf{y}, \mathcal{M}_m) &= \int_{\boldsymbol{\Theta}_1} \pi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2^*|\mathbf{y}, \mathcal{M}_m) d\boldsymbol{\theta}_1 \\ &= \int_{\boldsymbol{\Theta}_1} \pi(\boldsymbol{\theta}_2^*|\boldsymbol{\theta}_1, \mathbf{y}, \mathcal{M}_m) \pi(\boldsymbol{\theta}_1|\mathbf{y}, \mathcal{M}_m) d\boldsymbol{\theta}_1 \\ &\approx \frac{1}{S} \sum_{s=1}^S \pi(\boldsymbol{\theta}_2^*|\boldsymbol{\theta}_1^{(s)}, \mathbf{y}, \mathcal{M}_m), \end{aligned}$$

where $\boldsymbol{\theta}_1^{(s)}$ are the posterior draws of $\boldsymbol{\theta}_1$ from the Gibbs sampling algorithm.

The generalization to more blocks can be seen in [40] and [92, Chap. 7]. In addition, the extension to the Metropolis-Hastings algorithm can be seen in [43], and [92, Chap. 7].

10.5.3 Gelfand-Dey method

We can use the Gelfand-Dey method [72] when we want to calculate the Bayes factor to compare non-nested models, models where the Savage-Dickey density ratio is hard to calculate, or the Chib's methods are difficult to implement. The Gelfand-Dey method is very general, and can be used in virtually any model [126, Chap. 5].

Given a probability density function $q(\boldsymbol{\theta})$, whose support is in Θ , then

$$\mathbb{E} \left[\frac{q(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|\mathcal{M}_m)p(\mathbf{y}|\boldsymbol{\theta}_m, \mathcal{M}_m)} \middle| \mathbf{y}, \mathcal{M}_m \right] = \frac{1}{p(\mathbf{y}|\mathcal{M}_m)},$$

where the expected value is with respect to the posterior distribution given the model \mathcal{M}_m (see Exercise 12).

The critical point is to select a good $q(\boldsymbol{\theta})$. [83] recommends to use $q(\boldsymbol{\theta})$ equal to a truncated multivariate normal density function with mean and variance equal to the posterior mean ($\hat{\boldsymbol{\theta}}$) and variance ($\hat{\Sigma}$) of $\boldsymbol{\theta}$. The truncation region is $\hat{\Theta} = \left\{ \boldsymbol{\theta} : (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \hat{\Sigma}^{-1} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \leq \chi^2_{1-\alpha}(K) \right\}$, where $\chi^2_{1-\alpha}(K)$ is the $(1 - \alpha)$ percentile of the Chi-squared distribution with K degrees of freedom, K is the dimension of $\boldsymbol{\theta}$. We can pick small values of α , for instance, $\alpha = 0.01$.

Observe that

$$\mathbb{E} \left[\frac{q(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|\mathcal{M}_m)p(\mathbf{y}|\boldsymbol{\theta}_m, \mathcal{M}_m)} \middle| \mathbf{y}, \mathcal{M}_m \right] \approx \frac{1}{S} \sum_{s=1}^S \left[\frac{q(\boldsymbol{\theta}^{(s)})}{\pi(\boldsymbol{\theta}^{(s)}|\mathcal{M}_m)p(\mathbf{y}|\boldsymbol{\theta}_m^{(s)}, \mathcal{M}_m)} \right],$$

where $\boldsymbol{\theta}_m^{(s)}$ are draws from the posterior distribution.

Observe that we can calculate the marginal likelihoods of the models in Chapters 6, 7, 8 and 9 using the Chib's methods and the Gelfand-Dickey method.

Example: Simulation exercise

Let's check the performance of the Chib's method and Gelfand-Dey method to calculate the marginal likelihood, and consequently, the Bayes factor in a setting where we can get the analytical solution of the marginal likelihood. In particular, the Gaussian linear model with conjugate prior (see Section 3.3).

Let's assume that the data generating process is $y_{it} = 0.7 + 0.3x_{i1} + 0.7x_{i2} - 0.2x_{i3} + 0.2x_{i4}\mu_i$, where $x_{i1} \sim B(0.3)$, $x_{ik} \sim N(0, 1)$, $k = 2, \dots, 4$, and $\mu_i \sim N(0, 2^2)$, $i = 1, 2, \dots, 500$. Let's set $H_0 : \beta_4 = 0$ (model \mathcal{M}_1) versus $H_1 : \beta_4 \neq 0$ (model \mathcal{M}_2).

Let's assume that $\boldsymbol{\beta}_{m0} = \mathbf{0}_{m0}$, $\mathbf{B}_{m0} = 0.5\mathbf{I}_m$, $\alpha_0 = \delta_0 = 4$. The dimensions of $\mathbf{0}_{m0}$ and \mathbf{I}_m are 4 for model \mathcal{M}_1 and 5 for \mathcal{M}_2 . In addition, let's assume equal prior probabilities.

We know from Section 3.3 that the marginal likelihood is

$$p(\mathbf{y}|\mathcal{M}_m) = \frac{\delta_{m0}^{\alpha_{m0}/2}}{\delta_{mn}^{\alpha_{mn}/2}} \frac{|\mathbf{B}_{mn}|^{1/2}}{|\mathbf{B}_{m0}|^{1/2}} \frac{\Gamma(\alpha_{mn}/2)}{\Gamma(\alpha_{m0}/2)},$$

where $\mathbf{B}_{mn} = (\mathbf{B}_{m0}^{-1} + \mathbf{X}_m^\top \mathbf{X}_m)^{-1}$, $\boldsymbol{\beta}_{mn} = \mathbf{B}_{mn}(\mathbf{B}_{m0}^{-1}\boldsymbol{\beta}_{m0} + \mathbf{X}_m^\top \mathbf{X}_m \hat{\boldsymbol{\beta}}_m)$, $\alpha_{mn} = \alpha_{m0} + N$, and $\delta_{mn} = \delta_{m0} + (\mathbf{y} - \mathbf{X}_m \hat{\boldsymbol{\beta}}_m)^\top (\mathbf{y} - \mathbf{X}_m \hat{\boldsymbol{\beta}}_m) + (\hat{\boldsymbol{\beta}}_m - \boldsymbol{\beta}_{m0})^\top ((\mathbf{X}_m^\top \mathbf{X}_m)^{-1} + \mathbf{B}_{m0})^{-1} (\hat{\boldsymbol{\beta}}_m - \boldsymbol{\beta}_{m0})$, $m = 1, 2$ are the indices of the models.

The log marginal likelihoods for models \mathcal{M}_1 and \mathcal{M}_2 are -1089.82 and -1087.94, respectively. This implies a $2 \times \log(BF_{21}) = 3.75$ which means positive evidence against model \mathcal{M}_1 (see Table 1.1).

We calculate the log marginal likelihood using the Chib's method taking into account that

$$\log(p(\mathbf{y}|\mathcal{M}_m)) = \log(p(\mathbf{y}|\boldsymbol{\theta}^*, \mathcal{M}_m)) + \log(\pi(\boldsymbol{\theta}^*|\mathcal{M}_m)) - \log(\pi(\boldsymbol{\theta}^*|\mathbf{y}, \mathcal{M}_m)),$$

where $p(\mathbf{y}|\boldsymbol{\theta}^*, \mathcal{M}_m)$ is the value of a normal density with mean $\mathbf{X}_m \boldsymbol{\beta}_m^*$ and variance $\sigma_m^{2*} \mathbf{I}_N$ evaluated at \mathbf{y} . In addition, $\log(\pi(\boldsymbol{\theta}^*|\mathcal{M}_m)) = \log(\pi(\boldsymbol{\beta}_m^*|\sigma_m^{2*})) + \log(\pi(\sigma_m^{2*}))$, where the first term is the density of a normal with mean $\boldsymbol{\beta}_{m0}$ and variance matrix $\sigma^{2*} \mathbf{B}_{m0}$ evaluated at $\boldsymbol{\beta}_m^*$, and the second term is the density of an inverse-gamma with parameters $\alpha_{m0}/2$ and $\delta_{m0}/2$ evaluated at σ_m^{2*} . Finally, the third term in the right hand of the previous expression is $\log(\pi(\boldsymbol{\theta}^*|\mathbf{y}, \mathcal{M}_m)) = \log(\pi(\boldsymbol{\beta}_m^*|\sigma_m^{2*}, \mathbf{y})) + \log(\pi(\sigma_m^{2*}|\mathbf{y}))$, where the first term is the density of a normal with mean $\boldsymbol{\beta}_{mn}$ and variance matrix $\sigma_m^{2*} \mathbf{B}_{mn}$ evaluated at $\boldsymbol{\beta}_m^*$, and the second term is the density of an inverse-gamma with parameters $\alpha_{mn}/2$ and $\delta_{mn}/2$ evaluated at σ_m^{2*} . We use the modes of the posterior draws of $\boldsymbol{\beta}_m$ and σ_m^2 as reference values.

We get the same value, up to two decimals, for the log marginal likelihood of the restricted and unrestricted models using the Chib's method and the analytical expression. Thus, $2 \times \log(BF_{21}) = 3.75$, that is, positive evidence against model \mathcal{M}_1 (see Table 1.1).

