BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING
FAKULTA STROJNÍHO INŽENÝRSTVÍ
INSTITUTE OF MATHEMATICS
ÚSTAV MATEMATIKY

BAYESIAN PARAMETER ESTIMATE USING MCMC

BAYESOVSKÝ ODHAD PARAMETRU POMOCÍ MCMC

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR
AUTOR PRÁCE
Jakub Straka

SUPERVISOR
VEDOUCÍ PRÁCE
Ing. Pavel Hrabec, Ph.D.

BRNO 2025

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

Assignment Bachelor's Thesis

Institut:	Institute of Mathematics
Student:	**Jakub Straka**
Degree programm:	Mathematical Engineering
Branch:	no specialisation
Supervisor:	**Ing. Pavel Hrabec, Ph.D.**
Academic year:	2024/25

As provided for by the Act No. 111/98 Coll. on higher education institutions and the BUT Study and Examination Regulations, the director of the Institute hereby assigns the following topic of Bachelor's Thesis:

Bayesian paremeter estimate using MCMC

Brief Description:

A common limitation of Bayesian estimates is the computability of the posterior probability distribution. One way to at least numerically calculate this distribution is simulation using 'Monte Carlo Markov chains' (MCMC).

Bachelor's Thesis goals:

– Review of basic MCMC techniques
– Application of this technique for Bayesian estimation (at least on simulated data)

Recommended bibliography:

BROOKS, S., GELMAN, A., JONES, G., & Meng, X.-L. (Eds.). Handbook of Markov Chain Monte Carlo (1st ed.). Chapman and Hall/CRC, 2011. https://doi.org/10.1201/b10905

Deadline for submission Bachelor's Thesis is given by the Schedule of the Academic year2024/25

In Brno,

L. S.

doc. Mgr. Petr Vašík, Ph.D. Director of the Institute	doc. Ing. Jiří Hlinka, Ph.D. FME dean

Abstract

This thesis aims to estimate parameters whilst using Markov chain Monte Carlo (MCMC) methods. First of all, the thesis summarises chapters like Statistical Inference, introduction into Markov chains, linear models with the Bayesian approach, as well as the basic theory for MCMC simulations, namely the Metropolis-Hastings algorithm and convergence statistics. The insights from these chapters are utilised at real-world data – properties in Brno. It has been specified that we shall estimate the price of these properties given an array of parameters. Consequently, the results from the simulations are compared and finally, an estimate (point and interval) of prices of hypothetical properties in Brno are presented.

Abstrakt

Cílem této bakalářské práce je odhad parametrů pomocí metod Markov chain Monte Carlo (MCMC). Nejprve je představen potřebný teoretický aparát, jmenovitě některé statě z inferenční statistiky, základy Markovských řetězců, lineární modely (pojato Bayesovsky) a v neposlední řadě také základy z teorie MCMC – algoritmus Metropolis-Hastings a konvergenční statistiky. Tyto poznatky jsou použity na reálných datech nemovitostí ve městě Brno, kde bylo specifikováno jako cíl odhad jejich ceny na základě specifikovaných parametrů nemovitostí. Následně jsou výsledky simulací porovnány, a nakonec je vytvořen odhad (bodový i intervalový) ceny hypotetických nemovitostí v Brně.

Keywords

Bayesian statistics, Markov chain Monte Carlo, Metropolis-Hastings Algorithm, linear regression model, parameter estimate

Klíčová slova

Bayesovská statistika, Markov chain Monte Carlo, algoritmus Metropolis-Hastings, lineární regresní model, odhad parametru

Straka, J.: *Bayesian parameter estimate using MCMC*, Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2025. 59 pages. Supervisor of bachelor's thesis Ing. Pavel Hrabec, Ph.D.

Rozšířený abstrakt

Tato bakalářská práce se zabývá Bayesovským odhadem parametrů pomocí techniky Markov chain Monte Carlo (MCMC). Cílem je získat *aposteriorní rozdělení* parametrů, z něhož lze odvodit klíčové charakteristiky, jako jsou střední hodnota či směrodatná odchylka jednotlivých parametrů. *Aposteriorní rozdělení* je určeno na základě zvoleného *apriorního rozdělení* a *věrohodnosti*, která je odvozena z naměřených dat.

Práce nejprve představuje teoretické základy nezbytné pro pochopení metod MCMC. Následně je z veřejně dostupných zdrojů vytvořen dataset obsahující informace o parametrech 171 bytů v Brně. Tato data slouží jako motivační příklad pro aplikaci MCMC simulací. Kvalita navržených modelů je posuzována pomocí koeficientu determinace R^2 pro Bayesovské modely a metodou Leave-one-out cross-validation (LOO-CV). Práce je strukturována do tří hlavních bloků.

První blok (kapitoly 2 až 5) zahrnuje teoretické základy metod MCMC. Druhá kapitola se věnuje principům inferenční statistiky, kde jsou objasněny pojmy *apriorní rozdělení*, *aposteriorní rozdělení* a *věrohodnost*. Třetí kapitola představuje základy teorie *Markovových řetězců*, zaměřené na řetězce s diskrétním časem a případně spojitou množinou stavů, které jsou klíčové pro algoritmy MCMC. Dále jsou definovány pojmy *stacionární rozdělení* a *limitní rozdělení* Markovova řetězce. Kapitola 4 popisuje *algoritmus Metropolis-Hastings* a statistiky konvergence, jako jsou $\hat{R}$ a *ESS*. Teoretický blok uzavírá kapitola o *Bayesovských lineárních modelech*.

Druhý blok se zaměřuje na získání dat, jejich analýzu a simulace pomocí metod MCMC. V kapitole 6 jsou představena data a analyzován vztah mezi prediktory a závislou proměnnou. Na základě této analýzy je vytvořen lineární model pro predikci cen nemovitostí v Brně. Následně je diskutován výběr parametrů *apriorního rozdělení*, přičemž byla zvolena *normální*, *log-normální*, *exponenciální* a *rovnoměrné* rozdělení. Kapitola 7 popisuje algoritmy a čtyři různé strategie simulací, lišící se délkou a procentem vyřazených vzorků. Simulace jsou implementovány v jazyce *Python* s využitím knihovny *PyMC*, specializované na Bayesovské modelování. Tato kapitola zahrnuje grafické výstupy simulací, včetně grafů *aposteriorního rozdělení* parametrů, a hodnoty parametrů, jako jsou střední hodnota, směrodatná odchylka a konvergenční statistiky.

Poslední blok této bakalářské práce pojednává o analýze navrhovaných strategií a také predikcí budoucích pozorování. Těmito budoucími pozorováními máme na mysli hypoteticky vytvořené byty s předem danými parametry.

Práce poskytuje stručný přehled technik MCMC a na reálných datech demonstruje predikce parametrů modelu, doplněné o grafy a tabulky. Pozoruhodné je, že výpočetně jednodušší strategie (v práci označené jako *„netrpělivá“* a *„konzervativní“*) dosahují srovnatelné přesnosti s časově a výpočetně náročnější *„rozsáhlou“* strategií. Pomocí metod LOO-CV a R^2 bylo zjištěno, že nejvhodnější modely využívají *normální* nebo *rovnoměrné* rozdělení, přičemž v těchoto modelech hodnota R^2 přesahuje 0.85.

Na závěr je pomocí *„rozsáhlé“* simulace provedena predikce cen hypotetických nemovitostí v Brně, a to jak bodově, tak intervalově.

I, hereby, declare that the bachelor's thesis *Bayesian parameter estimate using MCMC* has been written exclusively by the author under the guidance of supervisor Ing. Pavel Hrabec, Ph.D. and with the aid of the literature listed in references.

Jakub Straka

I would like to express my profound gratitude to my supervisor Ing. Pavel Hrabec, Ph.D. for his valuable pieces of advice, friendly attitude as well as patience during the consultations. His guidance proved valuable when confronted with various hardships that occurred during the completion of this thesis.

Finally, yet importantly, I would like to convey my gratefulness to my family for supporting me throughout my studies.

Jakub Straka

Contents

1 Introduction

A common limitation of Bayesian estimates is the computational complexity of the posterior probability distribution. One way to at least numerically compute this distribution is through simulation using "Markov chain Monte Carlo" (MCMC).

The main aim of this thesis is to provide an estimate of parameters on at least simulated data. This goal has been upgraded, which means we have acquired publicly available data on the property prices in Brno. Furthermore, we shall introduce a linear model which predicts the prices of the properties. It must be clarified that this is not the main objective of this thesis, as this example is solely motivational. The model itself could have been improved, which would result in a more complex problem, thus making the results less transparent.

Another target of this thesis is to conduct research on basic MCMC techniques. This matter is described in the first part of this thesis.

This thesis can be roughly divided into three parts. The first part describes the theory crucial to perform an MCMC simulation. The second chapter introduces basic concepts of statistical inference, such as prior and posterior distribution as well as likelihood. Since we want to predict a parameter, the posterior predictive distribution is also introduced. The third chapter focuses on the basic introduction to Markov chains. In our instance, discrete-time Markov chains are considered. In addition, the idea of stationary (and limiting) distribution is presented as we shall later leverage this in simulations to come. Since we want to predict continuous parameters, it is essential to define (discrete-time) Markov chains in a continuous state space. Chapter 4 discusses the basic MCMC algorithm – Metropolis-Hastings. The fourth chapter provides the definitions of convergence statistics, which are used when evaluating the convergence of an MCMC simulation. Chapter 5 terminates this theoretical part, as the linear models with the Bayesian approach are explained.

The second part analyses the acquired data. Chapter 6 thoroughly describes the dataset and its properties. It also provides an explanation for the setup, in particular the parameters for prior distributions. In chapter 7, the simulation as well as the algorithm are described. Note that the simulation is executed in programming language *Python* whilst mainly utilising the library *PyMC*. Various strategies for MCMC simulations are proposed, which all vary in the percentage of discarded samples (*burn-in* phase), length or number of parallel chains used.

Finally, in the last part of this thesis (Chapter 8), the results of various simulations are compared according to R^2 and *LOO-CV* (both measuring the performance of a model) are compared. Moreover, the predictions of hypothetical properties located in Brno (provided the necessary parameters) are shown.

2 Statistical Inference

In order to understand the concept of Markov chain Monte Carlo, a few chapters from Statistics shall be introduced. Chapters about prior and posterior distributions of random variables will particularly prove to be useful later in this thesis.

2.1 Prior Distribution

The distribution of a parameter before any further data are observed is called the prior distribution of the parameter. [3]

Definition 2.1 (Prior distribution)**.** Suppose that one has a statistical model with parameter θ. If one treats θ as random, then the distribution that one assigns (either p.f. or p.d.f.) to θ before observing the other random variables of interest is called its prior distribution. [3]

Note. When the parameter space[1] (Ω) is at most countable (meaning its cardinality is less or equal than $|\aleph_0|$), the given distribution is discrete, which implies the prior p.f. of θ. In the case of the cardinality of Ω greater than $|\aleph_0|$, one refers to a continuous distribution, thus we call prior p.d.f of θ.

The notation which will be used to represent the prior distribution is $\xi(\theta)$. As can be seen, the prior distribution is a function of a parameter θ (or even of a vector $\boldsymbol{\theta}$) which we want to estimate.

2.2 Posterior Distribution

Definition 2.2 (Posterior distribution)**.** Let $\theta \in \Omega$ be a parameter and let $X_1, \ldots, X_n$ be observed data, then a conditional distribution of θ given $X_1, \ldots, X_n$ is called the posterior distribution. [3]

Theorem 2.3. *Suppose we have a prior p.d.f. or p.f. of θ denoted by $\xi(\theta)$ and random variables $X_1, \ldots, X_n$ which form a distribution $f(x|\theta)$. Then the posterior p.f. or p.d.f. is:*

$$\xi(\theta|\boldsymbol{x}) = \frac{f(x_1|\theta)\cdots f(x_n|\theta)\xi(\theta)}{g_n(\boldsymbol{x})} \quad \text{for } \theta \in \Omega. \tag{2.1}$$

where $g_n(\boldsymbol{x})$ is marginal joint p.d.f. or p.f. of $X_1, \ldots, X_n$. [3]

One can see a strong resemblance between the expression from theorem 2.3 and Bayes' theorem.

It should be noted that the exact value of the parameter θ is unknown, and it is to be further estimated. Should we wish to improve our estimation of the parameter θ, which is at this point given by the function $\xi(\theta|x)$, an iterative approach can be established. This is possible only if we have another set of independent observations, otherwise this iterative approach is invalid. This means that our (to this very moment) posterior distribution shall be declared as a prior distribution, and the computation of our updated posterior distribution can be performed again. This iterative process can continue until our requirements (on accuracy) are met.

[1] Parameter space (represented by symbol Ω) is a set of all possible values of a particular parameter (in our case θ) or a vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)^T$.

Function $g_n(\boldsymbol{x})$ can be interpreted as prior predictive distribution (noted as $\varphi(\boldsymbol{x})$), i.e. distribution of (yet unknown) random vector $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ before the data are considered.

$$\varphi(\boldsymbol{x}) = g_n(\boldsymbol{x}) = \int_\Omega f_n(\boldsymbol{x}|\theta)\xi(\theta)\,\mathrm{d}\theta, \tag{2.2}$$

where $f_n(\boldsymbol{x}|\theta) = \prod_{i=1}^n f(x_i|\theta)$ and Ω denotes parameter space. [6]

2.2.1 Posterior Predictive Distribution

After a dataset has been observed (vector $\boldsymbol{x}$), an unknown variable $\tilde{\boldsymbol{x}}$ can be estimated. Let $\varphi(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ be a posterior predictive function. In order to obtain the exact form of this particular function, we can use equation 2.2 and *update* prior distributions to posterior (as they are conditional on the observed vector $\boldsymbol{x}$).

$$\varphi(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \int_\Omega f_n(\tilde{\boldsymbol{x}}|\theta)\xi(\theta|\boldsymbol{x})\,\mathrm{d}\theta,$$

where $\xi(\theta|\boldsymbol{x})$ addresses posterior distribution of parameter θ given the data $\boldsymbol{x}$ and $f_n(\tilde{\boldsymbol{x}}|\theta)$ is analogical to the previous predictive distribution. [6]

2.3 Likelihood

Definition 2.4 (Likelihood function)**.** One refers to a likelihood function when the joint p.d.f. (or p.f.) of the observations in random sample $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ (where $f(x_i|\theta)$ is p.d.f. (or p.f.) of each $X_i = x_i$) is regarded as a function of θ for given values of vector $\boldsymbol{X}$. [3] [4]

$$\mathcal{L}(\theta|\boldsymbol{x}) = \prod_{i=1}^n f(x_i|\theta). \tag{2.3}$$

Note. The likelihood function does not have to be normalised; therefore, any function proportionate to 2.3 shall also be declared the likelihood function.

Let's recall equation 2.1 for the reason that we can replace it with the following relation:

$$\xi(\theta|\boldsymbol{x}) \propto \mathcal{L}(\theta|\boldsymbol{x})\xi(\theta),$$

which states that posterior distribution is proportionate to the product of likelihood and prior distribution.

Since posterior distribution is p.d.f. (or in discrete case p.f.) the following condition must be obeyed: $\int_\Omega \xi(\theta|\boldsymbol{x})\,\mathrm{d}\theta = 1$ (for discrete parameter space: $\sum_\Omega \xi(\theta|\boldsymbol{x}) = 1$). The normalising constant comes from integrating (or summing) the product of likelihood and prior distribution over the region Ω, which yields the equation 2.1 back.

3 Markov Chains

As the name of the method (Markov chain Monte Carlo) suggests, it is crucial to introduce the basics of Markov chains.

This chapter only describes the matter of discrete-time Markov chains (on a continuous state space), as they will be used later in this thesis. The case when we consider continuous-time for Markov chains shall be omitted.

