Università della Svizzera italiana

Faculty of Economics

Università della Svizzera italiana

Faculty of Economics

CHANGE POINT ANALYSIS IN FINANCIAL TIME SERIES

Author:
Laima Masaleviciute

Supervisors:
Prof. Antonietta Mira
Prof. Stefano Peluso

This thesis is submitted for the degree of
Master of Science in Economics, Major in Finance

February, 2022

Contents

Acknowledgements

I express my deepest gratitude to my professors A. Mira (Università della Svizzera italiana, USI) and S. Peluso (Università degli Studi di Milano-Bicocca) for allowing me to join their research team during the work on this thesis. This thesis would not be complete without their underlying trust, feedback, and support.

I am also extremely grateful to my boyfriend Martin Odermatt for two matters: (1) providing technical support required to run the experiments, such as lending out his laptops and backing up my machine, and (2) his moral support during my ups and downs. Also, many thanks to my neighbor, Carolyne, for her time in reviewing and correcting this work.

During my studies at USI, I had the pleasure of meeting my friends and best study colleagues, Samanta and Francesco. Their passion for finance and quantitative topics has always motivated me to take up personal challenges like this thesis. I sincerely thank them for sharing their passions with me. In addition, I would like to sincerely thank my career mentor and friend, Christelle, who patiently listened to me and directed my thoughts in the right direction.

The remote support from the Masalevičius family in Lithuania was particularly helpful during the studies. My parents, Roma and Laimonas, and my brothers, Povilas, Dovydas, and Martynas, have supported me morally and sent their positive vibes through countless video calls. I would also like to extend my gratitude to my beloved grandmother Joana, who supported me with her constant warm wishes and deepest prayers. Thanks to my dear godparents, Lina and Dainius, and their daughter Evelina, who have sent their love through gift packages.

I am also very thankful to my second family in Switzerland. Special thanks go to

my boyfriend's parents, Frena and Peter, beloved auntie Antonia with her partner Mike and the family Reymont. They have always wished me good luck and success during my study years.

Big thanks should also go to my 'gurus' Magdalena, Irma, Monika, and Andželika, who have encouraged me and had the incredible patience to listen to my struggles. I also thank my friends from Zurich - Vale, Vali, Peter, Jenni, and Manuel. I always felt and appreciated your support.

Last but not least, a special thanks go to my ex-boss Markus and his wife, Champa. I thank them for all their support before and during my master's studies and for my future career.

Dedication

I dedicate this work to my parents,
Roma Masalevičienė and Laimonas Masalevičius,
for their *unconditional* love and support.

Abstract

When modeling long economic and financial time series, several problems of parameter estimation arise, often due to the ignorance of structural breaks. To overcome these problems, change point methods can be incorporated into the models.

In this thesis, we implement the sparse change point vector autoregressive model (*CP-VAR(p)*) using the Just Another Gibbs Sampler (JAGS) program. This model is known for its rich parameterization, which can lead to a large number of parameters when dealing with multiple regimes. To mitigate this problem, [Dufays et al., 2021] proposed shrinkage priors to shrink the parameters of irrelevant regimes to zero. The aim of this study is to use the sparse *CP-VAR(p)* model to detect the number of change points in mean parameters in time series.

To evaluate the robustness of the model in detecting change points, we first perform a simulation study on a bivariate *CP-VAR(1)* data generation process using JAGS. We can identify two out of three change points in a simulated time series of $T = 500$. However, our model had slightly poorer change point detection performance in the shorter time series of T =120. Also, our model can identify the absence of the change points in most cases. Unfortunately, the computational power of our machine limits the scope of a more extensive simulation study.

We then apply the model to two financial and economic datasets that include monthly observations between 2013 and 2022. The first dataset includes log returns of stylized stock indices (value and growth) and changes in the Consumer Price Index (CPI), while the second dataset includes variables from energy markets and two macro variables. In both experiments, we set the lag to p=1 and find that no change point is detected with the methodology used. However, we identify some interesting and significant variables.

In summary, our study demonstrates the effectiveness of our sparse *CP-VAR(p)* model in detecting change points on simulated data. We also show the capabilities of our model in identifying the change points and significant parameter values in time series of financial and economic data.

List of Tables

List of Figures

Chapter 1

Introduction

We begin this chapter by describing the context of our work. We then set the goals to be achieved by the end of the thesis and provide an outline of our work.

1.1 Context

The analysis performed in this thesis has evolved thanks to a recent research paper written by [Dufays et al., 2021]. The research "Sparse change-point VAR models" was published in the Journal of Applied Econometrics. The researchers identified several problems in the paper related to modeling long time series, i.e., bias in parameter estimates, misleading interpretation, and severe forecasting errors. The issues are linked to the ignorance of structural breaks that are occasionally present in macroeconomic and financial time series. To overcome these problems, change point methods can be incorporated into the models.

The change point models for the univariate case are well studied and covered in the literature by [Chib, 1998], [Hashem Pesaran et al., 2006], [Koop and Potter, 2007], [Giordani and Kohn, 2008], [Maheu and Gordon, 2008], and [Peluso et al., 2019]. However, other difficulties arise when working in a multivariate setting. Breaks in different time series may occur at different times. Also, the large number of parameters, even in small models, makes model estimation a difficult task.

Despite the above-mentioned issues, [Dufays et al., 2021] state the importance of modeling structural breaks in the multivariate setting. For example, some variables may significantly affect other variables before the break occurs. Or, some variables may become significant only after the break occurs. Several relevant studies can be found in the literature related to the analysis of the structural breaks in the multivariate setting, [Pettenuzzo and Timmermann, 2011], [Chib and Kang, 2013], [Maheu and Song, 2018].

The research paper of our interest proposes a sparse change point vector autoregressive model, the *sparse CP-VAR(p)*, for applications in economics and finance. In the proposed model, the structural breaks are subject to several considerations:

1. breaks are allowed to appear in the autoregressive parameters, the off-diagonal VAR parameters, or the covariance matrix parameters,

2. the breaks in the mean parameters can appear separately from the breaks in the covariance matrix,

3. the number of breaks in the model is directly inferred thanks to the introduction of the shrinkage priors.

Related to the last point, the inclusion of shrinkage priors provides some additional benefits, such as:

1. imply the number of breaks and regimes in the model directly, without the need to compute the marginal likelihood,

2. shrink irrelevant *CP-VAR(p)* parameters towards zero, thus addressing the issue of over-parameterization.

1.2 Goals

G1: Implementation of the Sparse *CP-VAR(p)* Model in the JAGS Program

We analyze the model developed by [Dufays et al., 2021] and investigate how it can be implemented using high-performance statistical software (JAGS). JAGS is designed to perform Bayesian analysis using Markov Chain Monte Carlo (MCMC) simulations. [Lee, 2019] addresses two reasons why the computational Bayesian methods are rarely adopted: (1) the implementation of Bayesian models requires "nontrivial algorithms", and (2) the difficulty of extending and applying the existing methods found in the literature. The JAGS software allows the implementation of a Bayesian model as a graphical model that can be easily read, interpreted, and extended. We detail the implementation of the model in Chapter 3.

G2: Test the model's performance and robustness through the simulation study

After the implementation of the model, we aim to test the model's performance on simulated data where the true parameters are known. The simulation study shows how well our model is able to detect the change points. It also allows us to check how our model performs in a setting similar to the one we face with real data. We discuss model testing in Chapter 4.

G3: Investigate the interrelationships between financial and economic variables through the defined experiments

We aim to identify the interrelationships between financial and economic variables. The interrelationships are specified by the sparse *CP-VAR(p)* model through the breakpoints and auto-regressive parameters. We are interested in investigating the stock and commodity markets. These two markets behave differently at different stages of the business cycle as identified by the [Credit Suisse, 2019] report. We also believe that macroeconomic variables can help to introduce significant parameters in our study. Chapter 5 details the investigation.

1.3 Outline of the Thesis

Chapter 2: Background
Provides an overview of the statistical methods used throughout the thesis. Introduces how and where these methods are applied in the thesis.

Chapter 3: Methodology
Describes the statistical model developed by [Dufays et al., 2021]. Presents the JAGS implementation methodology of the sparse *CP-VAR(p)*. Introduces the MCMC diagnostic tools.

Chapter 4: Simulation Study
Describes the simulated experiments and provides a summary of the results.

Chapter 5: Experimentation
Presents the finance and economics applications with real data. Introduces the research questions to be answered. Includes the descriptions of the selected data and the experimental setup. Summarizes the experimental results and answers the research questions.

Chapter 6: Conclusions
Reiterates the achievements of the thesis. Reflects on the main results of the simulation study and the experiments with real data. Presents aspects to be considered in future work.

Chapter 2

Background

This chapter describes the basic theoretical concepts used throughout the thesis. We start with change point models and the purpose of their use. We present the fundamentals of Bayesian computational methods and later go into the theory of Markov Chain Monte Carlo (MCMC) simulations.We describe another topic relevant to the development of this thesis - the multivariate regression technique - vector autoregression (VAR). Finally, we introduce how these methods will be used in our thesis work.

2.1 Change Point Models

The problem of finding a change point is encapsulated in finding two components: (1) the number of change points in an ordered sequence, and (2) their locations. Further analysis can be done by finding the magnitude of the parameter that experienced the change, or other parameters that are affected by the change point. In the literature, change point models can be found under various names such as change detection, breakpoints, structural breaks, segmentation, or regime-switching models. The applications of change point models are countless: from quality control to behavioral studies to econometric research. Change point models belong to the large class of statistical models known as *non-linear models*, [Chatfield and Xing, 2020].

Change point models can also be found in the literature based on many different combinations of statistical techniques. The paper [van den Burg and Williams, 2020] distinguishes such change point models:

1. online vs. offline (sequential data vs. fixed data),
2. univariate vs. multivariate,
3. model-based vs. non-parametric,
4. Bayesian vs. frequentist.

The first literature on change point detection models can be found in the early 1950s. These applications were focused on industrial quality control. [Page, 1954] and [Page, 1955] attempted to detect a change in the mean of normally distributed variables. One of the earliest Bayesian change point models found is focused on object tracking problems accounting for changes in direction when the change points "obey an arbitrary specified a priori probability distribution". This method was only relevant to the case study where the size of changes is normally distributed with mean zero, [Chernoff and Zacks, 1964]. A single change point model with a Bayesian analytical approach was developed by [Smith, 1975], where he analyzed cases of the normal and binomial distributions. Later Bayesian methods found in the literature use sampling methods, such as Monte Carlo Markov chains, for change point inference. [Chib, 1998], for example, introduced a sampling procedure of the latent state variable indicating the current regime. The change points are then inferred from the discrete states. Other cases of change point analysis are examined through the prism of regressions. An alternative model for change point analysis found in earlier work is described by [Holbert, 1982] as the switching linear model.

We focus on Bayesian change point analysis in the multivariate VAR(p) offline setting. The model, which will be discussed later, is much more complex than the examples described below.In the following sections, we provide the theoretical background for a thorough understanding of the methodological part of this thesis. If the reader is fairly familiar with the topic, we suggest fast forward to the methodological part.

2.2 Fundamentals of Bayesian inference

The Bayes theorem is named after the English mathematician, Thomas Bayes (1702-1761). His famous theorem was discovered and published only after his death thanks to his friend, Richard Price [T. Bayes, 1763]. Another figure of Bayesian statistics was Pierre-Simon Laplace (1749-1827). He discovered and developed the methodology independently from Bayes, [Dale, 1985].

The essence of Bayesian analysis is the reallocation of credibility over the range of possible options after observing evidence. A meaningful result is obtained about each option after the process is repeated many times. It is in our interest to identify the credibility of the parameters lying in the data model.

The reallocation of credibility can be expressed mathematically by the concept of *conditional probability*. It is yet worth looking at the example from [Kruschke, 2015] in Figure 2.1 to illustrate this fact. The top panel shows the table of probabilities of persons having different hair and eye color. The probabilities in the middle of the table refer to *joint probabilities*, or the probability of having both different hair and eye colors. Whereas the *marginal probabilities*, the ones that lie in the margins of the table, refer to the sum of the joint probabilities across rows or columns, respectively. Note that the sum of row marginals (or, column marginals) is equal to one.

The *conditional probability* gives a probability of an outcome given that another outcome has occurred. If we denote the hair color as parameter c for the columns and the eye color as r for the rows, we can compute the conditional probability. This answers the question: given a particular eye color, r, what is the probability that a person has a particular hair color, c? In short, the conditional probability is the joint probability divided by the marginal probability of the conditioning event. Since the marginal probability is the sum of the joint probabilities across cells, we can further write the conditional probability as:

$$p(c|r) = \frac{p(r,c)}{p(r)} = \frac{p(r,c)}{\sum_{c^*} p(r,c^*)} \tag{2.1}$$

$$p(c|r) = \frac{p(r,c)}{p(r)} = \frac{p(r,c)}{\int_c p(r,c)dc} \tag{2.2}$$

where c^* denotes every possible hair color option. If the row variables are continuous, we need to replace the sum with the integral. This operation is called *marginalizing over c* or *integrating out* the variable c. We can do the same operation again, but instead of c, we can marginalize r. The second panel in Figure 2.1 displays this relation between conditional and marginal probabilities in a visual way.

We can relate conditional probabilities directly to Bayesian inference. Suppose the hair colors, c, are our parameter values of interest, and the eye colors, r, are the observed evidence. We can now update our knowledge about the probability of a person having a particular hair color given the evidence about eye color. This is a trivial book-based example. But suppose we are interested in working with economic and financial data. The data is our evidence, which lies in the rows of the table r. Then, assume that we want to find the autoregressive parameters of the *VAR(p)* model lying in the columns on the table, c. Using Bayesian inference, we can derive the posterior probabilities of the autoregressive parameters, the credible parameters, given the model and the data. We will introduce the *VAR(p)* model in a later section. Nevertheless, these two hypothetical examples demonstrate the essence of Bayesian inference.

The well-known Bayes formula is derived in the next steps. First, we recall the definition of conditional probability and write:

$$p(c|r) = \frac{p(r,c)}{p(r)}$$
$$p(r|c) = \frac{p(r,c)}{p(c)}$$

Next, both sides of the above equations are multiplied by $p(r)$ for the first equation and $p(c)$ for the second equation. We get the following expressions:

$$p(c|r)p(r) = p(r,c)$$
$$p(r|c)p(c) = p(r,c)$$

We then equate the two expressions:

$$p(c|r)p(r) = p(r|c)p(c)$$

Furthermore, we divide both sides of the above equation by $p(r)$ and arrive at:

$$p(c|r) = \frac{p(r|c)p(c)}{p(r)} \tag{2.3}$$

Finally, the denominator is rewritten as the sum of the conditional and marginal probabilities:

$$p(c|r) = \frac{p(r|c)p(c)}{\sum_{c^*} p(r|c^*)p(c^*)} \tag{2.4}$$

The last two equations, 2.3 and 2.4, define the *Bayes' Rule*.

A generalized example is given in the third panel of Figure 2.1, where the evidence, or data, is noted with the variable D. The variable θ refers to a generic parameter of interest. This panel presents the generic problem setting where the row variables are resemble the data and the column variables define the parameter values.

In the literature, Bayes' rule is more often found in the standardized notation:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{2.5}$$

where the denominator in the discrete case can be written as:

$$p(D) = \sum_{\theta^*} p(D|\theta^*)p(\theta^*)$$

and the denominator in the continuous case has this form:

$$p(D) = \int p(D|\theta^*)p(\theta^*)d\theta^*$$

The Bayes' rule in equation 2.5 provides the updated knowledge, the posterior,

Eye color	Hair color				Marginal (Eye color)
	Black	Brunette	Red	Blond	
Brown	0.11	0.20	0.04	0.01	0.37
Blue	0.03	0.14	0.03	0.16	0.36
Hazel	0.03	0.09	0.02	0.02	0.16
Green	0.01	0.05	0.02	0.03	0.11
Marginal (hair color)	0.18	0.48	0.12	0.21	1.0

Row	Column			Marginal
	...	c	...	
$\vdots$		$\vdots$		
r	$\cdots$	$p(r, c) = p(r\|c)\, p(c)$	$\cdots$	$p(r) = \sum_{c^*} p(r\|c^*)\, p(c^*)$
$\vdots$		$\vdots$		
Marginal		$p(c)$		

Data	Model parameter			Marginal
	...	θ value	...	
$\vdots$		$\vdots$		$\vdots$
D value	$\cdots$	$p(D, \theta) = p(D\|\theta)\, p(\theta)$	$\cdots$	$p(D) = \sum_{\theta^*} p(D\|\theta^*)\, p(\theta^*)$
$\vdots$		$\vdots$		$\vdots$
Marginal	...	$p(\theta)$	...	

Figure 2.1: Joint and marginal probabilities in form of a table.

noted as $p(\theta|D)$. The prior information is encoded in the term $p(\theta)$. The evidence, or data, is summarized by $p(D)$, which is also called a *marginal likelihood*. The term $p(D|\theta)$, also known as *likelihood*, is "the probability that the data D could be generated by the model with the parameter value θ", [Kruschke, 2015]. One can think of this expression as if the data were fixed and parameter θ was variable and unknown. In contrast, the parameters of common probability distributions are

thought to be fixed and data were variable. Therefore, θ^* notes that every value of the parameter is taken, while θ takes a specific value.

Knowing the mathematical expression of Bayes' rule from 2.5, we can therefore identify the main steps for simple Bayesian data analysis:

1. Understand the problem and define the parameters of interest, θ,
2. Find the *likelihood* function, $p(D|\theta^*)$,
3. Choose the appropriate priors, $p(\theta)$
4. Compute the *marginal likelihood of the data*, $p(D)$,
5. Combine the components into a mathematical model,
6. Collect the data, D, and apply the model to obtain the posterior, $p(\theta|D)$,
7. Analyze the results.

[Kruschke, 2015] provides several compelling examples showing some insights about the influence of model elements on the posterior distributions. In Figure 2.2, on the top panel, the same prior is printed on different scales. The top left panel shows how the prior distribution aligns with the posterior when the sample is small. On the other hand, as the sample size increases, the posterior becomes more similar to the likelihood than to the prior. The bottom panel shows the contrasts between the weak and strong priors with two sample sizes. The left panel depicts that a flat prior has little or no effect on the posterior when the sample size is small. The right panel illustrates the strong prior with a sharp peak that significantly affects the likelihood even when the sample size is large.

2.3 Monte Carlo Markov Chain (MCMC)

Computations in Bayesian models can be complex, if not impossible. Often just computing the *marginal likelihood*, $p(D)$, from Bayes' formula in Equation 2.5 can

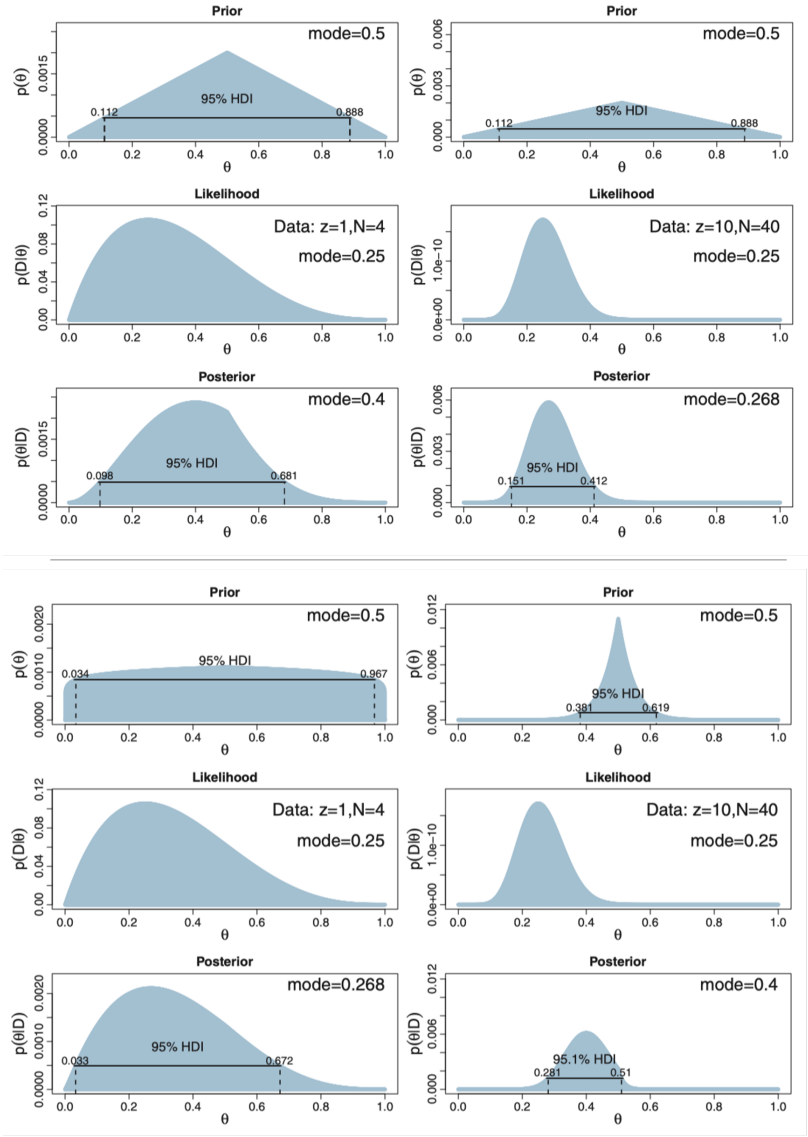

Figure 2.2: Effects of sample size and prior distribution on the posterior

be a difficult exercise. Integration in the continuous case may not be possible in realistic and complex models.

When researchers could not use the computing power of machines in the past, the model components were chosen so that integration would be an achievable task.

For example, *conjugate* prior distributions were such that it was possible to obtain a closed-form expression for the posterior. The models tended to have simple likelihood functions for the same reason.

As computing power increased, integral approximation methods became more visible. The numerical approximation of the integral made it possible to build large models with hundreds of parameters, as is often the case in real-world problems.

In addition to the integral stochastic approximation technique, Markov chain Monte Carlo (MCMC) methods have become increasingly popular for approximating the posterior. The reason is that these techniques do not require computing the integral that appears in the denominator of Bayes' formula, Equation 2.5. These methods involve "randomly sampling a large number of representative combinations of parameter values from the posterior distribution", [Kruschke, 2015]. The part of the name 'Monte Carlo' refers to any simulation of random samples. This name appeals to many card and dice games from the famous casino. The two sampling methods introduced later are the types of Monte Carlo simulation. The part of the name 'Markov Chain' is dedicated to a mathematician, Andrey Markov (1856 - 1922). The first-order Markov process is a process where the new value in the process can depend only on the current value, but not on any other value in the process. In other words, this process has no information about its history.

MCMC techniques play a fundamental role in the data analysis project of this thesis. Therefore, the following sections will provide an overview of two major sampling techniques, namely the Metropolis algorithm and Gibbs sampling. Using these methods, a complex Bayesian analysis can be performed using modern software and a personal computer.

2.3.1 Metropolis Algorithm

The Metropolis algorithm is named after one of the authors of a paper published in the Journal of Chemical Physics, [Metropolis et al., 1953]. The algorithm allows the approximation of the posterior distribution, $p(\theta|D)$, by a large representative sample of parameter values, θ, from this distribution. Then, the collection of sampled θ's

is used to estimate the form of the posterior. As the sample size increases, the approximated posterior becomes more accurate and, on average, tends to converge to the true values.

The Metropolis method is useful because it does not require calculating the *marginal likelihood* of the data, which overcomes the complexity of the *Bayes' rule*. This algorithm is interesting for Bayesian analysis because it allows exploring the target distribution $P(\theta)$, or the posterior, without normalizing it by $p(D)$. The target posterior distribution is proportional to the *likelihood* times the *prior*, $p(D|\theta)p(\theta)$. Thus, the algorithm essentially requires only the *prior* and the *likelihood* to be able to approximate the posterior.

The Metropolis method consists of several steps:

1. generating a parameter value from the *proposal distribution* to obtain a proposed position, $\theta_{proposed}$,

2. evaluating the target distribution at the proposed value, $P(\theta_{proposed})$, and computing the p_{move},

3. generating random values from a uniform distribution, *U(0,1)*, and accepting or rejecting the move according to the probability, p_{move},

4. the steps are repeated many times in order to generate a Markov chain through the parameter space and obtain the approximated density of the posterior.

These steps will now be described in more detail.

First, the *proposal distribution* is defined as the range of possible moves to be proposed and the probability of each proposal. The form of the proposal must be such that it is able to explore the parameter space well. A simple example of a proposal distribution is a normal distribution centered on the current position, $\theta_{current}$. The scale parameter in the proposal distribution allows one to define the size of the step in the random-walk process.

Second, after obtaining the proposed move $\theta_{proposed}$ from the proposal distribution, the algorithm has to accept or reject the move. This decision is made according

to the probability calculated by the expression:

$$p_{move} = min(\frac{P(\theta_{proposed})}{P(\theta_{current})}, 1)$$

Third, the random value is generated from a uniform distribution between 0 and 1. If the generated value is between 0 and p_{move}, the move is accepted. Obviously, if $P(\theta_{proposed}) > P(\theta_{current})$, the probability of a move is equal to 1, thus, any uniformly generated value will be accepted and the move will occur. After the move is accepted, the previously proposed position becomes the current position in the next step. If the move is rejected, the current position is still the current position in the next step.

Finally, after repeating the previous steps many times, the random walk process stabilizes. Then, the approximated posterior distribution can be obtained. The described process and the example in Figure 2.3 are retrieved from [Kruschke, 2015]. The bottom panel shows the possible values of the target distribution. The random walk process is shown in the middle, where one of the two proposed moves (move left or right) is accepted. Finally, the histogram of the visited position in the random walk is displayed at the top panel.

The disadvantage of the Metropolis algorithm is the selection and tuning of the proposal distribution. Improper tuning of the parameters in the proposal distribution can result in a large proportion of samples being rejected. This is because the steps of a random walk through the parameter space are either too large or too small.

Note that the Metropolis algorithm we described is a special case of the general Metropolis-Hastings algorithm, which was developed in 1970, [Hastings, 1970]. The original Metropolis algorithm is limited to a symmetric proposal distribution. An example of a symmetric distribution is a normal distribution. The assumption of the symmetric distribution was dropped in the generalized Metropolis-Hastings algorithm.