We calculate the log marginal likelihood using the Gelfand-Dey method taking into account that

$$\log \left[\frac{q(\boldsymbol{\theta}^{(s)})}{\pi(\boldsymbol{\theta}^{(s)}|\mathcal{M}_m)p(\mathbf{y}|\boldsymbol{\theta}_m^{(s)}, \mathcal{M}_m)} \right] = \log(q(\boldsymbol{\theta}^{(s)})) - \log(\pi(\boldsymbol{\theta}^{(s)}|\mathcal{M}_m)) - \log(p(\mathbf{y}|\boldsymbol{\theta}_m^{(s)}, \mathcal{M}_m)),$$

where $q(\boldsymbol{\theta}^{(s)})$ is the truncated multivariate normal density of Subsection 10.5.3 evaluated at $\boldsymbol{\theta}^{(s)} = [\boldsymbol{\beta}^{(s)\top} \ \sigma^{2(s)}]^\top$, which is the s -th posterior draw of the Gibbs sampling algorithm, such that $\boldsymbol{\theta}^{(s)}$ satisfies the truncation restriction. $\log(\pi(\boldsymbol{\theta}^{(s)}|\mathcal{M}_m)) = \log(\pi(\boldsymbol{\beta}_m^{(s)}|\sigma_m^{2(s)})) + \log(\pi(\sigma_m^{2(s)}))$, where the first term is the density of a normal with mean $\boldsymbol{\beta}_{m0}$ and variance matrix $\sigma^{2(s)} \mathbf{B}_{m0}$ evaluated at $\boldsymbol{\beta}_m^{(s)}$, and the second term is the density of an inverse-gamma with parameters $\alpha_{m0}/2$ and $\delta_{m0}/2$ evaluated at $\sigma_m^{2(s)}$. The third term $p(\mathbf{y}|\boldsymbol{\theta}^{(s)}, \mathcal{M}_m)$ is the value of a normal density with mean $\mathbf{X}_m \boldsymbol{\beta}_m^{(s)}$ and variance $\sigma_m^{2(s)} \mathbf{I}_N$ evaluated at \mathbf{y} .

The log marginal likelihoods of the restricted and unrestricted models using the Gelfand-Dey method are -1087.43 and -1084.53, respectively. This implies $2 \times \log(BF_{21}) = 5.80$, which is positive evidence in favor of the unrestricted model.

We see in this example that these methods give good approximations to the true marginal likelihoods. However, the Chib's method did a better job

than the Gelfand-Dey method. In addition, the computational demand in the Gelfand-Dey method is by far larger than the Chib's method. We can see this because the Chib's method requires just evaluation at the modes given that we have the marginal posterior of σ^2 in this example, whereas the Gelfand-Dey method requires many evaluations based on the posterior draws. However, we should take in mind that the Gelfand-Dey method is more general.

The following code shows how to do this calculations.

R code. Simulation exercise: Bayes factors

```

1 set.seed(010101)
2 N <- 500; K <- 5; K2 <- 3
3 B <- c(0.7, 0.3, 0.7, -0.2, 0.2)
4 X1 <- rbinom(N, 1, 0.3)
5 X2 <- matrix(rnorm(K2*N), N, K2)
6 X <- cbind(1, X1, X2)
7 Y <- X%*%B + rnorm(N, 0, sd = 2)
8 # Hyperparameters
9 d0 <- 4
10 a0 <- 4
11 b0 <- rep(0, K)
12 c0pt <- 0.5
13 LogMarLikLM <- function(X, c0){
14   K <- dim(X)[2]; N <- dim(X)[1]
15   # Hyperparameters
16   B0 <- c0*diag(K); b0 <- rep(0, K)
17   # Posterior parameters
18   bhat <- solve(t(X)%*%X)%*%t(X)%*%Y
19   Bn <- as.matrix(Matrix::forceSymmetric(solve(solve(B0) + t
20     (X)%*%X)))
21   bn <- Bn%*%(solve(B0)%*%b0 + t(X)%*%X%*%bhat)
22   dn <- as.numeric(d0 + t(Y)%*%Y+t(b0)%*%solve(B0)%*%b0-t(bn
23     )%*%solve(Bn)%*%bn)
24   an <- a0 + N
25   # Log marginal likelihood
26   logpy <- (N/2)*log(1/pi)+(a0/2)*log(d0)-(an/2)*log(dn) +
27     0.5*log(det(Bn)/det(B0)) + lgamma(an/2)-lgamma(a0/2)
28   return(-logpy)
29 }
30 LogMarM2 <- -LogMarLikLM(X = X, c0 = c0pt)
31 LogMarM1 <- -LogMarLikLM(X = X[,1:4], c0 = c0pt)
32 BF12 <- exp(LogMarM1-LogMarM2)
33 BF12; 1/BF12
34 2*log(1/BF12}

```

R code. Simulation exercise: Bayes factors

```

1 # Chib's method
2 sig2Post <- MCMCpack::rinvgamma(S, an/2, dn/2)
3 BetasGibbs <- sapply(1:S, function(s){MASS::mvrnorm(n = 1,
4   mu = bn, Sigma = sig2Post[s]*Bn)})
5 mode_continuous <- function(x){
6   density_est <- density(x)
7   mode_value <- density_est$x[which.max(density_est$y)]
8   return(mode_value)
9 }
10 # Unrestricted model
11 BetasMode <- apply(BetasGibbs, 1, mode_continuous)
12 Sigma2Mode <- mode_continuous(sig2Post)
13 VarModel <- Sigma2Mode*diag(N)
14 MeanModel <- X%*%BetasMode
15 LogLik <- mvtnorm::dmvnorm(c(Y), mean = MeanModel, sigma =
16   VarModel, log = TRUE, checkSymmetry = TRUE)
17 LogPrior <- mvtnorm::dmvnorm(BetasMode, mean = rep(0, K),
18   sigma = Sigma2Mode*cOpt*diag(K), log = TRUE,
19   checkSymmetry = TRUE)+log(MCMCpack::dinvgamma(Sigma2Mode
20   , a0/2, d0/2))
21 LogPost1 <- mvtnorm::dmvnorm(BetasMode, mean = bn, sigma =
22   Sigma2Mode*Bn, log = TRUE, checkSymmetry = TRUE)
23 LogPost2 <- log(MCMCpack::dinvgamma(Sigma2Mode, an/2, dn/2))
24 LogMarLikChib <- LogLik + LogPrior -(LogPost1 + LogPost2)
25 # Restricted model
26 anRest <- N + a0; XRest <- X[, -5]
27 KRest <- dim(XRest)[2]; B0Rest <- cOpt*diag(KRest)
28 BnRest <- solve(solve(B0Rest)+t(XRest)%*%XRest)
29 bhatRest <- solve(t(XRest)%*%XRest)%*%t(XRest)%*%Y
30 b0Rest <- rep(0, KRest)
31 bnRest <- BnRest%*%(solve(B0Rest)%*%b0Rest+t(XRest)%*%XRest%
32   *%bhatRest)
33 dnRest <- as.numeric(d0 + t(Y-XRest)%*%bhatRest)%*%(Y-XRest)%*
34   %bhatRest)+t(bhatRest - b0Rest)%*%solve(solve(t(XRest)%*
35   %XRest)+B0Rest)%*%(bhatRest - b0Rest)
36 sig2PostRest <- MCMCpack::rinvgamma(S, anRest/2, dnRest/2)
37 BetasGibbsRest <- sapply(1:S, function(s){MASS::mvrnorm(n =
38   1, mu = bnRest, Sigma = sig2PostRest[s]*BnRest)})
39 BetasModeRest <- apply(BetasGibbsRest, 1, mode_continuous)
40 Sigma2ModeRest <- mode_continuous(sig2PostRest)
41 VarModelRest <- Sigma2ModeRest*diag(N)
42 MeanModelRest <- XRest%*%BetasModeRest
43 LogLikRest <- mvtnorm::dmvnorm(c(Y), mean = MeanModelRest,
44   sigma = VarModelRest, log = TRUE, checkSymmetry = TRUE)
45 LogPriorRest <- mvtnorm::dmvnorm(BetasModeRest, mean = rep
46   (0, KRest), sigma = Sigma2ModeRest*cOpt*diag(KRest), log
47   = TRUE, checkSymmetry = TRUE)+log(MCMCpack::dinvgamma(
48   Sigma2ModeRest, a0/2, d0/2))
49 LogPost1Rest <- mvtnorm::dmvnorm(BetasModeRest, mean =
50   bnRest, sigma = Sigma2ModeRest*BnRest, log = TRUE,
51   checkSymmetry = TRUE)
52 LogPost2Rest <- log(MCMCpack::dinvgamma(Sigma2ModeRest,
53   anRest/2, dnRest/2))
54 LogMarLikChibRest <- LogLikRest + LogPriorRest -
55   LogPost1Rest + LogPost2Rest)
56 BFChibs <- exp(LogMarLikChibRest - LogMarLikChib)
57 BFChibs; 1/BFChibs; 2*log(1/BFChibs)