3.1 Discrete-time Markov Chains

Definition 3.1 (Stochastic process in discrete time)**.** A stochastic process in discrete time is a family $\{X_t;\ t \in T\}$, where $T = \mathbb{N}_0$ and the set of all possible values of random variables X_t, $\forall t \in T$ is referred as state space S of the process. [5] [16]

Definition 3.2 (Discrete-time Markov chain)**.** A stochastic process $\{X_t;\ t \in T\}$ defined on probability space $(\Omega, \mathcal{F}, P)$ [2] on a countable state space S is a continuous-time Markov chain if and only if:

$$\Pr(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = \Pr(X_{n+1} = j | X_n = i). \tag{3.1}$$

for all $n \in \{0, 1, \ldots\}$ and for all $i_0, \ldots, i_{n-1}, i, j$, where $\Pr(X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) > 0$. [5] [15]

Note. From now on, a simpler version of the notation will be used, meaning that the transition probabilities[3] shall be denoted by:

$$p_{ij}(n, n+1) = \Pr(X_{n+1} = j | X_n = i), \quad i, j \in S,\ n = 0, 1, \ldots$$

Suppose a discrete-time Markov chain which satisfies the following property:

$$p_{ij}(n, n+1) = p_{ij}, \quad \text{for each } i, j \in S.$$

Then, one talks about a time-homogeneous Markov chain, which implies that the time dynamics do not evolve throughout time (do not depend on the index n). [5] [15]

Determining the *initial distribution* (denoted by $\phi(i)$) is necessary to describe the starting conditions of a given Markov chain fully.

$$\phi(i) = \Pr(X_0 = i),\ i \in S, \quad \sum_{i \in S} \phi(i) = 1.$$

Now we can define a transition matrix for every transition probability $p_{ij},\ i, j \in S$, such that $\boldsymbol{P}(t) = \{p_{ij}\}_{i,j \in S}$. Note that for every row in matrix $\boldsymbol{P}$, the following condition must be satisfied: $\sum_{j \in S} p_{ij} = 1$. [16]

[2] Where Ω stands for event-space, $\mathcal{F}$ denotes σ-algebra and P depicts probability function.

[3] From an initial state i at time n to a final state j at time $n + 1$.

3.1.1 Stationary and Limiting Distribution

Definition 3.3 (Stationary distribution)**.** Let $\{X_t,\ t \in T\}$ be a discrete-time Markov chain, $\boldsymbol{P} = (p_{ij})_{i,j\in S}$ denotes matrix of transition probabilities, and set S is state space, then a probability vector $\bar{\boldsymbol{\pi}} = (\bar{\pi}_i)_{i\in S}$ is a *stationary (invariant) distribution* if $\forall t \in T$ and $j \in S$: [5] [16]

$$\bar{\pi}_j = \sum_{i\in S} \bar{\pi}_i p_{ij}. \tag{3.2}$$

An equation 3.2 can be rewritten only in terms of vectors and transition matrix $\boldsymbol{P}$. [5] [15]

$$\bar{\boldsymbol{\pi}} = \bar{\boldsymbol{\pi}}\boldsymbol{P}, \quad \text{where } \bar{\boldsymbol{\pi}} = \{\bar{\pi}_i : i \in S\}.$$

An equivalent definition of *stationary distribution* according to [15] states that a stochastic process $\{X_t,\ t \in T\}$ on a general state space is *stationary* if distributions X_n remains the same if the time is shifted by any amount of $n \geq 0$. Therefore, it complies:

$$(X_n, \ldots, X_{n+k}) = (X_0, \ldots, X_k), \quad \text{where } k \geq 1.$$

Definition 3.4 (Limiting distribution)**.** Suppose X_n is a Markov chain with transition probabilities p_{ij}. A probability distribution $\boldsymbol{\pi} = \{\pi_i : i \in S\}$ is called *limiting distribution* if and only if all $i \in S$ satisfy: [16] [15]

$$\lim_{n\to\infty} \Pr(X_n = i) = \pi_i,\ i \in S.$$

Since $\boldsymbol{\pi} = \{\pi_i : i \in S\}$ is a distribution, following condition must be fulfilled:

$$\sum_{i\in S} \pi_i = 1.$$

Theorem 3.5. *If a limiting distribution of a discrete-time Markov chain (homogeneous-time) exists, it shall be proclaimed as a stationary distribution.* [5]

3.2 Discrete-time Markov Chain on Continuous State Space

At this point, we shall define a discrete-time Markov chain on a continuous state space. It is obvious that the Markov property (3.1) in a continuous state space persists. Previously, the state space S was discrete, while in this chapter it will be continuous.

In this instance (S being continuous), the Markov chain is influenced by a transition kernel $K(x, A)$, where $x \in S$ and $A \subset S$. Let $\sigma(S)$ be a Borel σ-algebra on S, then the transition kernel $K : S \times \sigma(S) \to [0, 1]$ defines a Markov chain $\{X_t\}$ through a property:

$$\Pr(X_{t+1} \in A | X_t, \ldots, X_0) = K(X_t, A).$$

One can identify that the transition kernel $K(X_t, A)$ denotes the probability of the Markov chain moving in one step from state x into a set A. [10]

If the kernel $K(x, A)$ is well behaved (Markov chain can jump from an arbitrary state to any state in a finite number of steps) then the Markov chain will have a *stationary distribution* $\pi(x)$ such that:

$$\pi(A) = \int_S \pi(x)K(x, A)\,\mathrm{d}x, \quad A \in S.$$

4 Markov Chain Monte Carlo Algorithms

This chapter relates to the fundamentals of algorithms Markov chain Monte Carlo, which shall later be leveraged. The Metropolis-Hastings algorithm (only the generic variant) will be exclusively described.

4.1 Metropolis–Hastings Algorithm

This method aims to approximate a given target density f by introducing the Markov chain $\{X_t, t \geq 0\}$ so that the limiting distribution of this Markov chain is function f.

Definition 4.1 (Markov kernel)**.** Assume that $\mathcal{X}$ is a complete separable metric space (Polish space) with the σ-field $\mathcal{B}$ of its Borel subsets. We say that a Markov kernel on $(\mathcal{X}, \mathcal{B})$ is a mapping:

$$P : \mathcal{X} \times \mathcal{B} \to [0, 1], \quad \text{such that:}$$

1. $P(x, \cdot)$ is a probability measure on $(\mathcal{X}, \mathcal{B})$ for every $x \in \mathcal{X}$,
2. $P(\cdot, A)$ is $\mathcal{B}$-measurable mapping from $\mathcal{X}$ to [0,1] for every $A \in \mathcal{B}$. [9]

Let $Q(x, \cdot)$ be a Markov kernel whose transitions have a possibly unnormalised density, which implies:

$$Q(x, \mathrm{d}y) \propto q(x, y)\mathrm{d}y.$$

The conditional density (called the proposal distribution) $q(y|x)$ can be almost arbitrary in order to produce the limiting distribution, which ought to estimate the target density f. Also, let's denote the probability $\rho(x, y)$ of accepting a move from x to Y, where x is the current (newest) value, and Y is a proposed value that is chosen according to the $q(y|x)$ (respectively by $Q(x, \cdot)$). [13] [9] [14] [8]

According to [13] and [9], the basic algorithm is described below.

Algorithm 1 Metropolis-Hastings (one iteration)

Given x_t,

1. Generate proposal $Y_t \sim q(y|x_t)$.
2. Take:

$$X_{t+1} = \begin{cases} Y_t & \text{with probability } \rho(x_t, Y_t), \\ x_t & \text{with probability } 1 - \rho(x_t, Y_t), \end{cases}$$

where:

$$\rho(x, y) = \min\left\{\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}, 1\right\}.$$

When the ratio $\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)} \geq 1$, the proposed value of Y_t is certain to be accepted.

In case of proposal distribution q being symmetric (when $q(x|y) = q(y|x)$), the acceptance probability will be altered to $\rho(x, y) = \min\left\{\frac{f(y)}{f(x)}, 1\right\}$, thus is it only driven by the ratio of the target distribution. It is, therefore, independent of the function q. [13]

4.2 Convergence and Efficiency of Markov Chain

The essential idea behind an MCMC simulation is the convergence of a sampling distribution to a limiting (stationary) distribution when time reaches infinity (in an infinite number of samples).

Since we want to estimate the target distribution (especially the posterior distribution) in a finite time (in a finite number of iterations), one needs to introduce statistics which describe the rate and the efficiency of convergence. $\hat{R}$ statistic combined with ESS is the most common way of addressing the convergence and efficiency of simulated chains.

The purpose of those statistics is to determine whether a given simulation yields satisfactory answers. [12] [17]

4.2.1 $\hat{R}$ statistic

The motivation of $\hat{R}$ statistic is to answer whether the chains in an MCMC simulation mixed well. By the term *mixed well*, one refers to when all chains started from different initial points, explored the target distribution, and consequently all converged to the same stationary distribution.

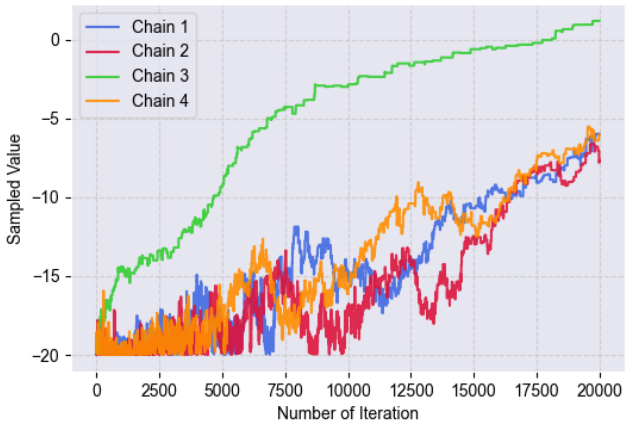

Figure 4.1: Chains Not Mixing Well

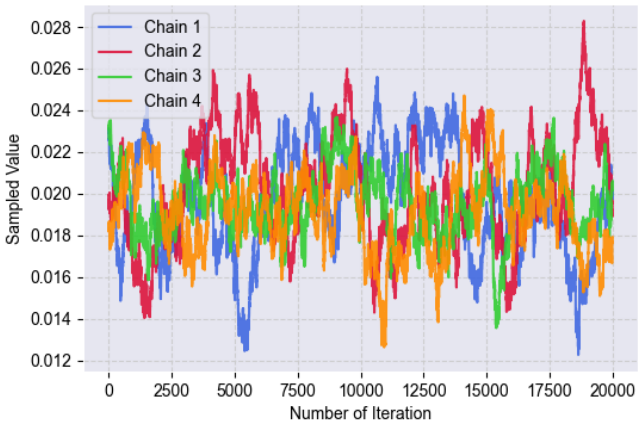

Figure 4.2: Chains Mixing Well

The statistic is computed by the square root of the weighted average of variance within-chain and variance between chains divided by the variance within-chain.

Let N be a number of samples drawn from a single chain, let M be a number of chains, $\theta^{(nm)}$ is n-th draw from chain m. Then W is a variance within-chain, and B is a variance between chains.

$$W = \frac{n}{m-1}\sum_{m=1}^{M} s_m^2, \quad \text{where} \quad s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}(\theta^{(nm)} - \bar{\theta}^{(\cdot m)})^2.$$

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{\theta}^{(\cdot m)} - \bar{\theta}^{(\cdot\cdot)})^2, \quad \text{where} \quad \bar{\theta}^{(\cdot m)} = \frac{1}{N}\sum_{n=1}^{N}\theta^{(nm)} \quad \text{and} \quad \bar{\theta}^{(\cdot\cdot)} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}^{(\cdot m)}. \tag{4.1}$$

Then $\hat{R}$ statistic is computed by following expression:

$$\hat{R} = \sqrt{\frac{\frac{N-1}{N}W + \frac{1}{N}B}{W}}. \tag{4.2}$$

It should be noted that for only one chain ($M = 1$), the expression for computing the variance between chains (4.1) is still valid because the chain will be split in half (this process is called *split-*$\hat{R}$), thus doubling the value of M. [17]

Provided the expression 4.2, one can observe that for $N \to \infty$, the value of $\hat{R}$ reaches 1.

According to [12] and [17], the conservative approach to deciding if all the chains in the simulation mixed well is set to:

$$\hat{R} < 1.01.$$

4.2.2 Effective Sample Size

A relatively small value of the $\hat{R}$ statistic ($\hat{R} \approx 1$) is not enough to derive the conclusion that an accurate output has been provided by an MCMC simulation. Another statistic needs to be addressed.

Generally speaking, an effective sample size (ESS) refers to a quantity of independent draws used that contain the same information as the total amount of draws.

For a single chain, the ESS can be computed with the following expression:

$$ESS_1 = \frac{N}{1 + 2\sum_{t=1}^{\infty} \rho_t}, \tag{4.3}$$

where N denotes a number of draws from one chain and ρ_t is autocorrelation at lag $t > 0$. [12] [17]

In order to define autocorrelation, it is required to introduce autocovariance at lag-t.

$$\gamma_t = \text{cov}\{g(X_i), g(X_{i+t})\},$$

where $X_1, X_2, \ldots$ is a Markov chain and $g(X_1), g(X_2), \ldots$ are functionals of this Markov chain. The autocorrelation function is defined: $\rho : t \mapsto \frac{\gamma_t}{\gamma_0}$, thus:

$$\rho_t = \frac{\text{cov}\{g(X_i), g(X_{i+t})\}}{\text{var}\{g(X_i)\}} = \frac{\sum_{i=1}^{n-t}[g(X_i) - \hat{\mu}_N][g(X_{i+t}) - \hat{\mu}_N]}{\sum_{i=1}^{n}[g(X_i) - \hat{\mu}_N]^2},$$

where $\hat{\mu}_N$ denotes a sample mean. [2]

For multiple chains (M is the number of used chains) the equation (4.3) for computing ESS is altered as follows:

$$ESS_M = \frac{NM}{1 + 2\sum_{t=1}^{\infty} \rho_t},$$

To ensure reliable results from an MCMC simulation, the value of ESS should surpass (in accordance with [12] [17]) a value of 400, thus:

$$ESS > 400.$$

5 Bayesian Approach to Linear Models

This chapter shall introduce briefly a Bayesian approach to linear models. Generally speaking, a linear regression answers the question of how exactly the data from X influence the response variable Y. A posterior predictive distribution is to be discussed, as well as other Bayesian aspects.

Suppose that we have observations (x_i, y_i), $i = 1, \ldots, n$ – in total n observations with interchangeable order. Variable $\boldsymbol{X} = (X_1, \ldots, X_k)$ are called explanatory variables (independent variables) and $\boldsymbol{Y} = (y_1, \ldots, y_n)^T$ is a response variable. In a general sense, the variables $X_1, \ldots, X_n$ can be either continuous or discrete, but in the case of the response variable $\boldsymbol{Y}$ we presume that it is continuous. [6] [1]

The simplest version of the linear model (normal linear model) ought to be introduced where the distribution of y_i given the data $\boldsymbol{X}$ is normal with its expected value following a linear expression:

$$\mathrm{E}(y_i|\boldsymbol{\beta}, \boldsymbol{X}) = \beta_1 x_{i1} + \cdots + \beta_k x_{ik}, \tag{5.1}$$

for $i = 1, \ldots, n$. It should be clarified that for many applications, the term x_{i1} is replaced by a constant 1, thus replacing the equation 5.1:

$$\mathrm{E}(y_i|\boldsymbol{\beta}, \boldsymbol{X}) = \beta_1 + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}. \tag{5.2}$$

Furthermore, to obtain the exact value of y_i, one must adjust the equation 5.2.

$$y_i = \beta_1 + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i, \quad \text{for } i = 1, \ldots, n.$$

In terms of matrices:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^T$ represents the vector of residuals (errors), the vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)^T$ is unknown, $\boldsymbol{X}$ is a known matrix of predictors (of dimensions $n \times k$) and $\boldsymbol{Y}$ is a vector of observed values ($n \times 1$). [6]

In the normal linear model framework the crucial statistical issues are:

1. Defining the variables X_i and $\boldsymbol{Y}$, possibly altering them by using transformations so that $\boldsymbol{Y}$ is reasonably linear as a function of $\boldsymbol{X} = (X_1, \ldots, X_n)$, where errors $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 \boldsymbol{I})$ [4].
2. Setting up the prior distribution parameters on a model which accurately reflects our knowledge about a particular parameter.

The aim of statistical inference is to estimate the parameters $\boldsymbol{\beta}$ and σ, conditional on X and Y.

That being said, we can derive the simplest case for linear regression whilst using the matrix notation. [6]

$$\boldsymbol{Y}|\boldsymbol{\beta}, \sigma, \boldsymbol{X} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}) \tag{5.3}$$

[4] By the variance σ^2 we mean $\sigma^2 = \mathrm{var}(y_i|\boldsymbol{\beta}, \boldsymbol{X}),\ \forall i$

5.1 Posterior predictive distribution

Another useful application of the regression model is to predict an outcome variable $\tilde{\boldsymbol{Y}}$ when we have observed a data $\tilde{\boldsymbol{X}}$ of explanatory variables.

The posterior predictive distribution $p(\tilde{\boldsymbol{Y}}|\boldsymbol{Y})$ describes the distribution of yet unobserved data (outcome) $\tilde{\boldsymbol{Y}}$ given the already considered outcome $\boldsymbol{Y}$.