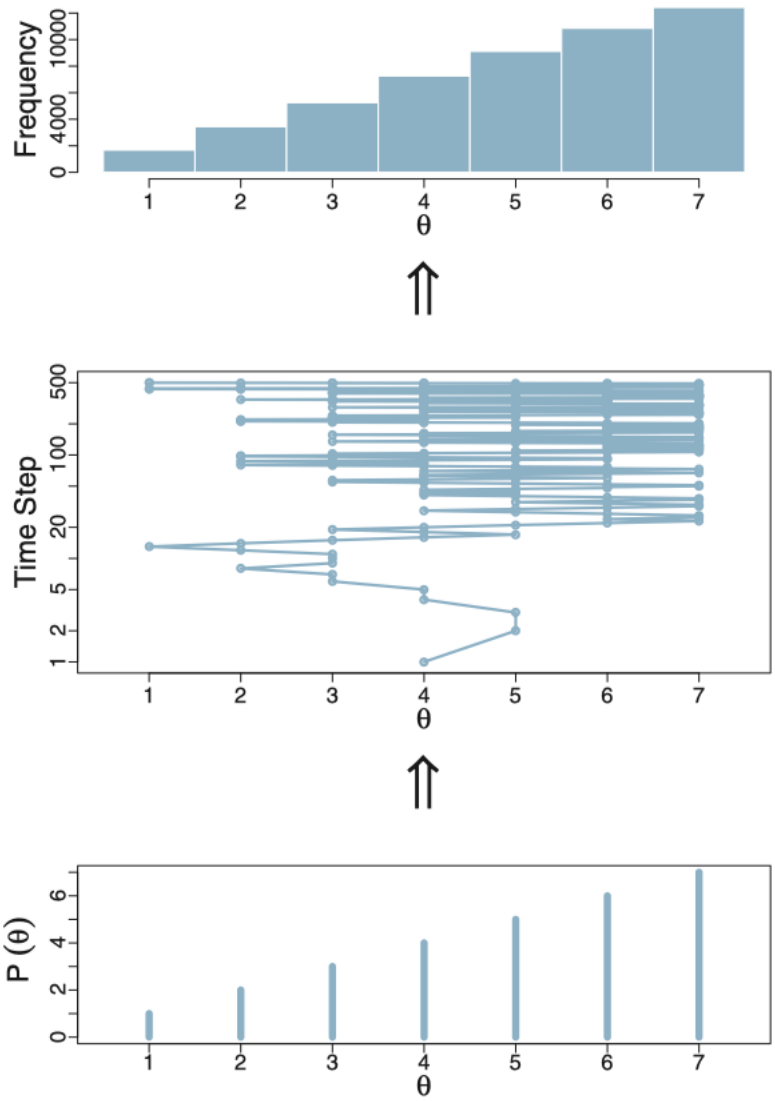

Figure 2.3: Example of the Metropolis algorithm in a discrete case.

2.3.2 Gibbs Sampling

Gibbs sampler will be used by the JAGS software to perform the analysis later described in the thesis. Therefore, it is important to draw our attention to this algorithm.

The Gibbs sampler is a special case of the Metropolis-Hastings algorithm. The algorithm was first introduced by the physicist Josiah Willard Gibbs (1893-1903). However, it was published about a hundred years later in the article by [Gelfand and Smith, 1990] written for the statistical community.

The technique is similar to the random walk through the parameter space as we have seen in the Metropolis algorithm. The difference between the Gibbs sampling and the Metropolis algorithm is how the proposed step is generated. In the fixed scan Gibbs sampler, the proposed parameters are selected in a looped sequence: $\theta_1, \theta_2, \theta_3, ..., \theta_1, \theta_2, \theta_3, ...$ However, it is necessary to generate a new proposed value directly from a so-called "fully conditional" distribution, i.e. the conditional distribution of the parameter to be updated obtained conditioning on the data and on all other parameters, $p(\theta_i|\theta_{j\neq i}, D)$. "Because the proposal distribution exactly mirrors the posterior probability for that parameter, the proposed move is always accepted", [Kruschke, 2015]. Indeed one can verify that using the full conditionals as proposals in a Metropolis-Hastings algorithm, the resulting acceptance probability is exactly equal to 1. In the two-parameter case, to generate a new value for θ_1, we need to be able to draw it from its conditional distribution conditioning on θ_2 and the data, $p(\theta_1|\theta_2, D)$. Next, given θ_1, we can draw a new value for θ_2, but conditioning on θ_1, $p(\theta_2|\theta_1, D)$. This process is well illustrated in Figure 2.4 from [Kruschke, 2015]. In the top panel, on the left, the θ_1 is sampled conditioning on θ_1 from $p(\theta_1|\theta_2, D)$. Conversely, the right figure shows the process of sampling θ_2 conditioning on θ_1 from $p(\theta_2|\theta_1, D)$. Each slice in the graphs illustrates the corresponding conditional distributions.

Thus, the algorithm obtains an approximation of the posterior distribution by sampling from all conditional distributions, $p(\theta_i|\theta_{j\neq i}, D)$. Sampling from full conditionals provides an advantage to the Gibbs sampler. No values are rejected as compared to the Metropolis algorithm, and there is no need to tune the proposal distribution. The posterior approximations are illustrated in Figure 2.4. The left panel displays the chain with every intermediate step, changing one parameter at a time. The same algorithm is shown on the right panel after the completed iterations along both parameters. These posterior approximations resemble the true posterior

in the top panel.

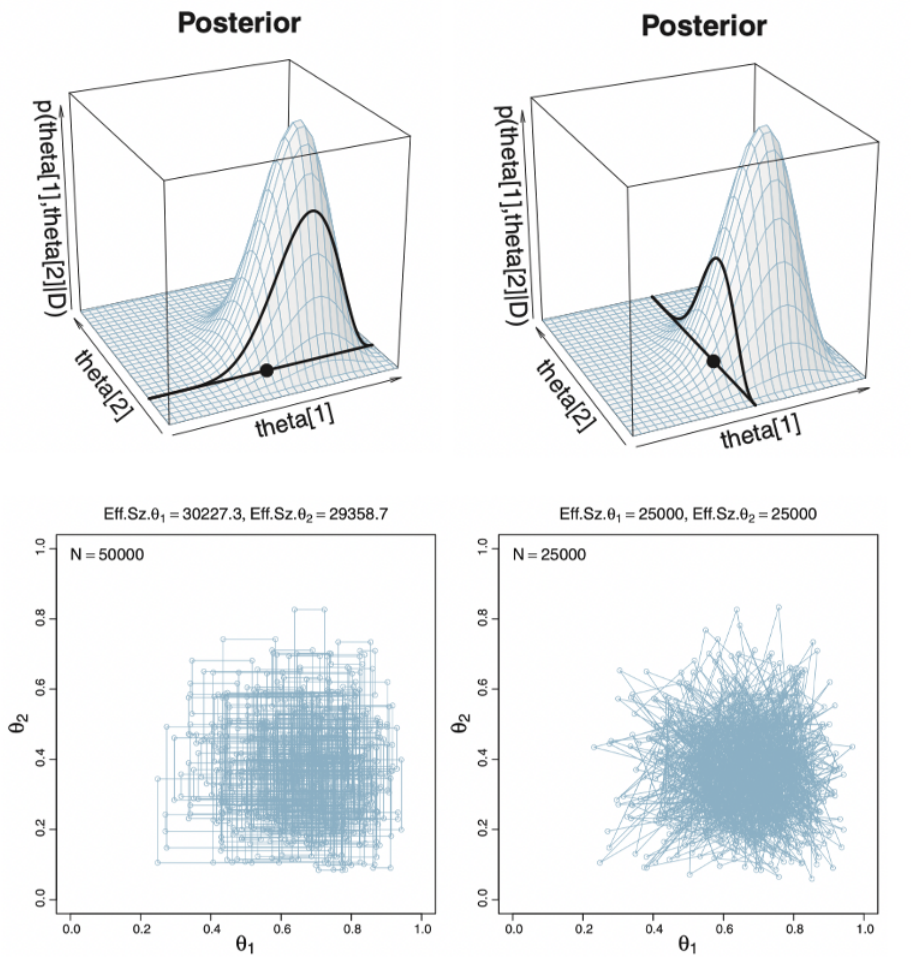

Figure 2.4: Illustration of the Gibbs sampling technique.

2.4 VAR Models

Multivariate models are useful when a researcher aims to understand the relationships between multiple time series. These interdependencies between the time series are described by the crosscorrelation function. Vector autoregression (VAR) is probably one of the most prominent models in the class of multivariate mod-

els that play an important role in this thesis. In this section, we often refer to [Chatfield and Xing, 2020], who provide an informative introduction to time series analysis, also covering multivariate models.

Multivariate models allow researchers to understand the complex structure behind the system being analyzed, which in turn provides an opportunity to make better predictions. Also, the multivariate models are very useful in economics, where the data are characterized by "*feedback* in a *closed-loop* system" where 'outputs' affect the 'inputs', [Chatfield and Xing, 2020]. The authors give an example from economics, where the rising prices impact the rise in wages, which further impacts the rise of prices.

Compared to the univariate case, the data analysis project is much more complex in the multivariate setting. The larger choice of candidate models is one of the reasons for complexity. The second reason is related to the rich parameterization. The frequentist *VAR(p)* models often suffer from the risk of overfitting and estimating imprecise parameters due to the large parameterization, [Elliott and Timmermann, 2013]. The Bayesian perspective of statistics overcomes such issues by allowing the flexible choice of priors with the shrinkage effect towards zero, [Chatfield and Xing, 2020]. Shrinkage priors will be discussed later in more detail.

From the definition of the *VAR(p)* model, we assume that the variables are stationary processes. In short, we assume that the variables are scaled to zero mean.

Before we generalize the model, we must first provide a VAR(1) model as an example. A *VAR(1)* model with $n = 2$ variables and $p = 1$ lags can be written in this form:

$$y_{1,t} = a_{10}1 + a_{11}y_{1,t-1} + a_{12}y_{2,t-1} + \varepsilon_{1,t} \tag{2.6}$$

$$y_{2,t} = a_{20}1 + a_{21}y_{1,t-1} + a_{22}y_{2,t-1} + \varepsilon_{2,t} \tag{2.7}$$

VAR(p) in matrix form:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 1 \\ y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{bmatrix}$$

Summarizing the model in more compact matrix notation, we have:

$$y_t = Ax_t + \varepsilon_t$$

where A is 2×3 matrix, $x_t^{'} = [1 \ y_{t-1}]$, $y_t \in \mathbb{R}^{2\times 1}$, and $\varepsilon_t \in \mathbb{R}^{2\times 1} \ \forall t$.

Extending the model to *VAR(p)* with n equations and p lagged values, we obtain the model with $(n \times p + 1) \times n$ parameters to estimate, $y_t \in \mathbb{R}^{n\times 1}$, and $\varepsilon_t \in \mathbb{R}^{n\times 1} \forall t$.

The variable p must be specified in the *VAR(p)* model as an input, or hyperparameter, which is one of the model identifiers. Different values of p yield different models that may produce similar results, but we are interested in a value of p that yields the optimal model. A useful model selection technique relies on *model-selection criteria*. One of the most commonly used criteria is known as *Akaike's Information Criterion, (AIC)*. The AIC "chooses the model with the best fit, as measured by the likelihood function, subject to a penalty term, to prevent overfitting, that increases with the number of parameters in the model"; see the first formula in Equation 2.8, [Chatfield and Xing, 2020].

$$\begin{aligned} AIC &= -2 \ ln(max. \ likelihood) + 2r \\ BIC &= -2 \ ln(max. \ likelihood) + (r + r \ln N) \\ Schwartz's \ IC &= -2 \ ln(max. \ likelihood) + (r \ln N) \end{aligned} \tag{2.8}$$

Another frequently used criterion is the *Bayesian Information Criterion, (BIC)*, which is similar to AIC but replaces the term $2r$ with $(r + r \ln N)$. This means that the addition of extra parameters is penalized more in BIC than in AIC; see the second formula in Equation 2.8. Similar to BIC, Schwartz's Bayesian criterion is

another possible option, which replaces $(r + r \ln N)$ with $(r \ln N)$, and is specified in the third line of Equation 2.8.

Some of these and other model-selection criteria can be quickly computed in R using the function `VARselect` from the `vars` package. Furthermore, the `vars` package also provides a function, `VAR`, which estimates the *VAR(p)* model based on the data and hyper-parameter p using Ordinary Least Squares (OLS) method. More about the `vars` package can be found in [Pfaff and Stigler, 2022].

2.5 Summary

We have introduced several important statistical methods: the change point technique, the basics of Bayesian statistics, MCMC simulations for estimating complex Bayesian models, and multivariate *VAR(p)* models. [Dufays et al., 2021] combined these methods into a master model and wrote the model-specific program to run the MCMC simulations. In the next chapter, we show that the implementation of these methods does not have to be complex. We introduce the statistical program JAGS, which enables us to write complex models in a simple and intuitive way. We also describe how the software helps us to evaluate MCMC convergence and interpret the parameter values.

Chapter 3

Methodology

The first goal of the thesis will be covered at the end of this chapter. Our goal is to implement the sparse *CP-VAR(p)* model in the JAGS program. To achieve the first goal, we divide this chapter into two parts. The first part covers the statistical specification of the model as in [Dufays et al., 2021]. The second part covers the coded specification of the model in the JAGS program.

3.1 Sparse Change Point Vector Autoregression

We begin this section by describing the model as it is defined by [Dufays et al., 2021]. We then describe the prior structure of the model and the ideas behind the calibration procedure of the priors. At the end of this part, we show the procedure for sampling the mean parameter.

3.1.1 Model Specification

The specification of the sparse *CP-VAR(p)* model is very similar to that of [Dufays et al., 2021], where the variable p denotes a number of lags in a *VAR(p)* model with $p \in [1, p]$.

Multivariate time series are denoted as:

$$y_{1:T} = \{y_1, ..., y_T\},\ y_t \in \mathbb{R}^{n \times 1}\ \forall t$$

The variable n specifies the number of equations in the *VAR(p)* model with $n \in [1, n]$.

A discrete date break vector is defined as:

$$\tau_{1:K} = \{\tau_1, ..., \tau_K\} \;\; s.t. \;\; \tau_0 \equiv 0 < \tau_1 < \tau_2 < ... < \tau_K < \tau_{K+1} = T$$

Here, K refers to the number of breaks, $i \in [0, K]$, and $K + 1$ defines the total number of regimes. Note that $t \in [\tau_i + 1, \tau_{i+1}]$, which describes a cycling through the regimes. In the first regime, the time flow starts at 1 and ends at the end of the first regime. For the last regime, the time flow starts at the first point in the last regime and ends at T, which is the last data point in the time series.

The sparse *CP-VAR(p)* model is specified in this form:

$$y_t = a_i^{(0)} + \sum_{l=1}^{p} a_i^{(l)} y_{t-l} + \varepsilon_t \tag{3.1}$$

$$= A_i x_t + \varepsilon_t \tag{3.2}$$

with $\varepsilon_t \sim N(0, \sum_i)$, $A_i = (a_i^{(0)}, a_i^{(1)}, ..., a_i^{(p)}) \in \mathbb{R}^{n \times k}$, $k = (n \times p) + 1$, $x_t = (1 \; y_{t-1}^{'} \; ... \; y_{t-p}^{'})$.

When working with a multivariate model, e.g. *VAR(p)*, even choosing low dimensions results in a model with a large number of parameters to be estimated. The number of parameters is further increased by the number of regimes imposed by the change point approach. [Dufays et al., 2021] propose the use of *shrinkage priors* to improve the model efficiency. Thus, the model is rewritten in the following form:

$$y_t = [A_0 + \sum_{i=1}^{K} \Delta A_i \mathbb{1}_{\{t > \tau_i\}}] x_t + \varepsilon_t \tag{3.3}$$

with $\varepsilon_t \sim N(0, \sum_t)$, $\Delta A_i = A_i - A_{i-1}$, where $\Delta A_i \in \mathbb{R}^{n \times k}$.

Although we are working with parameter differences from a modeling perspective, we aggregate the parameters into the standard *VAR(p)* form for interpretation purposes. The matrices of the autoregression parameters for each regime $K+1$ (with

K breaks) are specified as:

$$A_m^* = (A_0 + \sum_{i=1}^{m} \Delta A_i)$$

with $m \in [1, K]$, where A_0 specifies the first regime and A_m is each subsequent regime. In the results discussion, the collection of such matrices will be defined as A^*. [Dufays et al., 2021] give two reasons why working with changes in parameters is more advantageous than working with absolute parameter values. First, "having dependent priors over regimes complicates the estimation; see [Koop and Potter, 2007]". The second reason supports the use of shrinkage priors. Choosing zero as the mean parameter in first differences where there is no break allows easier deployment of the shrinkage hierarchy.

3.1.2 Number of Breaks

[Dufays et al., 2021] propose a simple and efficient approach regarding the parameter K. Because of the shrinkage property in the prior specification, the number of regimes can be arbitrarily chosen as long as it is large enough. The authors of the paper suggest setting $K = \lceil lnT \rceil$, "a small but conservative number", since we are interested in finding few but significant change points. This specification allows K to grow with the length of the time series, but imposes that $K < T$. As a result, we can ensure computational ease and MCMC convergence without actually computing the marginal likelihood.

Furthermore, the breaks are modeled by the latent variable s, which denotes the *states*. Such break dynamics have been provided in [Chib, 1998]. The state variable $s_t \in [1, K+1]$ is modeled for each variable t in the time series, $s_{1:T} = s_1, s_2, ..., s_T$. We provide the following hierarchical prior structure for s_t in our model.

$$s_{t+1} = s_t + Bern(p_t)$$
$$p_t \sim Beta(K,\ T - K)$$

Specifying the prior in this way allows the variable either to stay in the current regime s_t or to jump to the next one with a probability specified by the *Beta* distribution, as suggested by [Chib, 1998]. Moreover, the chosen parameters of the *Beta* distribution have an important interpretation. K can be referred to as the number of successes and $T-K$ as the number of failures in recognizing the current state, s_t.

3.1.3 Shrinkage Prior and its Calibration Procedure

$$Vec(A_0)|\Gamma_0 \sim \mathcal{N}(0,\ diag\{\phi_0^{(1)}, ..., \phi_0^{(nk)}\}) \tag{3.4}$$

$$Vec(A_i)|\Gamma_i \sim \mathcal{N}(0,\ diag\{\phi_i^{(1)}, ..., \phi_i^{(nk)}\}) \tag{3.5}$$

where the diagonal terms refer to covariance matrices, Γ_0 and $\Gamma_i,\ \forall i \in [1, K]$

$$\phi_i^{(j)}|z_i^{(j)}, (\sigma_i^{(j)})^2 = c_{z_i^{(j)}}^{(j)}\ (\sigma_i^{(j)})^2 \tag{3.6}$$

$$(\sigma_i^{(j)})^2 \sim \mathcal{IG}(\nu_1^{(j)}/2,\ \nu_2^{(j)}/2) \tag{3.7}$$

$$z_i^{(j)} \sim Bern(1-\omega_i^{(j)}) \tag{3.8}$$

$$\omega_i^{(j)} \sim Beta(\alpha_{\omega_i^{(j)}},\ \beta_{\omega_i^{(j)}}) \tag{3.9}$$

provided $\forall i \in [0, K]$ and $\forall j \in [1, nk]$. In the prior hierarchy specification, the following hyperparameters need to be specified: $c_{z_i^{(j)}}^{(j)}$, $\nu^{(j)}$, and $\alpha_{\omega_i^{(j)}}$, $\beta_{\omega_i^{(j)}}$.

Since $z_i^{(j)}$ comes from *Bernoulli*, it can only take the values 0 or 1. So we have $c_0^{(j)}$ and $c_1^{(j)}$. Choosing $c_0^{(j)} < c_1^{(j)}$ as such allows the creation of *spike* and *slab*. Since a small value of $c_0^{(j)}$ reduces the variance and the larger value of $c_1^{(j)}$ increases it, these specifications establish the spikes and slabs in the prior parameter values, respectively. Thus, we can say that $c_0^{(j)}$ acts as a spike and $c_1^{(j)}$ as a slab. Furthermore, we choose $c_0 = 1/10000$ and $c_1 = 1$ parameters as suggested by [Malsiner-Walli and Wagner, 2016] for all j values.

The variance parameters $(\sigma_i^{(j)})^2$ have an *Inverse-Gamma* distribution. We also choose $\nu^{(j)}$ according to the [Malsiner-Walli and Wagner, 2016] suggestion, setting $\nu_1^{(j)}/2 = 5$, which corresponds to "a $t-$distribution with 10 degrees of freedom", and

$\nu_2^{(j)}/2 = 4$.

However, the parameters $\alpha_{\omega_i^{(j)}}$ and $\beta_{\omega_i^{(j)}}$ need to be calibrated appropriately. [Dufays and Rombouts, 2018], who developed a similar model in a univariate setting, provide background on the calibration procedure. The penalty parameter used in the calibration procedure of the shrinkage prior "act as tuning parameter (i.e. 'λ') in standard shrinkage regressions". Although the tuning parameter is usually chosen through a cross-validation process, the penalty parameter in our model can be chosen using economic or statistical theory. It is implied that marginal distribution of $a_i^{(j)}|a_j^{2MU}, b_j^{2MU}$ mimics a mixture of two uniform distributions, *2MU*, that is defined by three parameters: a^{2MU}, b^{2MU}, such that $0 < a^{2MU} < b^{2MU}$ and P which is the penalty parameter mentioned above. The $2MU$ is defined as such:

$$a_i^{(j)}|a_j^{2MU}, b_j^{2MU} \sim \omega^{2MU}(P)U[\frac{-a_j^{2MU}}{2}, \frac{a_j^{2MU}}{2}] + (1 - \omega^{2MU}(P))U[\frac{-b_j^{2MU}}{2}, \frac{b_j^{2MU}}{2}] \tag{3.10}$$

with the weight $\omega^{2MU}(P)$:

$$\omega^{2MU}(P) = \frac{a_j^{2MU}(1 - e^P)}{b_j^{2MU}e^P + a_j^{2MU}(1 - e^P)} \tag{3.11}$$

being obtained from:
$logf(a_i^{(j)} = \frac{b_j^{2MU}}{2} \mid a_j^{2MU}, b_j^{2MU}, P) - logf(a_i^{(j)} = 0 \mid a_j^{2MU}, b_j^{2MU}, P) = P$

Due to its relation to the BIC, the penalty parameter P is chosen as:

$$P_\pi = -(ln\frac{\pi}{1 - \pi} + lnT) \tag{3.12}$$

with $\pi \in [0.5,\ 1]$. P is the amount of penalty the log-likelihood function experiences when the underlying parameter is far from zero or falls in the region $[\frac{-b_j^{2MU}}{2}, \frac{b_j^{2MU}}{2}]$ $[\frac{-a_j^{2MU}}{2}, \frac{a_j^{2MU}}{2}]$. When $P \to \infty$, the model becomes a standard VAR model without any breaks. Thus, by setting P appropriately, we can imply break occurrence in the prior, because it controls the weight, Equation 3.11, which further

controls the $2MU$ components in Equation 3.10.

In the appendix of [Dufays et al., 2021], the motivation for the choice of parameters (a_j^{2MU}, b_j^{2MU}, and P) in the *2MU* distribution is detailed. Here we show only the final computations:

$$\begin{aligned} a_j^{2MU} &= \frac{1}{3}\sigma_{a_0}^{(j)}, \\ b_j^{2MU} &= \frac{a_j^{2MU}(19TK+1)}{K}, \\ P &= P_{0.95}, \\ \omega^{2MU}(P_{0.95}) &= \frac{a_j^{2MU}(1-e_{0.95}^P)}{b_j^{2MU}e_{0.95}^P + a_j^{2MU}(1-e_{0.95}^P)} \end{aligned}$$

$\sigma_{a_0}^{(j)}$ is a standard error of the OLS estimate of $a_0^{(j)}$, namely the standard *VAR(p)* model without breaks. This setup avoids "a strong prior in favor of detecting breaks".

Finally, the calibration procedure of $\alpha_{\omega_i^{(j)}}$ and $\beta_{\omega_i^{(j)}}$, which are needed for the Equation 3.9, is finalized by:

$$\begin{aligned} \omega_i^{(j)} &\sim Beta(a_{\omega_i^{(j)}}, b_{\omega_i^{(j)}}), \\ \{a_{\omega_i^{(j)}}, b_{\omega_i^{(j)}}\} &= argmin(\ P[\omega_i^{(j)} \in [\omega^{2MU}(P_{0.75}), \omega^{2MU}(P_{0.99})]] - 0.999)^2 \end{aligned}$$

The hyperparameters are provided for any $i \in [0, K]$ and $j \in [1, nk]$.

In our model, we choose an average value of $\sigma_{a_0}^{(j)}$, which further simplifies the prior specification by dropping the j indices from the above equations.

3.1.4 Prior on the Variance-Covariance Matrix

In contrast to [Dufays et al., 2021], we keep the variance-covariance matrix of the time series constant over time. We choose the Wishart distribution on the precision matrix, which is the inverse of the variance-covariance matrix. In the case of the multivariate normal, the Wishart distribution on the precision matrix serves as a

conjugate prior, [Plummer, 2017].

$$\mathcal{T} \sim Wishart(R,\ k) \tag{3.13}$$

where R and k are parameters of shape and degrees of freedom, respectively. The prior on $\sum$ is implicitly defined as $\sum = \mathcal{T}^{-1}$. R is a $n \times n$ positive-definite matrix and the degrees of freedom are $k \geq n$. Furthermore, when R is diagonal and $k = n + 1$, the correlation parameters ρ_{ij} for $i \neq j$ have a uniform distribution on the interval $[-1,\ 1]$, [Plummer, 2017].

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}$$

Since the expectation of $\mathcal{T}$ is kR^{-1}, the prior on the variance-covariance matrix is R/k. The larger values of k, $k > n + 1$, support the belief that "the elements of the multivariate normal distribution are independent", [Plummer, 2017].

For the prior on the precision matrix, we define $R = k\sum$, where $\sum = diag(1)$ and $k = n + 1$. This setting expresses a prior belief that our variables have variances of 1 and correlation parameters that are uniformly distributed on the interval $[-1,\ 1]$.

3.1.5 Sampling the Mean Parameter

First, we write our sparse *CP-VAR(p)* model previously found in Equation 3.3:

$$\begin{aligned} y_t &= [A_0 + \sum_{i=1}^{K} \Delta A_i \mathbb{1}_{\{t>\tau_i\}}]x_t + \varepsilon_t \\ &= A\tilde{x_t} + \varepsilon_t \end{aligned} \tag{3.14}$$

with $\varepsilon_t \sim N(0, \sum_t)$ and $A = (A_0\ \Delta A_1\ ...\ \Delta A_K) \in \mathbb{R}^{n \times \bar{k}}$.

Before sampling the mean parameter, we introduce the unit vector:

$$e_K(i) = (\ \underbrace{1\ 1\ ...\ 1}_{\text{number of } i} 0\ ...\ 0\ )' \in \mathbb{R}^{K+1 \times 1}$$

Next, we multiply the unit vector by the previously defined x_t. Recall that x_t is the collection of lagged y_t variables.

$$\tilde{x_t} = e_K(s_t) \otimes x_t = (x_t' \mathbb{1}_{s_t \geq 1} \; x_t' \mathbb{1}_{s_t \geq 2} \; ... \; x_t' \mathbb{1}_{s_t \geq K+1} \;)' \in \mathbb{R}^{\bar{k} \times 1}$$

where $\bar{k} = (K+1)((n \times p) + 1)$.

Then, we are able to define $\tilde{X}$ and the rest of the variables:

$$\tilde{X} = (\tilde{x_1} \; \tilde{x_2} \; ... \; \tilde{x_T}) \in \mathbb{R}^{\bar{k} \times T}$$
$$Y = (y_1 \; y_2 \; ... \; y_T) \in \mathbb{R}^{n \times T}$$
$$E = (\varepsilon_1 \; \varepsilon_2 \; ... \; \varepsilon_T) \in \mathbb{R}^{n \times T}$$

Finally, we can write our model from Equation 3.14 in the matrix form:

$$Y = A\tilde{X} + E \tag{3.15}$$

Note that the original *CP-VAR(p)* parameters are obtained through the following expression:

$$A_m^* = (A_0 + \sum_{i=1}^{m} \Delta A_i)$$

with $m \in [1, K]$, where A_0 specifies the first regime and A_m is each subsequent regime. The collection of these matrices is indicated with A^*.