```

R code. Simulation exercise: Bayes factors

```

1 # Gelfand-Dey method
2 GDmarglik <- function(ids, X, Betas, MeanThetas, VarThetas,
3   sig2Post){
4   K <- dim(X)[2]; Thetas <- c(Betas[ids,], sig2Post[ids])
5   Lognom <- (1/(1-alpha))*mvtnorm::dmvnorm(Thetas, mean =
6     MeanThetas, sigma = VarThetas, log = TRUE, checkSymmetry
7     = TRUE)
8   Logden1 <- mvtnorm::dmvnorm(Betas[ids,], mean = rep(0, K),
9     sigma = sig2Post[ids]*cOpt*diag(K), log = TRUE,
10    checkSymmetry = TRUE) + log(MCMCpack::dinvgamma(sig2Post
11    [ids], a0/2, d0/2))
12  VarModel <- sig2Post[ids]*diag(N)
13  MeanModel <- X%*%Betas[ids,]
14  Logden2 <- mvtnorm::dmvnorm(c(Y), mean = MeanModel, sigma
15    = VarModel, log = TRUE, checkSymmetry = TRUE)
16  LogGDid <- Lognom - Logden1 - Logden2
17  return(LogGDid)
18 }
19 sig2Post <- MCMCpack::rinvgamma(S, an/2, dn/2)
20 Betas <- LaplacesDemon::rmvt(S, bn, Hn, an)
21 Thetas <- cbind(Betas, sig2Post)
22 MeanThetas <- colMeans(Thetas); VarThetas <- var(Thetas)
23 iVarThetas <- solve(VarThetas)
24 ChiSQ <- sapply(1:S, function(s){(Thetas[s,]-MeanThetas)%*%
25   iVarThetas%*%(Thetas[s,]-MeanThetas)})
26 alpha <- 0.01; criticalval <- qchisq(1-alpha, K + 1)
27 idGoodThetas <- which(ChiSQ <= criticalval)
28 pb <- winProgressBar(title = "progress bar", min = 0, max =
29   S, width = 300)
30 InvMargLik2 <- NULL
31 for(s in idGoodThetas){
32   LogInvs <- GDmarglik(ids = s, X = X, Betas = Betas,
33     MeanThetas = MeanThetas, VarThetas = VarThetas, sig2Post
34     = sig2Post)
35   InvMargLik2 <- c(InvMargLik2, LogInvs)
36   setWinProgressBar(pb, s, title=paste( round(s/S*100, 0), "%",
37     "done"))
38 }
39 close(pb); mean(InvMargLik2)
40 # Restricted model
41 anRest <- N + a0; XRest <- X[,-5]
42 KRest <- dim(XRest)[2]; B0Rest <- cOpt*diag(KRest)
43 BnRest <- solve(solve(B0Rest)+t(XRest)%*%XRest)
44 bhatRest <- solve(t(XRest)%*%XRest)%*%t(XRest)%*%Y
45 b0Rest <- rep(0, KRest)
46 bnRest <- BnRest%*%(solve(B0Rest)%*%b0Rest+t(XRest)%*%XRest%*
47   *%bhatRest)
48 dnRest <- as.numeric(d0 + t(Y-XRest%*%bhatRest)%*%(Y-XRest%*
49   %bhatRest)+t(bhatRest - b0Rest)%*%solve(solve(t(XRest)%*%
50   %XRest)+B0Rest)%*%(bhatRest - b0Rest))
51 HnRest <- as.matrix(Matrix::forceSymmetric(dnRest*BnRest/
52   anRest))
53 sig2PostRest <- MCMCpack::rinvgamma(S, anRest/2, dnRest/2)
54 BetasRest <- LaplacesDemon::rmvt(S, bnRest, HnRest, anRest)
55 ThetasRest <- cbind(BetasRest, sig2PostRest)
56 MeanThetasRest <- colMeans(ThetasRest)
57 VarThetasRest <- var(ThetasRest)
58 iVarThetasRest <- solve(VarThetasRest)

```

R code. Simulation exercise: Bayes factors

```

1 ChiSQRest <- sapply(1:S, function(s){(ThetasRest[s,]-
  MeanThetasRest)%*%iVarThetasRest%*%(ThetasRest[s,]-
  MeanThetasRest)})
2 idGoodThetasRest <- which(ChiSQRest <= criticalval)
3 pb <- winProgressBar(title = "progress bar", min = 0, max =
  S, width = 300)
4 InvMargLik1 <- NULL
5 for(s in idGoodThetasRest){
6   LogInvs <- GDmarglik(ids = s, X = XRest, Betas = BetasRest
  , MeanThetas = MeanThetasRest, VarThetas = VarThetasRest
  , sig2Post = sig2PostRest)
7   InvMargLik1 <- c(InvMargLik1, LogInvs)
8   setWinProgressBar(pb, s, title=paste( round(s/S*100, 0), "%
    done"))
9 }
10 close(pb); summary(coda::mcmc(InvMargLik1))
11 mean(InvMargLik1)
12 BFFD <- exp(mean(InvMargLik2)-mean(InvMargLik1))
13 BFFD; mean(1/BFFD); 2*log(1/BFFD)

```

10.6 Summary

In this chapter we introduced Bayesian model average in generalized linear models. In the case of linear Gaussian models, we perform BMA using three approaches: the Bayesian information criterion approximation with the Occam's window, the Markov chain Monte Carlo model composition algorithm, and the conditional Bayes factors when taking into account endogeneity. In the case of other generalized linear models, logit, gamma and Poisson, we show how to use the BIC approximation to perform BMA. In addition, we show how to perform dynamic Bayesian model average in state-space models, this setting is based on forgetting parameters to facilitate computation. Finally, we present alternative ways to calculate the marginal likelihood: the Savage-Dickey density ration, Chib's method and Gelfand-Dey method, which are really useful when the BIC approximation does not perform a good job due to small or moderate sample sizes.

10.7 Exercises

1. The Gaussian linear model specifies $\mathbf{y} = \alpha \mathbf{i}_N + \mathbf{X}_m \boldsymbol{\beta}_m + \boldsymbol{\mu}_m$ such that $\boldsymbol{\mu}_m \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n)$, and \mathbf{X}_m does not have the column of ones. Assuming that $\pi(\sigma^2) \propto 1/\sigma^2$, $\pi(\alpha) \propto 1$, and $\boldsymbol{\beta}_m | \sigma^2 \sim N(\mathbf{0}_{k_m}, \sigma^2 (\mathbf{g}_m \mathbf{X}_m^\top \mathbf{X}_m)^{-1})$.

- Show that the posterior conditional distribution of $\boldsymbol{\beta}_m$ is $N(\boldsymbol{\beta}_{mn}, \sigma^2 \mathbf{B}_{mn})$, where $\boldsymbol{\beta}_{mn} = \mathbf{B}_{mn} \mathbf{X}_m^\top \mathbf{y}$ and $\mathbf{B}_{mn} = ((1 + g_m) \mathbf{X}_m^\top \mathbf{X}_m)^{-1}$.
- Show that the marginal likelihood associated with model \mathcal{M}_m is proportional to

$$p(\mathbf{y} | \mathcal{M}_m) \propto \left(\frac{g_m}{1 + g_m} \right)^{k_m/2} \left[(\mathbf{y} - \bar{y} \mathbf{i}_N)^\top (\mathbf{y} - \bar{y} \mathbf{i}_N) - \frac{1}{1 + g_m} (\mathbf{y}^\top \mathbf{P}_{X_m} \mathbf{y}) \right]^{-(N-1)/2},$$

where all parameter are indexed to model \mathcal{M}_m , $\mathbf{P}_{X_m} = \mathbf{X}_m (\mathbf{X}_m^\top \mathbf{X}_m)^{-1} \mathbf{X}_m$ is the projection matrix on the space generated by the columns of \mathbf{X}_m , and \bar{y} is the sample mean of \mathbf{y} .

Hint: Take into account that $\mathbf{i}_N^\top \mathbf{X}_m = \mathbf{0}_{k_m}$ due to all columns being centered with respect to their means.

2. **Determinants of export diversification I**

[108] use BMA to study the determinants of export diversification. Use the dataset *10ExportDiversificationHHI.csv* to perform BMA using the BIC approximation and MC3 to check if these two approaches agree.

3. **Simulation exercise of the Markov chain Monte Carlo model composition continues**

Program an algorithm to perform MC3 where the final S models are unique. Use the simulation setting of Section 10.2 increasing the number of regressors to 40, this implies approximately $1.1e+12$ models.

4. **Simulation exercise of IV BMA continues**

Use the simulation setting with endogeneity in Section 10.2 to perform BMA based on the BIC approximation and MC3.

5. **Determinants of export diversification II**

Use the datasets *11ExportDiversificationHHI.csv* and *12ExportDiversificationHHIInstr.csv* to perform IV BMA assuming that the log of per capita gross domestic product is endogenous (*avglgdpcap*). See [108] for details.