There are two components of uncertainty directly related to $p(\tilde{\boldsymbol{Y}}|\boldsymbol{Y})$. First, being represented by the variance σ^2, which translates to a "noise" which is not accounted for by $\boldsymbol{X\beta}$ (the model itself). The other is tied to the posterior uncertainty, which means it is caused by the fact that the response variable $\boldsymbol{Y}$ is finite (the dataset has essentially a finite length n). However, this uncertainty can be omitted because it diminishes as the sample size n gets progressively larger ($n \to \infty$).

Therefore, for normal distribution (the distribution can be arbitrary) we obtain:

$$\tilde{\boldsymbol{Y}} \sim N(\tilde{\boldsymbol{X}}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}).$$

Furthermore, let us consider the future observation $\tilde{\boldsymbol{Y}}$ provided the variance σ^2, then the expected value of $\tilde{\boldsymbol{Y}}$ would be:

$$E(\tilde{\boldsymbol{Y}}|\sigma, \boldsymbol{Y}) = E(\tilde{\boldsymbol{X}}\boldsymbol{\beta}|\sigma, \boldsymbol{Y}) = \tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}},$$

where vector $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_k)^T$ represents the prediction of vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)^T$. [6]

6 Estimation of the Prices of Properties in Brno

To use the theory listed above, we shall investigate a real-world problem, which implies using a real-world dataset. An estimation of the prices of properties in Brno shall be utilised as a motivational example.

6.1 Acquiring Data

To make our model more comprehensive, our dataset consists of only furnished flats located in the Brno-City District. All data were extracted from a website [24] during Q1/2025.

The data we are extracting from the website are: the *price* of the property, the *area* of the property, the *age* of the property, its location (*district index*), and two categorical variables whether the property has a *balcony* or a *garage*.

In total, there are six variables to be considered (five independent and one dependent).

6.1.1 Price

The dependent variable *price* of a property is recorded in Czech crowns (CZK). Given that this particular variable seems to be distributed log-normally, we shall take a logarithm (base e) of the price column in our dataset. Furthermore, this implies that we will predict the order-of-magnitude change, not an absolute change, which makes the prediction more transparent.

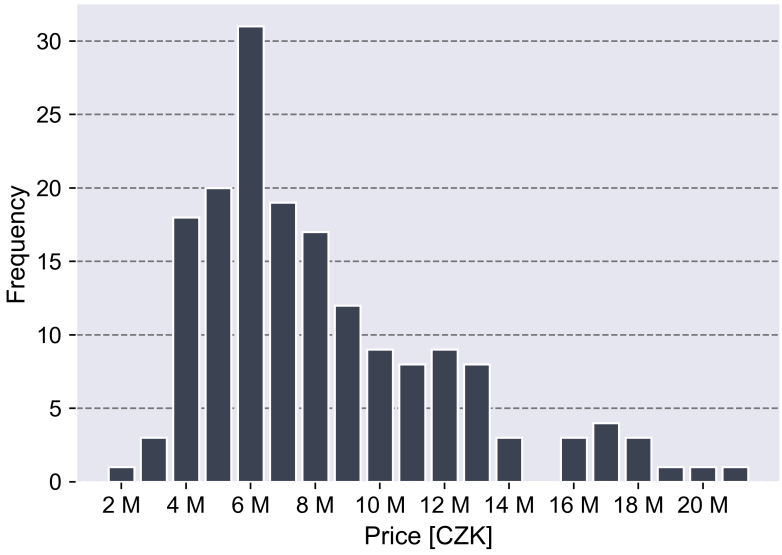

Figure 6.1: Histogram of the Prices

Now, using the log-transformed *price* variable, one might object that it roughly follows a normal distribution.

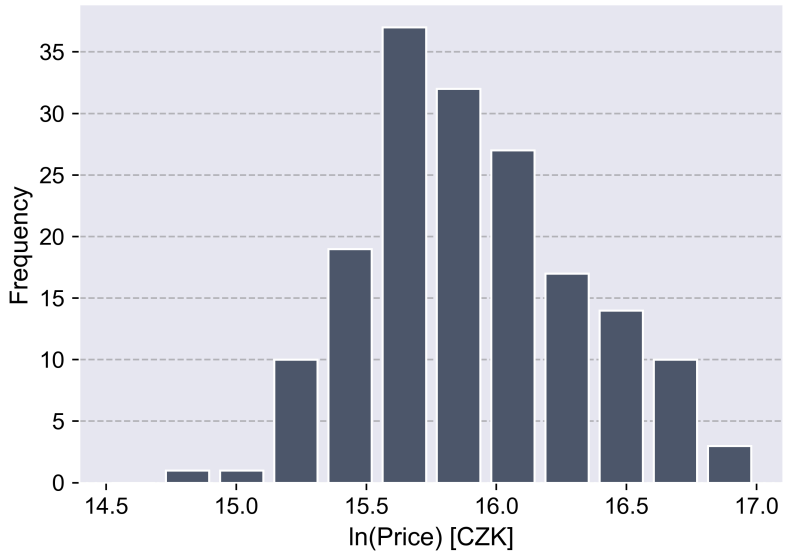

Figure 6.2: Histogram of the Logged Prices

6.1.2 Area of a Property

Another considered variable is the *area* of the property (measured in m^2). This variable represents living quarters inside each flat, meaning this number does not include a balcony, a terrace, or any other outside area.

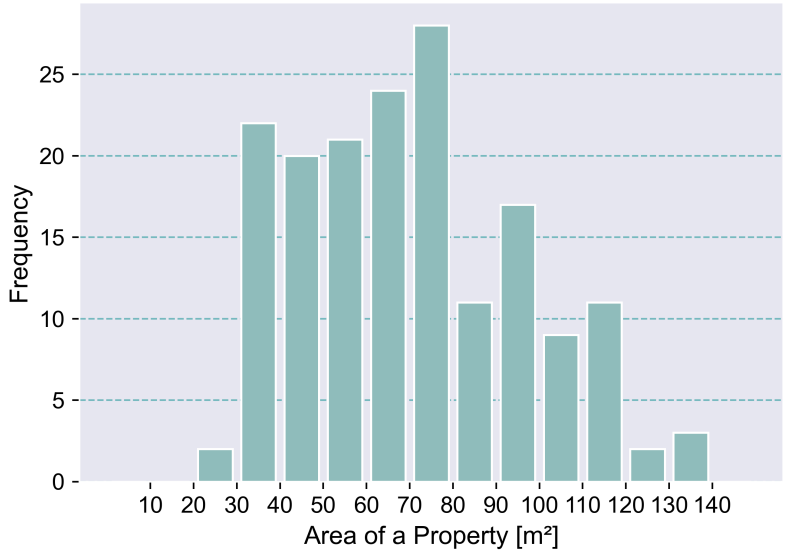

Figure 6.3: Histogram of Area

6.1.3 Age of a Property

When determining the *age* of the property, we shall compare the date (to be precise, a year) once a certificate of occupancy has been obtained or a thorough (or a complete) reconstruction has taken place. Even though we only compare years, we shall treat this variable as continuous.

Figure 6.4: Histogram of Property Age

6.1.4 Balcony

The first of two categorical variables is whether a particular property has a *balcony* (either enclosed or not) or a terrace.

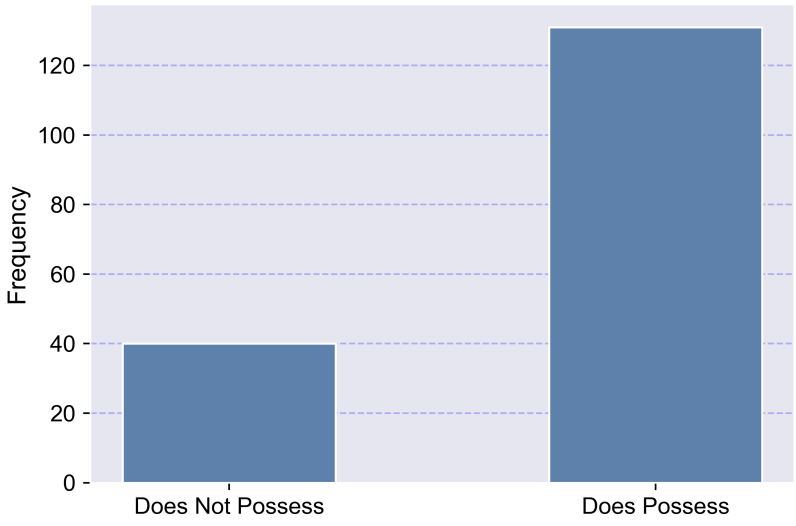

Figure 6.5: Histogram of Presence of a Balcony

6.1.5 Garage

The second categorical variable describes the ownership of a *garage* space or a dedicated parking space in the vicinity. In some cases, one might have opted for buying a garage or parking space for additional funds, but these examples have been omitted from the dataset.

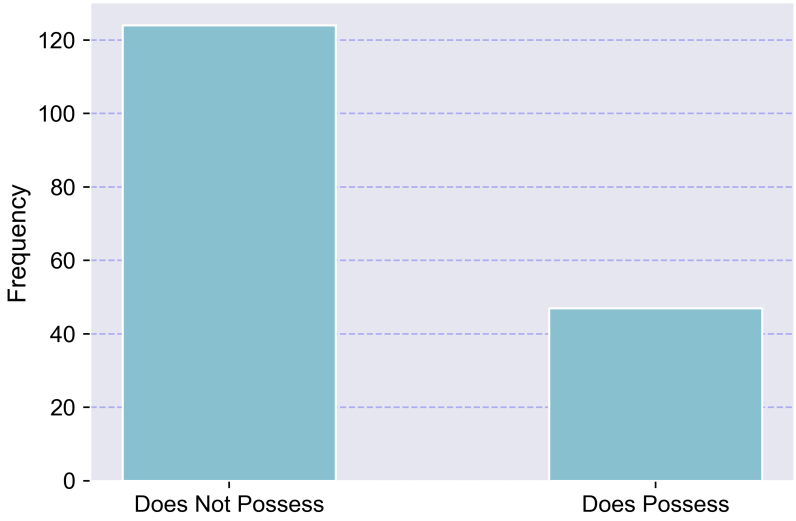

Figure 6.6: Histogram of Presence of a Garage

6.1.6 District

In terms of quarters in Brno, we have to determine an index (*district index*) which depicts the general appeal of each district in Brno itself. In order to fulfil this task, three support variables (scores) are considered (*safety*, *average price per* 1 m^2 and *distance*).

The *safety* index is computed using data of a crime rate and population in each quarter in Brno and the data is skewed so that the data are distributed from 0 to 100.

$$\text{crimerate}_i = \frac{\text{crimes}_i}{\text{population}_i} \cdot 100\,000,$$

where variable *crimes* depicts the number of committed crimes (contravention is not considered in this matter). The data, considering the crime and inhabitants (for each quarter in Brno) were extracted from publicly available sources [22] and [19].

$$\text{sft}_i = \left(1 - 10 \cdot \frac{\text{crimerate}_i}{\text{population}_i}\right) \cdot 100.$$

Another variable, which contributes to a *district* variable is *average price per* 1 m^2 (obtained from [20]). Once again, this variable is altered such that the "score" lies between 0 and 100.

$$\text{ppm}_i^2 = \frac{\text{average price per 1m}_i^2}{\max_{i \in D}(\text{average price per 1m}_i^2)} \cdot 100,$$

where set D signifies our dataset.

The *distance* from a centre of Brno [5] is the last variable that is considered in a compound variable *district*. By distance we mean the shortest path (in a straight line) between 2 points on a sphere computed with [21]. The following formula is used to compute the "score" for the distance:

$$\text{dst}_i = \left[1 - \frac{\text{distance}_i}{\max_{i \in D}(\text{distance}_i)}\right] \cdot 100.$$

Finally, an equation has been established to determine an appeal for each district. This equation artificially models an appeal, where the most significant "score" (according to the author) is considered the variable *average price per* $1\ m^2$, therefore, the most "weight" is dedicated to this variable in the computation. This can be attributed to the fact that the market clearly shows in which parts of Brno people are willing to pay more/less for the properties.

$$\text{district}_i = \frac{0.5 \cdot \text{sft}_i + \text{ppm}_i^2 + 0.5 \cdot \text{dst}_i}{2},$$

where sft_i denotes a *safety* index in each district, ppm_i^2 stands for the *average price per* $1\ m^2$ and dst_i indicates a *distance* (measured in km) from the centre of Brno for an observation i in our dataset D.

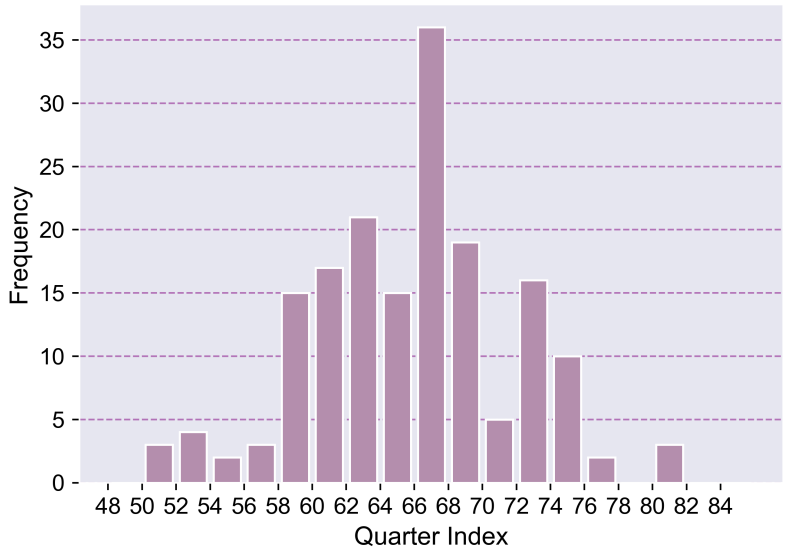

Figure 6.7: Histogram of District Index

[5] The term *distance from the centre of Brno* has been defined as a distance from the middle of Freedom Square to a given property.

6.2 Statistical Model

In order to make such a prediction, a statistical model must be introduced. Before the construction of the model itself, we shall first delve into the analysis of a response variable (variable *price*) given the independent variables. Then, a linear model, which portrays the estimation of prices, is defined. At the end of this subsection, we shall define the families as well as the parameters of distributions which will be used as priors in MCMC simulations.

6.2.1 Analysis of Variables

First of all, an analysis of the variables shall be performed. This implies that we will analyse every independent variable with a response variable (in our case – *price*), and the relationship between the two given variables will be determined.

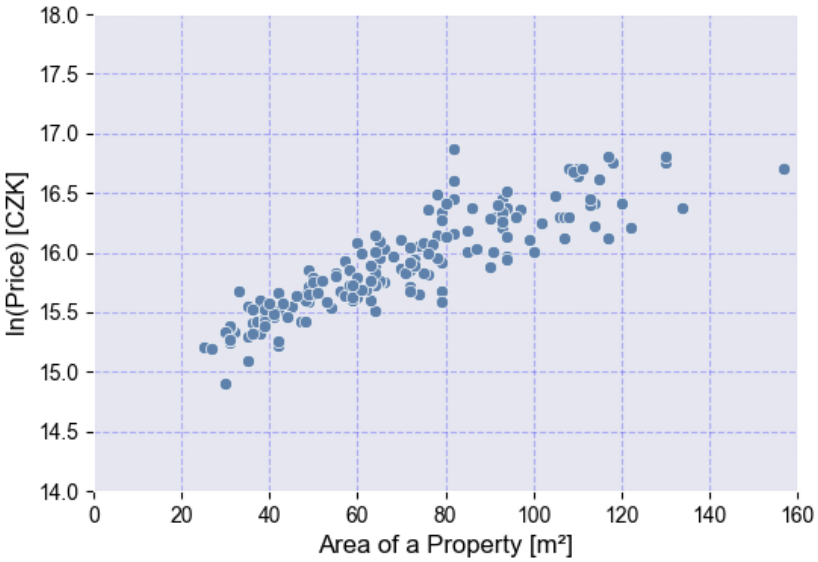

Figure 6.8: Scatter Plot: Price vs. Area

One can observe (from figure 6.8) a non-linear relationship. Two ways of describing this matter will be taken into consideration.

Firstly, an approximation with a second-degree polynomial ($y = ax^2 + bx + c$).

The second way to assess this matter is to utilise a logarithm (natural logarithm, to be precise) with a linear term ($y = a \ln x + bx + c$).

After simulations have been executed (an MCMC simulation with given parameters – three in total), we are in a position to evaluate the better approach (out of the two mentioned earlier). It should be noted that both models have a similar value of R^2.

The fundamental problem of the quadratic model is that for large values of *area* (around 190 to 200 m^2), the price decreases, which contradicts the fact that an increase in the *area* always increases (to some extent) the overall price of the property. This problem can be easily solved by introducing constraints since our dataset mainly contains flats with an area between 20 m^2 and 140 m^2.