3.2 MCMC in the JAGS Program

In this part of the research, we decide to modify the original model of [Dufays et al., 2021] in two aspects: (1) to exclude the time-varying covariance matrix with its hierarchical priors, and (2) to facilitate the calibration procedure of the shrinkage priors by averaging the standard errors (SEs) from the standard *VAR* model. The average of the SEs serves as one of the input parameters in the optimization problem. The optimized results provide α and β hyperparameters for the shrinkage prior. We focus

on identifying the number of regimes with their associated parameters and exclude the predictive densities, which are further investigated in [Dufays et al., 2021].

Before providing the coded description of the sparse *CP-VAR(p)* model, we introduce the two programming environments, JAGS (Just Another Gibbs Sampling) and R. We use both to implement the model. The introduction to these environments should help to understand the model specification and its implementation, since both systems communicate with each other.

After providing the coded descriptions of the model in both languages, R and JAGS, we review several MCMC diagnostic measures. These measures help to access the MCMC representativeness, accuracy, and efficiency that affect the parameter estimates of our interest.

3.2.1 JAGS and R Environments

Just Another Gibbs Sampling, or JAGS, is an open-source program for statistical inference of Bayesian hierarchical methods, [Depaoli et al., 2016]. Despite the name, Gibbs sampling within JAGS is extended with other algorithms for sampling from the posterior distribution, [Coro, 2013].

JAGS is a partial clone of its predecessor program known as BUGS (Bayesian Inference Using Gibbs sampling), developed in the late 1980s, [Plummer, 2003], [Depaoli et al., 2016]. The BUGS program and language use directed acyclic graphs (DAGs) for Bayesian model specification. The acyclic graph is the graph that has no cycle or no path that leads to the same node twice.

A directed graph typically contains a set of nodes and a set of directed edges (arrows) connecting the nodes, which specify the relationships. Different types of nodes (stochastic, deterministic, constant, and observed nodes) can be used in the graph. We provide the DAG of our sparse *CP-VAR(p)* model in Figure 3.2. The DAG's structure represents or model specified in terms of the prior and the likelihood using the hierarchical structure. The figure is created according to the methodology described in [Höhna et al., 2014]. It is equivalent to the coded JAGS model in the Appendix A. In general, DAGs are easiest to read from bottom to top.

The DAG methodology has been preserved in JAGS for its convenience and flexibility in specifying the coded model. After converting the DAG into a JAGS/BUGS code, we can run the MCMC. The BUGS code of the sparse *CP-VAR(p)* model can be found in Appendix A. The two signs, '$\sim$' and '$\leftarrow$', are between objects in the model code. These relations are used to describe the DAG through the BUGS language. The '$\sim$' defines a stochastic relation that associates the probability distribution function with a particular node. The logical relations expressed with the '$\leftarrow$' symbol say "how to calculate the values of the parameters of the probability distribution functions in terms of the values of the node's parents", [Thomas, 2006]. Furthermore, the `for` loops are used to express the repetition of an action.

JAGS is typically used in conjunction with the R environment. The direct interface with R is shown in Figure 3.1 taken from [Kruschke, 2015]. We will use the `runjags` package to run MCMC, which is a successor to `rjags`, [Plummer, 2003], [Plummer, 2017], [Denwood, 2016].

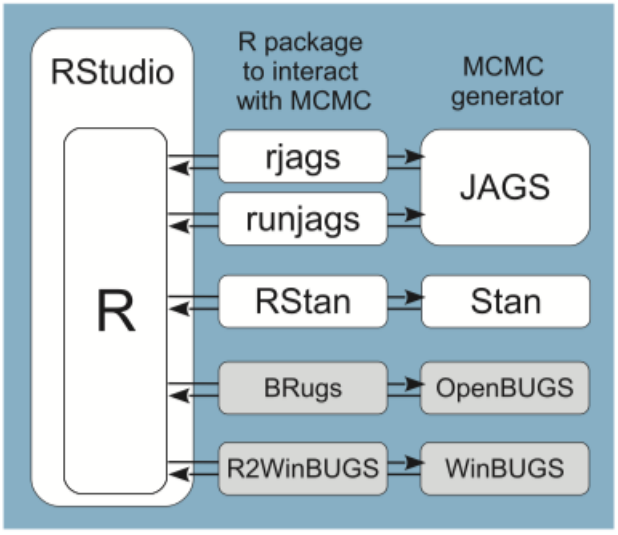

Figure 3.1: JAGS access through R using the `runjags` package.

3.2.2 Coded Model Specification

The sparse *CP-VAR(p)* model is implemented in R and JAGS code in several steps:

1. write the statistical model using JAGS/BUGS syntax in a `.txt` file,
2. import and prepare the data in R,
3. calibrate the hyperparameters in R,
4. pass the JAGS/BUGS coded model and the prepared data in JAGS using the `runjags` package,
5. analyze the results in R.

Although we provide the full R code in Appendix B, we present here the usage of the main JAGS function `run.jags()` from the `runjags` package. In the R code snippet Listing 3.1, we first collect all the input parameters into a `list` format. Then, we initialize three chains with different pseudorandom number generators. We use the `monitor` variable to specify which parameters we want JAGS to return in the summary statistics. In the next step, we write `run.jags()`, passing the JAGS/BUGS model stored in the `.txt` file, the modules JAGS needs to load, the variables to monitor, the data, the three initialized chains, and the sampling parameters (burn-in, thinning, and number of samples per chain). After the function has finished running the model, we retrieve the summary results with the `add.summary()` function. To visually check if the chains have converged well, it is useful to combine the three chains and create trace and density plots for each parameter of interest.

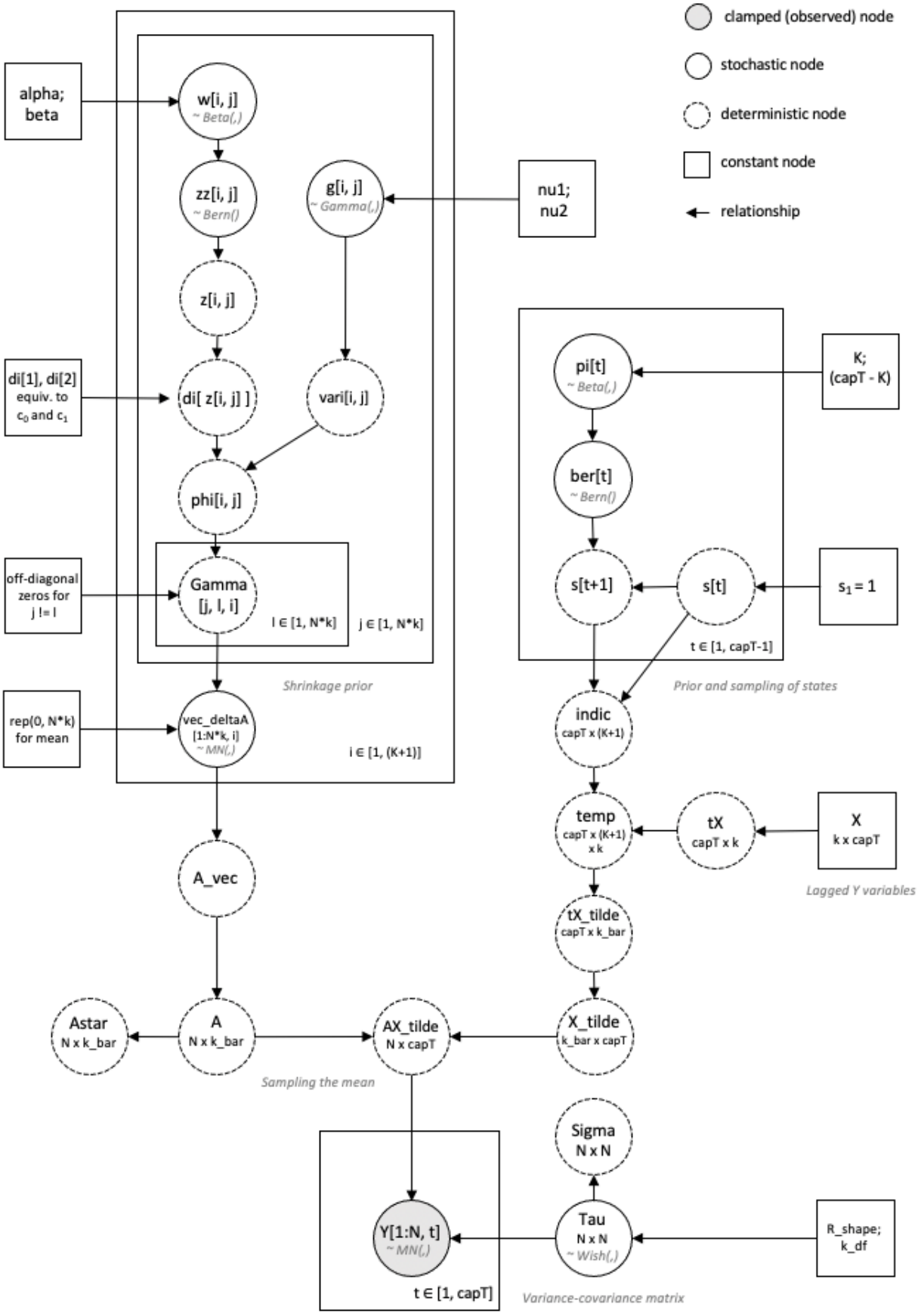

Figure 3.2: Directed acyclic graph of our sparse *CP-VAR(p)* model.

```
# 1. Prepare the data for JAGS
dat <- dump.format(list(Y=Y, X=X, capT=capT, N=N,
                        K=K, k=k, k_bar=k_bar,
                        alpha=a_cal, beta=b_cal,
                        di=di, nu1=nu[1], nu2=nu[2],
                        R_shape=R_shape, k_df=k_df ))

# 2. Initialize chains
inits1 <- dump.format(c(init_list, list(
   .RNG.name="base::Super-Duper", .RNG.seed=99999 )))
inits2 <- dump.format(c(init_list, list(
   .RNG.name="base::Wichmann-Hill", .RNG.seed=1234 )))
inits3 <- dump.format(c(init_list, list(
   .RNG.name="base::Mersenne-Twister", .RNG.seed=6666 )))

# 3. Tell JAGS which latent variables to monitor
monitor = c("Astar", "Sigma", "s")

# 4. Run the function that fits the models using JAGS
results <- run.jags(model="models/model-5-final.txt", modules="glm",
                    monitor=monitor, data=dat, n.chains=3,
                    inits=c(inits1, inits2, inits3),
                    plots = FALSE, method="parallel",
                    burnin=burnin, sample=sample, thin=thin)

# 5. Read the summary of the results
res0 <- add.summary(results)
res <- res0$summaries

# 6. Combine the MCMC chains
chains <- rbind(results$mcmc[[1]], results$mcmc[[2]], results$mcmc
    [[3]])
```

Listing 3.1: Excerpt of R code to initiate and start the JAGS program

3.2.3 MCMC Diagnostics

To achieve a good representation of the posterior distribution, [Kruschke, 2015] identifies three MCMC goals - representativeness, accuracy, and efficiency.

The chain is representative if it explores the full range of the posterior, mixes well, and overlaps with the other parallel chains in the MCMC. There are visual and numerical methods to check MCMC representativeness. The *trace plot* of each parameter chain helps to visually inspect the convergence of each parameter chain. The chains are converged if there are no chains that are stuck around similar values for an extended period of time. *Density plots* of parallel chains can also serve as a visual convergence check. Chains are converged if the density plots overlap well. From a numerical point of view, the Gelman & Rubin statistic can be used to check convergence, [Gelman and Donald, 1992]. The measure we refer to is called the *potential scale reduction factor* or *psrf* value. The *psrf* value tells "how much variance there is between chains relative to how much variance there is within chains," [Kruschke, 2015]. If the *psrf* is close to 1, it means that the chain has nearly converged to the posterior distribution. In general, it is acceptable to have *psrf* values below 1.1 for all parameters. On the other hand, larger *psrf* values indicate MCMC instability and improper convergence to the posterior. In the JAGS summary statistics, *psrf* is simply noted as `psrf`.

To obtain accurate estimates, the MCMC chain should be sufficiently long. For this reason, we need a measure of the accuracy (and length) of a chain that accounts for *autocorrelation*, or clumpiness, within the chain. We are interested in the measure that tells "how much independent information there is in the autocorrelated chain", [Kruschke, 2015]. The *effective sample size* was proposed by [Kass et al., 1998] because of this relationship, see Equation 3.16.

$$ESS = \frac{N}{1 + 2\sum_{k=1}^{\inf} ACF(k)} \tag{3.16}$$

[Kruschke, 2015] suggests that ESS should be as large as 10,000 to obtain accurate estimates from the posterior using the 95% HDI limits. Since this value is based on

practical experience, it is more of a suggestion than a requirement. We will try to achieve an ESS of at least 1,000 or more for each parameter of interest in our MCMCs. In the JAGS summary statistics, both ESS and auto-correlation values can be found under the keywords `SSeff` and `AC.x`, respectively.

The efficiency of the MCMC depends on the computing power and time available to run the model. Since we use a standard MacBook Pro to run our MCMCs, we are limited to the computing power of the machine.

3.2.4 Interpretation of the MCMC Estimates

After completing our MCMC diagnostics, we want to evaluate and interpret our parameter estimates.

The advantage of Bayesian computational methods is that we obtain the full distribution of parameter estimates. However, sometimes it is more convenient to have a single value as a result. The Bayesian point estimates are obtained by taking one of the following summary statistics from the parameter chains: mean, median, or mode. Following the approach of [Dufays et al., 2021], we will take the median of the chains for the interpretation of the results.

We will look not only at the median estimates of the parameter distribution, but also at the *high-density interval*, or HDI. The HDI summarizes the points of the distribution that have the most credibility. It sets percentage limits on either side of a distribution to separate the points in the distribution that are more credible from those that are less credible. For example, the 95% HDI of a normal distribution sets limits from -1.96 to +1.96 in the tails of the distribution. The areas under the density curve and between these limits cover 95% of the total area under the curve.

Another useful feature of the HDIs is the uncertainty measures of the parameter estimates. Narrow HDI limits indicate higher accuracy, while wide limits indicate how uncertain the estimates are. Typically, larger sample sizes provide us with more certainty about the estimated parameters, and so do the narrower HDIs.

In models like ours, it is interesting to find the parameter estimates with larger magnitudes that indicate the strength of the relationship between the variables.

Moreover, we want to know whether the parameter estimate is close to or far from zero. To explain the significance of the parameter, we will check whether or not our 95% HDI includes zero as a credible parameter value. [Kruschke, 2015] provides more details on parameter precision and interpretation of results.

Chapter 4

Simulation Study

This chapter aims to implement the second goal of this thesis set in Section 1.2. This goal focuses on testing the performance and robustness of the model through the simulation study. We complete it by establishing a set of research questions and answering them with the results from the simulated experiments.

4.1 Research Questions

The simulated experiments are designed to answer two research questions.

- **RQ 4.1:** How does the model perform in a setting similar to the one in Experimentation, Chapter 5?
- **RQ 4.2:** How well can our model detect multiple change points in the model's parameters?

4.2 Data Generation Process

We use the data generation processes DGP A and C for the simulation experiments, which can be found in [Dufays et al., 2021] and [Preuss et al., 2015]. The DGP C contains four regimes in mean parameter values, while the DGP A is built from a

single regime; see Figure 4.1. We will use DGPs (A and C) in the first simulation experiment and only DGP C in the second. DGP B will not be used in our evaluations because our model does not include a time-varying covariance matrix. In addition, we initialized the parameter chains with OLS parameter estimates to achieve faster convergence. Also, the three parallel chains are set with different pseudorandom number generators, which allows for evaluating the chain convergence from a numerical perspective using Gelman & Rubin statistics. The code used to simulate the DGPs can be found in Appendix B.

DGP A

Regime	1	2	3	4	#Regimes
time	$1 \leq t < \lfloor T/4 \rfloor$	$\lfloor T/4 \rfloor \leq t < \lfloor 2T/4 \rfloor$	$\lfloor 2T/4 \rfloor \leq t < \lfloor 3T/4 \rfloor$	$\lfloor 3T/4 \rfloor \leq t$	
A_t	$\begin{pmatrix} 0.4 & 0.6 & 0 \\ 0 & 0.3 & 0.8 \end{pmatrix}$	$\begin{pmatrix} 0.4 & 0.6 & 0 \\ 0 & 0.3 & 0.8 \end{pmatrix}$	$\begin{pmatrix} 0.4 & 0.6 & 0 \\ 0 & 0.3 & 0.8 \end{pmatrix}$	$\begin{pmatrix} 0.4 & 0.6 & 0 \\ 0 & 0.3 & 0.8 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$
Σ_t	$\begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}$	1

DGP B

Regime	1	2	3	4	#Regimes
time	$1 \leq t < \lfloor T/4 \rfloor$	$\lfloor T/4 \rfloor \leq t < \lfloor 2T/4 \rfloor$	$\lfloor 2T/4 \rfloor \leq t < \lfloor 3T/4 \rfloor$	$\lfloor 3T/4 \rfloor \leq t$	
A_t	$\begin{pmatrix} 0.4 & 0.6 & 0 \\ 0 & 0 & 0.9 \end{pmatrix}$	$\begin{pmatrix} 0.4 & 0.6 & 0 \\ 0 & 0 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.4 & 0.6 & -0.25 \\ 0 & 0 & 0.9 \end{pmatrix}$	$\begin{pmatrix} 0.4 & 0.6 & -0.25 \\ 0 & 0 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 4 \end{pmatrix}$
Σ_t	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 2.5 & 0.3 \\ 0.3 & 2.5 \end{pmatrix}$	$\begin{pmatrix} 2.5 & 0.3 \\ 0.3 & 2.5 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0.3 \\ 0.3 & 0.5 \end{pmatrix}$	3

DGP C

Regime	1	2	3	4	#Regimes
time	$1 \leq t < \lfloor 0.5T \rfloor$	$\lfloor 0.5T \rfloor \leq t < \lfloor 2T/3 \rfloor$	$\lfloor 2T/3 \rfloor \leq t < \lfloor 3T/4 \rfloor$	$\lfloor 3T/4 \rfloor \leq t$	
A_t	$\begin{pmatrix} 0 & 0.5 & 0.1 \\ 0 & 0.1 & 0.5 \end{pmatrix}$	$\begin{pmatrix} 0 & -0.5 & 0.1 \\ 0 & 0.1 & -0.5 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5 & 0.1 \\ 0 & 0.1 & 0.5 \end{pmatrix}$	$\begin{pmatrix} 0 & -0.5 & 0.1 \\ 0 & 0.1 & -0.5 \end{pmatrix}$	$\begin{pmatrix} 0 & 4 & 1 \\ 0 & 1 & 4 \end{pmatrix}$
Σ_t	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	1

Figure 4.1: Data generation processes found in [Dufays et al., 2021].

4.3 Simulation Experiment 1

We start the simulation experiment based on our first objective, to test the model's performance in a setting similar to the one in Chapter 5. We will perform this simulation experiment in two steps. First, we simulate the DGP C from the bivariate *CP-VAR(1)* to test the model's ability to detect the multiple change points in the

mean parameters. In the second step, we simulate from the DGP A to assess the model's performance when there are no true change points. The experimental setting is the same in both steps. The time series T=120 is composed of four regimes that last 60, 20, 10, and 30 time points. The MCMC setting consists of `burnin`=20,000, thin=2, and `sample`=20,000. JAGS will return 20,000 samples for each parallel chain after thinning and burn-in period. Since we have three parallel chains, we obtain 60,000 samples from JAGS for each parameter estimation. We repeat the simulation experiments twenty times in both cases.

When we run the simulation experiment on DGP C, we find the first out of three change points 18/20 times in the time series. The third change point is detected 4/20 times. The second change point is not found. In addition, we observe some detection variability around the true change point. The results are displayed in Table 4.1 and summarized in Figure 4.2. The figure shows the minimum, first quartile, median, third quartile, and maximum.

Table 4.1: Simulation experiment 1. Change points true vs. detected after running the MCMCs on 20 DGP C with T=120.

True/Found	**52**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**73**	**103**	**108**	**110**	**NA**	**Total**
61	1	1	1	1	1	3	3	3	1	1	1	1				2	20
81																20	20
91													2	1	1	16	20

At last, we run a simulation experiment with DGP A where no true change point exists. The simulation experiment has shown that model detected no change points in most cases.

4.4 Simulation Experiment 2

Since we could not identify the second change point and the third change point was detected only 4/20 times in the first simulation experiment, we suspect that the regimes' length and the length of the time series influence the model's performance.

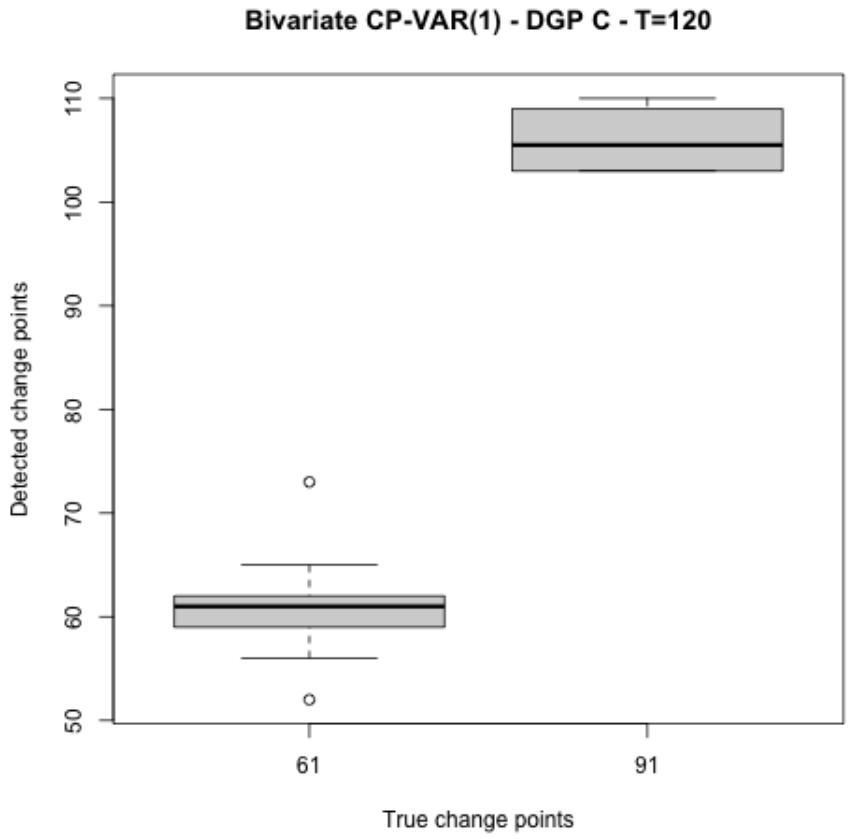

Figure 4.2: Simulation experiment 1. Box plot of the summary results of 20 DGP C with T=120.

In this simulation experiment, we extend the time series to T=500 and simulate the DGP C twenty times. Again, we use the bivariate *CP-VAR(1)* process. The time series T=500 consists of four regimes lasting 250, 83, 42, and 125 time points. We run the MCMCs with JAGS, setting `burnin`=10,000 and `thin`=2. JAGS returns `sample`=10,000 samples for each of the three parallel chains. We choose 10,000 samples instead of 20,000 as in the first simulation experiment simply because of the low computing power available on our machine.

At the end of this simulation experiment, we summarize the results in Table 4.2 and Figure 4.3. Within this experiment setting, we can identify the locations of two

out of three change points. Although the first change point at t=251 was found with some uncertainty in almost every experiment, detecting the remaining change points is unsuccessful. The second change point at t=334 is not detected. Finally, the last change point at t=376 was found 7 out of 20 times.

The box plot in Figure 4.3 shows the summary results in a visual form. We observe a much higher dispersion between the detected values around the third change point than the first one. Since the second change point was not found, it is excluded from the figure.

Table 4.2: Simulation experiment 2. Change points true vs. detected after running the MCMCs on 20 DGP C with T=500.

True/Found	241	242	246	247	250	251	253	254	255	257	258	260	264	375	376	383	388	415	443	NA	Total
251	1	1	1	1	1	1	2	4	1	1	1	1	1							3	20
334																				20	20
376														1	1	1	2	1	1	13	20

4.5 Summary of the Results

Recall the research questions set at the beginning of this chapter:

- **RQ 4.1:** How does the model perform in a setting similar to the one in Experimentation, Chapter 5?
- **RQ 4.2:** How well can our model detect multiple change points in the model's parameters?

RQ 4.1: Similar setting as in Experimentation, Chapter 5

Our first simulation experiment shows that in DGP C with T=120, we can detect two of three change points, namely the first and third. The second change point is not detected. Although the first change point is found in almost every simulation, the third change point is found only 4/20 times. When we switch to DGP A, where

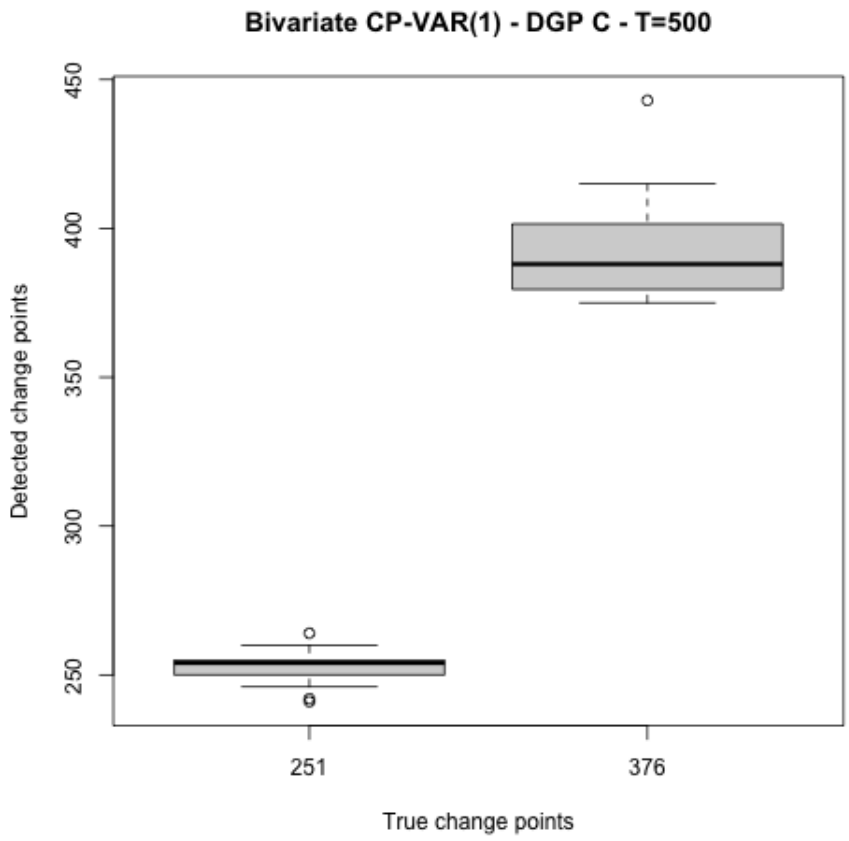

Figure 4.3: Simulation experiment 2. Box plot of the summary results on 20 DGP C with T=500.

there are no change points, we can identify their absence in most cases. We will use this experiment setting later when we work with the real data in Chapter 5.

RQ 4.2: Multiple change points detection

After extending the time series to T=500, we could identify two of the three change points again. However, the first change point is identified in most simulations, the second one is undetected, and the third change point is found 7/20 times. Increasing the time series length has improved the change point detection ability but only

slightly. We believe that the short lengths of the regimes around the second change point influence the detection. Furthermore, we believe that extending T will further improve detection accuracy. Unfortunately, an extensive simulation study requires greater MCMC efficiency, which is hardly achievable with the computational power of our machine.