6. Show that the link function in the case of the Bernoulli distribution is $\log\left(\frac{\theta}{1-\theta}\right)$.
7. [176, 177] perform variable selection using the file *13InternetMed.csv*. In this data set, the dependent variable is an indicator of Internet adoption (*internet*) for 5000 households in Medellín (Colombia) during the period 2006–2014. This dataset contains information about 18 potential determinants, which means 262144 (2^{18}) potential models just taking into account variable uncertainty (see these papers for details about the data set). Perform BMA using the logit link function using this data set.
8. [198] use the file *14ValueFootballPlayers.csv* to analyze the market value of soccer players in the most important leagues in Europe. In particular, there are 26 potential determinants of the market value (dependent variable) of a stratified sample of 335 soccer players in the five most important leagues in Europe (see [198] for details). Use this data set to perform BMA using the gamma distribution setting default values for Occam's window.
9. Use the dataset *15Fertile2.csv* from [222, p. 547] to perform BMA using the Poisson model with the log link. This data set has information about 1,781 women from Botswana in 1988 (for details, see <https://rdrr.io/cran/wooldridge/man/fertil2.html>, and take into account that we deleted some variables and omitted observations with NA values). The dependent variable is the number of children ever born (*ceb*), which is a count variable, as a function of 19 potential determinants.
10. Perform BMA in the logit model using MC3 and the BIC approximation using the simulation setting of Section 10.3.
11. Use the dataset *19ExchangeRateCOPUSD.csv* to estimate four different *state-space models* to explain the annual variation of the COP to USD exchange rate:

•Interest rate parity

$$\Delta e_t = \beta_{1t}^{IRP} + \beta_{2t}^{IRP}(i_{t-1}^{Col} - i_{t-1}^{USA}) + \mu_t^{IRP}$$

•Purchasing power parity

$$\Delta e_t = \beta_{1t}^{PPP} + \beta_{2t}^{PPP}(\pi_{t-1}^{Col} - \pi_{t-1}^{USA}) + \mu_t^{PPP}$$

•Taylor rule

$$\Delta e_t = \beta_{1t}^{Taylor} + \beta_{2t}^{Taylor}(\pi_{t-1}^{Col} - \pi_{t-1}^{USA}) + \beta_{2t}^{Taylor}(g_{t-1}^{Col} - g_{t-1}^{USA}) + \mu_t^{IRP}$$

•Money supply

$$\Delta e_t = \beta_{1t}^{Money} + \beta_{2t}^{Money}(g_{t-1}^{Col} - g_{t-1}^{USA}) + \beta_{2t}^{Money}(m_{t-1}^{Col} - m_{t-1}^{USA}) + \mu_t^{Money}$$

where varTRM (Δe_t) is the annual variation rate of the exchange rate COP to USD, TES_COL10 (i_t^{Col}) and TES_USA10 (i_t^{USA}) are the annual return rates of Colombian and USA public debts in 10 years, inflation_COL (π_t^{Col}) and inflation_USA (π_t^{USA}) are the annual inflation rates, varISE_COL (g_t^{Col}) and varISE_USA (g_t^{USA}) are annual variation of economic activity indices, and varCOL_M3 (m_t^{Col}) and varUSA_M3 (m_t^{USA}) are annual variations of money supply. We have monthly variations between January 2006 and November 2023.

Perform Bayesian model averaging using these four models to explain the annual variation of the exchange rate, get the posterior model probabilities, and plot the posterior mean and credible interval of $\beta_{2t}^{\text{Money}}$.

12. Perform a simulation of the dynamic logistic model, where there are 7 ($2^3 - 1$, the model without regressors is excluded) competing models originated from 3 regressors, $x_{tk} \sim N(0.5, 0.8^2)$, $k = 2, 3, 4$, and $\beta_1 = 0.5$, β_{2t} is a sequence from 1 to 2 in steps given by $1/T$, $\beta_{3t} = \begin{cases} -1, & 1 < t \leq 0.5T \\ 0, & 0.5T < t \leq T \end{cases}$, and $\beta_4 = 1.2$. Then, $\mathbf{x}_t^\top \boldsymbol{\beta}_t = \beta_1 + \beta_{2t}x_{2t} + \beta_{3t}x_{3t} + \beta_4x_{4t}$, where $P[Y_t = 1 | \mathbf{x}_t, \boldsymbol{\beta}_t] = \exp(\mathbf{x}_t^\top \boldsymbol{\beta}_t) / (1 + \exp(\mathbf{x}_t^\top \boldsymbol{\beta}_t))$, $t = 1, 2, \dots, 1100$. Use the function *logistic.dma* from the *dma* package to get the posterior model probabilities setting the forgetting parameter of the models equal to 0.99, and then to 0.95. Compare the results.
13. Show that

$$\mathbb{E} \left[\frac{q(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} | \mathcal{M}_m) p(\mathbf{y} | \boldsymbol{\theta}_m, \mathcal{M}_m)} \middle| \mathbf{y}, \mathcal{M}_m \right] = \frac{1}{p(\mathbf{y} | \mathcal{M}_m)},$$

where the expected value is with respect to the posterior distribution given the model \mathcal{M}_m , and $q(\boldsymbol{\theta})$ is the proposal distribution whose support is Θ .



Part III

Advanced methods: A brief introduction



11

Non-parametric and semi-parametric models

11.1 Additive non-parametric structure

11.1.1 Partial linear model

11.2 Hierarchical models

11.2.1 Finite mixtures

11.2.2 Dirichlet processes



12

Machine learning

12.1 Cross validation and Bayes factors

12.2 Regularization

The linear normal model using the conjugate family is ridge regression [102]. We can use empirical Bayes to select the scale parameter of the prior covariance matrix of the location parameters, which is in turn the regularization parameter in the ridge regression (see my class notes in MSc in Data Science and Analytic).

12.2.1 Bayesian LASSO

12.2.2 Stochastic search variable selection

12.2.3 Non-local priors

[110] R package: mombf (Model Selection with Bayesian Methods and Information Criteria) link: <https://cran.r-project.org/web/packages/mombf/index.html>