In contrast, the main issue of the logarithmic model is that it is quite arduous to define the prior distribution. For this particular reason, we opt for a polynomial approach. This indicates

that with larger values of *area*, the effect on *price* has diminishing returns. A residual plot between the variable *price* and *area* is visualised below.

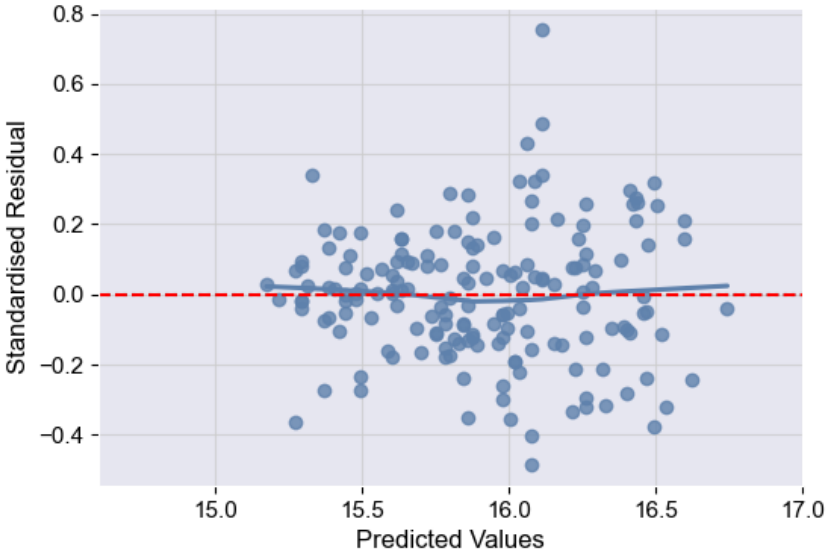

Figure 6.9: Residual Plot: Price vs. Area

Another variable in which the dependency must be found is the *age* of the property. One can anticipate that with the property being older, the *price* is going to decrease. When examining the figure 6.10, it is quite challenging to discover a pattern that the data follow. For this reason, a linear model ($y = ax + b$) is used.

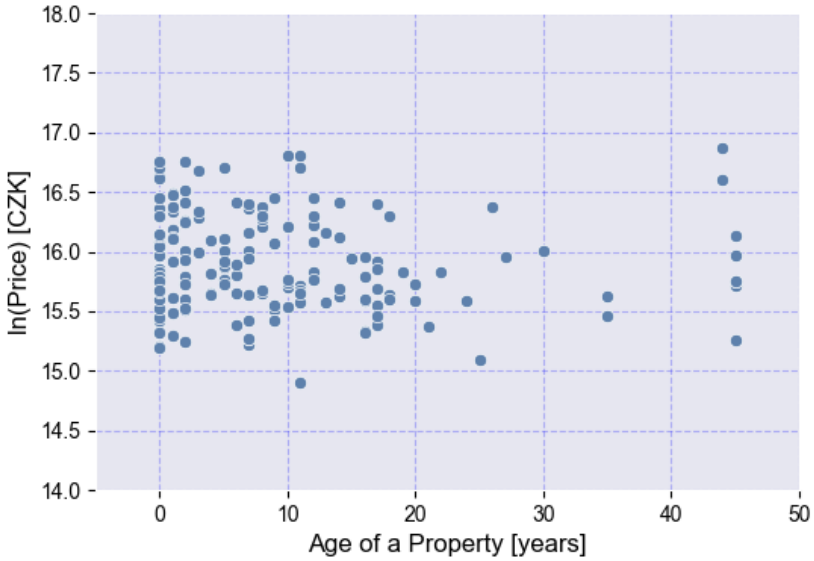

Figure 6.10: Scatter Plot: Price vs. Age

From the residual plot (figure 6.11), one can notice a slight non-linear behaviour (visualised by a blue line in the plot). Since addressing the relationship is quite a complex task, we shall conservatively stick to the linear model.

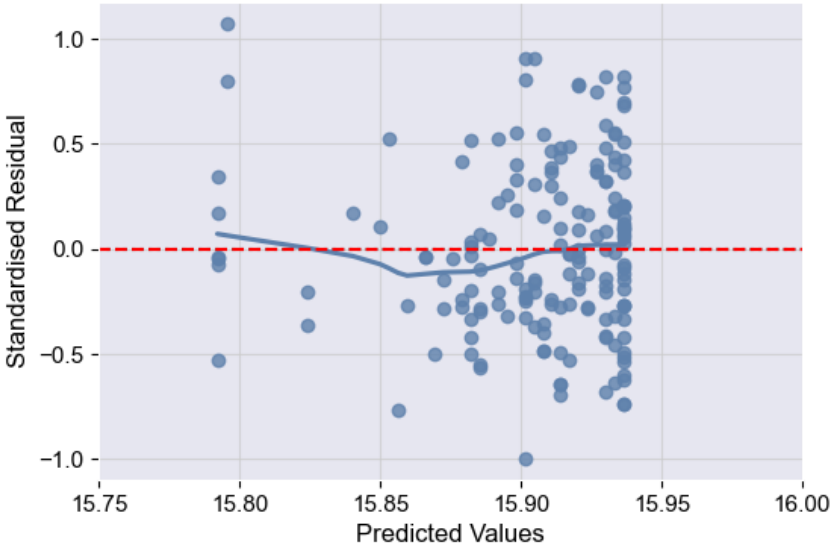

Figure 6.11: Residual Plot: Price vs. Age

Subsequently, the relationship between *price* and *district* is yet to be uncovered. Once again, one can predict an increase in the district index, as more appealing quarters (denoted by a higher value of the *district index*) are expected to have more expensive properties.

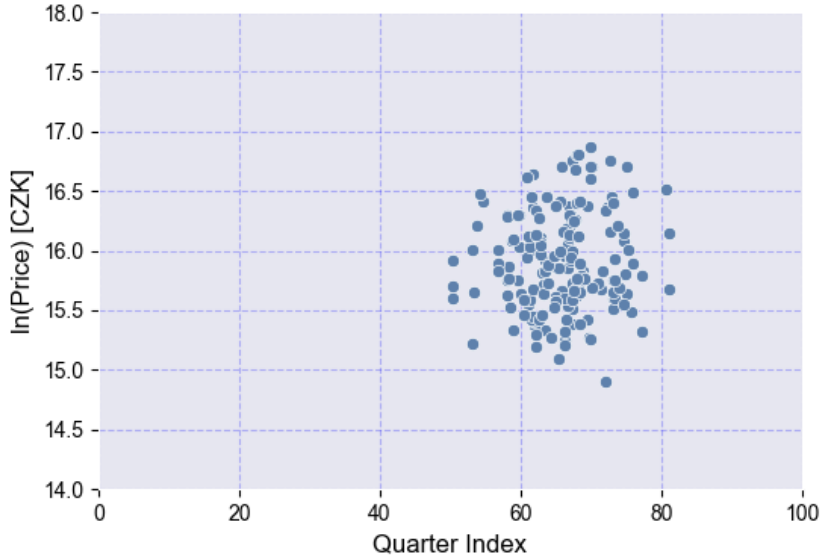

Figure 6.12: Scatter Plot: Price vs. District Index

From figure 6.12, one cannot see a clear pattern in the scatter plot. For this reason, the same strategy of utilising the linear model as before (when the *age* variable was discussed) is going to be established, therefore, $y = ax + b$.

The residual plot (figure 6.13) of variables *price* and *district* shows a good fit, which implies that we shall adhere to this particular model.

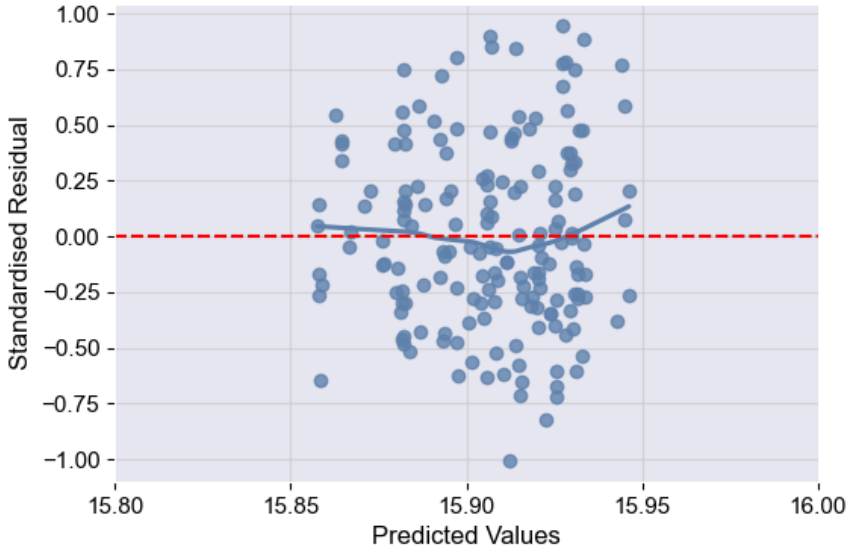

Figure 6.13: Residual Plot: Price vs. District Index

Finally, when speaking about both categorical variables *balcony* and *garage* we can simulate their influence on the *price* by adding a single parameter for both variables (since both variables assume values of either zero or one). Furthermore, based on the plots 6.14 and 6.15 both variables (*balcony* and *garage*) are projected to increase the *price* (which is in accordance with the consensus – flats with a balcony or with a garage space are generally more expensive than those that do not possess such amenities).

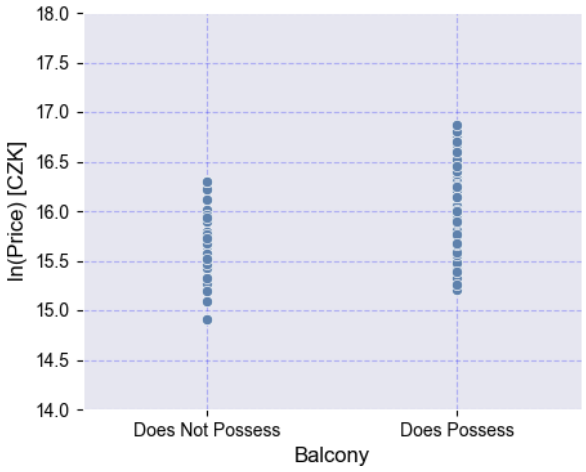

Figure 6.14: Scatter Plot: Price vs. Balcony

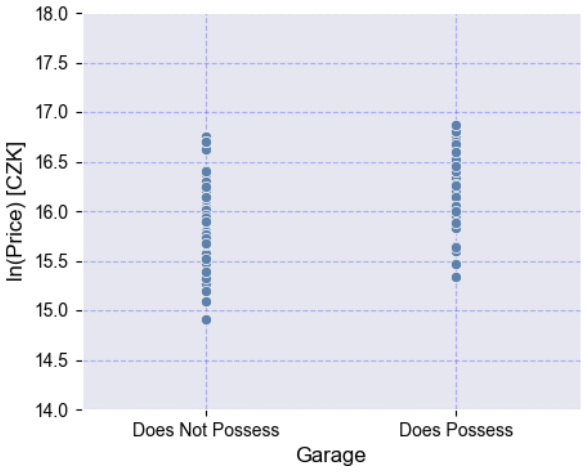

Figure 6.15: Scatter Plot: Price vs. Garage

To avoid inconveniences in future predictions, we shall introduce a set of constraints which is caused by the characteristics of our dataset.

Variable	Constraint
area	$[20, 140]$ m^2
age	$[0, 45]$ years
balcony	$\{0, 1\}$
garage	$\{0, 1\}$
district	$[0, 100]$ index

Table 6.1: The Constraints of Variables

6.2.2 Defining the Model

As we have discussed the relationship between each explanatory variable, we shall define our model for predicting property prices.

$$\text{price}_i = \alpha + \beta_0 \cdot \text{area}_i + \beta_1 \cdot \text{area}_i^2 + \beta_2 \cdot \text{age}_i + \beta_3 \cdot \text{balcony}_i + \beta_4 \cdot \text{garage}_i + \beta_5 \cdot \text{district}_i + \varepsilon_i, \tag{6.1}$$

where ε_i, $i \in D$ denotes residuals (error term, which is not explained by the model).

Therefore, one can derive the expected value (described in chapter 5) of variable *price* (from equation 6.1) given the vector of parameters $\boldsymbol{\beta} = (\alpha, \beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^T$ and the observed data $\boldsymbol{X}$:

$$\text{E}(\text{price}|\boldsymbol{\beta}, \boldsymbol{X}) = \alpha + \beta_0 \cdot \text{area} + \beta_1 \cdot \text{area}^2 + \beta_2 \cdot \text{age} + \beta_3 \cdot \text{balcony} + \beta_4 \cdot \text{garage} + \beta_5 \cdot \text{district}. \tag{6.2}$$

By the matrix of predictors $\boldsymbol{X}$, we are reffering to:

$$\boldsymbol{X} = \begin{bmatrix} 1 & \text{area}_1 & \text{area}_1^2 & \text{age}_1 & \text{balcony}_1 & \text{garage}_1 & \text{district}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \text{area}_N & \text{area}_N^2 & \text{age}_N & \text{balcony}_N & \text{garage}_N & \text{district}_N \end{bmatrix},$$

where the dimensions of the matrix $\boldsymbol{X}$ are (171×7), as the length of the dataset is 171.

6.2.3 Defining Prior Distributions

Since we want to predict a vector of parameters using Bayesian statistics and MCMC techniques (specified in chapters 2 and 4), the prior distribution must be defined in order to fulfil this task.

To achieve different outcomes from the simulation, various distributions shall be introduced. We will use unbounded as well as bounded distributions (bounded below or bounded below and above).

These distributions shall be used as priors:

Normal	$\mathcal{N}(\mu, \sigma^2),\ \sigma > 0$
Log-normal	$\mathcal{LN}(\mu, \sigma^2),\ \sigma > 0$
Uniform	$U(a, b),\ b > a$
Exponential	$\text{Exp}(\lambda),\ \lambda > 0$

Table 6.2: Used Prior Distributions

Now that we have sorted out which distribution to use, we have to determine the values for each distribution, as they shall be later used to compute the posterior distribution and define the initial positions in our MCMC simulation.

Variable	Change in Price
area	positive
age	negative
balcony	positive
garage	positive
district	positive

Table 6.3: The Impact on the Price

Table 6.3 depicts whether the positive increment of a given variable changes the final *price*. It should be clear that an increase in the *area* of a flat ought to increase the overall *price* of the flat when taking into consideration our constraints (described in table 6.1). Similarly, the *age* of the flat generally decreases the price in our observed data. Likewise, in this situation, restrictions will be applied, as our dataset only contains values from 0 to 45 years of age. In addition, the possession of a *balcony* or *garage* space is expected to greatly increase the price. Lastly, it is expected that the more appealing the *district*, the higher the prices.

As the column of variable *price* in our dataset has all its values logarithmically scaled (by natural logarithm), one can roughly predict the interval of each parameter in the vector $\boldsymbol{\beta} = (\alpha, \beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^T$ with relative ease.

For clarification, table 6.4 denotes which parameter corresponds to the variable since the possible intervals for each parameter shall be discussed.

Variable	Parameters
area	β_0, β_1
age	β_2
balcony	β_3
garage	β_4
district	β_5

Table 6.4: Variables and Their Parameters

We can expect the parameters β_0 (linear) and β_5 to be between 0 and 1 since a small change in the variable (for example, an increment of 1 m^2) does not change the order of magnitude of a *price*. Moreover, when discussing the feasible interval of the parameter β_1 (quadratic), one can anticipate that the value would be between 0 and -1, as can be seen in figure 6.8.

The exact same thought (as in the case of β_0) can be applied to the parameter β_2, although in this case, one can expect the change to be negative, and therefore the interval for β_2 would be $[-1, 0]$.

When discussing both categorical variables, which are tied to β_3 and β_4, one can anticipate much greater changes than in previous cases. Yet again, these changes only change the total price of a flat by tens of per cent, therefore, the same applies for β_0 (one can safely predict that $\beta_3 > \beta_0$ and $\beta_4 > \beta_0$).

To estimate the interval for the intercept α, it has been observed from the scatter plots that when factoring in only one of the independent variables, the value of the intercept is between 14 and 16. Since we want to conservatively predict the value of the intercept, we shall make the interval wider. The upper bound is set to 20, given that the most expensive property in our dataset costs less than 22 000 000 CZK, therefore $17 > \ln(22\,000\,000) \approx 16.91$.

When discussing the lower bound for the intercept, we have set the value to -20, as we do not possess any convincing knowledge of whether the value of the intercept α is positive or negative.