Overall, we used the Gelman & Rubin statistics to evaluate the convergence diagnostics of the chains in both simulation experiments. The autoregressive parameters' chains in most experiments converged appropriately; the *psrf* values remained below 1.1. However, there were some cases where two change points were detected, but the convergence was not achieved. These experiments have been re-run on a larger sample size to achieve the appropriate parameter convergence. On the other hand, the chains of the state's parameters showed poorer convergence in terms of *psrf* value (larger than 1.1). Improving the convergence of such chains (and the detection accuracy of the change points) would require a much larger sample size and higher machine computing power.

Chapter 5

Experimentation

The third goal aims to investigate the interrelationships between financial and economic variables through the defined experiments. We achieve this goal by defining research questions, which will be answered with the results of each experiment. First, we present research questions discussed in this chapter. Then, we describe the data selection process and the rationale behind each experiment. Finally, we end the chapter by presenting the results and answering the research questions.

5.1 Research Questions

We conduct two experiments using our model on data from the financial and economic fields. In both experiments, we are interested in finding answers to the following research questions:

- **RQ 5.1:** Does the model detect any change points in the real data? If yes, do these points have any meaningful interpretation?
- **RQ 5.2:** Are there any significant parameter values within the regimes? If yes, what is the interpretation of the parameters?

5.2 Data Selection

Before we begin the experimentation, we identify the financial and economic time series used in both experiments. Since we use indices that are composed of numerous financial and physical assets, we provide the technical characteristics of each index. We also identify the economic macro-variables chosen for both experiments. A description of the theory behind the data sets is also provided so that we can interpret the parameters in our model.

The list of the indices and the macro variable are given in Table 5.1.

Table 5.1: Variables used in the experiments.

Experiment	**Short Name**	**Long Name**
Experiment 1	SPXPVTR	S&P 500 Pure Value TR
	SPXPGTR	S&P 500 Pure Growth TR
	Unemp Rate	Unemployment Rate, de-seasonalized, standardized
	CPI	Consumer Price Index, de-seasonalized
Experiment 2	SPGSCLTR	S&P GSCI Crude Oil Index TR
	SPGSNGTR	S&P GSCI Natural Gas Index TR
	IXETR	S&P 500 Energy Select Sector TR
	CPI	Consumer Price Index, de-seasonalized

We download the daily price levels of the indices over the past 10 years from the *S&P Global* website and resample them at monthly intervals. It is common to compute the logarithms of the first differences of the prices, $logP_t - logP_{t-1} = log[\frac{P_t}{P_{t-1}}]$, since stock prices can be modeled as a random walk. The new time series is commonly called *log returns*, [Chatfield and Xing, 2020]. It is also common to assume that such a time series has (approximately) normal, independent, and identical distribution. We now have 10 years of monthly log returns.

The macro variables are retrieved from the *U.S. Bureau of Labor Statistics* website for the same period. We download the monthly changes in the Consumer Price Index

(CPI) and the Unemp Rate. Both variables are already deseasonalized, which allows us to observe only the seasonally unaffected trend. Since the unemployment rate shows a strong positive trend, we have standardized the variable by subtracting the mean and dividing it by the standard deviation. This step is performed to obtain a better mixing of the chains, since *VAR(p)* models work best with the stationary time series.

Additional details and technical characteristics of each index are provided in the following subsections.

5.2.1 Value vs. Growth Investment Styles

In modern investment history, value and growth styles are arguably the most entrenched investment strategies.

Value investing ideas were first developed by Benjamin Graham, "a thinker, writer, and teacher in the field of investment analysis" in the early 20th century. The core idea of value investing is to purchase common stocks that are priced below their intrinsic value. Graham's ideas were documented in [Graham and Dodd, 1934]. His influence is still present among investment professionals today. One of the most famous value investors is Warren Buffet, a former student of B. Graham.

To identify value stocks, the first step is to determine the fundamental or intrinsic value of the company. This can be done by performing a fundamental analysis of the companies, such as valuation using multiples (market-to-book or price-to-book (P/B), price-earnings (P/E), etc.), or by using a variety of discounting models. Because of the extensive fundamental analysis performed prior to investing, this strategy tends to withstand market fluctuations and perform well during economic downturns. In summary, value firms tend to show "low price-to-book ratios, low price-to-earnings ratios, above-average levels of dividend yield, and market prices believed to be below the issues' intrinsic values", [Bodie et al., 2018]

In contrast to the value style, the growth style focuses on companies with attractive profit opportunities. These opportunities usually arise when a company decides to reinvest its earnings in projects that will become valuable in the future. Typical

characteristics of such firms are "low current yield, high price-to-book ratios, and high price-to-earnings ratios".

Figure 5.1 shows the price levels of total return indices for pure value and pure growth strategies over the past ten years. We also provide the market index level for comparison.

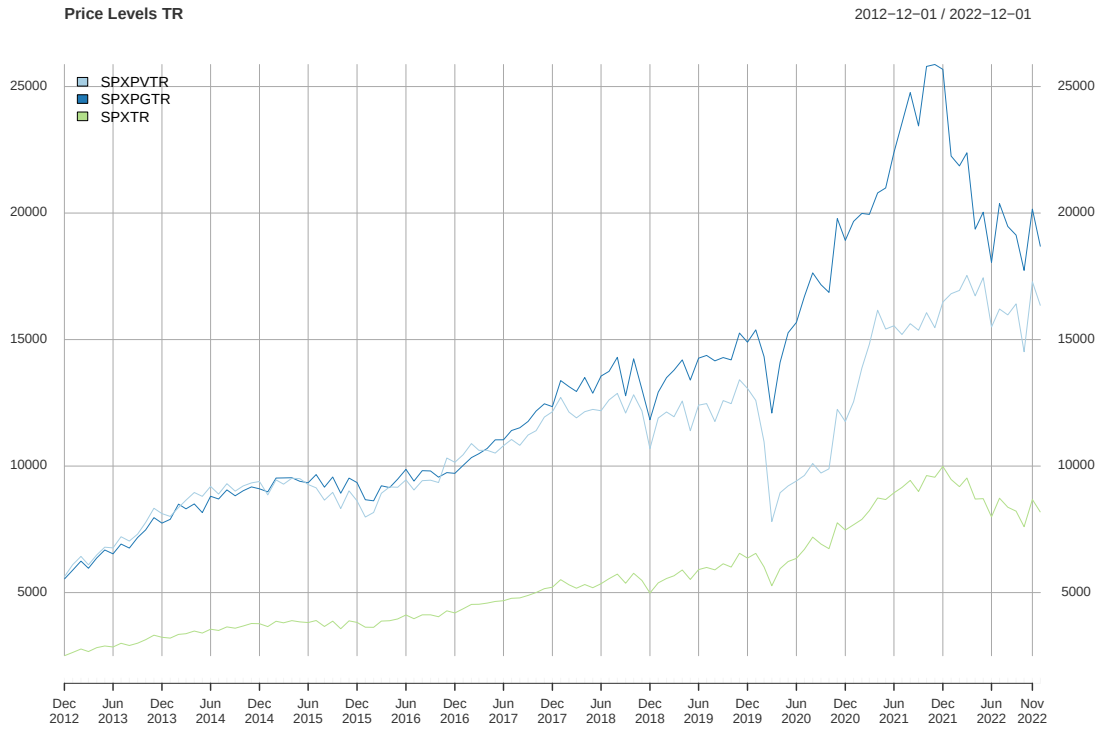

Figure 5.1: Monthly price levels of S&P 500 Index (SPXTR), S&P 500 Pure Value Index (SPXPVTR), and S&P 500 Pure Growth Index (SPXPGTR).

However, there are several approaches to differentiating between value and growth opportunities, which we will refer to as investment styles. S&P500 Global has created indices that help establish a benchmark for these stylized strategies. For example, the S&P500 Pure Value and S&P500 Pure Growth indices divide the universe of S&P500 stocks approximately equally, with no overlapping stocks. This means that the two styles complement each other. The S&P 500 Pure Value index is de-

rived using multiple measures such as "book value, earnings and sales to price", see [S&P Global, 2022b]. Whereas the S&P 500 Growth index is derived from "the sales growth, the ratio of earnings change to price, and momentum", [S&P Global, 2022a]. In our experiments, we will rely on the S&P Global metrics and use their indices. We use the total return version of the indices because "Gross Total Return (TR) versions reinvest regular cash dividends at the close on the ex-date without consideration for withholding taxes, [S&P Global, 2022d].

5.2.2 Energy Markets

There are several ways to invest in the energy markets. For simplicity, we highlight two investment approaches: stocks involved in energy production and distribution, and physical commodities. Stocks can be traded individually or through funds, for example, exchange-traded funds (ETFs). Commodities, or raw materials, are traded through futures or forward contracts.

Futures contracts are standardized derivative contracts traded on exchanges, while forwards are over-the-counter contracts. The following discussion will focus on futures and stocks. Note that there is no exchange of money at the time the futures contract is signed. The money exchange is required only on the day of contract maturity, [Bodie et al., 2018]. Thus, the amount of money committed on the futures contract can earn the collateral yield until the day of contract maturity.

The energy equity sector is generally defined by the Global Industry Classification Standard (GICS). It is the eighth largest sector in the S&P500 index by market capitalization, accounting for approximately 5%, [S&P Global, 2023]. The two largest constituents of the energy index are Exxon Mobil Corp and Chevron Corp, companies primarily involved in the oil and natural gas business. Based on [Bellucci, 2018] report, energy sector companies mainly engaged in:

- oil and gas exploration, production, refining, marketing, storage, and transportation;
- drilling;

- equipment manufacturing;
- Manufacture and extraction of products related to energy production.

While energy exploration companies tend to move between the stock market and commodities, commodities tend to behave differently than stocks, see Figure 5.2. The figure shows that the two commodity indices (crude oil and natural gas) in blue follow a rather different trajectory than the stock market S&P 500 in dark green. The energy stock index (light green) has the characteristics of both stocks and commodities. It moves between the commodities and the stock market index.

We also observe that commodities (blue trajectories) tend to follow the Consumer Price Index (pink trajectory), which signals the link between commodities and inflation (CPI is a measure of inflation). This link between inflation and commodities persists because "raw materials and their associated costs are key inputs in the basket of goods used in compiling many consumer price indices", [Credit Suisse, 2019].

In the second experiment, we take the S&P 500 Energy Select Sector, the market-capital-weighted industry index built out of the S&P500 universe of stocks. In other words, the S&P 500 Energy Select Sector Index is a sub-index of the S&P500 index. S&P GSCI Crude Oil Index TR and S&P GSCI Natural Gas Index TR are the other two indices taken for the second experiment. They represent the benchmark returns from crude oil and natural gas investments. These indices are constructed using a different methodology than the one from equities. Since commodities are traded in the market through derivative products, these two indices are composed of an ample number of futures contracts, [S&P Global, 2022c]. "The futures contract calls for delivery of a commodity at a specified delivery or maturity date, for an agreed upon price [...] to be paid at contract maturity", [Bodie et al., 2018]. The S&P methodological guidelines state that a futures contract "must be a physical commodity and may not be a financial instrument (e.g., securities, currencies, interest rates, etc.)", [S&P Global, 2022c]. Each futures contract included in the index must meet strict criteria. Technical characteristics, denomination currency, transparency of prices, volume and liquidity, and trading venue are just some criteria examples. The compositions of the indices are reviewed every quarter to ensure that contracts

are still eligible to be included in the indices. Furthermore, the contracts are rolled every month within the predefined period. We use the total returns version of indices that include collateral yield, or risk-free return, on the committed investment, [S&P Global, 2022c].

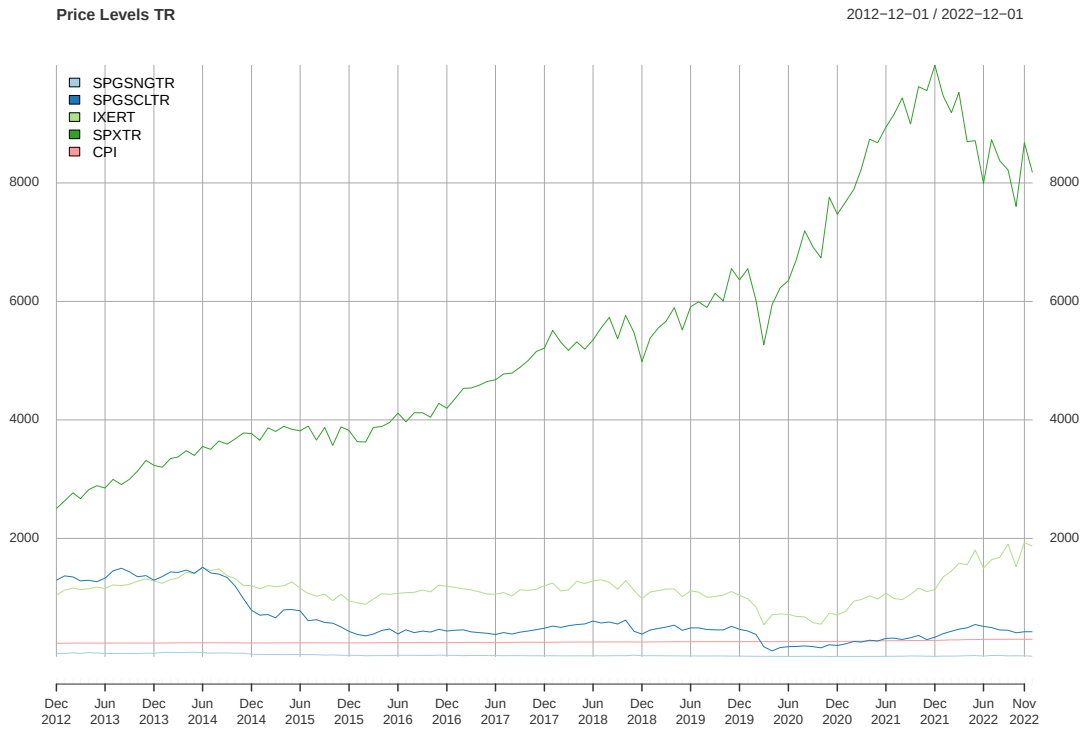

Figure 5.2: Monthly price levels of S&P GSCI Natural Gas Index TR (SPGSNGTR), S&P GSCI Crude Oil Index TR (SPGSCLTR), S&P 500 Energy Select Sector Index (IXERT), S&P 500 Index TR (SPXTR) and, Consumer Price Index (CPI).

5.3 Experimental Setup

Recall $VAR(1)$ model from the system of Equations 2.6 and extend it to $n = 4$:

$$\begin{aligned} y_{1,t} &= a_{10}1 + a_{11}y_{1,t-1} + a_{12}y_{2,t-1} + a_{13}y_{3,t-1} + a_{14}y_{4,t-1} + \varepsilon_{1,t} \\ y_{2,t} &= a_{20}1 + a_{21}y_{1,t-1} + a_{22}y_{2,t-1} + a_{23}y_{3,t-1} + a_{24}y_{4,t-1} + \varepsilon_{2,t} \\ y_{3,t} &= a_{30}1 + a_{31}y_{1,t-1} + a_{32}y_{2,t-1} + a_{33}y_{3,t-1} + a_{34}y_{4,t-1} + \varepsilon_{3,t} \\ y_{4,t} &= a_{40}1 + a_{41}y_{1,t-1} + a_{42}y_{2,t-1} + a_{43}y_{3,t-1} + a_{44}y_{4,t-1} + \varepsilon_{4,t} \end{aligned} \tag{5.1}$$

The compact matrix notation gives:

$$y_t = Ax_t + \varepsilon_t$$

where A is 4×5 matrix, $x_t^{'} = [1\ y_{t-1}]$, $y_t \in \mathbb{R}^{4\times1}$, and $\varepsilon_t \in \mathbb{R}^{4\times1}\ \forall t$.

In our sparse *CP-VAR(1)* models, we estimate $K + 1$ of these A matrices, which give us the parameter values of interest. Note that in our sparse *CP-VAR(p)* model specification, we use A^* to denote the autoregressive parameter matrix and A to denote the first differences in autoregressive parameters between the regimes. Since the Bayesian model gives us the distribution of each parameter value, we take the median of the chains when discussing the results.

In JAGS, we initialize the parameters with *VAR(p)* OLS estimates for all regimes. We then initialize three parallel chains with different random number generators. We also set `burnin=20,000`, `samples=20,000`, and `thin=2`. This setting takes every second value in each chain, discards the first 20,000 samples during the burn-in phase, and returns 20,000 samples. In total, we have 60,000 samples per experiment. This setup ensures that all *psrf* values in both experiments are in the acceptable range (<1.1).

5.4 Experiments

In the following sections, we provide detailed descriptions of the experiments and the interpretation of the estimated parameter matrix A^* obtained from both experiments.

5.4.1 Experiment 1. Value vs. Growth

In the first experiment, we consider the S&P500 universe further divided into value and growth portfolios. These are the S&P 500 Pure Value and S&P 500 Pure Growth portfolios. Since there are no overlapping components in the two index portfolios, they can be considered complementary. We further investigate whether the monthly log returns of these two strategies and the changes in the macro variables have lagged relationships with each other. These are the monthly changes in the CPI and the unemployment rate in the United States. After running the experiment, our model does not identify change points, so we only discuss the results obtained within the single regime.

$n = \{\text{SPXPVTR, SPXPGTR, Unemp-rate, CPI}\}$ is equivalent to $n = \{1, 2, 3, 4\}$ in the Equation 5.1.

Figure 5.3 shows the variables chosen for the first experiment over time. The data includes every month from January 2013 to December 2022, corresponding to 120 data points. Note that the stock indices show log returns, the unemployment rate, and the CPI are the time series of the monthly changes of the deseasonalized rates. In addition, the unemployment rate has been standardized to remove the positive trend.

The matrix $A^*_{1,1CP}$ in 5.2 provides the median values of each parameter in the first regime for the first experiment.

$$A^*_{1,1CP} = \begin{bmatrix} 0.0031 & 0.0070 & -0.1140 & 0.0029 & -0.0005 \\ 0.0032 & -0.0236 & -0.0102 & 0.0030 & 0.0012 \\ 0.0049 & \mathbf{-2.8076} & -0.2779 & \mathbf{0.8722} & -0.2752 \\ 0.0846 & 1.2554 & -0.1934 & 0.0022 & \mathbf{0.5560} \end{bmatrix} \tag{5.2}$$

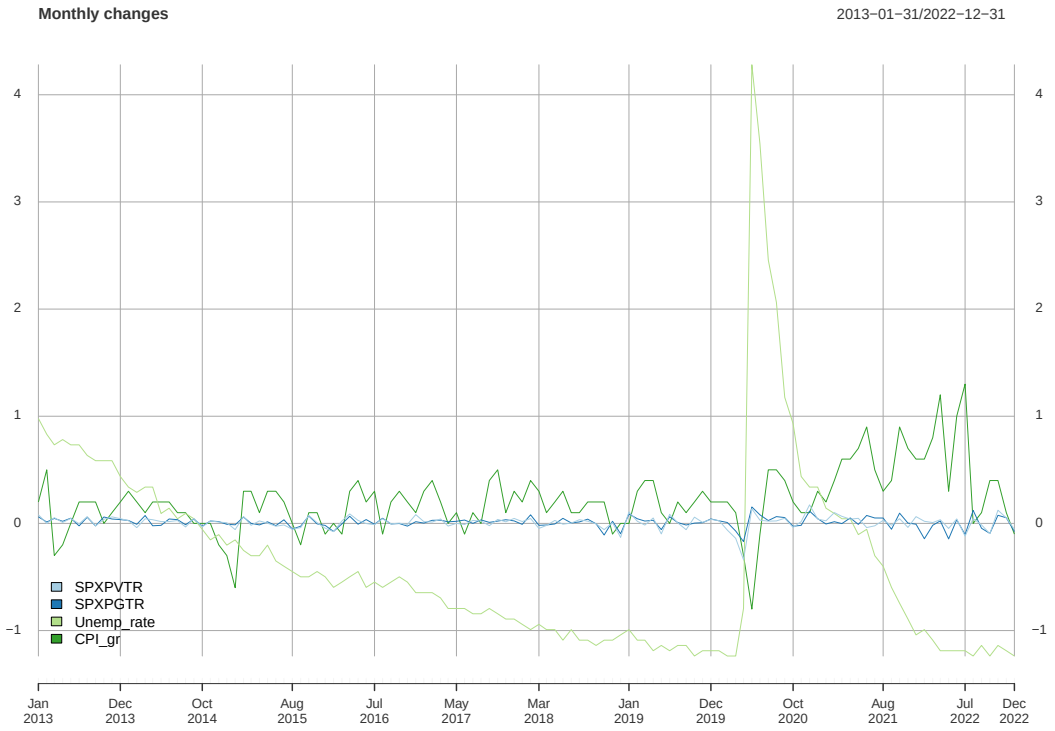

Figure 5.3: Experiment 1. Monthly changes in variables.

$$A^*_{1OLS} = \begin{bmatrix} \mathbf{0.0189} & 0.1027 & \mathbf{-0.3673} & \mathbf{0.0174} & -0.0065 \\ \mathbf{0.0210} & -0.0932 & -0.2051 & \mathbf{0.0191} & -0.0063 \\ 0.0355 & \mathbf{-4.6768} & 0.8197 & \mathbf{0.8857} & \mathbf{-0.2773} \\ \mathbf{0.0965} & \mathbf{2.4127} & \mathbf{-1.4484} & 0.0132 & \mathbf{0.5307} \end{bmatrix} \tag{5.3}$$

From the results in Table 5.2, the density plots in Figure 5.5, and the $A^*_{1,1CP}$ matrix in Equation 5.2, we observe several significant parameters. The significant parameter values are indicated in bold. These parameters are found to be away from zero, which means that zero falls outside the HDI 95% confidence interval.

Our attention is first drawn to the third equation in the model, which represents

the changes in the unemployment rate. The second and fourth columns of the matrix $A^*_{1,1CP}$, namely SPXPVTR value index and Unemp-rate, affect the changes in the unemployment rate. Our model shows that the value stock index's lagged return negatively impacts the change in the unemployment rate (-2.8076). In other words, the rising value stock index positively affects the growth of the *employment* rate. We also see a positive persistence in the unemployment rate itself (0.8722). The positive value means that the growth of the unemployment rate in the previous month has a substantial impact on the growth of the unemployment rate in the current month. We also observe that the lagged change in the CPI has a slightly negative impact on the unemployment rate (-0.2752), but since the HDI includes zero at 95%, we cannot provide credible evidence that the parameter is significant.

Next, we observe a couple of interesting parameters in the CPI equation on the 4th row of the $A^*_{1,1CP}$ matrix in the Equation 5.2. The lagged log return of the SPXPVTR value index has a very positive impact on the change in the CPI (1.2554). Since the 95% HDI includes zero, we need more evidence to declare significance. In addition, the lagged CPI itself has a positive and significant impact on the change in the CPI (0.5560).

We compare the Bayesian estimates to OLS estimates. Specifically, we compare a single regime of our sparse *CP-VAR(1)* with standard OLS *VAR(1)*. The OLS method declares significance when the p-value falls below the 5% threshold, leading to rejecting the null hypothesis. In the Bayesian method, we declare parameters significant if the 95% HDI limits exclude zero. Both models produce comparable parameter values. The parameter values found to be significant in our sparse model are also found to be significant in the standard *VAR(1)*. However, we find many more significant parameter values in the standard OLS *VAR(1)* model than in our sparse *CP-VAR(1)* model. We might be able to find a few more significant parameters if the chains of our model were run longer to obtain a larger sample. The larger sample increases the accuracy of parameter estimates and narrows the HDI limits. The complete results from the sparse *CP-VAR(1)* are in Appendix C, and OLS results are in Appendix D.

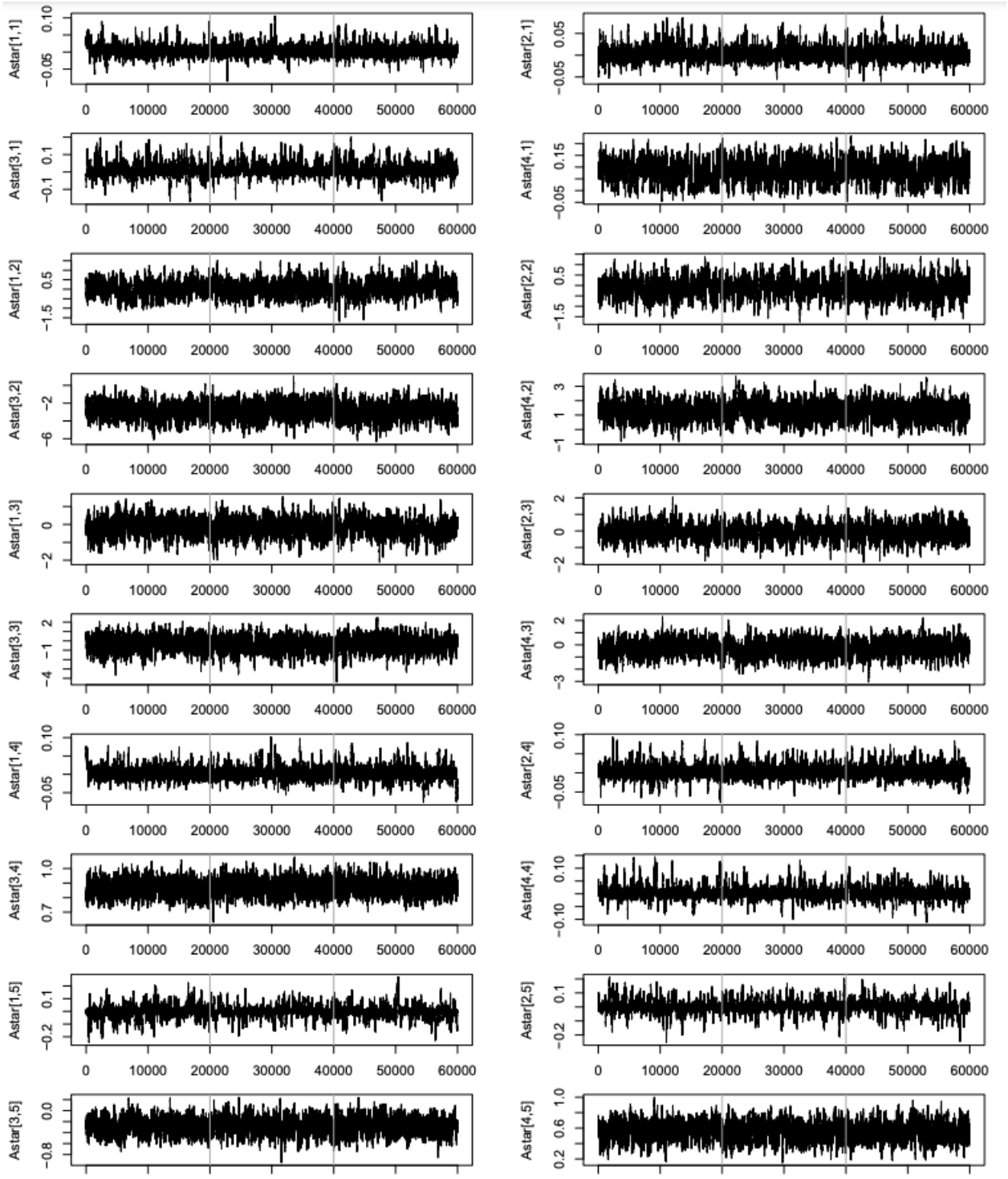

Figure 5.4: Experiment 1. Chain convergence of VAR(1) with n=4 in the first regime. Grey lines show the densities of the parallel chains.