12.3 Bayesian additive regression trees

12.4 Gaussian processes



13

Causal inference

13.1 Instrumental variables

13.1.1 Semi-parametric IV model

13.2 Regression discontinuity design

13.3 Regression kink design

13.4 Synthetic control

13.5 Difference in difference estimation

13.6 Event Analysis

13.7 Bayesian exponential tilted empirical likelihood

13.8 Double-Debiased machine learning causal effects



14

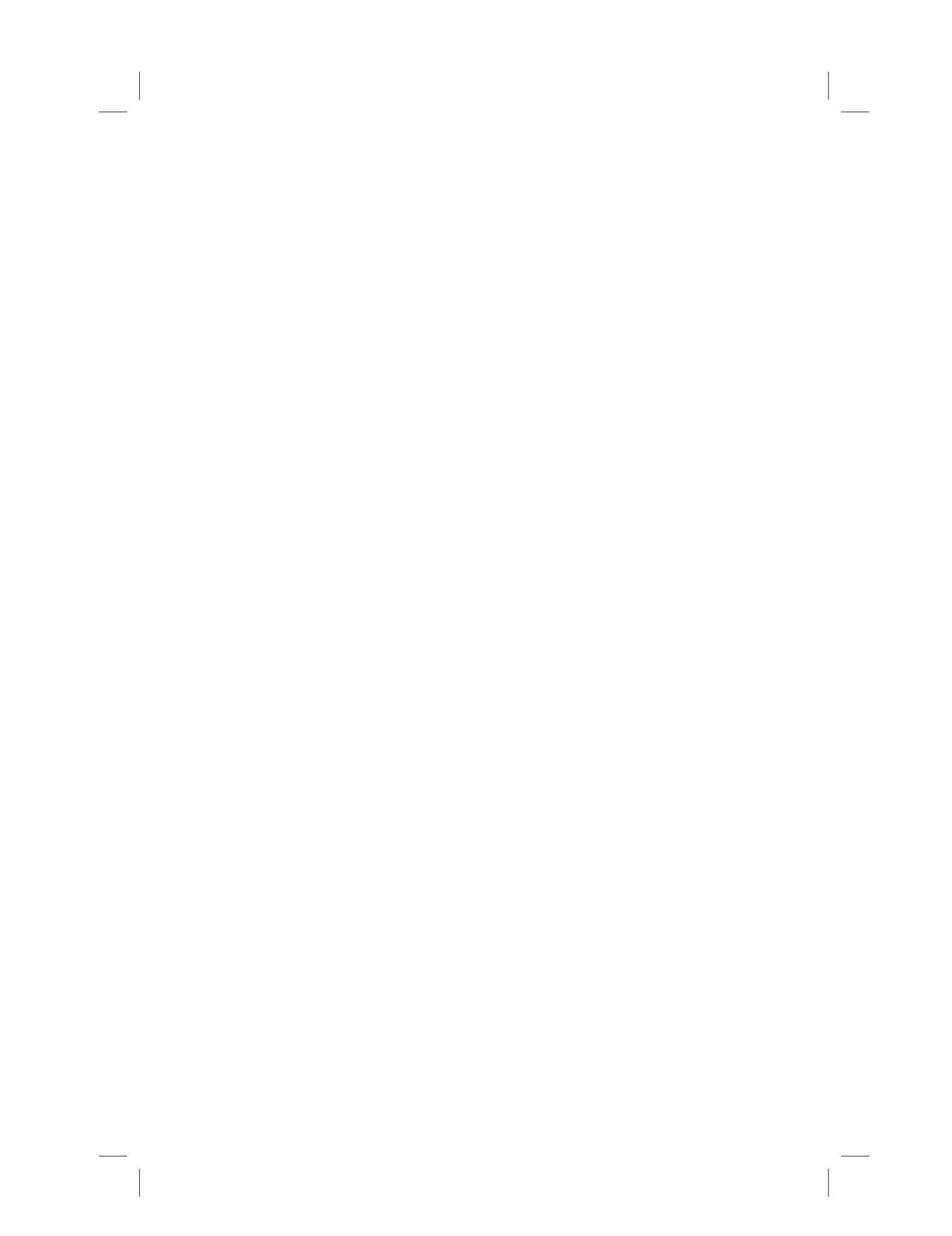
Approximation methods

14.1 Approximate Bayesian computation

14.2 Expectation propagation

14.3 Integrated nested Laplace approximations

14.4 Variational Bayes



Bibliography

- [1] D. Acemoglu, S. Johnson, and J. Robinson. The colonial origins of comparative development: An empirical investigation. *The American Economic Review*, 91(5):1369–1401, 2001.
- [2] J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.
- [3] Jim Albert. *Bayesian Computation with R*. Use R! Springer, New York, NY, 2nd edition, 2009.
- [4] Sungbae An and Frank Schorfheide. Bayesian analysis of dsge models. *Econometric reviews*, 26(2-4):113–172, 2007.
- [5] C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- [6] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- [7] R. Baath. *Package bayesboot*, 2018.
- [8] M. Barbieri and J. Berger. Optimal predictive model selection. *The Annals of Statistics*, 32(3):870–897, 2004.
- [9] M. Bayarri and J. Berger. P-values for composite null models. *Journal of American Statistical Association*, 95:1127–1142, 2000.
- [10] M. J. Bayarri and J. Berger. The interplay of bayesian and frequentist analysis. *Statistical Science*, 19(1):58–80, 2004.
- [11] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–416, 1763.
- [12] Thomas Bayes. LII. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. *Philosophical transactions of the Royal Society of London*, 53:370–418, 1763.

- [13] Daniel J Benjamin, James O Berger, Magnus Johannesson, Brian A Nosek, E-J Wagenmakers, Richard Berk, Kenneth A Bollen, Björn Brembs, Lawrence Brown, Colin Camerer, et al. Redefine statistical significance. *Nature human behaviour*, 2(1):6–10, 2018.
- [14] J. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, third edition edition, 1993.
- [15] J. Berger. The case for objective bayesian analysis. *Bayesian Analysis*, 1(3):385–402, 2006.
- [16] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.
- [17] James O Berger and Luis R Pericchi. The intrinsic bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122, 1996.
- [18] J. Bernardo and A. Smith. *Bayesian Theory*. Wiley, Chichester, 1994.
- [19] José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.
- [20] Peter J Bickel and Joseph A Yahav. Some contributions to the asymptotic theory of bayes solutions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 11(4):257–276, 1969.
- [21] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.
- [22] G. E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 71:791–799, 1976.
- [23] George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, 1st edition, 1976.
- [24] George EP Box. Robustness in the strategy of scientific model building. In *Robustness in statistics*, pages 201–236. Elsevier, 1979.
- [25] Kenneth P Burnham and David R Anderson. Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304, 2004.
- [26] Colin Cameron and Pravin Trivedi. *Microeometrics: Methods and Applications*. Cambridge, 2005.
- [27] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

- [28] O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.
- [29] Bradley P Carlin, Alan E Gelfand, and Adrian FM Smith. Hierarchical bayesian analysis of changepoint problems. *Journal of the royal statistical society: series C (applied statistics)*, 41(2):389–405, 1992.
- [30] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.
- [31] Chris K Carter and Robert Kohn. On gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.
- [32] G. Casella and E. Moreno. Objective Bayesian variable selection. *Journal of the American Statistical Association*, 101(473):157–167, 2006.
- [33] George Casella and Roger Berger. *Statistical inference*. CRC Press, 2024.
- [34] Joshua Chan, Gary Koop, Dale J Poirier, and Justin L Tobias. *Bayesian econometric methods*, volume 7. Cambridge University Press, 2019.
- [35] W. Chang. *Web Application Framework for R: Package shiny*. R Studio, 2018.
- [36] V. Chernozhukov and H. Hong. An MCMC approach to classical estimation. *Journal of Econometrics*, 115:293–346, 2003.
- [37] S. Chib. Bayes inference in the Tobit censored regression model. *Journal of Econometrics*, 51:79–99, 1992.
- [38] S. Chib and B. Carlin. On MCMC sampling in hierarchical longitudinal models. *Statistics and Computing*, 9:17–26, 1999.
- [39] Siddhartha Chib. Bayes regression with autoregressive errors: A gibbs sampling approach. *Journal of econometrics*, 58(3):275–294, 1993.
- [40] Siddhartha Chib. Marginal likelihood from the gibbs output. *Journal of the american statistical association*, 90(432):1313–1321, 1995.
- [41] Siddhartha Chib and Edward Greenberg. Bayes inference in regression models with arma (p, q) errors. *Journal of Econometrics*, 64(1-2):183–206, 1994.
- [42] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

- [43] Siddhartha Chib and Ivan Jeliazkov. Marginal likelihood from the metropolis-hastings output. *Journal of the American Statistical Association*, 96(453):270–281, 2001.
- [44] Siddhartha Chib and Todd A Kuffner. Bayes factor consistency. *arXiv preprint arXiv:1607.00292*, 2016.
- [45] Gerda Claeskens and Nils Lid Hjort. Model selection and model averaging. *Cambridge books*, 2008.
- [46] M. Clyde and E. George. Model uncertainty. *Statistical Science*, 19(1):81–94, 2004.
- [47] T. Conley, C. Hansen, and P. Rossi. Plausibly exogenous. *The Review of Economics and Statistics*, 94(1):260–272, 2012.
- [48] Johan Dahlin and Thomas B Schön. Getting started with particle metropolis-hastings for inference in nonlinear dynamical models. *Journal of Statistical Software*, 88:1–41, 2019.
- [49] A. P. Dawid, M. Musio, and S. E. Fienberg. From statistical evidence to evidence of causality. *Bayesian Analysis*, 11(3):725–752, 2016.
- [50] de Finetti. Foresight: its logical laws, its subjective sources. In H. E. Kyburg and H. E. Smokler, editors, *Studies in Subjective Probability*. Krieger, New York, 1937. p.55–118.
- [51] Piet De Jong and Neil Shephard. The simulation smoother for time series models. *Biometrika*, 82(2):339–350, 1995.
- [52] M. H. DeGroot. *Probability and statistics*. Addison-Wesley Publishing Co., London, 1975.
- [53] Marco Del Negro and F. Schorfheide. Forecasting with bayesian var models. In John Geweke, Gary Koop, and Herman van Dijk, editors, *The Oxford Handbook of Bayesian Econometrics*, pages 224–254. Oxford University Press, 2011.
- [54] Persi Diaconis and Donald Ylvisaker. Conjugate priors for exponential families. *The Annals of statistics*, 7(2):269–281, 1979.
- [55] J. M. Dickey and E. Gunel. Bayes factors from mixed probabilities. *Journal of the Royal Statistical Society: Series B (Methodology)*, 40:43–46, 1978.
- [56] James M Dickey. The weighted likelihood ratio, linear hypotheses on normal location parameters. *The Annals of Mathematical Statistics*, pages 204–223, 1971.
- [57] Peter. Diggle, P. Heagerty, Liang K-Y., and S. Zeger. *Analysis of longitudinal data*. Oxford University Press, 2002.

- [58] Thomas Doan, Robert Litterman, and Christopher Sims. Forecasting and conditional projection using realistic prior distributions. *Econometric reviews*, 3(1):1–100, 1984.
- [59] Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. *Sequential Monte Carlo methods in practice*, pages 3–14, 2001.
- [60] Arnaud Doucet, Adam M Johansen, et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.
- [61] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.
- [62] Y. D. Edwards and G. M. Allenby. Multivariate analysis of multiple response data. *Journal of Marketing Research*, 40:321–334, 2003.
- [63] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7:1–26, 1979.
- [64] Bradley Efron and Trevor Hastie. *Computer age statistical inference*, volume 5. Cambridge University Press, 2016.
- [65] Walter Enders. *Applied Econometric Time Series*. Wiley, Hoboken, NJ, 4th edition, 2014.
- [66] Carmen Fernandez, Eduardo Ley, and Mark FJ Steel. Benchmark priors for bayesian model averaging. *Journal of Econometrics*, 100(2):381–427, 2001.
- [67] R. Fisher. *Statistical Methods for Research Workers*. Hafner, New York, 13th edition, 1958.
- [68] Sylvia Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of time series analysis*, 15(2):183–202, 1994.
- [69] George M. Furnival and Robert W. Wilson. Regressions by leaps and bounds. *Technometrics*, 16(4):499–511, 1974.
- [70] P. Garthwaite, J. Kadane, and A. O'Hagan. Statistical methods for eliciting probability distributions. *Journal of American Statistical Association*, 100(470):680–701, 2005.
- [71] A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.
- [72] Alan E Gelfand and Dipak K Dey. Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(3):501–514, 1994.