Parameter	Estimate
α	$[-20, 20]$
β_0	$[0, 1]$
β_1	$[-1, 0]$
β_2	$[-1, 0]$
β_3	$[0, 1]$
β_4	$[0, 1]$
β_5	$[0, 1]$

Table 6.5: The Estimation of the Parameters

In the end, we shall estimate the parameters for the chosen distributions (table 6.2), which will be proclaimed as priors in the simulation. The goal is to find such parameters which satisfy the condition that 95 % (or more, in case of uniform distribution) of data happens to be in intervals specified in table 6.5.

To determine the parameters for normal distribution, we will use an empirical rule (approximately 95 % of all data are distributed within 2 standard deviations from the mean). This means that for the intervals $[0, 1]$ and $[-1, 0]$, we shall find a midpoint of these intervals (also defined in table 6.5), which is $\mu = 0.5$ or $\mu = -0.5$. Utilising the empirical rule $\mu \pm 2\sigma$ will allocate 95 % of the data. This implies that $\sigma = 0.25$, therefore $\sigma^2 = \frac{1}{16}$. A similar approach is established when computing the parameters of the normal distribution for α. In this case, the mean is $\mu = 0$, so the standard deviation is $\sigma = 10$, thus $\sigma^2 = 100$.

Another straightforward assessment is when discussing the lower (a) and upper bound (b) in a uniform distribution. By extracting the intervals of the estimates from table 6.5, we shall declare them as our parameters of uniform distributions.

When addressing the parameters of the exponential distribution, we have to use its p.d.f. in order to derive the parameter λ such that 95 % of the probability density is located within a specific interval.

$$0.95 = \int_a^b \lambda \exp(-\lambda x)\, \mathrm{d}x.$$

By computing this simple integral (or simply by utilising the c.d.f. of exponential distribution), one can easily derive the parameter λ for all parameters from the vector $\boldsymbol{\beta}$.

The most arduous task is to determine the parameters μ and σ for the log-normal distribution. Once again, we will employ a p.d.f. of log-normal distribution such that:

$$0.95 = \int_a^b \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right) \mathrm{d}x.$$

Which can be further simplified to (with the aid of the c.d.f. of log-normal distribution)[6]:

$$0.95 = \Phi\left(\frac{\ln b - \mu}{\sigma}\right) - \Phi\left(\frac{\ln a - \mu}{\sigma}\right) \Longrightarrow 0.95 = \Phi\left(\frac{\ln b - \mu}{\sigma}\right).$$

To eliminate the parameter μ, we will perform a substitution with mode $\hat{x} = \exp(\mu - \sigma^2)$, which yields $\mu = \ln(\hat{x}) + \sigma^2$. Which can be altered to:

$$\sigma^2 + \sigma \cdot \Phi^{-1}(0.95) + \ln\left(\frac{\hat{x}}{b}\right) = 0. \tag{6.3}$$

[6]The parameter a is chosen so that $\Phi(\frac{\ln a - \mu}{\sigma}) = 0$. In the case of parameter α, this means that the whole distribution is shifted along the x-axis so that $\ln a \to -\infty$.

In order to find out the values for σ, we have to solve the quadratic equation 6.3. The mode for the parameter α has been set as $\hat{x} = 0$ (in new coordinates: $\hat{x} = 20$) and for all parameters β_i we opted for $\hat{x} = \frac{1}{3}$. Subsequently, the mean is computed using: $\mu = \ln(\hat{x}) + \sigma^2$.

Table 6.6 depicts the information for all priors, which shall be used later in the MCMC simulation.

Parameter	$\mathcal{N}(\mu, \sigma^2)$	$\mathcal{LN}(\mu, \sigma^2)$	$U(a, b)$	Exp(λ)
α	$(0, 100)$	$(3.116728, 0.347844^2)$	$(-20, 20)$	$\frac{\ln 20}{40}$
β_0	$(\frac{1}{2}, \frac{1}{16})$	$(-0.83865, 0.50986^2)$	$(0, 1)$	$\ln 20$
β_1	$(-\frac{1}{2}, \frac{1}{16})$	$(-0.83865, 0.50986^2)$	$(-1, 0)$	$\ln 20$
β_2	$(-\frac{1}{2}, \frac{1}{16})$	$(-0.83865, 0.50986^2)$	$(-1, 0)$	$\ln 20$
β_3	$(\frac{1}{2}, \frac{1}{16})$	$(-0.83865, 0.50986^2)$	$(0, 1)$	$\ln 20$
β_4	$(\frac{1}{2}, \frac{1}{16})$	$(-0.83865, 0.50986^2)$	$(0, 1)$	$\ln 20$
β_5	$(\frac{1}{2}, \frac{1}{16})$	$(-0.83865, 0.50986^2)$	$(0, 1)$	$\ln 20$

Table 6.6: Parameters of the Priors

It should be noted that there is an exception for the log-normal and the exponential distribution due to the fact that, in some cases, those distributions have to be transformed, the given parameters of the distributions persist, though. By transformations, we mean either translation along the x-axis (in the case of log-normal and exponential distributions for the parameter α) or redefining a domain from $[0, \infty)$ to $(-\infty, 0]$ (in both cases for parameters β_1 and β_2).

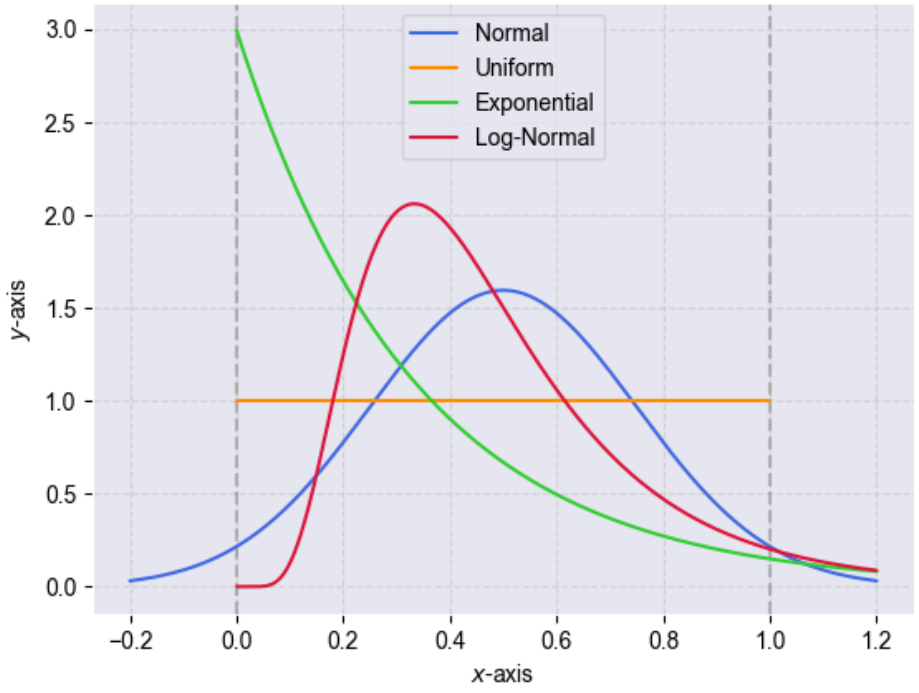

Figure 6.16: Used Priors for parameter β_0, β_3, β_4, β_5

7 Solution by Using MCMC Algorithm

This chapter summarises the principles of solving our problem, which have been defined earlier. The simulation is performed in programming language *Python* with the use of various libraries, most notably *PyMC*, which is widely used in Bayesian statistics and modelling.

In total, four types of simulation are considered. Firstly, a simulation which omits the burn-in stage is presented. Additionally, simulations with the burn-in stage are divided into three strategies. Impatient and conservative strategies are implemented to further compare the results of different fractions of samples that are discarded. In the end, an extensive sampling is conducted in order to obtain the most accurate solution.

The fundamental idea behind the strategies mentioned beforehand is to compare their results with an extensive sampling where it can be presumed that the answers are closest to the stationary distribution. For this reason, the number of samples (for strategies *without burn-in*, *impatient* as well as *conservative*) remains constant, only the percentage of discarded samples varies. In addition, the number of chains throughout those three strategies remains constant as well. According to [17], running at least four chains in simulations is recommended.

Furthermore, for each strategy, a diagnostic of whether the simulation converged sufficiently (determined with convergence statistics), as well as the values of predicted parameters, are presented in a table.

7.1 Burn-In Stage

The term burn-in signifies a procedure in which a portion of some initial iterations of an MCMC simulation is discarded. There is no recommended percentage of samples that ought to be removed. It is advised to observe the convergence statistics (namely $\hat{R}$ and ESS) when discussing the number of samples used and in the burn-in sequence. After the burn-in is executed, one proceeds normally with the sampling.

One has to bear in mind that this step is optional, and an MCMC simulation can be executed without a burn-in stage. [2]

7.2 Algorithm

As mentioned before (in chapter 6), in our simulation, various distributions shall be used, therefore, only the algorithm for normal distribution will be specified.

With the aid of [23], a sampling algorithm has been constructed.

Algorithm 2 Sampling – normal prior distribution

sampling(μ, σ, burn-in, samples, chains):
  Parameters $\boldsymbol{\beta} = (\alpha, \beta_0, \ldots, \beta_k)^T$ are established.
    $\beta_i \sim N(\mu_i, \sigma_i)$
  Expected value is computed according to our model.
    $\mu = \alpha + \beta_0 \cdot \text{area} + \beta_1 \cdot \text{area}^2 + \cdots + \beta_5 \cdot \text{district}$
  Likelihood of all observations is computed.
    likelihood = $\mathcal{L}(\mu, \text{observed data})$
  Metropolis-Hastings sampling is performed.
    trace = Metropolis(chains, tune, draws)
  Posterior predictive distribution is computed.
return trace

The input of this method consists of lists of μ and σ for every parameter in vector $\boldsymbol{\beta}$ as well as a number of samples that will be discarded (*burn-in*) and a number of *samples* that will be used to predict parameters $\alpha, \beta_0, \ldots, \beta_k$. In addition, an amount of parallel *chains* computed has also to be specified.

The method itself returns a *trace*, which can be interpreted as a collection of samples throughout the parameter space. Each sample holds information about the parameters $\alpha, \beta_0, \ldots, \beta_k$ and in total *trace* approximates the posterior distribution (target distribution).

Realisation with the *PyMC* library is effortless, thanks to the fact that all necessary methods have already been created. One particularly curious information about the implementation of Metropolis-Hastings in *PyMC* is that the variance of the distribution (in this case – the normal distribution) is dynamically changing. Meaning if the acceptance ratio (in chapter 4 denoted by $\rho(x, y)$) is too high, the proposals are chosen from a wider interval and vice versa (according to [23] the ideal acceptance ratio is between 20 % and 50 %).

Our algorithm also consists of a posterior predictive distribution, which is further used to determine future observations and their interval of confidence. In our case, this means that we will be able to predict an expected value of the price of a property, as well as its interval of confidence.

Finally, the whole concept of sampling is summarised in a pseudocode below:

Algorithm 3 MCMC Simulation

```
Simulation is performed for different priors:
    normal_sampling(parameters, burn-in, samples, chains)        ▷ Algorithm 2
    log_normal_sampling(parameters, burn-in, samples, chains)
    exponential_sampling(parameters, burn-in, samples, chains)
    uniform_sampling(parameters, burn-in, samples, chains)
for each sampling do
    Display parameters and convergence statistics,
    Compute R^2,
    Compute LOO.                                  ▷ Used in evaluation of results
end for
```

For clarification, the pseudocode for sampling (described earlier) is altered such that it would be able to make samples for multiple distributions (namely for log-normal, exponential and uniform distributions) because originally, it was described for normal distribution.

The input *parameters* signifies a *Python* dictionary, which stores parameters for each distribution. The rest (*burn-in*, *samples* and *chains*) are identical to the pseudocode before.

7.3 Sampling Without Burn-In Stage

The main idea of this simulation is to demonstrate that, in many cases, the absence of a burn-in stage contributes to less accurate results unless the starting point is not directly a realisation of the stationary distribution.

As in this scenario, a burn-in stage is omitted and, therefore, the value of *burn-in* is set to zero. For each chain (four in total), five hundred thousand samples will be generated.

A brief summary of chosen parameters (namely β_0, β_3 and β_5) of each prior distribution is displayed in the table below (a more detailed summary can be found in the appendix).

prior	parameter	mean	std	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Normal	β_0	−1.55409	1.12923	5.45	13.47	1.96053
	β_3	−0.32587	0.25281	97.06	527.6	1.02306
	β_5	0.27145	0.39501	6.55	62.29	1.62178
Log-Normal	β_0	0.09697	0.18509	6.36	32.92	1.66753
	β_3	0.15608	0.07849	21.91	75.15	1.11143
	β_5	0.04284	0.02706	5.84	20.16	1.81858
Exponential	β_0	0.08325	0.16150	6.03	19.10	1.74926
	β_3	0.08448	0.05546	889.2	166.1	1.03052
	β_5	0.04055	0.03165	6.48	16.44	1.64776
Uniform	β_0	0.07097	0.13745	7.56	57.44	1.46718
	β_3	0.07772	0.03937	61.57	97.77	1.03884
	β_5	0.02356	0.02831	7.27	27.64	1.50786

Table 7.1: Summary for simulation without burn-in stage

In this instance, the table contains values for *ESS*-bulk, which signifies the *Effective Sample Size* for values in the bulk of the distribution (related to the efficiency of mean), while *ESS*-tail denotes the *ESS* of samples for 5 % and 95 % quantiles. Tail-ESS is a useful measure for sampling efficiency in the tails of the distribution (related to the efficiency of variance) [18]. However, in both cases, we desire the value to be higher than 400.

From table 7.1 one, can observe that the desired thresholds ($ESS > 400$ and $\hat{R} < 1.01$) for convergence statistics have not been met in any category. Furthermore, the values for the mean of parameters vary greatly throughout the simulation.

This fact makes the simulation without the burn-in stage useless, as we cannot safely predict any parameters given the data (in table 7.1).

Lastly, the data provided from table 7.1 will be visualised as *trace* plots (for the chosen parameters). The *trace* plots depict the posterior distribution, which is a probability density function.

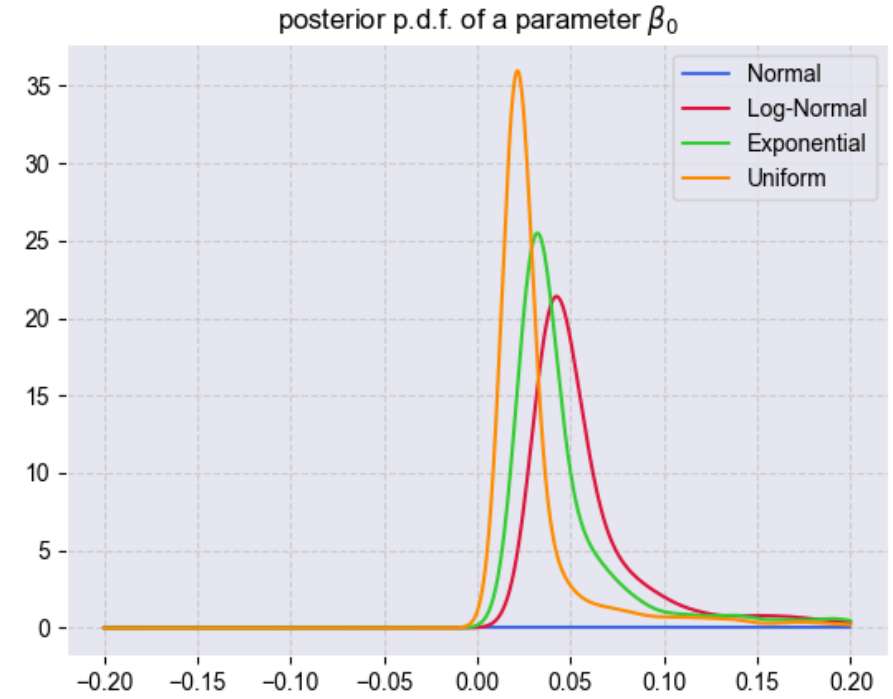

Figure 7.1: Parameter β_0 – without burn-in stage

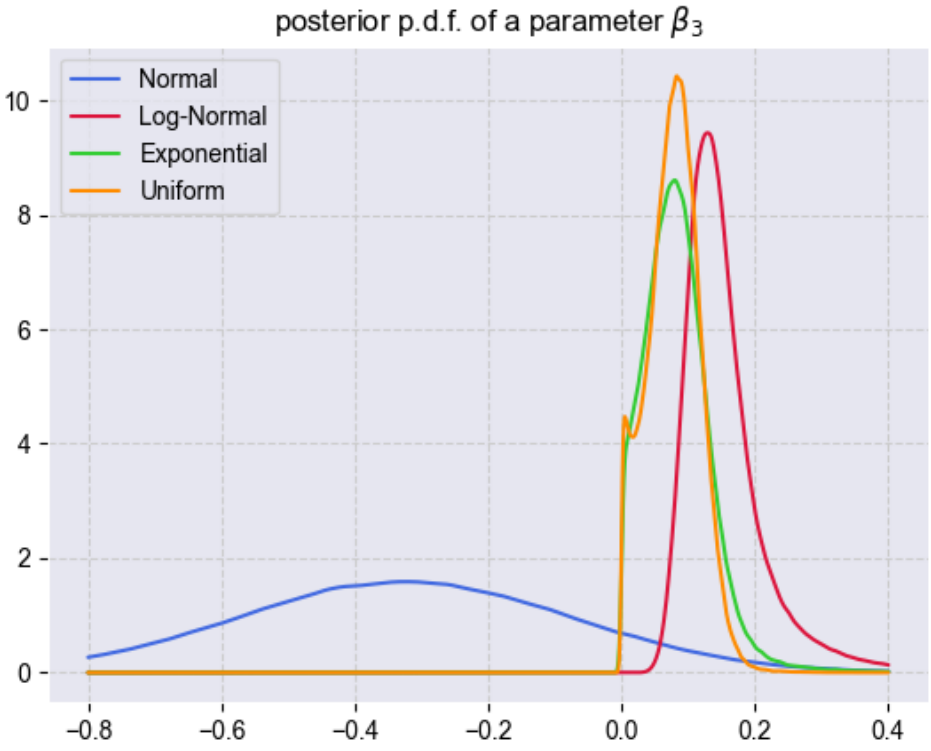

Figure 7.2: Parameter β_3 – without burn-in stage

One can observe that in the case of figures 7.1 and 7.2 for parameters β_0 and β_3 a contradiction occurs due to the fact that $\beta_0, \beta_3 < 0$ for normal distribution (according to our prior belief, it ought be $\beta_0, \beta_3 > 0$). This reveals an inappropriate simulation configuration.