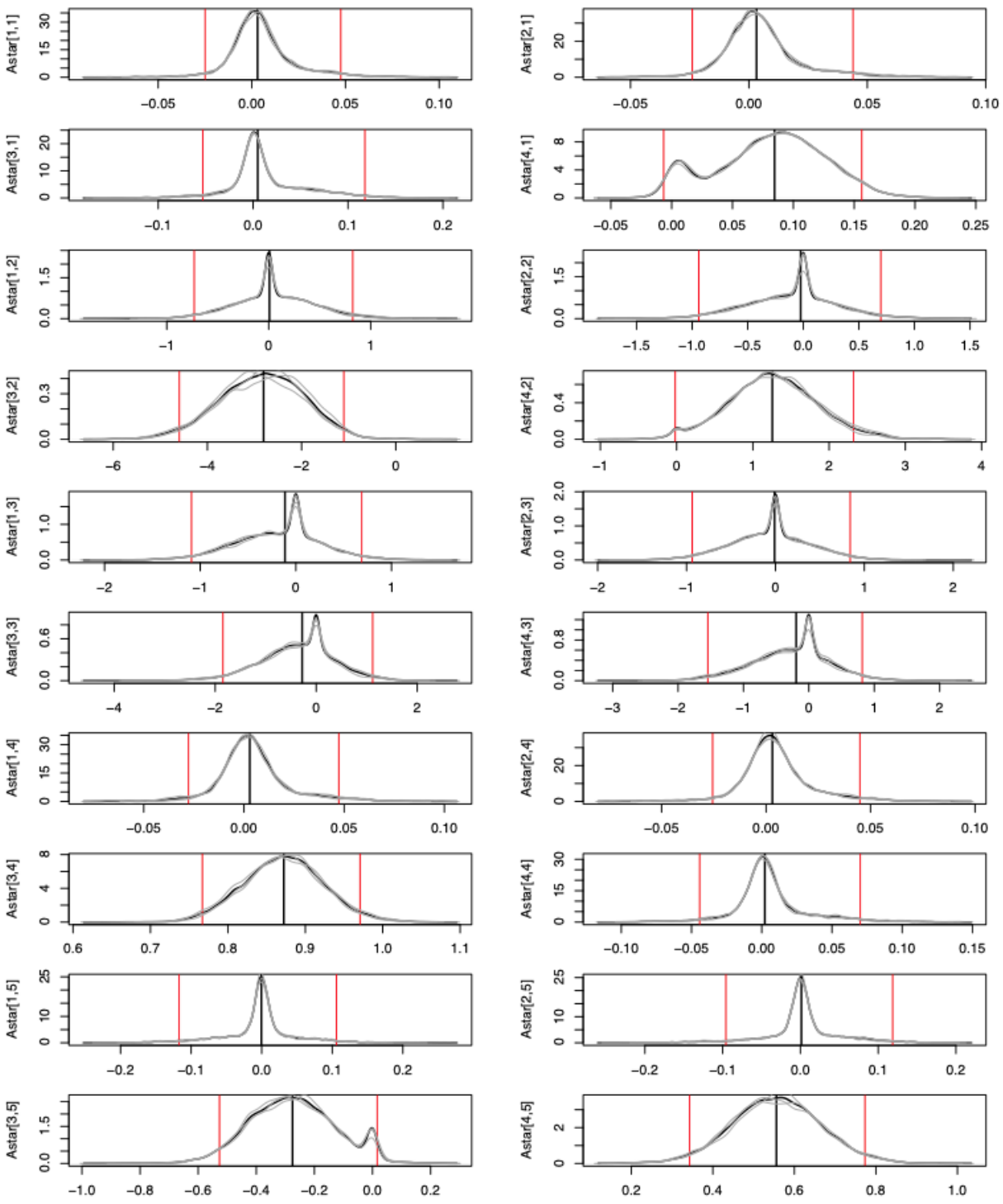

Figure 5.5: Experiment 1. Density plots of VAR(1) with n=4 in the first regime. Grey lines separate the parallel chains, and red lines indicate the 95% HDI.

Table 5.2: Experiment 1. JAGS summary results for the first regime. Significant parameters are shown in bold.

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC%ofSD	SSeff	AC.20	psrf
Astar[1,1]	-0.0249	0.0031	0.0475	0.0052	0.0169	0.0015	0.0006	3.7	734	0.7567	1.0027
Astar[2,1]	-0.0239	0.0032	0.0440	0.0052	0.0161	0.0023	0.0005	3.3	892	0.7364	1.0020
Astar[3,1]	-0.0529	0.0049	0.1178	0.0139	0.0399	0.0015	0.0018	4.4	512	0.8253	1.0042
Astar[4,1]	-0.0066	0.0846	0.1562	0.0810	0.0449	0.0902	0.0018	4.1	593	0.8075	1.0004
Astar[1,2]	-0.7304	0.0070	0.8241	0.0492	0.3806	0.0262	0.0141	3.7	732	0.7754	1.0083
Astar[2,2]	-0.9395	-0.0236	0.6974	-0.0937	0.4005	-0.0828	0.0140	3.5	821	0.7502	1.0021
Astar[3,2]	-4.5966	**-2.8076**	-1.1027	-2.8319	0.9008	-2.7713	0.0305	3.4	874	0.7472	1.0082
Astar[4,2]	-0.0185	1.2554	2.3212	1.2635	0.5866	1.2257	0.0188	3.2	971	0.7292	1.0034
Astar[1,3]	-1.0908	-0.1140	0.6883	-0.1714	0.4410	-0.1466	0.0145	3.3	924	0.7311	1.0032
Astar[2,3]	-0.9359	-0.0102	0.8406	-0.0525	0.4395	-0.0845	0.0153	3.5	825	0.7497	1.0031
Astar[3,3]	-1.8456	-0.2779	1.1183	-0.3503	0.7354	-0.2597	0.0222	3	1096	0.6941	1.0054
Astar[4,3]	-1.5442	-0.1934	0.8164	-0.2599	0.5840	-0.2070	0.0194	3.3	910	0.7246	1.0079
Astar[1,4]	-0.0277	0.0029	0.0473	0.0049	0.0172	0.0014	0.0006	3.4	844	0.7363	1.0089
Astar[2,4]	-0.0256	0.0030	0.0449	0.0048	0.0164	0.0016	0.0005	3.3	923	0.7236	1.0061
Astar[3,4]	0.7671	**0.8722**	0.9707	0.8720	0.0517	0.8739	0.0015	2.9	1160	0.6484	1.0060
Astar[4,4]	-0.0442	0.0022	0.0701	0.0055	0.0256	0.0012	0.0011	4.4	508	0.8211	1.0131
Astar[1,5]	-0.1170	-0.0005	0.1060	-0.0018	0.0485	0.0001	0.0021	4.3	545	0.8288	1.0038
Astar[2,5]	-0.0958	0.0012	0.1186	0.0031	0.0458	0.0000	0.0019	4.2	555	0.8322	1.0072
Astar[3,5]	-0.5268	-0.2752	0.0162	-0.2707	0.1475	-0.2738	0.0051	3.4	848	0.7357	1.0017
Astar[4,5]	0.3429	**0.5560**	0.7732	0.5565	0.1087	0.5600	0.0036	3.3	934	0.7199	1.0018

5.4.2 Experiment 2. Energy Markets

The second experiment examines the nuances of the energy markets. We regress the log returns of commodity indices (crude oil and natural gas), the energy stock index, and the changes in two macro variables on their own lagged data. We try to understand whether the commodity market has a lagged cross-relationship with the stock market, the macro variables, and itself. After running the experiment, our model again does not identify any change points in the mean parameter values in the time series. Thus, the results from the single regime will be presented.

$n = \{\text{SPGSCLTR, SPGSNGTR, IXETR, CPI}\}$ is equivalent to $n = \{1, 2, 3, 4\}$ in the Equation 5.1.

Figure 5.3 shows the variables chosen for the second experiment over time. The data includes every month from January 2013 to December 2022, corresponding to 120 data points. Note that the stock indices are transformed into log returns, and the CPI is the time series of monthly changes in deseasonalized rates.

The matrix $A^*_{1,2CP}$ in 5.4 provides the median values of each parameter in the first regime for the second experiment.

$$A^*_{1,2CP} = \begin{bmatrix} -0.0044 & 0.0094 & -0.0369 & 0.5540 & 0.0003 \\ -0.0160 & 0.1149 & -0.0816 & -0.3152 & 0.0777 \\ 0.0013 & -0.0293 & -0.0257 & -0.0068 & 0.0013 \\ \mathbf{0.1268} & 0.5698 & 0.1451 & 0.6632 & \mathbf{0.4186} \end{bmatrix} \tag{5.4}$$

$$A^*_{2OLS} = \begin{bmatrix} \mathbf{-0.0192} & -0.0369 & \mathbf{-0.0982} & \mathbf{0.6886} & 0.0205 \\ \mathbf{-0.0372} & 0.1405 & -0.1110 & \mathbf{-0.3880} & \mathbf{0.1197} \\ 0.0013 & \mathbf{-0.0720} & -0.0575 & \mathbf{-0.0071} & \mathbf{0.0082} \\ \mathbf{0.1254} & \mathbf{0.5695} & \mathbf{0.1672} & \mathbf{0.7772} & \mathbf{0.4214} \end{bmatrix} \tag{5.5}$$

We observe a few significant parameters that are far from zero. This is due to the results in Table 5.3, the density plots in Figure 5.8, and the $A^*_{1,2CP}$ matrix in Equation 5.4. The significant parameter values are indicated in bold.

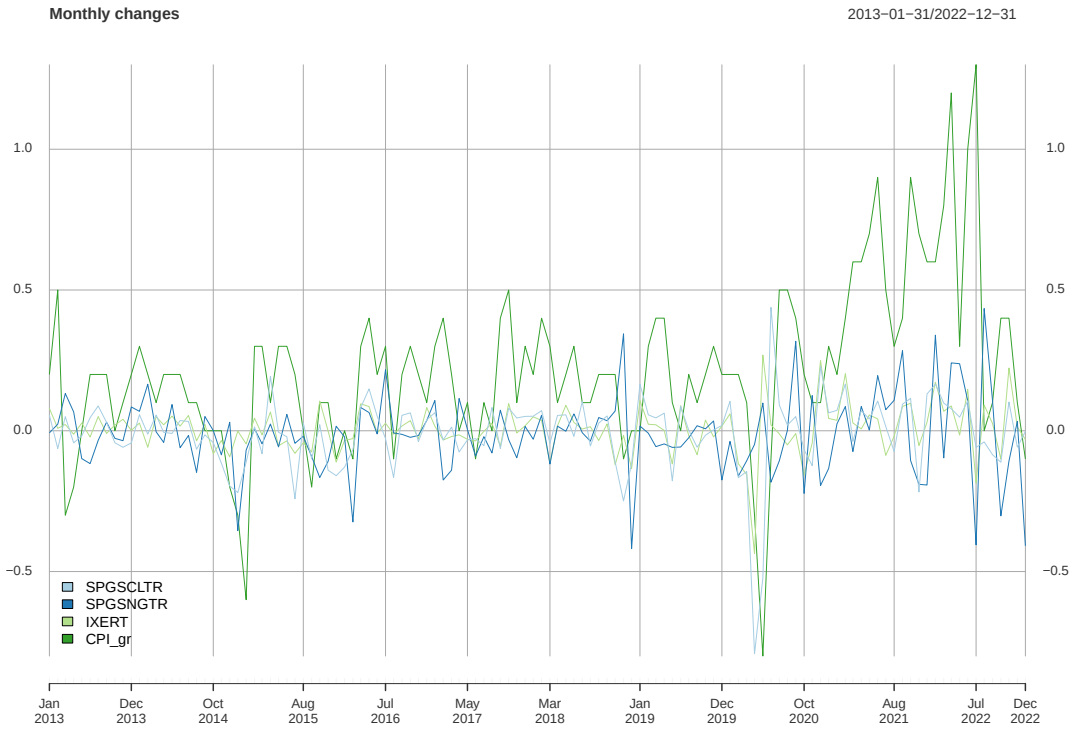

Figure 5.6: Experiment 2. Monthly changes in variables. Changes in the unemployment rate are standardized.

We first look at the parameter values of the 4th equation, the changes in the CPI, see Equation 5.4. Again, we see that the lagged change in the CPI positively affects the current change in the CPI itself by 0.4186. We also find a positive and significant intercept (0.1268). In the CPI equation, we find other rather large parameter values. It seems that the lagged log returns of the crude oil and energy stock index positively affect the current change in the CPI (0.5698 and 0.6632, respectively). Since the 95% HDI credible intervals include zeros, we need stronger evidence to declare significance.

In the first equation, we also find a meaningful parameter of 0.5635, which signals the positive impact of the lagged log return of the energy stock index on the return of the crude oil index. Finally, although most of the density mass is concentrated away from zero, with the HDI of 95%, we find zero on the far left of the confidence

interval.

We again compare OLS with the Bayesian technique. As in the first experiment, we observe that the parameter values that are significant from a p-value perspective in OLS methods are not necessarily significant in our sparse *CP-VAR(1)* model. Nevertheless, again, both models yield comparable parameter values. The complete results from the sparse *CP-VAR(p)* are in Appendix C, and OLS results are in Appendix D.

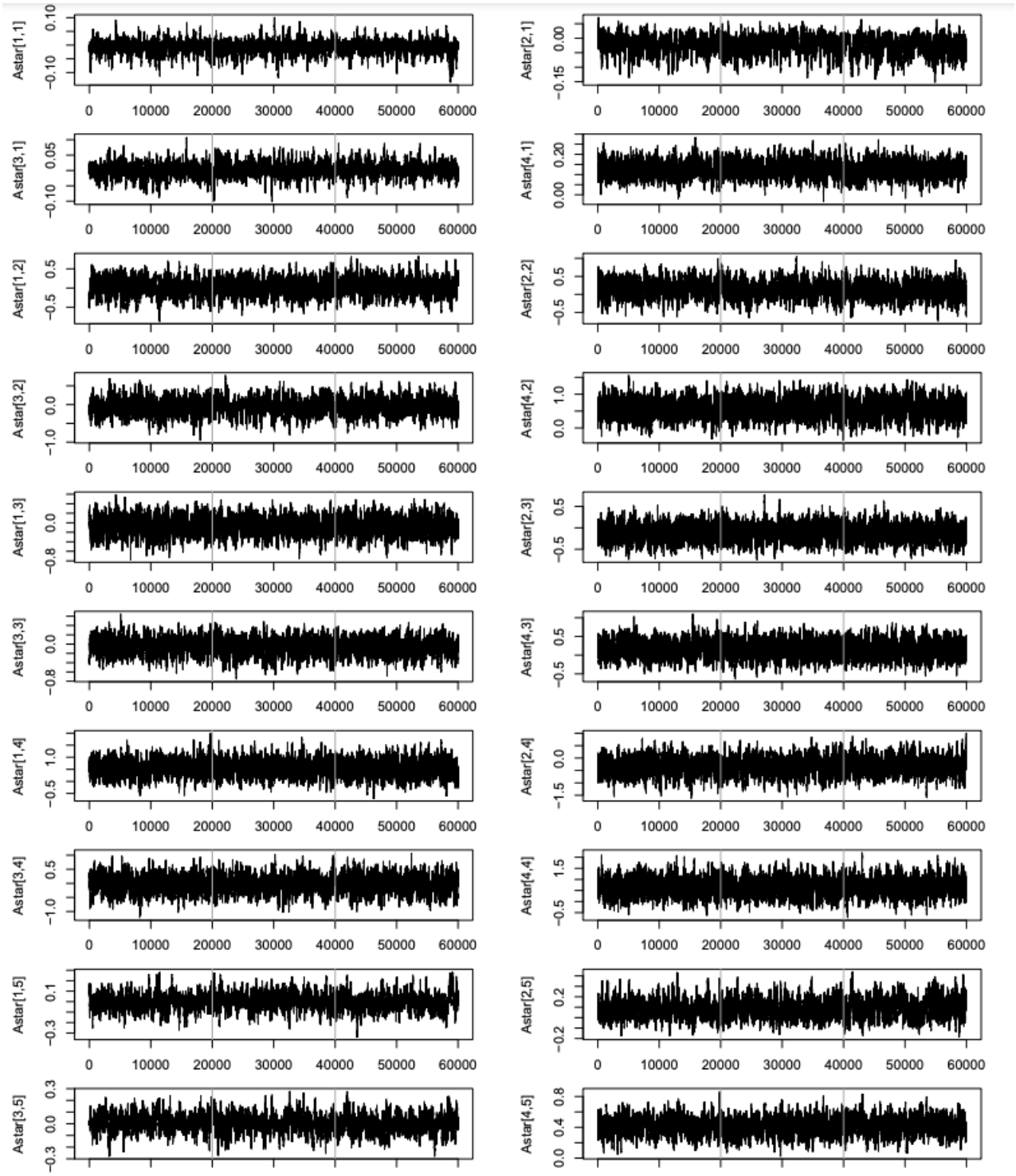

Figure 5.7: Experiment 2. Chain convergence of VAR(1) with n=4 in the first regime. Grey lines show the densities of the parallel chains.

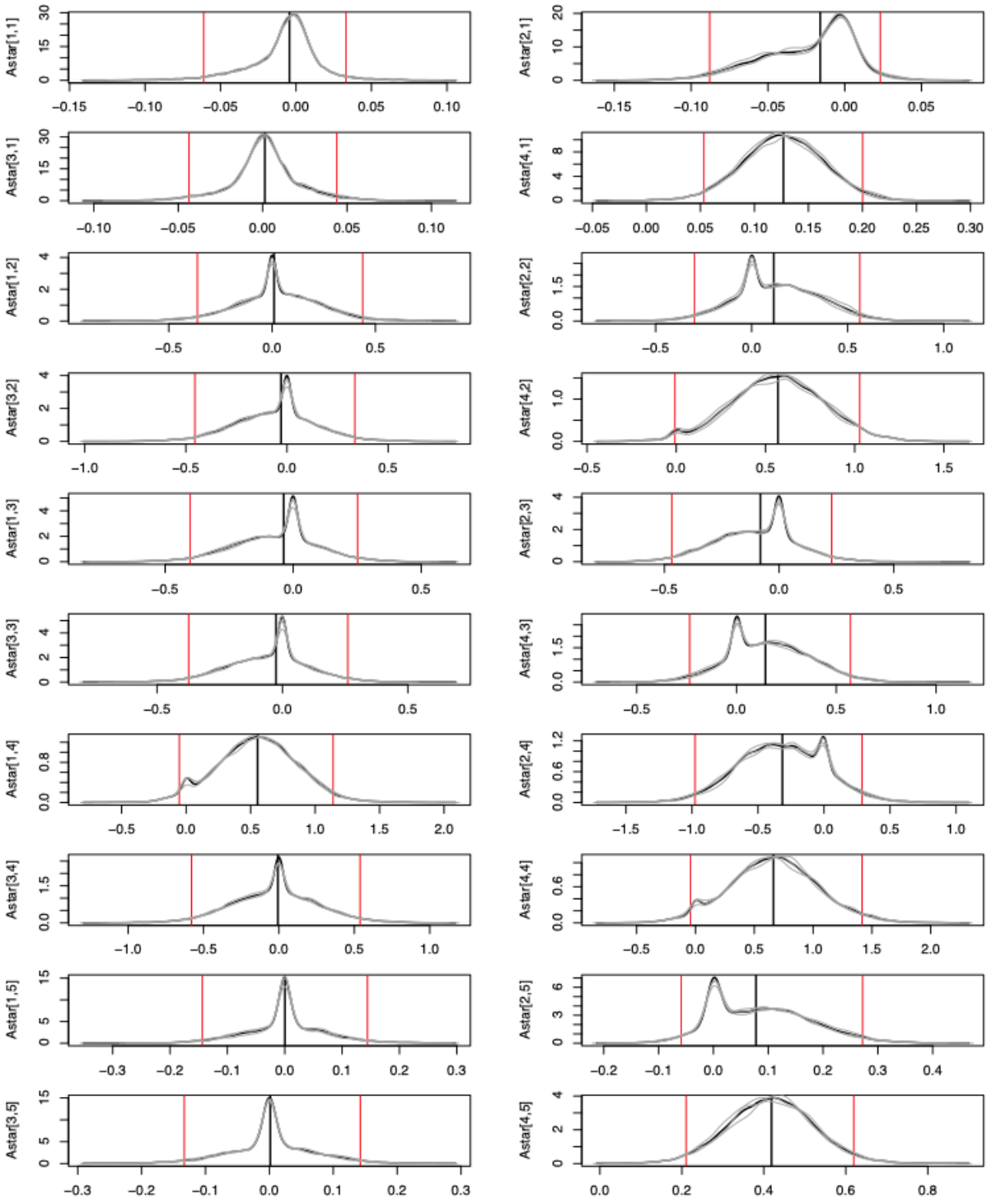

Figure 5.8: Experiment 2. Density plots of VAR(1) with n=4 in the first regime. Grey lines separate the parallel chains, and red lines indicate the 95% HDI.

Table 5.3: Experiment 2. JAGS summary results for the first regime. Significant parameters are shown in bold.

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC%ofSD	SSeff	AC.20	psrf
Astar[1,1]	-0.0611	-0.0044	0.0332	-0.0075	0.0222	-0.0023	0.0008	3.5	807	0.7503	1.0007
Astar[2,1]	-0.0879	-0.0160	0.0231	-0.0236	0.0298	-0.0065	0.0012	4.1	587	0.8060	1.0076
Astar[3,1]	-0.0436	0.0013	0.0440	0.0016	0.0199	0.0004	0.0006	3.2	982	0.7028	1.0035
Astar[4,1]	0.0532	**0.1268**	0.2002	0.1269	0.0375	0.1261	0.0012	3.2	955	0.6962	1.0044
Astar[1,2]	-0.3614	0.0094	0.4384	0.0288	0.1960	0.0267	0.0063	3.2	980	0.7281	1.0072
Astar[2,2]	-0.2974	0.1149	0.5631	0.1296	0.2204	0.0912	0.0074	3.4	877	0.7352	1.0094
Astar[3,2]	-0.4553	-0.0293	0.3355	-0.0540	0.1959	-0.0482	0.0066	3.4	881	0.7393	1.0015
Astar[4,2]	-0.0075	0.5698	1.0280	0.5676	0.2599	0.5838	0.0080	3.1	1059	0.6817	1.0033
Astar[1,3]	-0.4024	-0.0369	0.2515	-0.0638	0.1615	-0.0466	0.0042	2.6	1454	0.6200	1.0012
Astar[2,3]	-0.4678	-0.0816	0.2289	-0.0992	0.1760	-0.0655	0.0047	2.7	1407	0.6277	1.0013
Astar[3,3]	-0.3740	-0.0257	0.2611	-0.0528	0.1561	-0.0392	0.0045	2.9	1207	0.6720	1.0032
Astar[4,3]	-0.2344	0.1451	0.5711	0.1567	0.2056	0.0955	0.0054	2.6	1469	0.6236	1.0031
Astar[1,4]	-0.0531	0.5540	1.1394	0.5505	0.3123	0.5623	0.0089	2.8	1234	0.6556	1.0009
Astar[2,4]	-0.9774	-0.3152	0.2875	-0.3180	0.3267	-0.2088	0.0101	3.1	1052	0.6978	1.0055
Astar[3,4]	-0.5797	-0.0068	0.5379	-0.0223	0.2772	-0.0300	0.0086	3.1	1041	0.7065	1.0004
Astar[4,4]	-0.0426	0.6632	1.4164	0.6630	0.3761	0.6554	0.0113	3	1098	0.6844	1.0022
Astar[1,5]	-0.1436	0.0003	0.1443	0.0014	0.0670	-0.0006	0.0024	3.6	789	0.7684	1.0078
Astar[2,5]	-0.0587	0.0777	0.2721	0.0854	0.0892	0.0369	0.0035	4	635	0.8013	1.0034
Astar[3,5]	-0.1333	0.0013	0.1423	0.0032	0.0645	0.0007	0.0023	3.6	768	0.7691	1.0052
Astar[4,5]	0.2103	**0.4186**	0.6195	0.4182	0.1040	0.4199	0.0034	3.3	937	0.7240	1.0032

5.5 Summary of the Results

According to the Gelman & Rubin statistics, the chains of the parameter values converged appropriately in both experiments; the *psrf* values remained below 1.1. From a visual inspection perspective, the three parallel chains of each parameter overlap well in the trace and density plots.

Recall the research questions defined earlier.

- **RQ 5.1:** Does the model detect any change points in the real data? If yes, do these points have any meaningful interpretation?
- **RQ 5.2:** Are there any significant parameter values within the regimes? If yes, what is the interpretation of the parameters?

Next, we summarize the results of both experiments by answering our research questions.

RQ 5.1: Presence of Change Points

The results of both experiments with the sparse *CP-VAR(1)* model do not detect any change points. This finding means that there is only a single regime in the autoregressive parameters of the model. Furthermore, this is equivalent to having a standard *VAR(1)* model that can be estimated using the OLS frequentist method.

RQ 5.2: Significant Parameters

We find several significant parameter values in both experiments from our sparse *CP-VAR(1)* models. However, the significant parameters are only in the macro-variable equations and not the tradable asset returns.

We also provide a comparison of the results between Bayesian and frequentist methods. In other words, we compare the parameters from our sparse *CP-VAR(1)* and the OLS *VAR(1)*. Both models provide similar parameter values in terms of size. The parameters found to be significant in our sparse *CP-VAR(1)* are also significant in the OLS *VAR(1)*. We also observe that the OLS method returns significant parameter values even in the index equations.

chapterConclusions

Before we state our findings, let us reflect on our accomplishments. First, we implemented the sparse *CP-VAR(p)* developed by [Dufays et al., 2021] using the JAGS software. See the Methodology, Chapter 3. The program required us to specify the model using priors and likelihood using a DAG. The model specification

in JAGS enables us to run the MCMCs without writing the sampling program from scratch.

The simulation study, Chapter 4, puts our model to performance and robustness tests. First, we check the model's performance in the setting where T=120. This setting is similar to the one in Experimentation, Chapter 5. We also test the model's ability to find multiple change points when T=500.

In Chapter 5, Experimentation, we run our model on real financial and economic data. We aim to identify the lagged interrelationships between the log returns of stock and commodity indices and the changes in macro variables such as the CPI and the unemployment rate.

5.6 Findings

We present our findings from both chapters, the simulation study, Chapter 4, and the experiment, Chapter 5.

5.6.1 Simulation Study

The simulation study on bivariate sparse *VAR(1)* shows that our model can find multiple change points when T=500. However, multiple change points were found in only one-third of our experiments. Our model also fails to find a change point between two short regimes.

The change point detection was slightly worse in the time series of T=120. In this setting, we identified the first change point (18/20 times) and the third (4/20 times). In the test where no change points were present, our model could identify their absence in most simulated scenarios.

5.6.2 Experimentation

Both experiments with real data (n=4) detected a single regime (no change points) in the sparse *CP-VAR(1)*, which is equivalent to a standard *VAR(1)* model. Despite

the detection of a single regime, we were able to identify several significant variables in both experiments. However, the significance persists mainly in the macro variable equations, not the asset index ones. In addition, the comparison between the OLS and Bayesian estimates for the single regime shows that our sparse *CP-VAR(1)* finds fewer significant parameters than the OLS equivalent model.