- [73] A. Gelman and X. Meng. Model checking and model improvement. In Gilks, Richardson, and Speigelhalter, editors, *In Markov chain Monte Carlo in practice*. Springer US, 1996. Chapter 6, pp. 157–196.
- [74] A. Gelman, X. Meng, and H. Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, pages 733–760, 1996.
- [75] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, November 1992.
- [76] Andrew Gelman, John B Carlin, Hal S Stern, David Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2021.
- [77] Andrew Gelman et al. Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian analysis*, 1(3):515–534, 2006.
- [78] Andrew Gelman and Guido Imbens. Why ask why? forward causal inference and reverse causal questions. Technical report, National Bureau of Economic Research, 2013.
- [79] S Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- [80] E. George and R. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.
- [81] E. George and R. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339–373, 1997.
- [82] J. Geweke. *Bayesian Statistics*, chapter Evaluating the accuracy of sampling-based approaches to calculating posterior moments. Clarendon Press, Oxford, UK., 1992.
- [83] John Geweke. Using simulation methods for bayesian econometric models: inference, development, and communication. *Econometric reviews*, 18(1):1–73, 1999.
- [84] John Geweke. Getting it right: Joint distribution tests of posterior simulators. *Journal of the American Statistical Association*, 99(467):799–804, 2004.
- [85] John Geweke. *Contemporary Bayesian econometrics and statistics*, volume 537. John Wiley & Sons, 2005.
- [86] John Geweke, Gary Koop, and Herman K van Dijk. *The Oxford handbook of Bayesian econometrics*. Oxford University Press, USA, 2011.

- [87] Ryan Giordano, Runjing Liu, Michael I Jordan, and Tamara Broderick. Evaluating sensitivity to the stick-breaking prior in bayesian nonparametrics. *Bayesian Analysis*, 1(1):1–34, 2022.
- [88] I. J. Good. The bayes/non bayes compromise: A brief review. *Journal of the American Statistical Association*, 87(419):597–606, September 1992.
- [89] S. N. Goodman. Toward evidence-based medical statistics. 1: The P value fallacy. *Annals of internal medicine*, 130(12):995–1004, 1999.
- [90] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.
- [91] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.
- [92] Edward Greenberg. *Introduction to Bayesian econometrics*. Cambridge University Press, 2012.
- [93] Damodar N. Gujarati and Dawn C. Porter. *Basic Econometrics*. McGraw-Hill Education, New York, NY, 5th edition, 2009.
- [94] Paul Gustafson. Local robustness in bayesian analysis. In *Robust Bayesian Analysis*, pages 71–88. Springer, 2000.
- [95] Johannes Edmund Handschin and David Q Mayne. Monte carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International journal of control*, 9(5):547–559, 1969.
- [96] W. Hastings. Monte Carlo sampling methods using Markov chains and their application. *Biometrika*, 57:97–109, 1970.
- [97] P. Heidelberger and P. D. Welch. Simulation run length control in the presence of an initial transient. *Operations Research*, 31(6):1109–1144, 1983.
- [98] Lütkepohl Helmut. *New introduction to multiple time series analysis*. Springer, 2005.
- [99] J. Hoeting, D. Madigan, A. Raftery, and C. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999.
- [100] Daniel Hosszejni and Gregor Kastner. Modeling univariate and multivariate stochastic volatility in r with `stochvol` and `factorstochvol`. *Journal of Statistical Software*, 100(12):1–34, 2021.
- [101] Joseph G. Ibrahim and Purushottam W. Laud. On bayesian analysis of generalized linear models using jeffreys’s prior. *Journal of the American Statistical Association*, 86(416):981–986, 1991.

- [102] H. Ishwaran and J. S. Rao. Spike and slab variable selection: Frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.
- [103] Liana Jacobi, Chun Fung Kwok, Andrés Ramírez-Hassan, and Nhung Nghiem. Posterior manifolds over prior parameter regions: Beyond pointwise sensitivity assessments for posterior statistics from mcmc inference. *Studies in Nonlinear Dynamics & Econometrics*, 28(2):403–434, 2024.
- [104] Liana Jacobi, Dan Zhu, and Mark Joshi. Estimating posterior sensitivities with application to structural analysis of bayesian vector autoregressions, 2022.
- [105] H. Jeffreys. Some test of significance, treated by the theory of probability. *Proceedings of the Cambridge philosophy society*, 31:203–222, 1935.
- [106] H. Jeffreys. *Theory of Probability*. Oxford University Press, London, 1961.
- [107] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.
- [108] M. Jetter and A. Ramírez Hassan. Want export diversification? Educate the kids first. *Economic Inquiry*, 53(4):1765–1782, 2015.
- [109] V. E. Jhonson and D. Rossell. Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(498):649–660, 2012.
- [110] Valen E Johnson and David Rossell. Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(498):649–660, 2012.
- [111] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.
- [112] J. B. Kadane. Predictive and structural methods for eliciting prior distributions. In A Zellner, editor, *Bayesian Analysis in Econometrics and Statistics: Essays in honor of Harold Jeffreys*, pages 89–93. North-Holland Publishing Company,, Amsterdam, 1980.
- [113] Joseph Kadane and Lara Wolfson. Experiences in elicitation. *The Statistician*, 47(1):3–19, 1998.
- [114] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

- [115] N. Kantas, A. Doucet, S. S. Singh, and J. M. Maciejowski. An overview of sequential monte carlo methods for parameter estimation in general state-space models. *IFAC Proceedings Volumes*, 42(10):774–785, 2009.
- [116] Nikolas Kantas, Arnaud Doucet, Sumeetpal S Singh, Jan Maciejowski, Nicolas Chopin, et al. On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351, 2015.
- [117] G. Karabatsos. A menu-driven software package of Bayesian nonparametric (and parametric) mixed models for regression analysis and density estimation. *Behavior Research Methods*, 49:335–362, 2016.
- [118] A. Karl and A. Lenkoski. Instrumental variable Bayesian model averaging via conditional Bayes factor. Technical report, Heidelberg University, 2012.
- [119] R. Kass. Statistical inference: the big picture. *Statistical science*, 26(1):1–9, 2011.
- [120] Robert E. Kass and Adrian E. Raftery. Bayes factorss. *Journal of American Statistical Association*, 90(430):773–795, 1995.
- [121] Gregor Kastner and Sylvia Frühwirth-Schnatter. Ancillarity-sufficiency interweaving strategy (asis) for boosting mcmc estimation of stochastic volatility models. *Computational Statistics & Data Analysis*, 76:408–423, 2014.
- [122] Chang-Jin Kim and Charles R Nelson. Has the us economy become more stable? a bayesian approach based on a markov-switching model of the business cycle. *Review of Economics and Statistics*, 81(4):608–616, 1999.
- [123] Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.
- [124] G Koop, R León-Gonzalez, and R Strachan. Bayesian model averaging in the instrumental variable regression model. *Journal of Econometrics*, 171:237–250, 2012.
- [125] Gary Koop, Dimitris Korobilis, et al. Bayesian multivariate time series methods for empirical macroeconomics. *Foundations and Trends® in Econometrics*, 3(4):267–358, 2010.
- [126] Gary M Koop. *Bayesian econometrics*. John Wiley & Sons Inc., 2003.
- [127] Hideo Kozumi and Genya Kobayashi. Gibbs sampling methods for Bayesian quantile regression. *Journal of Statistical Computation and Simulation*, 81(11):1565—1578, 2011.

- [128] Fabian Krueger. *bvars: Bayesian Analysis of a Vector Autoregressive Model with Stochastic Volatility and Time-Varying Parameters*, 2022. R package version 1.1.
- [129] Tony Lancaster. *An introduction to modern Bayesian econometrics*. Blackwell Oxford, 2004.
- [130] P. Laplace. *Théorie Analytique des Probabilités*. Courcier, 1812.
- [131] Pierre Simon Laplace. Mémoire sur la probabilité de causes par les événements. *Mémoire de l'académie royale des sciences*, 1774.
- [132] E.L. Lehmann and George Casella. *Theory of Point Estimation*. Springer, second edition edition, 2003.
- [133] Alex Lenkoski, Theo S. Eicher, and Adrian Raftery. Two-stage Bayesian model averaging in endogeneous variable models. *Econometric Reviews*, 33, 2014.
- [134] Alex Lenkoski, Anna Karl, and Andreas Neudecker. *Package ivbma*, 2013.
- [135] Eduardo Ley and Mark FJ Steel. On the effect of prior assumptions in Bayesian model averaging with applications to growth regression. *Journal of Applied Econometrics*, 24(4):651–674, 2009.
- [136] D. V. Lindley. The philosophy of statistics. *The Statistician*, 49(3):293–337, 2000.
- [137] D. V. Lindley and L. D. Phillips. Inference for a Bernoulli process (a Bayesian view). *American Statistician*, 30:112–119, 1976.
- [138] Dennis V Lindley. A statistical paradox. *Biometrika*, 44(1/2):187–192, 1957.
- [139] Robert B Litterman. Forecasting with bayesian vector autoregressions—five years of experience. *Journal of Business & Economic Statistics*, 4(1):25–38, 1986.
- [140] J. S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995.
- [141] Gruber Luis and Kastner Gregor. *bayesianVARs: MCMC Estimation of Bayesian Vectorautoregressions*, 2024. R package version 1.0.5.
- [142] D. Madigan and A. E. Raftery. Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.
- [143] D. Madigan, J. C. York, and D. Allard. Bayesian graphical models for discrete data. *International Statistical Review*, 63(2):215–232, 1995.