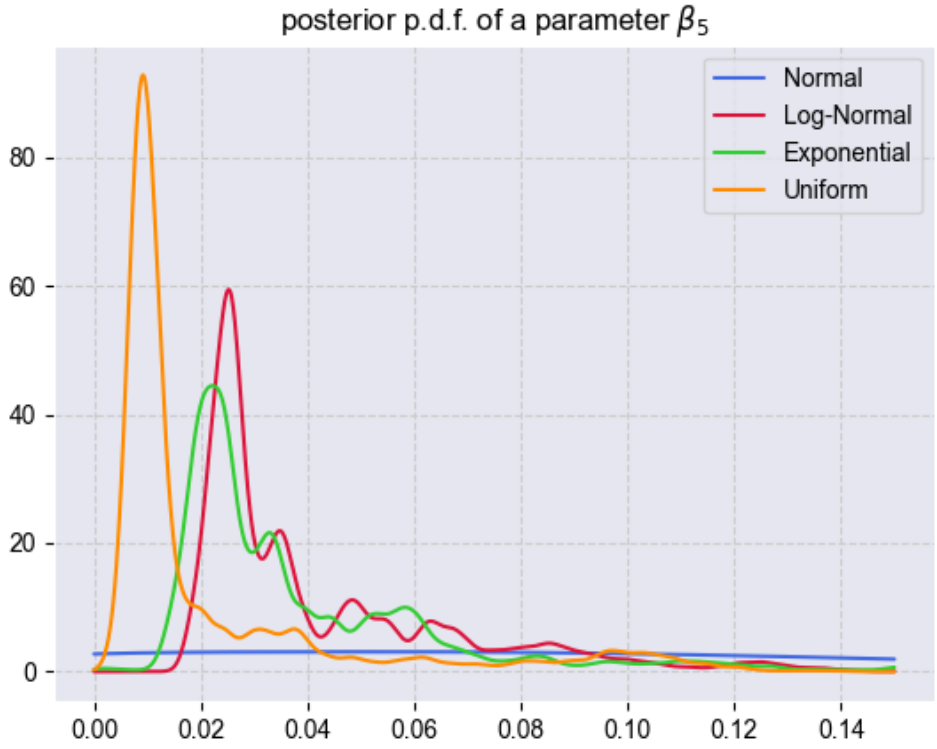

Figure 7.3: Parameter β_5 – without burn-in stage

When addressing *trace* plot 7.3, one can notice many irregularities in the traces, which may be caused by the absence of the burn-in stage. The traces contain initial samples, which do not reflect well our knowledge of the particular parameter. Those samples still explore the parameter space, and therefore, their value (at this point) varies from the stationary distribution that we are aiming for.

In all three cases, a variance between each prior information is notable (especially when comparing *normal* distribution to *log-normal* and the rest).

These problems can be solved by introducing a burn-in sequence or by prolonging the simulation itself.

7.4 Sampling With Burn-In Stage

In this instance, we shall focus on strategies where a burn-in stage is implemented. To evaluate the exact percentage of samples that will be discarded, it is necessary to introduce a trace plot of a simulation without the burn-in stage. By such observation of the diagnostics, two strategies will be utilised. Those strategies are "*impatient*", characterised by a lower percentage of eliminated samples, and "*conservative*" signified by a more cautious approach, therefore, the percentage shall be higher than in the case of "*impatient*" strategy.

In the end, an *extensive* simulation is executed to verify the credibility of both *impatient* and *conservative* strategies (whether a pseudo-convergence occurred). According to [2], it is recommended to run such a simulation.

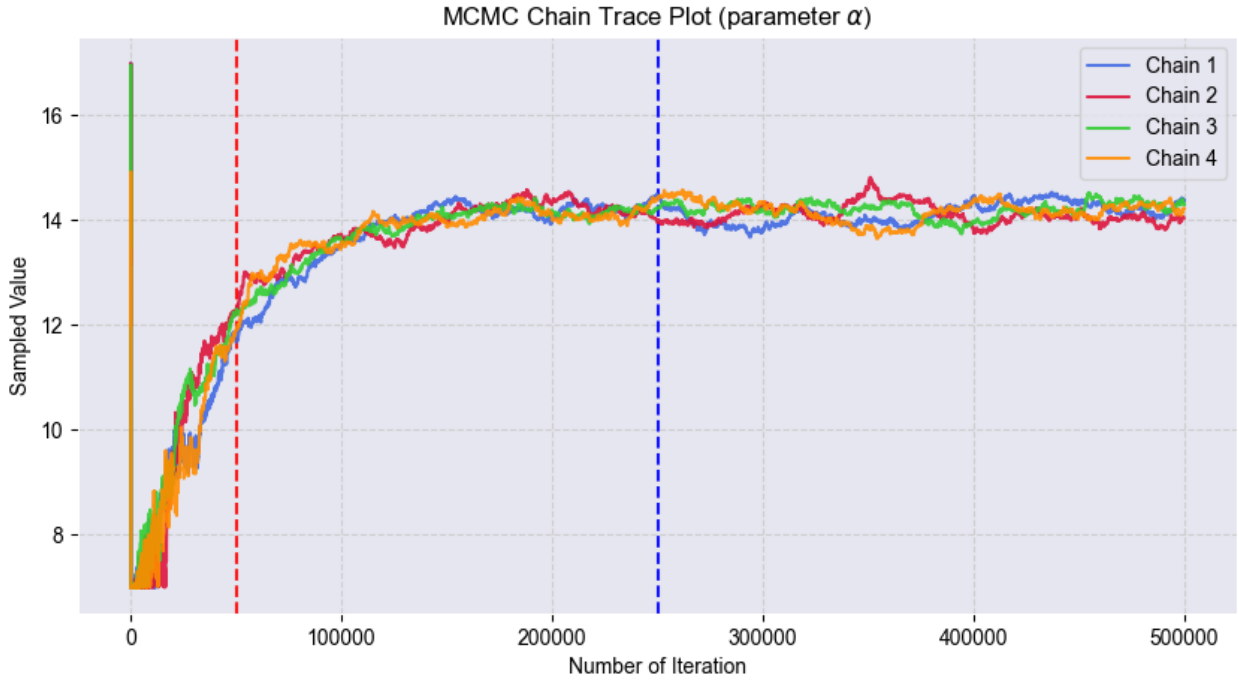

Figure 7.4: Trace plot for parameter α, prior: uniform

Based on diagnostics, it has been decided to select 10 % of samples as burn-in in the *impatient* strategy. When discussing the *conservative* strategy, this number has been selected to 50 %. In figures 7.4 and 7.5, the strategies are represented by dashed vertical lines (red – *impatient*, blue – *conservative*). The lines themselves denote the threshold between burn-in and samples that are used to compute the posterior distribution.

To put it differently, first $50,000$ (or $250,000$) of the samples are discarded, and then the simulation takes into consideration only the remaining $450,000$ ($250,000$) samples for *impatient* (*conservative*) strategy.

It should be noted that for both simulations (*conservative* and *impatient*), the overall total of samples is constant (set to $500,000$ – as in the case of simulation without the burn-in stage).

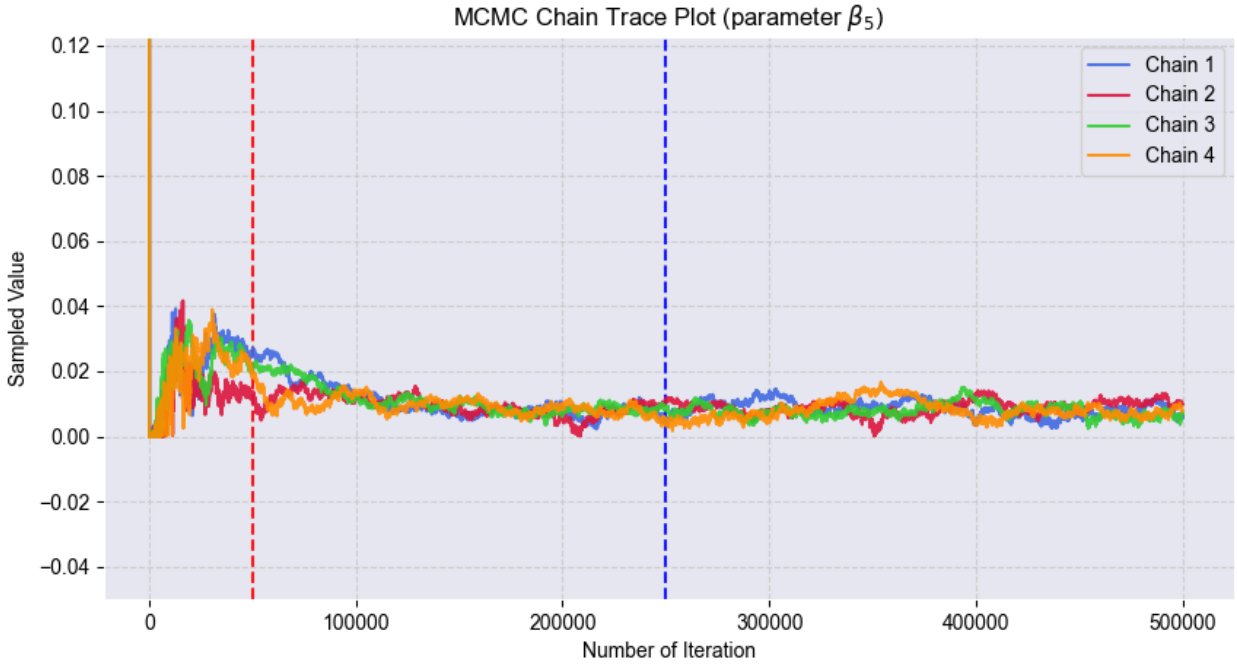

Figure 7.5: Trace plot for parameter β_5, prior: uniform

7.4.1 Impatient Strategy

As mentioned before, the so-called *impatient* strategy consists of 10 % samples that will be discarded. This implies that 50, 000 of the samples will no longer be used in the simulation, and the simulation shall proceed with 450, 000 samples throughout the four chains.

The table below summarises the same parameters (β_0, β_3 and β_5) as in the sampling without the burn-in sequence, whilst using different prior distributions.

prior	parameter	mean	std	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Normal	β_0	0.01987	0.00249	1048.1	2069.0	1.00538
	β_3	0.07644	0.03199	28460	97604	1.00024
	β_5	0.00799	0.00209	1379.8	3429.6	1.00193
Log-Normal	β_0	0.03244	0.00248	1553.5	2978.2	1.00082
	β_3	0.11662	0.02768	92412	154369	1.00009
	β_5	0.01559	0.00225	2037.9	3985.1	1.00270
Exponential	β_0	0.01892	0.00242	1067.0	2363.1	1.00334
	β_3	0.08853	0.03237	27254	76644	1.00013
	β_5	0.00834	0.00212	1493.1	3202.5	1.00189
Uniform	β_0	0.01973	0.00244	889.3	1840.8	1.00457
	β_3	0.08775	0.03180	26807	71222	1.00025
	β_5	0.00788	0.00211	1421.6	3069.6	1.00145

Table 7.2: Summary for *impatient* simulation

Under these circumstances, the results look promising according to convergence statistics. It should be noted that all values in the *ESS* columns obey our criterion: $ESS > 400$, the same applies to the $\hat{R}$, which satisfies: $\hat{R} < 1.01$.

However, an issue has evidently arisen when comparing *log-normal* prior distribution to the rest of the priors. Even though in the case of *log-normal* prior the convergence statistics of the results satisfy our set conditions, the values of all parameters (for a complete summary, visit appendix) differ significantly. This may be caused by overpowering information that is provided by a *log-normal* prior (the prior information is too restrictive), resulting in poor likelihood, thus providing faulty results.

The variance between *normal*, *exponential* and *uniform* priors are negligible (as can be seen in figures 7.6, 7.7 and 7.8). The exact precision of each model is discussed in chapter 8.

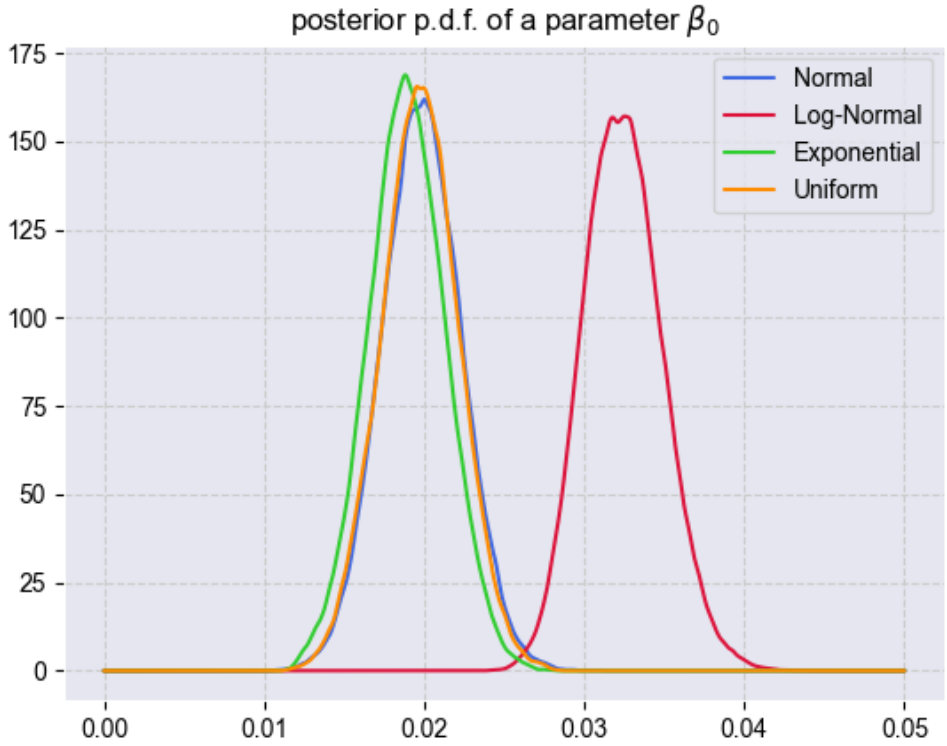

Figure 7.6: Parameter β_0 – *impatient* strategy

Yet again, the phenomenon of *log-normal* distribution providing different outputs can be observed in figures 7.7 and 7.8.