Although we could not identify the change points in the mean parameters in the experiments with real data, change points may be present in the covariance matrix. In fact, [Dufays et al., 2021] ran the experiment with their model on excess returns, dividend yields, and 1-month T-bills. They found that "most of the mean parameters are invariant over time and often insignificant in terms of the posterior probability." Instead, they identified the breaks in the covariance matrix. Unfortunately, the setup of our model does not allow us to infer the change points in the covariance matrix, which is a significant drawback of our model.

5.7 Further Work

We suggest several topics for further work. These suggestions are based on the limitations of our model and its MCMC inefficiency.

First, extending the model with a time-varying variance-covariance matrix would be useful. This improvement is especially relevant for the time series in economics and finance. Although [Dufays et al., 2021] included a time-varying variance-covariance matrix in their model, we removed it to limit the scope of the thesis.

The simplification of the calibration procedure may have added to the poorer performance of the model. Therefore, we suggest reverting to the original calibration procedure as specified in the [Dufays et al., 2021] research paper.

Another limitation of our study is the poor efficiency of the MCMC. In particular, the lack of computational power in the MCMC has reduced our ability to perform more extensive simulation studies. As a result, we suggest running the simulations on longer time series and many more times to test the model's capabilities better.

[Dufays et al., 2021] have also evaluated their model in forecasting applications. Therefore, our research can be further extended to forecasting applications as it is a

relevant topic in asset management.

Another interesting topic could be an exploration of modified versions of our model. The DAGs in BUGS/JAGS provide an easy way to build and adapt the Bayesian models. Thus, our model can be easily adapted by changing the priors and the likelihood specification in the model description. Furthermore, with JAGS, we could further explore how well change points are detected in other multivariate models, for example, in a vector autoregressive moving average (VARMA) model.

Appendix A

Appendix: JAGS Model

A.1 JAGS Model in BUGS Language

```
model {


# 2.2 Section: Specification of shrinkage prior
for (i in 1:(K+1)) {
  vec_deltaA[1:(N*k), i] ~ dmnorm.vcov(rep(0, N*k), Gamma[1:(N*k),
   1:(N*k), i])    # def vec_deltaA per regime, i in 1 to K
}

# when i=[1:K+1]
for (i in 1:(K+1)) {
    for (j in 1:(N*k)) {  # for j in 1:N*k, for regime i in 1:K+1
        for (l in 1:(N*k)) {
            Gamma[j, l, i] <- ifelse(l==j, phi[i, j], 0)
        }
        phi[i, j] <- di[z[i,j]] * vari[i, j]
        # d1=c0 and d2=c1 passed from R to JAGS
        vari[i, j] <- 1 / g[i, j]
        g[i, j] ~ dgamma(nu1/2, nu2/2) # nu1 and nu2 passed from R
   to JAGS
        zz[i, j] ~ dbern(1 - w[i, j])
```

```
            z[i, j] <- ifelse(zz[i, j]==0, 1, 2)
            # JAGS does not handle index 0
        w[i, j] ~ dbeta(alpha, beta) # passed from R to JAGS
    }
}


# 2.3 Section: Sampling break dynamics
# states
s[1] <- 1
for (t in 1:(capT)) {
    pi[t] ~ dbeta(K, (capT-K))
    ber[t] ~ dbern(pi[t])
    s[t+1] <- s[t] + ber[t]
}

# indicator function on states vector s, capT x (K+1)
for (t in 1:capT) {
 for (i in 1:(K+1)) {
     indic[t, i] <- ifelse(s[t]>=i, 1, 0)
  }
}


# 3.1 Section: Sampling the mean parameters
# manipulation step: make a vec out of vec_deltaA matrix
for (i in 1:(K+1)) {
  A_vec[((i-1)*N*k+1) : (i*N*k)] <- vec_deltaA[,i]
}

# create matrix A out of A_vec = A[1:N, 1:k_bar]
for (i in 1:k_bar) {
  A[1:N, i] <- A_vec[((i-1)*N+1) : (i*N)]
}

# create matrix Astar = N x k*(K+1), or N x k_bar
Astar[1:N, (0*k+1):(1*k)] <- A[, 1:k]
```

```
for (i in 1:K) {
  Astar[1:N, (i*k+1):((i+1)*k)] <- Astar[1:N, ((i-1)*k+1):(i*k)] + A
   [, (i*k+1):((i+1)*k)]
}

# create X_tilde = product between X and the indicator function,
   k_bar x T
tX <- t(X)
for (i in 1:(K+1)) {
  for (j in 1:k) {
    for (t in 1:capT) {
        foo[t, j, i]   <- tX[t, j] * indic[t, i]
    }
  }
  tX_tilde[1:capT, ((i-1)*k+1) : (i*k)] <- foo[1:capT, 1:k, i]
}

X_tilde <- t(tX_tilde)


# Likelihood

# Process model    A  N x k_bar    X_tilde  k_bar x capT
    # create matrix for the mean parameter
    AX_tilde <- A %*% X_tilde

    # create a precision matrix for Tau, invert it to obtain Sigma
    Tau[1:N, 1:N] ~ dwish(R_shape, k_df)
    Sigma[1:N, 1:N] <- inverse(Tau)

# Data model
    for (t in 1:capT) {
      Y[1:N, t] ~ dmnorm( AX_tilde[1:N, t], Tau[1:N, 1:N] )
    }


}
```

Appendix B

Appendix: R Code

B.1 R code. Example of the Simulation Study

```
# set working directory to the R file location
setwd("~/Library/Mobile Documents/")

# load the required libraries libraries
library(runjags)
library(rockchalk) # mvrnorm
library(vars) # VAR
library(xlsx) # write.xlsx
library(HDInterval) # hdi
library(dplyr) # select
library(stats)

# write a loop to run code for some number of times
for (l in 1:20) {

  folder_name <- "results/sim-study-C-120/"
  name_text <- paste0("sim-study-C-120_", l) # l

  # function to simulate data from VAR(1)
  sim_VAR1 <- function(T, n, vecA, vecSigma) {
    A <- matrix(vecA, nrow = n, ncol = (n+1), byrow = FALSE) # by
```

```
 column
22   Sigma <- matrix(vecSigma, nrow = n, ncol = n, byrow = FALSE) #
  by column
23   y <- matrix(NA, T, n)
24   y[1, ] <- A[, 1]
25   set.seed(l) # seed per simulation
26   for (t in 2:T) {
27     y[t, ] <- mvrnorm(1, as.matrix(A[, 1]) + A[, 2:(n+1)] %*% y[t
  -1, ], Sigma)
28   }
29   return(y)
30 }
31
32 # Simulate the bivariate VAR(1) process according to DGP C
  parameters; see Dufays (2021)
33 simT <- 120
34
35 DGP_C <- rbind(
36   sim_VAR1(T=floor(0.5*simT),            n=2, vecA=c(0, 0, .5, .1,
   .1, .5),   vecSigma = c(1, 0, 0, 1)),
37   sim_VAR1(T=floor((2*simT/3-0.5*simT)), n=2, vecA=c(0, 0, -.5,
  .1, .1, -.5), vecSigma = c(1, 0, 0, 1)),
38   sim_VAR1(T=floor((3*simT/4-2*simT/3)), n=2, vecA=c(0, 0, .5, .1,
   .1, .5),   vecSigma = c(1, 0, 0, 1)),
39   sim_VAR1(T=floor(simT/4),              n=2, vecA=c(0, 0, -.5,
  .1, .1, -.5), vecSigma = c(1, 0, 0, 1)) )
40 colnames(DGP_C) <- c("y1", "y2")
41
42 # plot the simulated data and write the pdf file into the folder
  location
43 pdf(file = paste0(folder_name, "sim_plot_", name_text, ".pdf"),
  width = 11.7, height = 8.3)
44 plot(x=seq(1,length(DGP_C[,1])), y=DGP_C[,1], type = "l", col="
  blue",
45      main = "Simulation", xlab="Time", ylab="y")
46 lines(x=seq(1,length(DGP_C[,2])), y=DGP_C[,2], type = "l", col="
  black")
```

```
47  dev.off()
48
49
50  # assign simulated data to a variable 'data'
51  data <- DGP_C
52
53
54  # OLS VAR estimates on the full data set
55  # select the VAR(p) lag according to the information criteria
56  VARselect(data, lag.max = 8, type = "const")[["selection"]]
57
58  # fit VAR(p) model, when p=1
59  VAR_model <- VAR(data, p=1, type = "const")
60
61  # obtain the VAR(1) results
62  res_VAR <- summary(VAR_model)
63
64  # obtain VAR(1) standard errors
65  SE_y1 <- res_VAR$varresult$y1$coefficients[,2]
66  SE_y2 <- res_VAR$varresult$y2$coefficients[,2]
67  SE_VAR <- c(SE_y1, SE_y2)
68
69  # obtain VAR(1) parameter estimates
70  E_y1 <- t(res_VAR$varresult$y1$coefficients[,1])
71  E_y2 <- t(res_VAR$varresult$y2$coefficients[,1])
72  E_VAR <- rbind(E_y1, E_y2)
73  E_VAR <- cbind(E_VAR[, 3], E_VAR[, 1:2])
74  VAR_est <- c(E_VAR)
75
76  # prepare data and other parameters for JAGS
77  burnin <- 20000
78  sample <- 20000
79  thin <- 2
80
81  N <- ncol(data) # n variables in the data
82  P <- 1          # p lags
83  k <- (N*P) + 1  # k parameters to estimate per regime
```

```
84
85  # formatting of variable Y
86  Time <- nrow(data)
87  Y <- t(data[(P+1):Time,]) # drop dates and first p values
88  capT <- ncol(Y)           # T, the lenght of the time series
89
90  K <- round(log(capT), 0)  # K, implied number of breaks
91  k_bar <- (K+1)*((N*P)+1)  # k_bar, total number of pars in CP-VAR(
     p)
92
93  # create a matrix X out of lagged variables of Y
94  X <- matrix(data = 0, nrow = (Time-P),  ncol = (N*P)) # N*P = 4*6
     = 24
95  for (p in 1:P) {
96    X[ ,((p-1)*N+1):(p*N)] <- data[(P-(p-1)):(Time-p), ]
97  }
98  colnames(X) <-   rep(colnames(data), P)
99  one_vec <- matrix(1, nrow = (Time-P), ncol = 1)
100 X <- t(cbind(one_vec, X))
101
102
103 # 2.2.1 Section: Calibration of the S&S hyper-parameters; see
     Dufays(2021)
104 # define penalty function
105 pen <- function(pi, capT) {
106   val <- -( log(pi/(1-pi)) + log(capT) )
107   return(val)
108 }
109
110 # simplify: take the mean of SEs from standard VAR model
111 sig_hat <- mean(SE_VAR)
112
113 a <- (1/3) * sig_hat
114 b <- (a * (19*T*K+1)) / K
115
116 pen99 <- pen(0.99, capT)
117 pen75 <- pen(0.75, capT)
```

```
118
119 w_99 <- a * (1-exp(pen99))
120 w_75 <- a * (1-exp(pen75))
121
122 # calibration of alpha and beta through the optimization function
123 calib_fct <- function(th) {
124   ( (pbeta(w_99, th[1], th[2]) - pbeta(w_75, th[1], th[2]) ) -
     0.999)^2
125 }
126
127 optim_res <- optim(par=c(1,1), fn=calib_fct)
128 a_cal <- optim_res$par[1]
129 b_cal <- optim_res$par[2]
130
131 # hyperparameters according to Malsinger-Walli & Wagner (2016)
132 di <- c((1/10000), 1) # r = c0=1/10000 , c1=1
133 nu <- c(10, 8)        # nu1=5 , nu2=4
134
135 # R/df = expectation of cov matrix
136 k_df <- N+1
137 R_shape <- diag(((1^2)*k_df), N, N)
138 Sig <-  R_shape/k_df # only for plotting
139
140
141 # 3.1 Section: Sampling the mean parameters
142 # the code of JAGS model is defined in .txt file
143
144 vec_deltaA <- matrix(0, nrow = N*k, ncol = (K+1))
145 for (i in 1:(K+1)) {
146   vec_deltaA[, i] <- VAR_est
147 }
148 init_list <- list(vec_deltaA = vec_deltaA)
149
150 # prepare the data for JAGS
151 dat <- dump.format(list(Y=Y, X=X, capT=capT, N=N,
152                         K=K, k=k, k_bar=k_bar,
153                         alpha=a_cal, beta=b_cal,
```

```
                          di=di, nu1=nu[1], nu2=nu[2],
                          R_shape=R_shape, k_df=k_df))

# initialize 3 chains with initial values and
# use different pseudo-random number generators
inits1 <- dump.format(c(init_list, list( .RNG.name="base::Super-
 Duper", .RNG.seed=99999 )))
inits2 <- dump.format(c(init_list, list( .RNG.name="base::Wichmann
 -Hill", .RNG.seed=1234 )))
inits3 <- dump.format(c(init_list, list( .RNG.name="base::Mersenne
 -Twister", .RNG.seed=6666 )))

# tell JAGS which latent variables to monitor
monitor = c("A", "Astar", "Sigma", "s")

# run the function that fits the models using JAGS
results <- run.jags(model="models/model-5-deltaA.txt", modules="
 glm",
                    monitor=monitor, data=dat, n.chains=3,
                    inits=c(inits1, inits2, inits3),
                    plots = FALSE, method="parallel",
                    burnin=burnin, sample=sample, thin=thin)

# read the summary of the results
res0 <- add.summary(results)
res <- res0$summaries

# combine the MCMC chains
chains <- rbind(results$mcmc[[1]], results$mcmc[[2]], results$mcmc
 [[3]])

chains_A <- as.matrix(as.data.frame(chains) %>% dplyr::select(
 starts_with("A[")))
chains_Astar <- as.matrix(as.data.frame(chains) %>% dplyr::select(
 starts_with("Astar[")))
chains_s <- as.data.frame(chains) %>% dplyr::select(starts_with("s
 ["))
```

```
183
184 s <- t(rbind(apply(chains_s, 2, median), hdi(chains_s, 0.95))) #
     median
185 colnames(s) <- c("median", "lower", "upper")
186
187 tau <- rep(0, (nrow(s)))
188 for (i in 1:(nrow(s)-1)) {
189   tau[i+1] <- ifelse( s[i+1, 1] > s[i, 1], (i+1), 0)
190 }
191 cp <- as.data.frame(tau[tau>0])
192 colnames(cp) <- "cp"
193
194 # summarize inputs
195 inputs <- data.frame(
196   label = c("simT", "capT", "N" ,"P", "K", "k", "k_bar", "sig_hat"
     , "a_cal", "b_cal", "nu1", "nu2", "c0", "c1", "burnin", "sample",
      "thin"),
197   par = c(simT, capT, N, P, K, k, k_bar, sig_hat, a_cal, b_cal, nu
     [1], nu[2], di[1], di[2], burnin, sample, thin))
198
199
200 # write all summary results in a workbook
201 xlsx_name <- paste0(folder_name, "results_", name_text, ".xlsx")
202 write.xlsx(as.data.frame(res), file = xlsx_name,
203            sheetName = "res_table", append = FALSE)
204 write.xlsx(cp, file = xlsx_name,
205            sheetName="cp", append=TRUE)
206 write.xlsx(inputs, file = xlsx_name,
207            sheetName="inputs", append=TRUE)
208 write.xlsx(data, file = xlsx_name,
209            sheetName="data", append=TRUE)
210 write.xlsx(Y, file = xlsx_name,
211            sheetName="Y", append=TRUE)
212 write.xlsx(X, file = xlsx_name,
213            sheetName="X", append=TRUE)
214
215
```

```
# write all the cahins in excel (optional)
# write.xlsx(chains, file = paste0(folder_name, "chains_",
 name_text, ".xlsx"), sheetName="chains", append=FALSE)


# plot the results and write the pdf files into the folder
 location
par_names <- colnames(chains)

# chains
pdf(file = paste0(folder_name, "chains_all_", name_text, ".pdf"),
 width = 8.3, height = 11.7)
par(mfrow=c(12,2), mai=c(0.3, 0.6, 0.1, 0.3)) #c(0.02, 0.6, 0.02,
 0.02))
for (i in 1:length(par_names)) {
  plot(x=seq(1,nrow(chains)), y=chains[, i], type = "l", col="
 black",
       ylab=paste(par_names[i])) #xaxt='n'
  abline(v=sample, col="darkgrey")
  abline(v=(2*sample), col="darkgrey") # separate the three
 parallel chains
}
dev.off()

# densities with 95% HDI
pdf(file = paste0(folder_name, "dens_all_", name_text, ".pdf"),
 width = 8.3, height = 11.7)
par(mfrow=c(12,2), mai=c(0.3, 0.6, 0.1, 0.3))
for (i in 1:length(par_names)) {
  plot(density(chains[, i]), ylab=paste(par_names[i]), main=" ",
 cex=0.75, lwd=1.5)
  abline(v=res[i, 2], col="black", lwd=1.5)
  abline(v=res[i, 1], col="red")
  abline(v=res[i, 3], col="red")
  lines(density(results[["mcmc"]][[1]][,i]), col="darkgrey")
  lines(density(results[["mcmc"]][[2]][,i]), col="darkgrey")
  lines(density(results[["mcmc"]][[3]][,i]), col="darkgrey")
```

```
  }
  dev.off()

  # ACF plot
  pdf(file = paste0(folder_name, "acf_A_", name_text, ".pdf"), width
    = 8.3, height = 11.7)
  par(mfrow=c(8,2), mai=c(0.3, 0.6, 0.1, 0.3))
  for (i in 1:length(colnames(chains_A))) {
    plot(acf(chains_A[, i], plot = FALSE), main="", ylab=paste(
   colnames(chains_A)[i]))
  }
  dev.off()

  pdf(file = paste0(folder_name, "acf_Astar_", name_text, ".pdf"),
   width = 8.3, height = 11.7)
  par(mfrow=c(8,2), mai=c(0.3, 0.6, 0.1, 0.3))
  for (i in 1:length(colnames(chains_Astar))) {
    plot(acf(chains_Astar[, i], plot = FALSE), main="", ylab=paste(
   colnames(chains_Astar)[i]))
  }
  dev.off()

  # summary results plot (chains, ECDF, density, ACF)
  pdf(file = paste0(folder_name, "sum_res_", name_text, ".pdf"),
   width = 8.3, height = 8.3)
  par(mfrow=c(10,2), mai = c(0.02, 0.6, 0.02, 0.02))
  plot(res0)
  dev.off()

}
```

Appendix C

Appendix: Experimentation Results

C.1 Experiment 1: Value vs. Growth

Table C.1: Experiment 1. JAGS summary results

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC	SSeff	AC.20	psrf
Astar[1,1]	-0.0249	0.0031	0.0475	0.0052	0.0169	0.0015	0.0006	3.7	734	0.7567	1.0027
Astar[2,1]	-0.0239	0.0032	0.0440	0.0052	0.0161	0.0023	0.0005	3.3	892	0.7364	1.0020
Astar[3,1]	-0.0529	0.0049	0.1178	0.0139	0.0399	0.0015	0.0018	4.4	512	0.8253	1.0042
Astar[4,1]	-0.0066	0.0846	0.1562	0.0810	0.0449	0.0902	0.0018	4.1	593	0.8075	1.0004
Astar[1,2]	-0.7304	0.0070	0.8241	0.0492	0.3806	0.0262	0.0141	3.7	732	0.7754	1.0083
Astar[2,2]	-0.9395	-0.0236	0.6974	-0.0937	0.4005	-0.0828	0.0140	3.5	821	0.7502	1.0021
Astar[3,2]	-4.5966	**-2.8076**	-1.1027	-2.8319	0.9008	-2.7713	0.0305	3.4	874	0.7472	1.0082
Astar[4,2]	-0.0185	1.2554	2.3212	1.2635	0.5866	1.2257	0.0188	3.2	971	0.7292	1.0034
Astar[1,3]	-1.0908	-0.1140	0.6883	-0.1714	0.4410	-0.1466	0.0145	3.3	924	0.7311	1.0032
Astar[2,3]	-0.9359	-0.0102	0.8406	-0.0525	0.4395	-0.0845	0.0153	3.5	825	0.7497	1.0031
Astar[3,3]	-1.8456	-0.2779	1.1183	-0.3503	0.7354	-0.2597	0.0222	3	1096	0.6941	1.0054
Astar[4,3]	-1.5442	-0.1934	0.8164	-0.2599	0.5840	-0.2070	0.0194	3.3	910	0.7246	1.0079
Astar[1,4]	-0.0277	0.0029	0.0473	0.0049	0.0172	0.0014	0.0006	3.4	844	0.7363	1.0089
Astar[2,4]	-0.0256	0.0030	0.0449	0.0048	0.0164	0.0016	0.0005	3.3	923	0.7236	1.0061
Astar[3,4]	0.7671	**0.8722**	0.9707	0.8720	0.0517	0.8739	0.0015	2.9	1160	0.6484	1.0060
Astar[4,4]	-0.0442	0.0022	0.0701	0.0055	0.0256	0.0012	0.0011	4.4	508	0.8211	1.0131
Astar[1,5]	-0.1170	-0.0005	0.1060	-0.0018	0.0485	0.0001	0.0021	4.3	545	0.8288	1.0038
Astar[2,5]	-0.0958	0.0012	0.1186	0.0031	0.0458	0.0000	0.0019	4.2	555	0.8322	1.0072
Astar[3,5]	-0.5268	-0.2752	0.0162	-0.2707	0.1475	-0.2738	0.0051	3.4	848	0.7357	1.0017
Astar[4,5]	0.3429	**0.5560**	0.7732	0.5565	0.1087	0.5600	0.0036	3.3	934	0.7199	1.0018
Astar[1,6]	-1.9664	0.0164	1.9602	0.0334	0.9659	0.1014	0.0290	3	1108	0.6959	1.0015
Astar[2,6]	-1.9505	0.0042	1.8343	-0.0227	0.9322	-0.0305	0.0287	3.1	1059	0.7029	1.0006
Astar[3,6]	-1.9681	0.0096	2.0063	0.0428	0.9806	0.0029	0.0319	3.3	943	0.7302	1.0007
Astar[4,6]	-1.8601	0.0921	2.0639	0.0872	0.9637	0.0511	0.0310	3.2	966	0.7245	1.0042
Astar[1,7]	-2.0041	0.0425	2.1289	0.0690	1.0242	0.0310	0.0328	3.2	974	0.7172	1.0007
Astar[2,7]	-2.2943	-0.0528	2.1537	-0.0762	1.0872	-0.0699	0.0359	3.3	918	0.7344	1.0041
Astar[3,7]	-5.5793	-2.8345	-0.1803	-2.8562	1.3473	-2.8561	0.0461	3.4	852	0.7481	1.0036
Astar[4,7]	-0.9895	1.2762	3.4631	1.2692	1.1061	1.2321	0.0333	3	1105	0.6927	1.0009
Astar[1,8]	-2.3299	-0.1836	2.0188	-0.1861	1.0848	-0.1492	0.0351	3.2	954	0.7327	1.0067
Astar[2,8]	-2.1208	-0.0352	2.0615	-0.0468	1.0442	-0.0631	0.0326	3.1	1028	0.7120	1.0051
Astar[3,8]	-2.8624	-0.3030	2.0482	-0.3475	1.2375	-0.2252	0.0381	3.1	1053	0.7051	1.0016
Astar[4,8]	-2.6126	-0.2444	1.9560	-0.2609	1.1378	-0.2261	0.0373	3.3	930	0.7219	1.0036
Astar[1,9]	-1.9967	0.0061	1.9216	0.0129	0.9649	-0.0122	0.0301	3.1	1029	0.6981	1.0093
Astar[2,9]	-1.8560	0.0019	1.8517	-0.0349	0.9318	0.0797	0.0284	3	1078	0.7026	1.0040
Astar[3,9]	-1.0004	0.8987	2.8690	0.9236	0.9561	0.9925	0.0291	3	1079	0.6997	1.0035
Astar[4,9]	-2.0995	-0.0076	1.9507	-0.0609	0.9873	0.0028	0.0309	3.1	1019	0.7085	1.0012
Astar[1,10]	-2.0390	0.0023	1.8852	-0.0055	0.9685	0.0384	0.0294	3	1087	0.6891	1.0074
Astar[2,10]	-2.0242	0.0004	1.8927	0.0013	0.9606	0.0027	0.0279	2.9	1188	0.6705	1.0010
Astar[3,10]	-2.3133	-0.2918	1.7491	-0.2862	0.9733	-0.3147	0.0296	3	1082	0.6972	1.0014
Astar[4,10]	-1.3265	0.5616	2.5431	0.5860	0.9729	0.5417	0.0283	2.9	1184	0.6761	1.0111
Astar[1,11]	-2.7209	0.0410	2.8082	0.0244	1.3615	0.0942	0.0422	3.1	1039	0.7103	1.0012
Astar[2,11]	-2.5757	-0.0027	2.7123	-0.0232	1.3281	0.0233	0.0413	3.1	1032	0.7088	1.0004
Astar[3,11]	-2.6273	0.0004	2.8204	0.0420	1.3578	-0.0575	0.0423	3.1	1030	0.7035	1.0010
Astar[4,11]	-2.7488	0.0800	2.8052	0.0488	1.3911	0.0982	0.0454	3.3	938	0.7304	1.0029
Astar[1,12]	-2.7560	0.0897	2.8003	0.0847	1.4145	0.1140	0.0459	3.2	950	0.7247	1.0032
Astar[2,12]	-3.1192	-0.0404	2.7991	-0.0576	1.4768	-0.0089	0.0484	3.3	930	0.7343	1.0023
Astar[3,12]	-6.3855	-2.8746	0.3123	-2.8746	1.6717	-2.8684	0.0577	3.5	839	0.7451	1.0029
Astar[4,12]	-1.6022	1.2885	4.1883	1.2723	1.4521	1.3558	0.0430	3	1138	0.6895	1.0004
Astar[1,13]	-2.9541	-0.1598	2.7115	-0.1680	1.4188	-0.1407	0.0445	3.1	1016	0.7115	1.0046
Astar[2,13]	-2.7418	-0.0038	2.7910	0.0105	1.3918	-0.0623	0.0422	3	1086	0.7045	1.0055
Astar[3,13]	-3.4644	-0.2816	2.6653	-0.3339	1.5605	-0.1931	0.0490	3.1	1015	0.7117	1.0014
Astar[4,13]	-3.2956	-0.2298	2.6559	-0.2789	1.5166	-0.2004	0.0479	3.2	1003	0.7142	1.0112
Astar[1,14]	-2.8264	-0.0562	2.5053	-0.0543	1.3374	-0.0261	0.0419	3.1	1017	0.6990	1.0056
Astar[2,14]	-2.8452	0.0066	2.6889	0.0016	1.3667	0.0167	0.0433	3.2	998	0.7149	1.0002
Astar[3,14]	-1.8937	0.8960	3.5132	0.8964	1.3661	0.9553	0.0430	3.1	1011	0.7119	1.0006
Astar[4,14]	-2.9252	-0.0650	2.7080	-0.0755	1.4032	-0.0596	0.0445	3.2	996	0.7153	1.0035
Astar[1,15]	-2.8502	0.0020	2.8151	0.0027	1.3871	-0.0409	0.0410	3	1143	0.6860	1.0033
Astar[2,15]	-2.6932	0.0109	2.6701	-0.0001	1.3360	0.0762	0.0402	3	1104	0.6902	1.0020
Astar[3,15]	-2.9936	-0.3003	2.4793	-0.2801	1.3741	-0.3523	0.0441	3.2	970	0.7184	1.0029
Astar[4,15]	-2.2404	0.5823	3.2011	0.5733	1.3519	0.5553	0.0418	3.1	1044	0.7057	1.0022