- [144] David Madigan, Jeremy York, and Denis Allard. Bayesian graphical models for discrete data. *International Statistical Review*, 63:215–232, 1995.
- [145] Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park. MCMCpack: Markov chain Monte Carlo in R. *Journal of Statistical Software*, 42(9):1–21, 2011.
- [146] Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park. *Package MCMCpack*, 2018.
- [147] Tyler H McCormick, Adrian E Raftery, David Madigan, and Randall S Burd. Dynamic logistic regression and dynamic model averaging for binary classification. *Biometrics*, 68(1):23–30, 2012.
- [148] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. London: Chapman and Hall, 1989.
- [149] R. McCulloch and P. Rossi. An exact likelihood analysis of the multinomial probit model. *Journal of Econometrics*, 64:207–240, 1994.
- [150] Robert E McCulloch, Nicholas G Polson, and Peter E Rossi. A bayesian analysis of the multinomial probit model with fully identified parameters. *Journal of econometrics*, 99(1):173–193, 2000.
- [151] Sharon Bertsch McGrayne. *The Theory That Would Not Die: How Bayes’ Rule Cracked the Enigma Code, Hunted Down Russian Submarines, & Emerged Triumphant from Two Centuries of C.* Yale University Press, 2011.
- [152] Karel Mertens and Morten O Ravn. A reconciliation of svar and narrative estimates of tax multipliers. *Journal of Monetary Economics*, 68:S1–S19, 2014.
- [153] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys*, 21:1087–1092, 1953.
- [154] Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, 1996.
- [155] Radford M. Neal. Mcmc using hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng, editors, *Handbook of Markov Chain Monte Carlo*, chapter 5, pages 113–162. Chapman and Hall/CRC, 2011.
- [156] J.A. Nelder and R.W.M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

- [157] J. Neyman and E. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society, Series A*, 231:289–337, 1933.
- [158] Kuschnig Nikolas, Vashold Lukas, Tomass Nirai, McCracken Michael, and Ng Serena. *BVAR: Hierarchical Bayesian Vector Autoregression*, 2022. R package version 1.0.5.
- [159] Agostino Nobile. Comment: Bayesian multinomial probit models with a normalization constraint. *Journal of Econometrics*, 99(2):335–345, 2000.
- [160] T. Park and G. Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.
- [161] G. Parmigiani and L. Inoue. *Decision theory principles and approaches*. John Wiley & Sons, 2008.
- [162] Luis Pericchi and Carlos Pereira. Adaptative significance levels using optimal decision rules: Balancing by weighting the error probabilities. *Brazilian Journal of Probability and Statistics*, 2015.
- [163] Giovanni Petris, Sonia Petrone, and Patrizia Campagnoli. Dynamic linear models. In *Dynamic Linear Models with R*, pages 31–84. Springer, 2009.
- [164] Martyn Plummer, Nicky Best, Kate Cowles, Karen Vines, Deepayan Sarkar, Douglas Bates, Russell Almond, and Arni Magnusson. *Output Analysis and Diagnostics for MCMC*, 2016.
- [165] Andy Pole, Mike West, and Jeff Harrison. *Applied Bayesian forecasting and time series analysis*. Chapman and Hall/CRC, 2018.
- [166] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.
- [167] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023.
- [168] A. Raftery. Bayesian model selection in social research. *Sociological Methodology*, 25:111–163, 1995.
- [169] Adrian Raftery, Jennifer Hoeting, Chris Volinsky, Ian Painter, and Ka Yee Yeung. *Package BMA*, 2012.
- [170] Adrian E Raftery. Bayesian model selection in structural equation models. *Sage Focus Editions*, 154:163–163, 1993.
- [171] Adrian E Raftery, Miroslav Kárný, and Pavel Ettler. Online prediction under model uncertainty via dynamic model averaging: Application to a cold rolling mill. *Technometrics*, 52(1):52–66, 2010.

- [172] Adrian E. Raftery, David Madigan, and Jennifer A. Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191, 1997.
- [173] A.E. Raftery and S.M. Lewis. One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. *Statistical Science*, 7:493–497, 1992.
- [174] A. Ramírez Hassan. The interplay between the Bayesian and frequentist approaches: a general nesting spatial panel data model. *Spatial Economic Analysis*, 12(1):92–112, 2017.
- [175] A. Ramírez Hassan, J. Cardona Jiménez, and R. Cadavid Montoya. The impact of subsidized health insurance on the poor in Colombia: Evaluating the case of Medellín. *Economía Aplicada*, 17(4):543–556, 2013.
- [176] Andrés Ramírez-Hassan. Dynamic variable selection in dynamic logistic regression: an application to internet subscription. *Empirical Economics*, 59(2):909–932, 2020.
- [177] Andrés Ramírez-Hassan and Daniela A Carvajal-Rendón. Specification uncertainty in modeling internet adoption: A developing city case analysis. *Utilities Policy*, 70:101218, 2021.
- [178] Andrés Ramírez-Hassan and David T Frazier. Testing model specification in approximate bayesian computation using asymptotic properties. *Journal of Computational and Graphical Statistics*, 33(3):1–14, 2024.
- [179] Andrés Ramírez-Hassan and Alejandro López-Vera. Welfare implications of a tax on electricity: A semi-parametric specification of the incomplete easi demand system. *Energy Economics*, 131:1–13, 2024.
- [180] R. Ramírez-Hassan, A. Guerra-Urzola. Bayesian treatment effects due to a subsidized health program: The case of preventive health care utilization in medellín (Colombia). *Empirical Economics*, Forthcoming, 2019.
- [181] F. Ramsey. Truth and probability. In Routledge and Kegan Paul, editors, *The Foundations of Mathematics and other Logical Essays*. New York: Harcourt, Brace and Company, London, 1926. Ch. VII, p.156–198.
- [182] A. Ramírez-Hassan and M. Graciano-Londoño. A guided tour of Bayesian regression. Technical report, Universidad EAFIT, 2020.
- [183] Silvio R Rendon. Fixed and random effects in classical and bayesian regression. *Oxford Bulletin of Economics and Statistics*, 75(3):460–476, 2013.

- [184] Sylvia Richardson and Peter J Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792, 1997.
- [185] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2nd edition, 2011.
- [186] Christian P Robert, George Casella, and George Casella. *Introducing monte carlo methods with r*, volume 18. Springer, 2010.
- [187] G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.
- [188] P. Rossi. *Package bayesm*, 2017.
- [189] Peter E Rossi, Greg M Allenby, and Rob McCulloch. *Bayesian statistics and marketing*. John Wiley & Sons, 2012.
- [190] Donald B. Rubin. Using the sir algorithm to simulate posterior distributions. In J. M. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics 3*, pages 395–402. Oxford University Press, 1988.
- [191] Donnald B. Rubin. The Bayesian bootstrap. *The Annals of Statistics*, 9(1):130–134, 1981.
- [192] L. J. Savage. *The foundations of statistics*. John Wiley & Sons, Inc., New York, 1954.
- [193] Robert Schlaifer and Howard Raiffa. *Applied statistical decision theory*. Wiley New York, 1961.
- [194] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.
- [195] Thomas Sellke, MJ Bayarri, and James O Berger. Calibration of p values for testing precise null hypotheses. *The American Statistician*, 55(1):62–71, 2001.
- [196] Steve Selvin. A problem in probability (letter to the editor). *The American Statistician*, 11(1):67–71, 1975.
- [197] Steve Selvin. A problem in probability (letter to the editor). *The American Statistician*, 11(3):131–134, 1975.
- [198] M. Serna Rodríguez, A. Ramírez Hassan, and A. Coad. Uncovering value drivers of high performance soccer players. *Journal of Sport Economics*, 20(6):819–849, 2019.