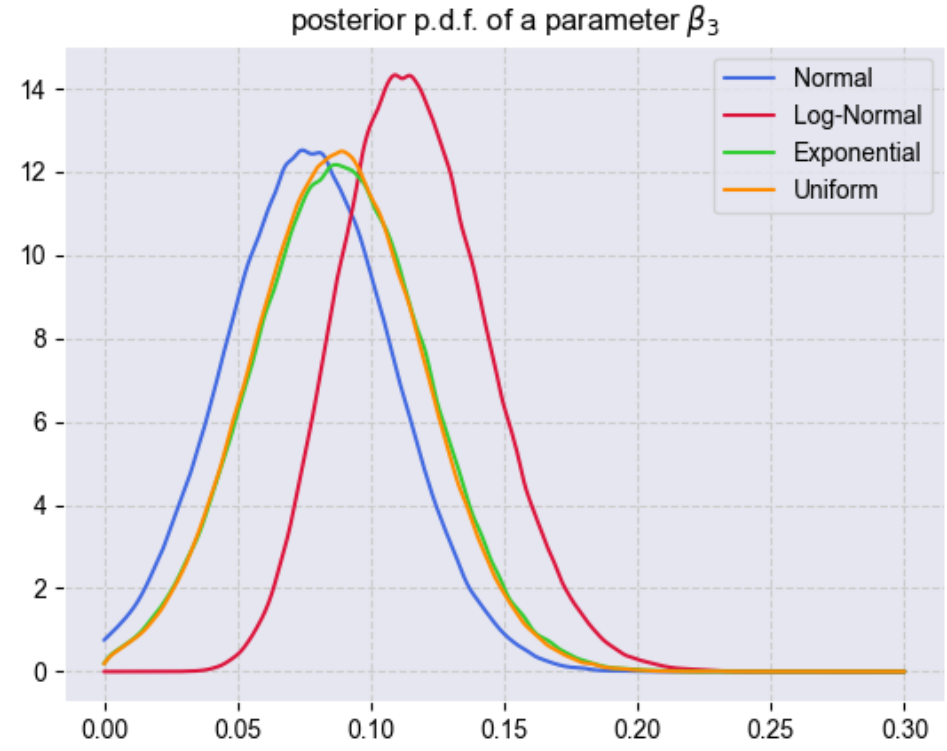

Figure 7.7: Parameter β_3 – *impatient* strategy

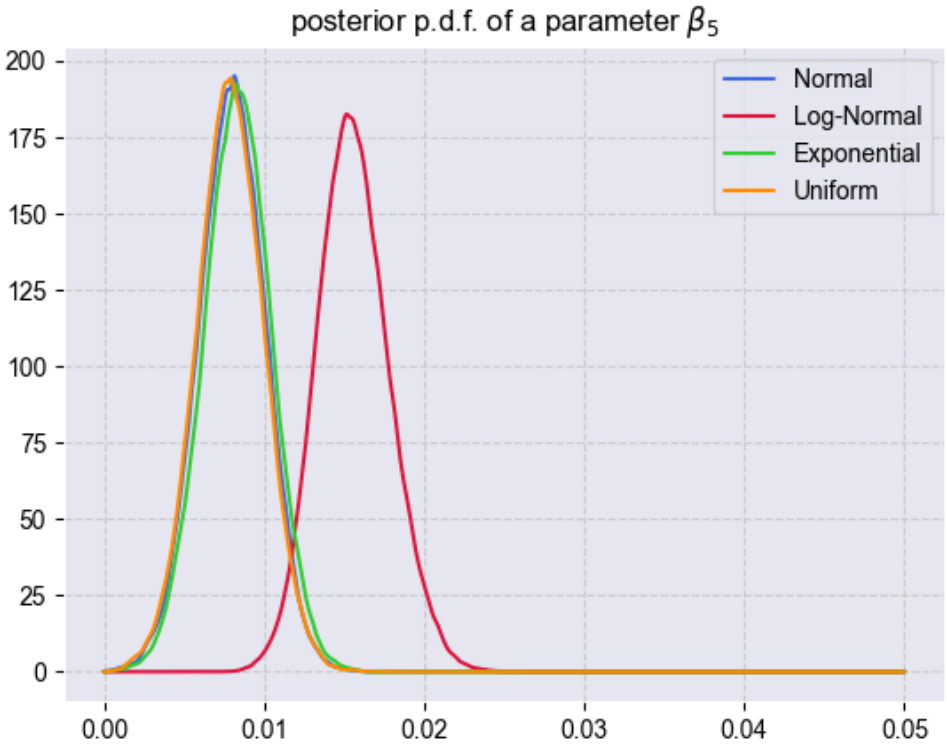

Figure 7.8: Parameter β_5 – *impatient* strategy

7.4.2 Conservative Strategy

Conservative strategy differs from *impatient* only in the percentage of samples dedicated to the burn-in stage. Through this procedure, the percentage is set to 50 %. This implies that four chains, as well as half a million samples for each chain (250, 000 samples are dedicated to the burn-in phase and 250, 000 of samples remain in the simulation), will be utilised.

Yet again a summary of parameters β_0, β_3 and β_5 is displayed.

prior	parameter	mean	std	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Normal	β_0	0.01963	0.00256	556.0	1068.8	1.00644
	β_3	0.07722	0.03199	15724	57339	1.00029
	β_5	0.00797	0.00214	828.9	1757.5	1.00349
Log-Normal	β_0	0.03238	0.00244	896.6	2022.7	1.00283
	β_3	0.11668	0.02771	51374	81805	1.00007
	β_5	0.01558	0.00222	984.7	2088.4	1.00366
Exponential	β_0	0.01900	0.00246	594.4	1456.3	1.00762
	β_3	0.08830	0.03242	14862	45200	1.00027
	β_5	0.00841	0.00217	803.3	1470.4	1.00840
Uniform	β_0	0.01968	0.00242	635.3	1455.9	1.00215
	β_3	0.08754	0.03167	19525	47401	1.00011
	β_5	0.00787	0.00201	799.6	1376.0	1.00286

Table 7.3: Summary for *conservative* simulation

One can recognise a resemblance with the answers from the *impatient* strategy. The values vary by only a slim margin. However, the issue with the *log-normal* prior persists. The cause of this occurrence is predicted to be the same as in the *impatient* approach.

Nonetheless, there seems to be a small distinction between *conservative* and *impatient* models as the values for *ESS* in most cases are lower for the *conservative* strategy. This is caused by the lower number of samples used to derive the posterior distribution.

Generally speaking, the drawback of the *conservative* strategy (compared to the *impatient* strategy) is that the statistic *ESS* is lower, therefore, more samples are needed for satisfactory results, which implies higher time complexity.

The main advantage of this approach is that, in theory, it should eliminate all infeasible samples. By infeasible samples, we mean samples that still explore the parameter space and have not yet converged to the target distribution.

Due to the fact that the results are almost the same (with respect to the *impatient* strategy), no trace plots will be shown.

7.4.3 Extensive Sampling

To ensure that the values from previous (shorter) simulations converged well, we shall implement an *extensive* simulation. By converging well, we refer to the fact that for all prior distributions, we obtain similar results. Moreover, by executing this simulation, we lower the chances of the simulation pseudo-converging.

Pseudo-convergence is a phenomenon in which a Markov chain can appear to have converged to its equilibrium distribution when it has not. This happens when parts of the state space are poorly connected by the Markov chain dynamics. By prolonging the simulation, we can reduce the chance that this will occur. [2]

Furthermore, an additional four chains will be added, making a total of eight chains to be computed.

Also, in terms of the percentage of samples dedicated to burn-in, a conservative approach has been adopted due to the fact in discards 50 % of samples, partially solving the problem with space complexity (to successfully execute the simulation needs to allocate lower tens of gigabytes). Therefore, the percentage has been set to 50 %. In addition, the sampling of each chain has been extended to four million samples, making this simulation ten times larger than the previous strategies in terms of the number of samples in a single chain. Note that the number of chains in this simulation is doubled compared to previous simulations (*without burn-in*, *impatient* and *conservative* strategies).

prior	parameter	mean	std	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Normal	β_0	0.01977	0.00245	9503.7	22258	1.00115
	β_3	0.07674	0.03193	227614	686379	1.00006
	β_5	0.00798	0.00212	12473	28693	1.00053
Log-Normal	β_0	0.03240	0.00247	13959	30751	1.00056
	β_3	0.11668	0.02768	757951	1288753	1.00001
	β_5	0.01558	0.00223	16592	33395	1.00071
Exponential	β_0	0.01890	0.00248	9480.3	20124	1.00057
	β_3	0.08861	0.03239	214430	606927	1.00003
	β_5	0.00834	0.00216	12425	26549	1.00057
Uniform	β_0	0.01968	0.00243	9101	20023	1.00094
	β_3	0.08771	0.03171	216002	634502	1.00006
	β_5	0.00786	0.00210	12772	26780	1.00051

Table 7.4: Summary for *extensive* simulation

When comparing the results from *extensive* sampling (table 7.4) with *impatient* (table 7.2) or *conservative* strategy (table 7.3) we can safely declare that all simulations (except the simulation *without burn-in*) provide reliable outputs, as a minimal variance between the strategies can be measured.

One can detect that in the case of *extensive* sampling, the demands on convergence statistics are fulfilled by a great margin. The values of *ESS* have to be larger than 400 for all values, and the $\hat{R}$ must be less than 1.01.

In this instance, the *trace* plot will not be displayed as the similarity with the result from *impatient* simulation is substantial. Also, it would be highly computationally intensive.

8 Evaluating Results

This chapter focuses on evaluating which prior distribution provides the best solution to our initial problem. We would like to choose the model which generalises the best unseen data. In order to answer such a question, a *LOO* statistic (Leave-One-Out) alongside R^2 for Bayesian models will be introduced. In conclusion, a prediction of three hypothetical properties (flats) in Brno will be performed.

8.1 Leave-One-Out

Leave-one-out cross-validation (LOO-CV) is a method for determining the general performance of a particular model. This is achieved by training the data on all observations except observation y_{-i} in a model.

Expected log predictive density (elpd) is used as a measure of the predictive performance of a model.

$$\text{elpd}_{\text{loo}} = \sum_{i=1}^{n} \log p_M(y_i|y_{-i}) = \sum_{i=1}^{n} \log \int p_M(y_i|\theta) p_M(\theta|y_{-i})\, \mathrm{d}\theta,$$

where $p_M(y_i|\theta)$ is the likelihood and $p_M(\theta|y_{-i})$ is the posterior distribution for θ without observation y_i of a model M. [11]

To further evaluate which model is the most reliable (in terms of statistic LOO-CV), we will only focus on *impatient*, *conservative* and *extensive* strategies as the results of the simulation *without burn-in* have not been satisfactory.

strategy	Normal	Log-normal	Exponential	Uniform
Impatient	62.96	27.84	61.07	63.16
Conservative	63.06	27.70	61.11	63.09
Extensive Sampling	63.03	27.98	61.03	63.19

Table 8.1: Summary of LOO-CV for all simulations

It can be observed that when the *log-normal* distribution is chosen as a prior, the model derived from this prior is the least suitable. Furthermore, *normal*, *uniform*, and *exponential* priors provide similar performance in predicting future data, where *normal* and *uniform* models perform the best (according to LOO-CV).

When comparing the models across the strategies, one can detect only negligible differences. It appears that the length of the simulation does not have any influence on the value of LOO-CV.

8.2 Computation of R^2

Another measure which shall be used in determining which model is the most accurate in answering our initial problem (prediction of property prices in Brno) is R^2.

Since the model works on the principles of Bayesian statistics, it is necessary to introduce R^2 for Bayesian regression models.

Instead of using point predictions $\hat{y}_n$, we shall use expected values conditional on the unknown parameters $\boldsymbol{\beta}$.

$$y_n^{\text{pred}} = E(\tilde{y}_n|\boldsymbol{\beta}, \boldsymbol{X_n}),$$

where $\tilde{y}_n$ denotes future observations from the model with predictors $\boldsymbol{X_n}$. Then R^2 is defined by the following expression:

$$R^2 = \frac{\text{var}(y_n^{\text{pred}})}{\text{var}(y_n^{\text{pred}}) + \text{var}(\varepsilon)}, \quad \text{where}$$

$\text{var}(y_n^{\text{pred}}) = \frac{1}{N-1}\sum_{n=1}^{N}(y_n^{\text{pred}} - \overline{y^{\text{pred}}})^2$ is the variance of the modelled predictive mean and $\text{var}(\varepsilon) = \frac{1}{N-1}\sum_{n=1}^{N}(\varepsilon_n)^2$ is the modelled residual variance (residuals: $\varepsilon_n = \tilde{y} - y_n^{\text{pred}}$). [7]

By running every feasible simulation (simulation *without burn-in* is not considered), one can determine R^2 for all suitable settings (different priors and simulation strategies).

strategy	Normal	Log-normal	Exponential	Uniform
Impatient	0.85059	0.80759	0.84275	0.85061
Conservative	0.85051	0.80746	0.84269	0.85052
Extensive Sampling	0.85049	0.80769	0.84257	0.85045

Table 8.2: Summary of R^2 for all simulations

It should be addressed that no contradiction occurs between the comparison of statistics *LOO* and R^2 since similar information about the models can be derived from tables 8.2 and 8.1.

Once again, we obtained the information that the model in which *log-normal* prior has been used provides the weakest fit for the data, as the value of R^2 is the lowest of all the proposed priors. Yet again, *normal* and *uniform* models both seem to be the most reliable, as the value of R^2 (in both cases) surpassed 0.85.

Lastly, the length of the simulation does not appear to influence the final value of R^2 in any way.

8.3 Prediction of Prices

This chapter shall focus on a prediction of the prices of hypothetical flats located in Brno. The prediction will involve point estimation as well as confidence interval estimation. To resolve this matter, a posterior predictive distribution will be used.

In this scenario, three different properties shall be considered. We opt for specific parameters (of properties) such that they cover as much parameter space as possible (new, old, large, without balcony, etc.) whilst taking into consideration our constraints (described in chapter 6.2.1).

The first property (property A) is characterised by a large area. It possesses a balcony, and it is situated in a reasonably attractive neighbourhood (namely Brno-Veveří or Brno-Bystrc).

Then, property B is a medium-sized brand-new flat located in a lucrative district (specifically Brno-Pisárky) while having a balcony as well as a garage space.

Lastly, property C is on the cheaper side (small area, the oldest of all). Therefore, it does appear to have neither a balcony nor a garage space. This tiny flat would hypothetically be positioned in not-so-appealing districts of Brno (for example, Brno-Chrlice).

Properties	Area[m^2]	Age[years]	Balcony	Garage	District[idx]
Property A	100	10	yes	no	65
Property B	65	0	yes	yes	82
Property C	25	35	no	no	55

Table 8.3: Parameters of hypothetical properties in Brno

Provided information on the posterior performance of the models, only *normal* and *uniform* models shall be analysed, since they both achieved the best fit for the problem (according to *LOO* and R^2).

To further evaluate the *price* of hypothetically chosen properties, an *extensive* sampling has been used, as it should be the most accurate simulation in theory (in reality, all simulations performed similar posterior performance). The output of the *posterior predictive distribution* given the data from the hypothetical properties is listed below.

Property	Prior	Mean[CZK]	Confidence Interval[CZK]
Property A	Normal	11, 445, 700	[8, 270, 560; 15, 794, 130]
	Uniform	11, 493, 000	[8, 315, 950; 15, 875, 570]
Property B	Normal	10, 478, 500	[7, 522, 610; 14, 617, 080]
	Uniform	10, 429, 800	[7, 488, 300; 14, 541, 240]
Property C	Normal	3, 229, 500	[2, 303, 820; 4, 536, 090]
	Uniform	3, 216, 400	[2, 289, 540; 4, 504, 450]

Table 8.4: Prediction of the Prices

Note that the results (in table 8.4) have to be transformed using price = exp(log_price), since the dataset has been logarithmically scaled, therefore the output from the simulation is the log of the actual price.

9 Conclusion

In this thesis, we have studied the techniques of MCMC simulation. It involves chapters from statistical inference (more thoroughly described in [3] and [6]), introduction into Markov chains (inspiration from [16], [15] and [10]), basic algorithms of Markov chain Monte Carlo (mainly used [2]) as well as Bayesian linear models (more can be found in [6]). Chapters 2 to 5, which contain the described theory, are necessary for comprehending the core of the simulations.

Afterwards, the problem of estimating the prices of properties in Brno is introduced, thus, chapter 6 describes the relationship between dependent and independent variables together with the creation of the linear model used in the upcoming simulations. Additionally, the chapter also includes a discussion of the parameters of the chosen priors. We opted for normal, log-normal, exponential and uniform distribution as priors to achieve variety, therefore, different outcomes from the simulations are anticipated.

Chapter 7 addresses the core of the simulation, which is performed by the programming language *Python* with the aid of library *PyMC*. Different sampling strategies are suggested, specifically the strategy *without the burn-in stage*, *impatient* and *conservative* strategy, each offering a different ratio of discarded samples. These procedures share identical lengths. The credibility of these strategies is proved by an *extensive* simulation (16 times longer than other strategies), which is used as a reference.