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC	SSeff	AC.20	psrf
Astar[1,16]	-3.2739	0.0679	3.3746	0.0228	1.6586	0.0690	0.0531	3.2	975	0.7199	1.0020
Astar[2,16]	-3.2183	-0.0153	3.1946	-0.0433	1.6199	0.0163	0.0502	3.1	1043	0.7061	1.0015
Astar[3,16]	-3.4330	-0.0243	3.3175	0.0204	1.6861	-0.0858	0.0521	3.1	1049	0.6990	1.0005
Astar[4,16]	-3.2808	0.0744	3.3127	0.0472	1.6541	0.1359	0.0510	3.1	1052	0.7093	1.0022
Astar[1,17]	-3.2946	0.0689	3.4288	0.0708	1.6931	0.0318	0.0528	3.1	1029	0.7083	1.0034
Astar[2,17]	-3.4950	-0.0402	3.4363	-0.0486	1.7567	-0.0338	0.0566	3.2	963	0.7228	1.0037
Astar[3,17]	-6.5959	-2.8769	0.9141	-2.8782	1.8864	-2.9609	0.0633	3.4	889	0.7407	1.0007
Astar[4,17]	-2.2051	1.2700	4.6299	1.2524	1.7358	1.3184	0.0526	3	1091	0.6960	1.0009
Astar[1,18]	-3.6295	-0.1275	3.0889	-0.1368	1.7020	-0.0351	0.0524	3.1	1054	0.7101	1.0035
Astar[2,18]	-3.3903	0.0041	3.4022	0.0122	1.7280	0.0358	0.0541	3.1	1021	0.7150	1.0033
Astar[3,18]	-3.8777	-0.2968	3.3463	-0.3046	1.8230	-0.3143	0.0559	3.1	1063	0.7001	1.0009
Astar[4,18]	-3.9387	-0.2935	3.1626	-0.3296	1.7904	-0.3319	0.0555	3.1	1040	0.7125	1.0059
Astar[1,19]	-3.2296	-0.0574	3.3572	-0.0358	1.6493	-0.0801	0.0512	3.1	1039	0.7101	1.0029
Astar[2,19]	-3.3155	0.0083	3.2628	-0.0197	1.6498	0.0839	0.0496	3	1105	0.6959	1.0004
Astar[3,19]	-2.3951	0.9107	4.1741	0.9188	1.6633	0.9179	0.0519	3.1	1026	0.7129	1.0002
Astar[4,19]	-3.3154	-0.0660	3.4754	-0.0507	1.7115	-0.0404	0.0553	3.2	957	0.7222	1.0008
Astar[1,20]	-3.3352	0.0023	3.5262	0.0023	1.7251	-0.0059	0.0543	3.1	1011	0.7118	1.0007
Astar[2,20]	-3.1234	0.0631	3.3614	0.0648	1.6325	-0.0076	0.0486	3	1130	0.6930	1.0021
Astar[3,20]	-3.6519	-0.3631	2.9979	-0.3507	1.6830	-0.3191	0.0531	3.2	1004	0.7102	1.0005
Astar[4,20]	-2.7448	0.5943	3.8795	0.5883	1.6635	0.6444	0.0529	3.2	988	0.7117	1.0011
Astar[1,21]	-3.6976	0.0301	3.9039	0.0213	1.9037	-0.0325	0.0608	3.2	981	0.7201	1.0021
Astar[2,21]	-3.6353	-0.0881	3.6473	-0.0949	1.8546	-0.0641	0.0566	3.1	1075	0.7009	1.0016
Astar[3,21]	-3.8658	-0.0245	3.9640	0.0064	1.9664	-0.1072	0.0594	3	1095	0.7013	1.0010
Astar[4,21]	-3.8569	0.0364	3.7534	0.0111	1.9085	0.0867	0.0584	3.1	1068	0.7032	1.0029
Astar[1,22]	-3.6697	0.0493	3.8864	0.0615	1.9341	0.0356	0.0583	3	1101	0.6935	1.0064
Astar[2,22]	-4.0814	-0.0237	3.9151	-0.0649	2.0244	0.0451	0.0657	3.2	950	0.7261	1.0025
Astar[3,22]	-7.1114	-2.8999	1.3533	-2.9149	2.1384	-2.8471	0.0726	3.4	867	0.7385	1.0018
Astar[4,22]	-2.8112	1.2747	5.0646	1.2577	1.9938	1.3080	0.0623	3.1	1026	0.7049	1.0005
Astar[1,23]	-4.0552	-0.0802	3.7171	-0.1036	1.9702	-0.0559	0.0613	3.1	1033	0.7146	1.0018
Astar[2,23]	-3.8008	0.0058	3.9361	0.0109	1.9788	0.0178	0.0617	3.1	1028	0.7169	1.0010
Astar[3,23]	-4.2437	-0.2839	3.7321	-0.2829	2.0321	-0.3512	0.0625	3.1	1057	0.7008	1.0022
Astar[4,23]	-4.3853	-0.3130	3.5851	-0.3226	2.0329	-0.2625	0.0621	3.1	1073	0.7024	1.0078
Astar[1,24]	-3.8422	-0.0365	3.7235	-0.0191	1.8984	-0.0864	0.0599	3.2	1005	0.7085	1.0024
Astar[2,24]	-3.6937	-0.0431	3.8163	-0.0404	1.9066	-0.0594	0.0585	3.1	1063	0.6973	1.0010
Astar[3,24]	-2.7438	0.9247	4.6768	0.9215	1.8870	0.8834	0.0583	3.1	1048	0.7040	1.0011
Astar[4,24]	-3.9618	-0.0817	3.8282	-0.0687	1.9440	-0.1213	0.0618	3.2	988	0.7189	1.0025
Astar[1,25]	-3.8890	0.0053	4.0717	-0.0026	2.0012	-0.0377	0.0628	3.1	1016	0.7105	1.0003
Astar[2,25]	-3.7295	0.0685	3.6977	0.0634	1.8894	0.1454	0.0561	3	1135	0.6901	1.0028
Astar[3,25]	-4.3323	-0.4180	3.4464	-0.4091	1.9864	-0.4270	0.0636	3.2	976	0.7169	1.0013
Astar[4,25]	-3.2733	0.6311	4.4575	0.6206	1.9297	0.7134	0.0609	3.2	1003	0.7144	1.0017
Astar[1,26]	-4.0271	0.0329	4.3669	0.0390	2.1240	-0.0006	0.0669	3.2	1007	0.7146	1.0011
Astar[2,26]	-4.1853	-0.1134	3.9915	-0.1043	2.0846	-0.2179	0.0642	3.1	1055	0.6986	1.0004
Astar[3,26]	-4.3852	-0.0298	4.2403	-0.0037	2.1591	-0.1241	0.0658	3	1077	0.6969	1.0020
Astar[4,26]	-4.2580	-0.0104	4.0773	-0.0177	2.1106	-0.0320	0.0636	3	1101	0.6944	1.0026
Astar[1,27]	-4.1735	0.0258	4.2969	0.0134	2.1705	0.0477	0.0660	3	1080	0.6981	1.0060
Astar[2,27]	-4.5424	-0.0468	4.2288	-0.1047	2.2560	0.0157	0.0733	3.2	947	0.7279	1.0011
Astar[3,27]	-7.4645	-2.8595	1.8671	-2.8907	2.3297	-2.6871	0.0745	3.2	979	0.7281	1.0060
Astar[4,27]	-3.2230	1.2192	5.5000	1.2185	2.2214	1.2249	0.0685	3.1	1050	0.7087	1.0012
Astar[1,28]	-4.3865	-0.0925	4.3143	-0.0915	2.2141	-0.0448	0.0701	3.2	996	0.7184	1.0010
Astar[2,28]	-4.2850	0.0498	4.4940	0.0534	2.2181	0.0642	0.0701	3.2	1001	0.7197	1.0011
Astar[3,28]	-4.8227	-0.2670	4.0434	-0.2722	2.2517	-0.2336	0.0697	3.1	1044	0.7015	1.0033
Astar[4,28]	-4.8629	-0.3328	4.0108	-0.3485	2.2522	-0.4284	0.0688	3.1	1070	0.7033	1.0040
Astar[1,29]	-4.1328	0.0116	4.3646	0.0235	2.1534	0.0781	0.0674	3.1	1020	0.7102	1.0056
Astar[2,29]	-4.1698	-0.0859	4.3655	-0.0710	2.1598	-0.0190	0.0660	3.1	1070	0.6981	1.0010
Astar[3,29]	-3.3270	0.9466	5.1262	0.9362	2.1371	0.8582	0.0657	3.1	1059	0.7081	1.0022
Astar[4,29]	-4.3583	-0.0815	4.2692	-0.0709	2.1714	-0.1297	0.0685	3.2	1005	0.7152	1.0045
Astar[1,30]	-4.5099	0.0373	4.2491	0.0287	2.2026	0.1005	0.0697	3.2	999	0.7105	1.0002
Astar[2,30]	-4.0983	0.0746	4.2153	0.0749	2.1224	0.0417	0.0638	3	1105	0.6968	1.0032
Astar[3,30]	-4.7578	-0.4239	3.8212	-0.4156	2.1869	-0.4593	0.0696	3.2	988	0.7139	1.0010
Astar[4,30]	-3.7747	0.6084	4.8972	0.6002	2.1586	0.7155	0.0667	3.1	1047	0.7049	1.0025

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC	SSeff	AC.20	psrf
Sigma[1,1]	0.0414	0.0540	0.0690	0.0546	0.0071	0.0528	0.0000	0.5	45032	0.0083	1.0000
Sigma[2,1]	-0.0075	0.0022	0.0123	0.0022	0.0050	0.0020	0.0000	0.4	49465	0.0091	1.0001
Sigma[3,1]	-0.0165	0.0035	0.0238	0.0036	0.0102	0.0036	0.0001	0.6	27693	0.0238	1.0001
Sigma[4,1]	-0.0126	0.0016	0.0155	0.0016	0.0071	0.0018	0.0000	0.6	26264	0.0170	1.0000
Sigma[1,2]	-0.0075	0.0022	0.0123	0.0022	0.0050	0.0020	0.0000	0.4	49465	0.0091	1.0001
Sigma[2,2]	0.0403	0.0531	0.0678	0.0537	0.0071	0.0523	0.0000	0.5	35642	0.0178	1.0003
Sigma[3,2]	-0.0167	0.0032	0.0235	0.0033	0.0102	0.0033	0.0001	0.6	25312	0.0261	1.0000
Sigma[4,2]	-0.0127	0.0009	0.0149	0.0009	0.0070	0.0010	0.0000	0.6	28830	0.0133	1.0001
Sigma[1,3]	-0.0165	0.0035	0.0238	0.0036	0.0102	0.0036	0.0001	0.6	27693	0.0238	1.0001
Sigma[2,3]	-0.0167	0.0032	0.0235	0.0033	0.0102	0.0033	0.0001	0.6	25312	0.0261	1.0000
Sigma[3,3]	0.1664	0.2188	0.2813	0.2215	0.0298	0.2135	0.0002	0.8	17719	0.0423	1.0003
Sigma[4,3]	-0.0456	-0.0159	0.0121	-0.0163	0.0147	-0.0145	0.0001	0.8	16502	0.0364	1.0001
Sigma[1,4]	-0.0126	0.0016	0.0155	0.0016	0.0071	0.0018	0.0000	0.6	26264	0.0170	1.0000
Sigma[2,4]	-0.0127	0.0009	0.0149	0.0009	0.0070	0.0010	0.0000	0.6	28830	0.0133	1.0001
Sigma[3,4]	-0.0456	-0.0159	0.0121	-0.0163	0.0147	-0.0145	0.0001	0.8	16502	0.0364	1.0001
Sigma[4,4]	0.0802	0.1052	0.1348	0.1064	0.0142	0.1032	0.0001	0.7	18620	0.0271	1.0001
s[1]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[2]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[3]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[4]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[5]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[6]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[7]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[8]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[9]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[10]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[11]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[12]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[13]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[14]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[15]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[16]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[17]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[18]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[19]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[20]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[21]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[22]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[23]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[24]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[25]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[26]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[27]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[28]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[29]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[30]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[31]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[32]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[33]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[34]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[35]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[36]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[37]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[38]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[39]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[40]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[41]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[42]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[43]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[44]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[45]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[46]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[47]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[48]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[49]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[50]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC	SSeff	AC.20	psrf
s[51]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[52]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[53]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[54]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[55]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[56]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[57]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[58]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[59]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[60]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[61]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[62]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[63]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[64]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[65]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[66]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[67]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[68]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[69]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[70]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[71]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[72]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[73]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[74]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[75]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[76]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[77]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[78]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[79]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[80]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[81]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[82]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[83]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[84]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[85]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[86]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[87]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[88]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[89]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[90]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[91]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[92]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[93]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[94]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[95]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[96]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[97]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[98]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[99]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[100]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[101]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[102]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[103]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[104]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[105]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[106]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[107]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[108]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[109]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[110]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[111]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[112]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[113]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[114]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[115]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[116]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[117]	1	1	1	1	0	1	N/A	N/A	N/A	N/A	N/A
s[118]	1	1	1	1.0000	0.0058	1	5.9E-05	1	9587	N/A	1.2910
s[119]	1	1	1	1.0001	0.0115	1	5.2E-05	0.4	49587	-0.0001	1.0298
s[120]	1	1	1	1.0427	0.2021	1	8.2E-04	0.4	60440	-0.0025	1.0000

C.2 Experiment 2: Energy Markets

Table C.2: Experiment 2. JAGS summary results

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC	SSeff	AC.20	psrf
Astar[1,1]	-0.0611	-0.0044	0.0332	-0.0075	0.0222	-0.0023	0.0008	3.5	807	0.7503	1.0007
Astar[2,1]	-0.0879	-0.0160	0.0231	-0.0236	0.0298	-0.0065	0.0012	4.1	587	0.8060	1.0076
Astar[3,1]	-0.0436	0.0013	0.0440	0.0016	0.0199	0.0004	0.0006	3.2	982	0.7028	1.0035
Astar[4,1]	0.0532	**0.1268**	0.2002	0.1269	0.0375	0.1261	0.0012	3.2	955	0.6962	1.0044
Astar[1,2]	-0.3614	0.0094	0.4384	0.0288	0.1960	0.0267	0.0063	3.2	980	0.7281	1.0072
Astar[2,2]	-0.2974	0.1149	0.5631	0.1296	0.2204	0.0912	0.0074	3.4	877	0.7352	1.0094
Astar[3,2]	-0.4553	-0.0293	0.3355	-0.0540	0.1959	-0.0482	0.0066	3.4	881	0.7393	1.0015
Astar[4,2]	-0.0075	0.5698	1.0280	0.5676	0.2599	0.5838	0.0080	3.1	1059	0.6817	1.0033
Astar[1,3]	-0.4024	-0.0369	0.2515	-0.0638	0.1615	-0.0466	0.0042	2.6	1454	0.6200	1.0012
Astar[2,3]	-0.4678	-0.0816	0.2289	-0.0992	0.1760	-0.0655	0.0047	2.7	1407	0.6277	1.0013
Astar[3,3]	-0.3740	-0.0257	0.2611	-0.0528	0.1561	-0.0392	0.0045	2.9	1207	0.6720	1.0032
Astar[4,3]	-0.2344	0.1451	0.5711	0.1567	0.2056	0.0955	0.0054	2.6	1469	0.6236	1.0031
Astar[1,4]	-0.0531	0.5540	1.1394	0.5505	0.3123	0.5623	0.0089	2.8	1234	0.6556	1.0009
Astar[2,4]	-0.9774	-0.3152	0.2875	-0.3180	0.3267	-0.2088	0.0101	3.1	1052	0.6978	1.0055
Astar[3,4]	-0.5797	-0.0068	0.5379	-0.0223	0.2772	-0.0300	0.0086	3.1	1041	0.7065	1.0004
Astar[4,4]	-0.0426	0.6632	1.4164	0.6630	0.3761	0.6554	0.0113	3	1098	0.6844	1.0022
Astar[1,5]	-0.1436	0.0003	0.1443	0.0014	0.0670	-0.0006	0.0024	3.6	789	0.7684	1.0078
Astar[2,5]	-0.0587	0.0777	0.2721	0.0854	0.0892	0.0369	0.0035	4	635	0.8013	1.0034
Astar[3,5]	-0.1333	0.0013	0.1423	0.0032	0.0645	0.0007	0.0023	3.6	768	0.7691	1.0052
Astar[4,5]	0.2103	**0.4186**	0.6195	0.4182	0.1040	0.4199	0.0034	3.3	937	0.7240	1.0032
Astar[1,6]	-1.9115	0.0097	1.9837	0.0415	0.9734	0.0258	0.0197	2	2443	0.4451	1.0015
Astar[2,6]	-1.9762	-0.0475	1.9200	-0.0636	0.9699	-0.1006	0.0208	2.1	2169	0.4854	1.0012
Astar[3,6]	-1.7916	0.0124	1.9821	0.0466	0.9497	-0.0458	0.0200	2.1	2262	0.4737	1.0030
Astar[4,6]	-1.7572	0.1455	2.1734	0.1629	0.9624	0.1424	0.0214	2.2	2020	0.5219	1.0023
Astar[1,7]	-1.8987	0.0593	2.0922	0.0534	0.9916	0.0762	0.0211	2.1	2203	0.4939	1.0019
Astar[2,7]	-1.8104	0.1484	2.1871	0.1460	0.9983	0.1285	0.0224	2.2	1985	0.5168	1.0031
Astar[3,7]	-2.0417	-0.0544	1.9188	-0.0670	0.9994	-0.0205	0.0214	2.1	2190	0.4861	1.0005
Astar[4,7]	-1.4377	0.5439	2.5677	0.5372	0.9997	0.5358	0.0213	2.1	2204	0.4825	1.0005
Astar[1,8]	-1.9600	-0.0534	1.9098	-0.0621	0.9595	-0.0685	0.0212	2.2	2046	0.5092	1.0016
Astar[2,8]	-2.0300	-0.0756	1.9924	-0.0592	1.0156	-0.0503	0.0221	2.2	2105	0.4947	1.0055
Astar[3,8]	-2.1594	-0.0543	1.8251	-0.0665	0.9993	-0.0072	0.0216	2.2	2136	0.4935	1.0015
Astar[4,8]	-1.7836	0.1820	2.1241	0.1741	0.9752	0.1797	0.0209	2.1	2169	0.4919	1.0012
Astar[1,9]	-1.4144	0.5783	2.7388	0.5835	1.0417	0.5431	0.0237	2.3	1939	0.5180	1.0005
Astar[2,9]	-2.3899	-0.3110	1.7801	-0.3104	1.0450	-0.2547	0.0240	2.3	1896	0.5304	1.0003
Astar[3,9]	-1.9644	-0.0164	2.0280	-0.0053	0.9963	-0.0289	0.0222	2.2	2009	0.5206	1.0035
Astar[4,9]	-1.4664	0.6530	2.6983	0.6498	1.0572	0.6853	0.0245	2.3	1859	0.5356	1.0019
Astar[1,10]	-1.9883	0.0040	1.9022	0.0093	0.9788	0.0762	0.0212	2.2	2122	0.4925	1.0012
Astar[2,10]	-1.9138	0.1103	2.1951	0.1146	1.0125	0.1160	0.0215	2.1	2210	0.4811	1.0003
Astar[3,10]	-2.0097	-0.0094	1.8893	-0.0385	0.9698	-0.0050	0.0213	2.2	2066	0.5015	1.0007
Astar[4,10]	-1.5177	0.4205	2.3344	0.4147	0.9657	0.3826	0.0205	2.1	2216	0.4897	1.0027
Astar[1,11]	-2.6936	0.0275	2.8013	0.0406	1.3742	0.0808	0.0289	2.1	2260	0.4743	1.0014
Astar[2,11]	-2.9020	-0.0467	2.7320	-0.0524	1.4109	-0.0119	0.0309	2.2	2091	0.5067	1.0014
Astar[3,11]	-2.5951	0.0155	2.6630	0.0403	1.3363	0.0065	0.0270	2	2444	0.4501	1.0017
Astar[4,11]	-2.5053	0.1656	2.7998	0.1747	1.3345	0.2050	0.0284	2.1	2210	0.4923	1.0013
Astar[1,12]	-2.6678	0.0852	2.9320	0.1015	1.4121	0.1034	0.0324	2.3	1900	0.5347	1.0013
Astar[2,12]	-2.5295	0.1122	2.9803	0.1246	1.3906	0.1358	0.0301	2.2	2140	0.4941	1.0041
Astar[3,12]	-2.7979	-0.1065	2.7623	-0.0941	1.4139	-0.1379	0.0310	2.2	2086	0.4986	1.0004
Astar[4,12]	-2.1738	0.5321	3.4016	0.5215	1.4106	0.4911	0.0307	2.2	2110	0.4953	1.0004
Astar[1,13]	-2.7991	-0.0569	2.6702	-0.0702	1.3745	-0.0276	0.0301	2.2	2085	0.4941	1.0002
Astar[2,13]	-2.8197	-0.0740	2.6682	-0.0710	1.3873	-0.0740	0.0293	2.1	2234	0.4786	1.0029
Astar[3,13]	-2.8357	-0.0587	2.7096	-0.0725	1.4085	-0.0622	0.0300	2.1	2210	0.4918	1.0005
Astar[4,13]	-2.6122	0.1455	2.7844	0.1423	1.3609	0.0749	0.0289	2.1	2219	0.4820	1.0008
Astar[1,14]	-2.2901	0.5474	3.4099	0.5601	1.4346	0.5736	0.0311	2.2	2133	0.4977	1.0007
Astar[2,14]	-3.0300	-0.2930	2.5700	-0.2721	1.4280	-0.3744	0.0315	2.2	2058	0.5042	1.0008
Astar[3,14]	-2.9388	-0.0732	2.7100	-0.0578	1.4272	-0.0632	0.0325	2.3	1925	0.5328	1.0014
Astar[4,14]	-2.2274	0.6392	3.4684	0.6326	1.4348	0.6470	0.0325	2.3	1950	0.5242	1.0019
Astar[1,15]	-2.6814	0.0134	2.8738	0.0201	1.3950	0.0080	0.0308	2.2	2051	0.5034	1.0021
Astar[2,15]	-2.6385	0.1561	2.9797	0.1523	1.4208	0.0963	0.0303	2.1	2198	0.4690	1.0002
Astar[3,15]	-2.7179	-0.0480	2.7086	-0.0455	1.3722	-0.0262	0.0292	2.1	2205	0.4809	1.0014
Astar[4,15]	-2.3607	0.3931	3.0812	0.3813	1.4027	0.4530	0.0313	2.2	2014	0.5124	1.0031

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC	SSeff	AC.20	psrf
Astar[1,16]	-3.3029	0.0099	3.3509	0.0315	1.6757	-0.0771	0.0349	2.1	2306	0.4688	1.0004
Astar[2,16]	-3.7053	-0.0627	3.2627	-0.0665	1.7360	-0.0845	0.0390	2.2	1979	0.5152	1.0010
Astar[3,16]	-3.1834	0.0322	3.3275	0.0316	1.6522	0.0437	0.0345	2.1	2294	0.4737	1.0017
Astar[4,16]	-3.1705	0.1465	3.3364	0.1605	1.6543	0.0700	0.0363	2.2	2074	0.5033	1.0007
Astar[1,17]	-3.3670	0.1159	3.4830	0.1248	1.7389	0.0988	0.0390	2.2	1986	0.5231	1.0011
Astar[2,17]	-3.4029	0.1320	3.4234	0.1373	1.7099	0.1033	0.0367	2.1	2166	0.4985	1.0050
Astar[3,17]	-3.4185	-0.0763	3.4236	-0.0679	1.7289	-0.0635	0.0378	2.2	2097	0.5032	1.0002
Astar[4,17]	-2.7996	0.5352	3.9289	0.5393	1.7040	0.5435	0.0363	2.1	2206	0.4792	1.0003
Astar[1,18]	-3.5020	-0.0670	3.2856	-0.0733	1.7135	-0.0787	0.0380	2.2	2035	0.5100	1.0003
Astar[2,18]	-3.3858	-0.0565	3.2916	-0.0515	1.6947	-0.0607	0.0355	2.1	2273	0.4768	1.0012
Astar[3,18]	-3.4245	-0.0813	3.4054	-0.0933	1.7248	-0.0691	0.0378	2.2	2084	0.5098	1.0005
Astar[4,18]	-3.0580	0.1608	3.6424	0.1266	1.6784	0.1595	0.0359	2.1	2187	0.4883	1.0009
Astar[1,19]	-2.9562	0.4987	3.8724	0.5075	1.7177	0.3813	0.0364	2.1	2229	0.4883	1.0007
Astar[2,19]	-3.6496	-0.3151	3.2398	-0.2941	1.7508	-0.2816	0.0389	2.2	2023	0.5096	1.0012
Astar[3,19]	-3.4508	-0.0278	3.2491	-0.0299	1.7031	-0.0761	0.0384	2.3	1969	0.5216	1.0009
Astar[4,19]	-2.8461	0.6608	4.0763	0.6475	1.7361	0.6929	0.0390	2.2	1978	0.5200	1.0018
Astar[1,20]	-3.3451	0.0225	3.3295	0.0213	1.6909	0.0336	0.0362	2.1	2185	0.4917	1.0020
Astar[2,20]	-3.3445	0.1264	3.5555	0.1267	1.7297	0.1259	0.0376	2.2	2115	0.4863	1.0011
Astar[3,20]	-3.3277	-0.0027	3.2602	-0.0156	1.6647	0.0452	0.0354	2.1	2208	0.4858	1.0020
Astar[4,20]	-2.8048	0.3869	3.8545	0.3971	1.6999	0.4371	0.0374	2.2	2069	0.4968	1.0054
Astar[1,21]	-3.7525	-0.0136	3.8956	0.0310	1.9213	0.0253	0.0400	2.1	2307	0.4696	1.0017
Astar[2,21]	-4.0224	-0.0656	3.8636	-0.0745	1.9827	-0.0163	0.0436	2.2	2071	0.4986	1.0018
Astar[3,21]	-3.7685	0.0250	3.8665	0.0226	1.9320	0.0453	0.0400	2.1	2330	0.4686	1.0007
Astar[4,21]	-3.8001	0.1466	3.8386	0.1412	1.9344	0.0937	0.0431	2.2	2011	0.5114	1.0009
Astar[1,22]	-3.9086	0.0991	3.9297	0.0987	1.9920	-0.0109	0.0441	2.2	2045	0.5162	1.0009
Astar[2,22]	-3.7749	0.1707	4.0692	0.1777	1.9822	0.1240	0.0435	2.2	2076	0.5045	1.0048
Astar[3,22]	-3.9170	-0.0771	3.9384	-0.0688	1.9886	-0.0808	0.0434	2.2	2099	0.5043	1.0010
Astar[4,22]	-3.2602	0.5429	4.5297	0.5459	1.9668	0.5818	0.0422	2.1	2173	0.4864	1.0005
Astar[1,23]	-3.9352	-0.0516	3.9152	-0.0569	1.9843	-0.0421	0.0440	2.2	2035	0.5089	1.0009
Astar[2,23]	-3.8820	-0.0611	3.9005	-0.0357	1.9724	-0.0705	0.0421	2.1	2199	0.4896	1.0015
Astar[3,23]	-4.0160	-0.0737	3.9203	-0.0812	2.0137	-0.0157	0.0448	2.2	2018	0.5154	1.0005
Astar[4,23]	-3.8216	0.2219	4.0661	0.1899	1.9743	0.3078	0.0439	2.2	2026	0.5043	1.0002
Astar[1,24]	-3.4520	0.4962	4.4267	0.5185	1.9901	0.4071	0.0436	2.2	2087	0.4959	1.0005
Astar[2,24]	-4.1750	-0.3357	3.6861	-0.3273	1.9952	-0.4462	0.0445	2.2	2013	0.5056	1.0002
Astar[3,24]	-3.9082	-0.0118	3.7295	-0.0180	1.9407	-0.0329	0.0417	2.2	2161	0.4981	1.0011
Astar[4,24]	-3.3785	0.6353	4.6279	0.6539	2.0162	0.6541	0.0461	2.3	1916	0.5227	1.0015
Astar[1,25]	-3.8083	0.0352	3.8859	0.0400	1.9648	0.0694	0.0424	2.2	2147	0.5002	1.0019
Astar[2,25]	-3.9956	0.1109	3.9390	0.1051	2.0090	0.1766	0.0439	2.2	2098	0.4933	1.0019
Astar[3,25]	-3.8472	0.0270	3.7023	0.0188	1.9212	0.1197	0.0408	2.1	2216	0.4865	1.0005
Astar[4,25]	-3.3831	0.3824	4.2948	0.4098	1.9627	0.3268	0.0428	2.2	2105	0.4968	1.0028
Astar[1,26]	-4.2541	0.0065	4.3544	0.0453	2.1667	0.0006	0.0454	2.1	2279	0.4705	1.0018
Astar[2,26]	-4.5479	-0.1025	4.1626	-0.0956	2.1979	-0.1412	0.0476	2.2	2133	0.4964	1.0026
Astar[3,26]	-4.2807	0.0311	4.1640	0.0250	2.1428	0.0383	0.0443	2.1	2339	0.4695	1.0005
Astar[4,26]	-4.0398	0.1486	4.3895	0.1449	2.1440	0.1631	0.0479	2.2	2001	0.5017	1.0007
Astar[1,27]	-4.2769	0.1375	4.4919	0.1407	2.2321	0.1136	0.0492	2.2	2059	0.5122	1.0020
Astar[2,27]	-4.1665	0.1422	4.5843	0.1493	2.2138	0.1829	0.0480	2.2	2128	0.4985	1.0042
Astar[3,27]	-4.3718	-0.0599	4.3937	-0.0398	2.2272	-0.1148	0.0487	2.2	2092	0.4981	1.0008
Astar[4,27]	-3.7202	0.5661	4.9656	0.5657	2.2012	0.5554	0.0471	2.1	2187	0.4853	1.0007
Astar[1,28]	-4.2897	-0.0609	4.4002	-0.0517	2.2068	-0.1084	0.0486	2.2	2058	0.5040	1.0004
Astar[2,28]	-4.5158	-0.0439	4.2440	-0.0335	2.2096	-0.0388	0.0475	2.2	2163	0.4931	1.0018
Astar[3,28]	-4.4039	-0.0343	4.4097	-0.0506	2.2407	0.0214	0.0499	2.2	2013	0.5113	1.0001
Astar[4,28]	-4.1739	0.2202	4.4956	0.1818	2.1875	0.2216	0.0473	2.2	2140	0.5011	1.0002
Astar[1,29]	-3.8872	0.5192	4.7735	0.5193	2.1982	0.4684	0.0468	2.1	2209	0.4872	1.0002
Astar[2,29]	-4.7534	-0.3646	4.0296	-0.3397	2.2181	-0.3790	0.0492	2.2	2030	0.5060	1.0003
Astar[3,29]	-4.2582	-0.0163	4.3954	-0.0035	2.1839	0.0541	0.0473	2.2	2135	0.4929	1.0010
Astar[4,29]	-3.7142	0.6789	5.1703	0.6763	2.2683	0.6389	0.0518	2.3	1917	0.5295	1.0017
Astar[1,30]	-4.2090	0.0388	4.3160	0.0563	2.1801	0.0021	0.0476	2.2	2099	0.5079	1.0021
Astar[2,30]	-4.2605	0.0882	4.4600	0.0894	2.2220	0.1461	0.0482	2.2	2124	0.4968	1.0018
Astar[3,30]	-4.1960	-0.0115	4.3103	-0.0091	2.1551	0.0790	0.0467	2.2	2132	0.4881	1.0004
Astar[4,30]	-4.0289	0.3589	4.6598	0.3902	2.2100	0.2789	0.0485	2.2	2078	0.5002	1.0034