- [199] Neil Shephard. Partial non-gaussian state space. *Biometrika*, 81(1):115–131, 1994.
- [200] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer, Cham, Switzerland, 4th edition, 2017.
- [201] Susan J Simmons, Fang Fang, Qijun Fang, and Karl Ricanek. Markov chain Monte Carlo model composition search strategy for quantitative trait loci in a Bayesian hierarchical model. *World Academy of Science, Engineering and Technology*, 63:58–61, 2010.
- [202] Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pages 1–48, 1980.
- [203] A. F. M. Smith. A General Bayesian Linear Model. *Journal of the Royal Statistical Society. Series B (Methodological)*, 35(1):67–75, 1973.
- [204] Adrian FM Smith and Alan E Gelfand. Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, 46(2):84–88, 1992.
- [205] Stan Development Team. shinystan: Interactive visual and numerical diagnostics and posterior analysis for Bayesian models., 2017. R package version 2.3.0.
- [206] Stan Development Team. Stan modeling language users guide and reference manual, 2024., 2024.
- [207] Stephen Stigler. Richard price, the first bayesian. *Statistical Science*, 33(1):117–125, 2018.
- [208] M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987.
- [209] Samuel Thomas and WanZhu Tu. Learning hamiltonian monte carlo in r. *The American Statistician*, 75(4):403–413, 2021.
- [210] Luke Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, pages 1701–1728, 1994.
- [211] Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.
- [212] A. Tversky and D. Kahneman. Judgement under uncertainty: heuristics and biases. *Science*, 185:1124–1131, 1974.
- [213] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

- [214] Isabella Verdinelli and Larry Wasserman. Computing bayes factors using a generalization of the savage-dickey density ratio. *Journal of the American Statistical Association*, 90(430):614–618, 1995.
- [215] Stephen G Walker. Modern bayesian asymptotics. *Statistical Science*, pages 111–117, 2004.
- [216] Stephen G. Walker. New approaches to bayesian consistency. *Annals of Statistics*, 32(5):2028–2043, 2004.
- [217] Ronald L. Wasserstein and Nicole A. Lazar. The ASA’s statement on p-values: context, process and purpose. *The American Statistician*, 2016.
- [218] Mike West and Jeff Harrison. *Bayesian forecasting and dynamic models*. Springer Science & Business Media, 2006.
- [219] R. Winkelmann. Health care reform and the number of doctor visits - An econometric analysis. *Journal of Applied Econometrics*, 19(4):455–472, 2004.
- [220] P. Woodward. BugsXLA: Bayes for the common man. *Journal of Statistical Software*, 14(5):1–18, 2005.
- [221] Jeffrey M Wooldridge. *Econometric analysis of cross section and panel data*. MIT press, 2010.
- [222] Jeffrey M. Wooldridge. *Introductory Econometrics: A Modern Approach*. Cengage Learning, Mason, Ohio: South-Western, fifth edition, 2012.
- [223] Jeffrey M. Wooldridge. *Introductory Econometrics: A Modern Approach*. Cengage Learning, Boston, MA, 6th edition, 2016.
- [224] Tomasz Woźniak. Bayesian vector autoregressions. *Australian Economic Review*, 49(3):365–380, 2016.
- [225] Tomasz Woźniak. *bsvars: Bayesian Estimation of Structural Vector Autoregressive Models*, 2024. R package version 3.1.
- [226] Wang Xiaolei and Tomasz Woźniak. *bsvarSIGNs: Bayesian SVARs with Sign, Zero, and Narrative Restrictions*, 2024. R package version 1.0.1.
- [227] A. Zellner. *Introduction to Bayesian inference in econometrics*. John Wiley & Sons Inc., 1996.
- [228] Arnold Zellner. On assessing prior distributions and bayesian regression analysis with g-prior distributions. *Bayesian inference and decision techniques*, 1986.
- [229] S. Ziliak. Guinnessometrics; the Economic Foundation of student’s t. *Journal of Economic Perspectives*, 22(4):199–216, 2008.

Appendix

TABLE 14.1
Libraries and commands in BEsmarter GUI.

Univariate models			
Model	Library	Command	Reference
Normal	MCMCpack	MCMCregress	[146]
Logit	MCMCpack	MCMClogit	[146]
Probit	bayesm	rpprobitGibbs	[188]
Multinomial(Mixed) Probit	bayesm	rmnpGibbs	[188]
Multinomial(Mixed) Logit	bayesm	rmnlIndepMetrop	[188]
Ordered Probit	bayesm	rordprobitGibbs	[188]
Negative Binomial(Poisson)	bayesm	rnegbinRw	[188]
Tobit	MCMCpack	MCMCtobit	[146]
Quantile	MCMCpack	MCMCquantreg	[146]
Bayesian bootstrap	bayesboot	bayesboot	[7]
Multivariate models			
Model	Library	Command	Reference
Multivariate	bayesm	rmultireg	[188]
Seemingly Unrelated Regression	bayesm	rsurGibbs	[188]
Instrumental Variable	bayesm	rivGibbs	[188]
Bivariate Probit	bayesm	rmvpGibbs	[188]
Hierarchical longitudinal models			
Model	Library	Command	Reference
Normal	MCMCpack	MCMChregress	[146]
Logit	MCMCpack	MCMChlogit	[146]
Poisson	MCMCpack	MCMChpoisson	[146]
Bayesian model averaging			
Model	Library	Command	Reference
Normal (BIC)	BMA	bicreg	[169]
Normal (MC ³)	BMA	MC3.REG	[169]
Normal (instrumental variables)	ivbma	ivbma	[134]
Logit (BIC)	BMA	bic.glm	[169]
Gamma (BIC)	BMA	bic.glm	[169]
Poisson (BIC)	BMA	bic.glm	[169]
Diagnostics			
Diagnostic	Library	Command	Reference
Trace plot	coda	traceplot	[164]
Autocorrelation plot	coda	autocorr.plot	[164]
Geweke test	coda	geweke.diag	[164]
Raftery & Lewis test	coda	raftery.diag	[164]
Heidelberger & Welch test	coda	heidel.diag	[164]

TABLE 14.2Datasets templates in folder *DataSim*.

Univariate models		
Model	Data set file	Data set simulation
Normal	11SimNormalmodel.csv	11SimNormal.R
Logit	12SimLogitmodel.csv	12SimLogit
Probit	13SimProbitmodel.csv	13SimProbit.R
Multinomial(Mixed) Probit	14SimMultProbmodel.csv	14SimMultinomialProbit.R
Multinomial(Mixed) Logit	15SimMultLogitmodel.csv	15SimMultinomialLogit.R
Ordered Probit	16SimOrderedProbitmodel.csv	16SimOrderedProbit.R
Negative Binomial(Poisson)	17SimNegBinmodel.csv	17SimNegBin.R
Tobit	18SimTobitmodel.csv	18SimTobit.R
Quantile	19SimQuantilemodel.csv	19SimQuantile.R
Bayesian bootstrap	41SimBootstrapmodel.csv	41SimBootstrap.R
Multivariate models		
Model	Data set file	Data set simulation
Multivariate	21SimMultivariate.csv	21SimMultReg.R
Seemingly Unrelated Regression	22SimSUR.csv	22SimSUR.R
Instrumental Variable	23SimIV.csv	23SimIV.R
Bivariate Probit	24SimMultProbit.csv	24SimMultProbit.R
Hierarchical longitudinal models		
Model	Data set file	Data set simulation
Normal	31SimLongitudinalNormal.csv	31SimLongitudinalNormal.R
Logit	32SimLongitudinalLogit.csv	32SimLongitudinalLogit.R
Poisson	33SimLongitudinalPoisson.csv	33SimLongitudinalPoisson.R
Bayesian model averaging		
Model	Data set file	Data set simulation
Normal (BIC)	511SimNormalBMA.csv	511SimNormalBMA.R
Normal (MC ³)	512SimNormalBMA.csv	512SimNormalBMA.R
Normal (instrumental variables)	513SimNormalBMAivYXW.csv 513SimNormalBMAivZ.csv	513SimNormalBMAiv.R
Logit (BIC)	52SimLogitBMA.csv	52SimLogitBMA.R
Gamma (BIC)	53SimGammaBMA.csv	53SimGammaBMA.R
Poisson (BIC)	53SimPoissonBMA.csv	53SimPoissonBMA.R

TABLE 14.3Real datasets in folder *DataApp*.

Univariate models		
Model	Data set file	Dependent variable
Normal	1ValueFootballPlayers.csv	log(Value)
Logit	2HealthMed.csv	Hosp
Probit	2HealthMed.csv	Hosp
Multinomial(Mixed) Probit	Fishing.csv	mode
Multinomial(Mixed) Logit	Fishing.csv	mode
Ordered Probit	2HealthMed.csv	MedVisPrevOr
Negative Binomial(Poisson)	2HealthMed.csv	MedVisPrev
Tobit	1ValueFootballPlayers.csv	log(ValueCens)
Quantile	1ValueFootballPlayers.csv	log(Value)
Bayesian bootstrap	1ValueFootballPlayers.csv	log(Value)
Multivariate models		
Model	Data set file	Dependent variable
Multivariate	4Institutions.csv	logpcGDP95 and PAER
Seemingly Unrelated Regression	5Institutions.csv	logpcGDP95 and PAER
Instrumental Variable	6Institutions.csv	logpcGDP95 and PAER
Bivariate Probit	7HealthMed.csv	$y = [\text{Hosp } \text{SHI}]'$
Hierarchical longitudinal models		
Model	Data set file	Dependent variable
Normal	8PublicCap.csv	log(gsp)
Logit	9VisitDoc.csv	DocVis
Poisson	9VisitDoc.csv	DocNum
Bayesian model averaging		
Model	Data set file	Dependent variable
Normal (BIC)	10ExportDiversificationHHI.csv	avghhi
Normal (MC ³)	10ExportDiversificationHHI.csv	avghhi
Normal (instrumental variables)	11ExportDiversificationHHI.csv 12ExportDiversificationHHIInstr.csv	avghhi and avgldpcap
Logit (BIC)	13InternetMed.csv	internet
Gamma (BIC)	14ValueFootballPlayers.csv	log market value
Poisson (BIC)	15Fertile2.csv	ceb