Finally, chapter 8 provides a comparison between the prior distribution. The main objective of this chapter is to choose the most suitable prior distribution which provides the best fit for our initial problem. This is achieved by introducing R^2 for Bayesian models and *LOO-CV* (Leave-One-Out Cross-Validation), which both measure the effectiveness of posterior prediction. From this comparison, *normal* and *uniform* proved as the most feasible while *log-normal* is the least suitable prior distribution from our model.

In the end, a prediction of future prices (of hypothetical properties) is conducted. This is achieved by a posterior predictive distribution, which we have implemented in our algorithm (more information is provided in chapter 7). Since both *normal* and *uniform* distributions, when chosen as prior, achieved almost identical accuracy, we shall use both in our price estimation. When discussing which strategy to utilise for the prediction, we opted for *extensive* sampling, which should be the most reliable according to theory [2] but also according to convergence statistics (more information can be found in chapter 7.4). The price estimates (point as well as interval) for the specified properties are listed in table 8.4.

References

[1] BOLSTAD, W., CURRAN, J. *Introduction to Bayesian Statistics*, 3rd ed. Wiley, 2016. ISBN 978-1-118-09315-8.

[2] BROOKS, S., GELMAN, A., JONES, G., MENG, X.-L. *Handbook of Markov Chain Monte Carlo*, 1st ed. CRC Press, 2011. ISBN 978-1-4200-7942-5.

[3] DEGROOT, M. H., SCHERVISH, M. J. *Probability and Statistics*. 4th ed. Pearson Education, Inc., 2012. ISBN 978-0-321-50046-5.

[4] DUKIC, V. "Review: Likelihood", *Bayesian Statistics and Computing*. Spring 2020, [online]. University of Colorado-Boulder, Boulder, CO. Available on: https://amath.colorado.edu/faculty/vdukic/Bayes/likprinc_lect2.pdf

[5] FAČEVICOVÁ, K., HRON., K., KUNDEROVÁ, P. *Markovovy řetězce a jejich aplikace*. 2nd ed., Univerzita Palackého v Olomouci, 2018. ISBN 978-80-244-5432-0.

[6] GELMAN, A., CARLIN, J. B., et al. *Bayesian Data Analysis*. 3rd ed., 2021, [online]. Available on: https://sites.stat.columbia.edu/gelman/book/BDA3.pdf

[7] GELMAN, A., GOODRICH, B., GABRY, J., VEHTARI, A. *R-squared for Bayesian Regression Models*. The American Statistician. 2018. https://doi.org/10.1080/00031305.2018.1549100

[8] GEYER, C. J. *Markov Chain Monte Carlo Lecture Notes*. Autumn 2005, [online]. University of Minnesota. Available on: https://www.stat.umn.edu/geyer/f05/8931/n1998.pdf

[9] LATUSZYŃSKI, K., NIEMIRO, W. *MCMC Notes*. Spring 2008.

[10] LI, Q. *Constructive Computation in Stachastic Models with Applications*. 1st ed., Tsinghua University Press, Beijing and Springer, 2010. ISBN 978-7-302-21501-1

[11] MAGNUSSON, M., ANDERSEN, M., JONASSON, J., VEHTARI, A. *Leave-One-Out Cross-Validation for Bayesian Model Comparison in Large Data*. 2020. 10.48550/arXiv.2001.00980.

[12] MOINS, T., ARBEL, J., DUTFOY, A., GIRARD, S. *On the Use of a Local $\hat{R}$ to Improve MCMC Convergence Diagnostic*. Bayesian Anal. 20 (1) 131 - 156, March 2025. https://doi.org/10.1214/23-BA1399

[13] ROBERT, C. P., CASELLA, G. *Introducing Monte Carlo Methods with R*. 1st ed. Springer, 2010. ISBN 978-1-4419-1575-7.

[14] ROBERTS, G. O., ROSENTHAL, J. S. *General state space Markov chains and MCMC algorithms*. Probability Surveys, 2004. Available on: https://doi.org/10.1214/154957804100000024

[15] SERFOZO, R. *Basics of Applied Stochastic Processes*. Springer-Verlag Berlin Heidelberg, 2009. ISBN 978-3-540-89331-8.

[16] TOLVER, A. *An Introduction to Markov Chains*. 1st ed., University of Copenhagen, 2016. ISBN 978-87-7078-952-3.

[17] VEHTARI, A., GELMAN, A., SIMPSON, D., CARPENTER, B., BÜRKNER, P. *Rank-Normalization, Folding, and Localization: An Improved $\hat{R}$ for Assessing Convergence of MCMC (with Discussion)*. Bayesian Anal. 16 (2) 667 - 718, June 2021. https://doi.org/10.1214/20-BA1221

[18] *Convergence and efficiency diagnostics for Markov Chains*. Online. Available at: https://mc-stan.org/rstan/reference/Rhat.html. [accessed 2025-05-12].

[19] *Data Brno: Inhabitants*. Online. Available at: https://data.brno.cz/pages/populace-brno-amc. [accessed 2025-03-01].

[20] *Data Brno: Property prices in Brno*. Online. Available at: https://data.brno.cz/pages/ceny-nemovitosti. [accessed 2025-03-01].

[21] *Google Maps*. Online. Available at: https://www.google.com/maps/. [accessed 2025-03-01].

[22] *Kriminalita Policie ČR*. Online. Available at: https://kriminalita.policie.gov.cz/. [accessed 2025-03-01].

[23] *PyMC*. Online. Available at: https://www.pymc.io/welcome.html. [accessed 2025-03-01].

[24] *Sreality*. Online. Available at: https://www.sreality.cz/. [accessed 2025-03-01].

Appendix

Without Burn-In Stage

	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Normal	α	47.15748	9.33081	33.27136	67.04255	9.15	75.30	1.34910
	β_0	−1.55409	1.12923	−3.92795	−0.59025	5.45	13.47	1.96053
	β_1	0.01059	0.01739	0.00404	0.02660	5.44	12.75	1.96805
	β_2	0.06239	0.08014	−0.05245	0.21548	29.41	77.21	1.08054
	β_3	−0.32587	0.25281	−0.81791	0.16626	97.06	527.61	1.02306
	β_4	0.62757	0.24704	0.14759	1.11273	333.60	2449.5	1.00767
	β_5	0.27145	0.39501	−0.08855	1.07346	6.55	62.29	1.62178
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Log-Normal	α	9.16866	5.74792	−4.69684	12.94079	5.81	11.92	1.82666
	β_0	0.09697	0.18509	0.03013	0.35436	6.36	32.92	1.66753
	β_1	−0.00056	0.00178	−0.00215	−0.00012	6.39	36.03	1.65984
	β_2	0.01156	0.01533	0.00271	0.03671	7.36	62.73	1.49264
	β_3	0.15608	0.07849	0.05947	0.28811	21.91	75.15	1.11143
	β_4	0.19549	0.07608	0.08734	0.32157	60.11	74.80	1.03912
	β_5	0.04284	0.02706	0.01720	0.09674	5.84	20.16	1.81858
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Exponential	α	9.80378	6.09077	−3.45711	13.60831	5.78	11.91	1.83153
	β_0	0.08325	0.16150	0.01972	0.31807	6.03	19.10	1.74926
	β_1	−0.00045	0.00130	−0.00191	−0.00005	6.09	21.24	1.73229
	β_2	0.00180	0.00603	0.00000	0.00703	17.44	65.60	1.14468
	β_3	0.08448	0.05546	0.00001	0.16730	889.19	166.13	1.03052
	β_4	0.13802	0.06663	0.00000	0.22388	40.78	68.07	1.07375
	β_5	0.04055	0.03165	0.01256	0.10895	6.48	16.44	1.64776
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Uniform	α	11.39741	6.27408	−4.84891	14.60849	6.62	17.99	1.60376
	β_0	0.07097	0.13745	0.01239	0.39519	7.56	57.44	1.46718
	β_1	−0.00037	0.00164	−0.00238	−0.00000	7.67	60.36	1.45497
	β_2	−0.00463	0.00683	−0.01595	−0.00000	85.00	63.30	1.05149
	β_3	0.07772	0.03937	0.00000	0.13986	61.57	97.77	1.03884
	β_4	0.13214	0.05333	0.00000	0.20156	32.51	71.63	1.07645
	β_5	0.02356	0.02831	0.00285	0.09781	7.27	27.64	1.50786

Table 9.1: Summary for simulation without burn-in stage

With Burn-In Stage

Impatient Strategy

	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Normal	α	14.18741	0.16955	13.85637	14.51855	934.67	2331.7	1.00436
	β_0	0.01987	0.00249	0.01498	0.02476	1048.1	2069.0	1.00538
	β_1	−0.00005	0.00002	−0.00008	−0.00002	1080.8	2271.1	1.00531
	β_2	−0.00305	0.00118	−0.00535	−0.00074	61435	185784	1.00006
	β_3	0.07644	0.03199	0.01364	0.13938	28460	97604	1.00024
	β_4	0.16001	0.03099	0.09867	0.22030	32862	131877	1.00014
	β_5	0.00799	0.00209	0.00391	0.01209	1379.8	3429.6	1.00193
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Log-Normal	α	13.16633	0.18773	12.78009	13.52095	1433.8	2464.5	1.00263
	β_0	0.03244	0.00248	0.02773	0.03738	1553.4	2978.2	1.00082
	β_1	−0.00013	0.00002	−0.00016	−0.00010	1624.9	3189.9	1.00073
	β_2	0.00456	0.00088	0.00291	0.00634	36529	104592	1.00013
	β_3	0.11662	0.02768	0.06491	0.17177	92412	154369	1.00009
	β_4	0.17191	0.03250	0.10887	0.23578	89894	239866	1.00002
	β_5	0.01559	0.00225	0.01132	0.02019	2037.9	3985.1	1.00270
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Exponential	α	14.15784	0.16974	13.82716	14.49524	1048.3	2365.3	1.00341
	β_0	0.01892	0.00242	0.01407	0.02365	1067.0	2363.1	1.00334
	β_1	−0.00004	0.00002	−0.00007	−0.00001	1093.2	2522.7	1.00339
	β_2	0.00038	0.00036	0.00000	0.00109	280297	336033	1.00002
	β_3	0.08853	0.03237	0.02487	0.15181	27254	76644	1.00013
	β_4	0.16331	0.03163	0.10204	0.22620	38691	148405	1.00008
	β_5	0.00834	0.00212	0.00413	0.01247	1493.1	3202.5	1.00189
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Uniform	α	14.19309	0.16767	13.86489	14.52572	983.77	2331.1	1.00347
	β_0	0.01973	0.00244	0.01491	0.02449	889.26	1840.8	1.00457
	β_1	−0.00005	0.00002	−0.00008	−0.00002	914.10	1943.5	1.00453
	β_2	−0.00307	0.00115	−0.00532	−0.00081	59352	161823	1.00009
	β_3	0.08775	0.03180	0.02525	0.15012	26807	71222	1.00025
	β_4	0.15281	0.03112	0.09074	0.21292	31082	136338	1.00011
	β_5	0.00788	0.00211	0.00366	0.01198	1421.6	3069.6	1.00145

Table 9.2: Summary for impatient simulation

Conservative Strategy

	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Normal	α	14.19595	0.17469	13.84798	14.53543	601.02	1137.66	1.00449
	β_0	0.01963	0.00256	0.01451	0.02460	555.97	1068.8	1.00644
	β_1	−0.00005	0.00002	−0.00008	−0.00002	571.71	1119.6	1.00621
	β_2	−0.00304	0.00118	−0.00535	−0.00074	34179	113039	1.00013
	β_3	0.07722	0.03199	0.01348	0.13892	15724	57339	1.00029
	β_4	0.16048	0.03098	0.10041	0.22200	17855	84234	1.00031
	β_5	0.00797	0.00214	0.00383	0.01219	828.87	1757.5	1.00349
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Log-Normal	α	13.16941	0.18488	12.80644	13.53258	748.88	1616.4	1.00330
	β_0	0.03238	0.00244	0.02774	0.03723	896.56	2022.7	1.00283
	β_1	−0.00013	0.00002	−0.00016	−0.00010	940.91	2193.4	1.00274
	β_2	0.00456	0.00088	0.00291	0.00632	21956	76539	1.00009
	β_3	0.11668	0.02771	0.06411	0.17155	51374	81805	1.00007
	β_4	0.17203	0.03250	0.10825	0.23545	54468	149997	1.00008
	β_5	0.01558	0.00222	0.01125	0.01994	984.68	2088.4	1.00366
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Exponential	α	14.15071	0.17274	13.79955	14.48623	625.29	1063.2	1.01141
	β_0	0.01900	0.00246	0.01432	0.02381	594.40	1456.3	1.00762
	β_1	−0.00004	0.00002	−0.00007	−0.00001	612.03	1529.5	1.00733
	β_2	0.00038	0.00036	0.00000	0.00109	155732	189035	1.00002
	β_3	0.08830	0.03242	0.02353	0.15129	14862	45200	1.00027
	β_4	0.16341	0.03158	0.10120	0.22516	19273	93633	1.00023
	β_5	0.00841	0.00217	0.00405	0.01270	803.28	1470.4	1.00840
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Uniform	α	14.19607	0.16330	13.87371	14.51986	557.72	953.35	1.00540
	β_0	0.01968	0.00242	0.01505	0.02448	635.35	1455.9	1.00215
	β_1	−0.00005	0.00002	−0.00008	−0.00002	657.24	1510.9	1.00199
	β_2	−0.00307	0.00115	−0.00531	−0.00079	36806	95262	1.00006
	β_3	0.08754	0.03167	0.02455	0.14891	19525	47401	1.00011
	β_4	0.15304	0.03114	0.09179	0.21392	18700	86246	1.00009
	β_5	0.00787	0.00201	0.00397	0.01194	799.57	1376.0	1.00286

Table 9.3: Summary for conservative simulation

Extensive Strategy

	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Normal	α	14.19093	0.17050	13.85685	14.52690	9042.7	20491	1.00106
	β_0	0.01977	0.00245	0.01497	0.02461	9503.7	22258	1.00115
	β_1	−0.00005	0.00002	−0.00008	−0.00002	9810.6	23333	1.00110
	β_2	−0.00304	0.00117	−0.00534	−0.00073	479895	1464935	1.00002
	β_3	0.07674	0.03193	0.01389	0.13915	227614	686379	1.00006
	β_4	0.16023	0.03101	0.09952	0.22123	270949	976862	1.00003
	β_5	0.00798	0.00212	0.00382	0.01216	12473	28693	1.00053
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Log-Normal	α	13.16866	0.18650	12.79419	13.52567	12055	24301	1.00096
	β_0	0.03240	0.00247	0.02765	0.03732	13959	30751	1.00056
	β_1	−0.00013	0.00002	−0.00016	−0.00010	14606	32955	1.00054
	β_2	0.00456	0.00088	0.00290	0.00632	284083	816801	1.00005
	β_3	0.11668	0.02768	0.06438	0.17159	757951	1288753	1.00001
	β_4	0.17196	0.03251	0.10892	0.23596	741224	1877571	1.00001
	β_5	0.01558	0.00223	0.01129	0.02003	16592	33395	1.00071
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Exponential	α	14.15823	0.17349	13.81983	14.50085	8965.3	19284	1.00081
	β_0	0.01890	0.00248	0.01400	0.02376	9480.3	20124	1.00057
	β_1	−0.00004	0.00002	−0.00007	−0.00001	9719	20788	1.00056
	β_2	0.00038	0.00036	0.00000	0.00110	2265795	2972552	1.00000
	β_3	0.08861	0.03239	0.02468	0.15188	214430	606927	1.00003
	β_4	0.16344	0.03165	0.10150	0.22567	284756	1034811	1.00002
	β_5	0.00834	0.00216	0.00411	0.01259	12425	26550	1.00057
	parameter	mean	std	HDI–2.5%	HDI–97.5%	*ESS*-bulk	*ESS*-tail	$\hat{R}$
Uniform	α	14.19665	0.16869	13.86209	14.52454	9283.3	19167	1.00091
	β_0	0.01968	0.00243	0.01490	0.02444	9100.9	20023	1.00094
	β_1	−0.00005	0.00002	−0.00008	−0.00002	9360.6	20976	1.00091
	β_2	−0.00307	0.00115	−0.00531	−0.00079	466710	1312060	1.00001
	β_3	0.08771	0.03171	0.02517	0.14966	216002	634502	1.00006
	β_4	0.15298	0.03119	0.09157	0.21392	264971	972722	1.00002
	β_5	0.00786	0.00210	0.00371	0.01196	12772	26780	1.00051

Table 9.4: Summary for extensive sampling