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC	SSeff	AC.20	psrf
Sigma[1,1]	0.0505	0.0660	0.0844	0.0667	0.0088	0.0647	0.0000	0.5	34950	0.0196	1.0001
Sigma[2,1]	-0.0103	0.0024	0.0150	0.0024	0.0064	0.0022	0.0000	0.5	38227	0.0135	1.0002
Sigma[3,1]	-0.0045	0.0069	0.0187	0.0070	0.0059	0.0065	0.0000	0.5	43477	0.0180	1.0000
Sigma[4,1]	-0.0056	0.0089	0.0244	0.0090	0.0076	0.0088	0.0000	0.5	36619	0.0088	1.0001
Sigma[1,2]	-0.0103	0.0024	0.0150	0.0024	0.0064	0.0022	0.0000	0.5	38227	0.0135	1.0002
Sigma[2,2]	0.0529	0.0692	0.0888	0.0701	0.0093	0.0680	0.0000	0.5	40445	0.0054	1.0000
Sigma[3,2]	-0.0086	0.0032	0.0152	0.0033	0.0060	0.0029	0.0000	0.5	33901	0.0189	1.0001
Sigma[4,2]	-0.0163	-0.0008	0.0144	-0.0008	0.0078	-0.0009	0.0000	0.5	38005	0.0144	1.0000
Sigma[1,3]	-0.0045	0.0069	0.0187	0.0070	0.0059	0.0065	0.0000	0.5	43477	0.0180	1.0000
Sigma[2,3]	-0.0086	0.0032	0.0152	0.0033	0.0060	0.0029	0.0000	0.5	33901	0.0189	1.0001
Sigma[3,3]	0.0445	0.0584	0.0746	0.0591	0.0078	0.0577	0.0000	0.5	37572	0.0150	1.0000
Sigma[4,3]	-0.0088	0.0051	0.0194	0.0051	0.0072	0.0051	0.0000	0.5	38483	0.0123	1.0000
Sigma[1,4]	-0.0056	0.0089	0.0244	0.0090	0.0076	0.0088	0.0000	0.5	36619	0.0088	1.0001
Sigma[2,4]	-0.0163	-0.0008	0.0144	-0.0008	0.0078	-0.0009	0.0000	0.5	38005	0.0144	1.0000
Sigma[3,4]	-0.0088	0.0051	0.0194	0.0051	0.0072	0.0051	0.0000	0.5	38483	0.0123	1.0000
Sigma[4,4]	0.0748	0.0979	0.1255	0.0990	0.0132	0.0961	0.0001	0.5	35669	0.0130	1.0000
s[1]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[2]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[3]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[4]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[5]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[6]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[7]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[8]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[9]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[10]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[11]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[12]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[13]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[14]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[15]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[16]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[17]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[18]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[19]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[20]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[21]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[22]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[23]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[24]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[25]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[26]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[27]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[28]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[29]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[30]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[31]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[32]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[33]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[34]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[35]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[36]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[37]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[38]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[39]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[40]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[41]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[42]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[43]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[44]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[45]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[46]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[47]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[48]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[49]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[50]	1	1	1	1	0	1	NA	NA	NA	NA	NA

Parameter	Lower95	Median	Upper95	Mean	SD	Mode	MCerr	MC	SSeff	AC.20	psrf
s[51]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[52]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[53]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[54]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[55]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[56]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[57]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[58]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[59]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[60]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[61]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[62]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[63]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[64]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[65]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[66]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[67]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[68]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[69]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[70]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[71]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[72]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[73]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[74]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[75]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[76]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[77]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[78]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[79]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[80]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[81]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[82]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[83]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[84]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[85]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[86]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[87]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[88]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[89]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[90]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[91]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[92]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[93]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[94]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[95]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[96]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[97]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[98]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[99]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[100]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[101]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[102]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[103]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[104]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[105]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[106]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[107]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[108]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[109]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[110]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[111]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[112]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[113]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[114]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[115]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[116]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[117]	1	1	1	1	0	1	NA	NA	NA	NA	NA
s[118]	1	1	1	1.0003	0.0158	1	9.1E-05	0.6	30476	NA	1.2219
s[119]	1	1	1	1.0015	0.0383	1	2.0E-04	0.5	35495	-0.0015	1.0020
s[120]	1	1	1	1.0433	0.2038	1	8.3E-04	0.4	59567	0.0014	1.0001

Appendix D

Appendix: VAR(1) OLS Estimates

D.1 Experiment 1: VAR(1) OLS Results from R

```
> head(data, 3)
                 SPXPVTR     SPXPGTR Unemp_rate CPI_gr
2013-01-31  0.0810890720  0.06127741  0.9793539    0.2
2013-03-03 -0.0006853358  0.01268487  0.8315616    0.5
2013-03-31  0.0532545131  0.04648943  0.7330334   -0.3

> tail(data, 3)
               SPXPVTR     SPXPGTR Unemp_rate CPI_gr
2022-10-31  0.12264259  0.07574522  -1.139002    0.4
2022-12-01  0.05193244  0.05238894  -1.188266    0.1
2022-12-31 -0.05607845 -0.07631098  -1.237530   -0.1

> VARselect(data, lag.max = 10, type = "const")[["selection"]]
AIC(n)  HQ(n)  SC(n) FPE(n)
     3      1      1      3

> VAR_model <- VAR(data, p=1, type = "const")

> summary(VAR_model)

VAR Estimation Results:
```

```
==========================
Endogenous variables: SPXPVTR, SPXPGTR, Unemp_rate, CPI_gr
Deterministic variables: const
Sample size: 119
Log Likelihood: 363.404
Roots of the characteristic polynomial:
0.8039 0.5466 0.2295 0.193
Call:
VAR(y = data, p = 1, type = "const")

Estimation results for equation SPXPVTR:
========================================
SPXPVTR = SPXPVTR.l1 + SPXPGTR.l1 + Unemp_rate.l1 + CPI_gr.l1 +
    const

                Estimate Std. Error t value Pr(>|t|)
SPXPVTR.l1      0.102696   0.134472   0.764  0.44662
SPXPGTR.l1     -0.367345   0.159560  -2.302  0.02314 *
Unemp_rate.l1   0.017402   0.006365   2.734  0.00725 **
CPI_gr.l1      -0.006532   0.018554  -0.352  0.72543
const           0.018946   0.006902   2.745  0.00703 **
---
Signif. codes: 0    ***    0.001    **    0.01    *    0.05    .
    0.1         1

Residual standard error: 0.05759 on 114 degrees of freedom
Multiple R-Squared: 0.09413,  Adjusted R-squared: 0.06235
F-statistic: 2.962 on 4 and 114 DF,  p-value: 0.02272

Estimation results for equation SPXPGTR:
========================================
SPXPGTR = SPXPVTR.l1 + SPXPGTR.l1 + Unemp_rate.l1 + CPI_gr.l1 +
    const

                Estimate Std. Error t value Pr(>|t|)
SPXPVTR.l1     -0.093160   0.113721  -0.819 0.414383
SPXPGTR.l1     -0.205076   0.134938  -1.520 0.131333
```

```
Unemp_rate.l1  0.019128   0.005383   3.554 0.000554 ***
CPI_gr.l1     -0.006265   0.015691  -0.399 0.690455
const          0.021015   0.005837   3.600 0.000472 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .
     0.1         1

Residual standard error: 0.0487 on 114 degrees of freedom
Multiple R-Squared: 0.1409, Adjusted R-squared: 0.1108
F-statistic: 4.675 on 4 and 114 DF,  p-value: 0.00157

Estimation results for equation Unemp_rate:
===========================================
Unemp_rate = SPXPVTR.l1 + SPXPGTR.l1 + Unemp_rate.l1 + CPI_gr.l1 +
    const

               Estimate Std. Error t value Pr(>|t|)
SPXPVTR.l1    -4.67678    0.95282  -4.908 3.09e-06 ***
SPXPGTR.l1     0.81969    1.13058   0.725   0.4699
Unemp_rate.l1  0.88569    0.04510  19.639  < 2e-16 ***
CPI_gr.l1     -0.27728    0.13146  -2.109   0.0371 *
const          0.03552    0.04891   0.726   0.4692
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .
     0.1         1

Residual standard error: 0.4081 on 114 degrees of freedom
Multiple R-Squared: 0.8058, Adjusted R-squared: 0.799
F-statistic: 118.3 on 4 and 114 DF,  p-value: < 2.2e-16

Estimation results for equation CPI_gr:
=======================================
CPI_gr = SPXPVTR.l1 + SPXPGTR.l1 + Unemp_rate.l1 + CPI_gr.l1 + const

               Estimate Std. Error t value Pr(>|t|)
SPXPVTR.l1     2.41265    0.53903   4.476 1.81e-05 ***
SPXPGTR.l1    -1.44837    0.63960  -2.265  0.02543 *
```

```
Unemp_rate.l1  0.01317    0.02551   0.516  0.60667
CPI_gr.l1      0.53066    0.07437   7.135 9.56e-11 ***
const          0.09653    0.02767   3.489  0.00069 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .
    0.1         1

Residual standard error: 0.2309 on 114 degrees of freedom
Multiple R-Squared: 0.4268, Adjusted R-squared: 0.4066
F-statistic: 21.22 on 4 and 114 DF,  p-value: 4.256e-13

Covariance matrix of residuals:
            SPXPVTR   SPXPGTR Unemp_rate     CPI_gr
SPXPVTR    0.003317 0.0020302   0.003521  0.0014884
SPXPGTR    0.002030 0.0023721   0.003168  0.0007035
Unemp_rate 0.003521 0.0031681   0.166521 -0.0135066
CPI_gr     0.001488 0.0007035  -0.013507  0.0532942

Correlation matrix of residuals:
           SPXPVTR SPXPGTR Unemp_rate   CPI_gr
SPXPVTR     1.0000 0.72379     0.1498  0.11195
SPXPGTR     0.7238 1.00000     0.1594  0.06257
Unemp_rate  0.1498 0.15941     1.0000 -0.14337
CPI_gr      0.1119 0.06257    -0.1434  1.00000
```

D.2 Experiment 2: VAR(1) OLS Results from R

```
> head(data, 3)
               SPGSCLTR     SPGSNGTR      IXERT CPI_gr
2013-01-31  0.05520384 -0.007381374 0.07949059    0.2
2013-03-03 -0.06306333  0.023498442 0.00869202    0.5
2013-03-31  0.05016049  0.132373705 0.02218825   -0.3

> tail(data, 3)
                SPGSCLTR    SPGSNGTR       IXERT CPI_gr
2022-10-31  0.102155523 -0.10766691  0.22290437    0.4
```

```
2022-12-01 -0.058105923  0.03319082  0.01370603    0.1
2022-12-31 -0.002073272 -0.40890745 -0.03075270   -0.1

> VARselect(data, lag.max = 10, type = "const")[["selection"]]
AIC(n)  HQ(n)  SC(n) FPE(n)
     1      1      1      1

> VAR_model <- VAR(data, p=1, type = "const")

> summary(VAR_model)

VAR Estimation Results:
=========================
Endogenous variables: SPGSCLTR, SPGSNGTR, IXERT, CPI_gr
Deterministic variables: const
Sample size: 119
Log Likelihood: 341.473
Roots of the characteristic polynomial:
0.4449 0.2525 0.2023 0.2023
Call:
VAR(y = data, p = 1, type = "const")

Estimation results for equation SPGSCLTR:
=========================================
SPGSCLTR = SPGSCLTR.l1 + SPGSNGTR.l1 + IXERT.l1 + CPI_gr.l1 + const

             Estimate Std. Error t value Pr(>|t|)
SPGSCLTR.l1 -0.03686    0.11406  -0.323    0.747
SPGSNGTR.l1 -0.09822    0.08666  -1.133    0.259
IXERT.l1     0.68864    0.16845   4.088 8.13e-05 ***
CPI_gr.l1    0.02053    0.04230   0.485    0.628
const       -0.01918    0.01510  -1.271    0.206
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .
     0.1         1

Residual standard error: 0.1238 on 114 degrees of freedom
```

```
Multiple R-Squared: 0.1727, Adjusted R-squared: 0.1437
F-statistic: 5.949 on 4 and 114 DF,  p-value: 0.0002199

Estimation results for equation SPGSNGTR:
=========================================
SPGSNGTR = SPGSCLTR.l1 + SPGSNGTR.l1 + IXERT.l1 + CPI_gr.l1 + const

             Estimate Std. Error t value Pr(>|t|)
SPGSCLTR.l1  0.14053    0.12429   1.131   0.2606
SPGSNGTR.l1 -0.11099    0.09443  -1.175   0.2423
IXERT.l1    -0.38800    0.18357  -2.114   0.0367 *
CPI_gr.l1    0.11967    0.04609   2.596   0.0107 *
const       -0.03718    0.01645  -2.260   0.0257 *
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .
     0.1         1

Residual standard error: 0.1349 on 114 degrees of freedom
Multiple R-Squared: 0.1331, Adjusted R-squared: 0.1027
F-statistic: 4.376 on 4 and 114 DF,  p-value: 0.002498

Estimation results for equation IXERT:
======================================
IXERT = SPGSCLTR.l1 + SPGSNGTR.l1 + IXERT.l1 + CPI_gr.l1 + const

              Estimate Std. Error t value Pr(>|t|)
SPGSCLTR.l1 -0.072025   0.080062  -0.900    0.370
SPGSNGTR.l1 -0.057524   0.060827  -0.946    0.346
IXERT.l1    -0.007140   0.118243  -0.060    0.952
CPI_gr.l1    0.008228   0.029689   0.277    0.782
const        0.001256   0.010597   0.118    0.906

Residual standard error: 0.08689 on 114 degrees of freedom
Multiple R-Squared: 0.02183,  Adjusted R-squared: -0.01249
F-statistic: 0.636 on 4 and 114 DF,  p-value: 0.6379

Estimation results for equation CPI_gr:
```

```
========================================
CPI_gr = SPGSCLTR.l1 + SPGSNGTR.l1 + IXERT.l1 + CPI_gr.l1 + const

             Estimate Std. Error t value Pr(>|t|)
SPGSCLTR.l1  0.56947    0.20097   2.834  0.00544 **
SPGSNGTR.l1  0.16721    0.15269   1.095  0.27577
IXERT.l1     0.77724    0.29681   2.619  0.01003 *
CPI_gr.l1    0.42145    0.07453   5.655 1.17e-07 ***
const        0.12543    0.02660   4.715 6.89e-06 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .
     0.1         1

Residual standard error: 0.2181 on 114 degrees of freedom
Multiple R-Squared: 0.4883, Adjusted R-squared: 0.4704
F-statistic:  27.2 on 4 and 114 DF,  p-value: 7.467e-16

Covariance matrix of residuals:
          SPGSCLTR   SPGSNGTR    IXERT     CPI_gr
SPGSCLTR 0.015323  0.0024726 0.007102  0.0091640
SPGSNGTR 0.002473  0.0181964 0.003368 -0.0008454
IXERT    0.007102  0.0033679 0.007550  0.0052630
CPI_gr   0.009164 -0.0008454 0.005263  0.0475711

Correlation matrix of residuals:
          SPGSCLTR SPGSNGTR  IXERT   CPI_gr
SPGSCLTR   1.0000  0.14808 0.6603  0.33943
SPGSNGTR   0.1481  1.00000 0.2873 -0.02873
IXERT      0.6603  0.28735 1.0000  0.27771
CPI_gr     0.3394 -0.02873 0.2777  1.00000
```

Appendix E

Bibliography

[Bellucci, 2018] Bellucci, L. (2018). Sector Primer Series: Energy. Technical report, S&P Dow Jones Indices, A Division of S&P500 Global.

[Bodie et al., 2018] Bodie, Z., Kane, A., and Marcus, A. (2018). *Investments*. McGraw-Hill Education, eleventh edition.

[Chatfield and Xing, 2020] Chatfield, C. and Xing, H. (2020). *The Analysis of Time Series. An Introduction with R*. Chapmann & Hall / CRC Press, 7 edition.

[Chernoff and Zacks, 1964] Chernoff, H. and Zacks, S. (1964). Estimating the current mean of a Normal distribution which is subjected to changes in time. Technical report, Stanford University and Technion, Israel Institute of Technology.

[Chib, 1998] Chib, S. (1998). Estimation and comparison of multiple change-point models. *Journal of Econometrics*, 86(2):221–241.

[Chib and Kang, 2013] Chib, S. and Kang, K. (2013). Change Points in Affine Arbitrage-free Term Structure Models. *Journal of Financial Economics*, 11(2):302–334.

[Coro, 2013] Coro, G. (2013). A Lightweight Guide on Gibbs Sampling and JAGS Gibbs Sampling with JAGS : Behind the Scenes. Technical report, Istituto di Scienza e Tecnologie dell'Informazione A. Faedo, Pisa, Italy.

[Credit Suisse, 2019] Credit Suisse (2019). Business Cycle Risk: Diversifying With Commodities. Technical report, Credit Suisse.

[Dale, 1985] Dale, A. I. (1985). *A History of Inverse Probability. From Thomas Bayes to Karl Pearso*. Springer, 2 edition.

[Denwood, 2016] Denwood, M. J. (2016). runjags: An R package providing interface utilities, model templates, parallel computing methods and additional distributions for MCMC models in JAGS. *Journal of Statistical Software*, 71(9).

[Depaoli et al., 2016] Depaoli, S., Clifton, J., and Cobb, P. (2016). Just Another Gibbs Sampler (JAGS): Flexible Software for MCMC Implementation. *Journal of Educational and Behavioral Statistics*, 41(6):628–649.

[Dufays et al., 2021] Dufays, A., Li, Z., Rombouts, J. V., and Song, Y. (2021). Sparse change-point VAR models. *Journal of Applied Econometrics*, 36(6):703–727.

[Dufays and Rombouts, 2018] Dufays, A. and Rombouts, J. (2018). Relevant parameter changes in structural break models. Technical report, Universit´e Laval and ESSEC Business School.

[Elliott and Timmermann, 2013] Elliott, G. and Timmermann, A. (2013). *Handbook of Economic Forecasting*, volume 2B. Elsevier.

[Gelfand and Smith, 1990] Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

[Gelman and Donald, 1992] Gelman, A. and Donald, B. R. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–511.

[Giordani and Kohn, 2008] Giordani, P. and Kohn, R. (2008). Efficient Bayesian inference for multiple change-point and mixture innovation models. *Journal of Business and Economic Statistics*, 26(1):66–77.

[Graham and Dodd, 1934] Graham, B. and Dodd, D. L. (1934). *Security Analysis*. The McGraw-Hill Companies, Inc.

[Hashem Pesaran et al., 2006] Hashem Pesaran, M., Pettenuzzo, D., and Timmermann, A. (2006). Forecasting time series subject to multiple structural breaks. *Review of Economic Studies*, 73(4):1057–1084.

[Hastings, 1970] Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109.

[Höhna et al., 2014] Höhna, S., Heath, T. A., Boussau, B., Landis, M. J., Ronquist, F., and Huelsenbeck, J. P. (2014). Probabilistic graphical model representation in phylogenetics. *Systematic Biology*, 63(5):753–771.

[Holbert, 1982] Holbert, D. (1982). A Bayesian Analysis Of A Switching Linear Model. *Journal of Econometrics*, 19:77–87.

[Kass et al., 1998] Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. M. (1998). Markov Chain Monte Carlo in Practice: A Roundtable Discussion. 52(2):93–100.

[Koop and Potter, 2007] Koop, G. and Potter, S. M. (2007). Estimation and forecasting in models with multiple breaks. *Review of Economic Studies*, 74:763–789.

[Kruschke, 2015] Kruschke, J. K. (2015). *Doing Bayesian Data Analysis. A Tutorial with R, JAGS, and Stan*. Elsevier, 2 edition.

[Lee, 2019] Lee, M. D. (2019). A simple and flexible Bayesian method for inferring step changes in cognition. *Behavior Research Methods*, 51(2):948–960.

[Maheu and Gordon, 2008] Maheu, J. and Gordon, S. (2008). Learning, Forecasting and Structural Breaks. *Journal of Applied Econometrics*, 23:553–583.

[Maheu and Song, 2018] Maheu, J. and Song, Y. (2018). An efficient Bayesian approach to multiple structural change in multivariate time series. *Journal of Applied Econometrics*, 33:251–270.

[Malsiner-Walli and Wagner, 2016] Malsiner-Walli, G. and Wagner, H. (2016). Comparing Spike and Slab Priors for Bayesian Variable Selection. *Austrian Journal of Statistics*, 40(4).

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

[Page, 1954] Page, E. (1954). Continuous Inspection Schemes. *Biometrika*, 41(1):100–115.

[Page, 1955] Page, E. (1955). A Test for a Change in a Parameter Occurring at an Unknown Point. *Biometrika*, 42(3):523–527.

[Peluso et al., 2019] Peluso, S., Chib, S., and Mira, A. (2019). Semiparametric multivariate and multiple change-point modeling. *Bayesian Analysis*, 14(3):727–751.

[Pettenuzzo and Timmermann, 2011] Pettenuzzo, D. and Timmermann, A. (2011). Predictability of stock returns and asset allocation under structural breaks. *Journal of Econometrics*, 164(1).

[Pfaff and Stigler, 2022] Pfaff, B. and Stigler, M. (2022). Package ‘ vars ’. Technical report.

[Plummer, 2003] Plummer, M. (2003). JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling. Technical report.

[Plummer, 2017] Plummer, M. (2017). JAGS Version 4.3.0 user manual. Technical report.

[Preuss et al., 2015] Preuss, P., Puchstein, R., and Dette, H. (2015). Detection of Multiple Structural Breaks in Multivariate Time Series. *Journal of the American Statistical Association*, 110(510):654–668.

[Smith, 1975] Smith, A. (1975). A Bayesian Approach to Inference about a Change-Point in a Sequence of Random Variables. *Biometrika*, 62(2):407.

[S&P Global, 2022a] S&P Global (2022a). S&P 500 Pure Growth. Technical report, S&P Dow Jones Indices, A Division of S&P500 Global.

[S&P Global, 2022b] S&P Global (2022b). S&P 500 Pure Value. Technical report, S&P Dow Jones Indices, A Division of S&P500 Global.

[S&P Global, 2022c] S&P Global (2022c). S&P GSCI Methodology. Technical report, S&P Dow Jones Indices, A Division of S&P500 Global.

[S&P Global, 2022d] S&P Global (2022d). S&P U.S. Indices Methodology. Technical Report June, S&P Dow Jones Indices, A Division of S&P500 Global.

[S&P Global, 2023] S&P Global (2023). S&P500. Technical report, S&P Dow Jones Indices, A Division of S&P500 Global.

[T. Bayes, 1763] T. Bayes, R. P. (1763). An Essay towards solving a Problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, F.R.S. communicated by Mr. Price, in a Letter to John Canton, A.M.F.R.S. *Philosophical Transactions of the Royal Society of London*, 53:370–418.

[Thomas, 2006] Thomas, A. (2006). The BUGS Language. *The Newsletter of the R Project News*, 6(1):7–11.

[van den Burg and Williams, 2020] van den Burg, G. J. J. and Williams, C. K. I. (2020). An Evaluation of Change Point Detection Algorithms. Technical report, The University of Edinburgh and The Alan Turing Institute.