Journal of Statistical Software

February 2024, Volume 108, Issue 2. doi: 10.18637/jss.v108.i02

The R Package markets: Estimation Methods for Markets in Equilibrium and Disequilibrium

Pantelis Karapanagiotis
EBS Business School

Abstract

Market models constitute a significant cornerstone of empirical applications in business, industrial organization, and policymaking macroeconomics. The econometric literature proposes various estimation methods for markets in equilibrium, which entail a market-clearing structural condition, and disequilibrium, which are described based on a structural short-side rule. Nonetheless, maximum likelihood estimations of such models are computationally demanding, and software providing simple, out-of-the-box methods for estimating them is scarce. Therefore, applications rely on project-specific implementations for estimating these models, which hinders research reproducibility and result comparability. This article presents the R package **markets**, which provides a common interface with generic functionality simplifying the estimation of models for markets in equilibrium and disequilibrium. The package specializes in estimating demanded, supplied, and aggregated market quantities and absolute, normalized, and relative market shortages. Its functionality is exemplified via an empirical application using a classic dataset of United States credit for housing starts. Moreover, the article details the scope and design of the implementation and provides statistical measurements of the computational performance of its estimation functionality gathered via large-scale benchmarking simulations. The **markets** package is free software distributed under the Expat license as part of the R software ecosystem. It comprises a set of estimation and analysis tools that are not directly available from either alternative R packages or other statistical software projects.

Keywords: disequilibrium, marginal effects, market clearing, maximum likelihood, short side rule, shortages.

1. Introduction

Demand and supply estimations are among the most common objectives of econometric work in policy-making, business, and finance. Foundational economic reasoning suggests that the

combination of these two market forces determines observed market outcomes, with various theories proposing different rules through which demand and supply translate to traded quantities and prices. These rules rely on different identification conditions, leading to distinct implications for the estimated market fundamentals. Shared among all approaches is that prices and quantities, simultaneously determined in a market system, represent different sides of a single coin.

With this shared characteristic as its focal point, the usage scope of R (R Core Team 2023) package **markets** is to provide a common estimation interface for market models with different structural assumptions. Moreover, the package's design goal is to provide harmonized post-estimation analysis functionality. The variability of identification conditions found in commonly used market models and the computational difficulties relating to their estimations prevented other software from providing a standard interface with unifying model functionality. As a consequence, recent empirical results obtained from market models with computationally demanding estimation methods rely on project specific implementations (e.g., Carbó-Valverde, Rodríguez-Fernández, and Udell 2016; Loberto and Zollino 2018), using various optimization tools and methods with nonstandardized starting values and tolerances. This situation makes project replication and cross-project comparability difficult.

The primary purpose of using market models is to obtain estimates of price elasticities, demanded/supplied quantities, and shortages/surpluses. The **markets** package offers a variety of well-established methods to obtain estimates of elasticities and market quantities. In addition, it provides a unified set of analysis and visualization tools for the obtained market fits irrespective of their underlying structural assumptions. Furthermore, **markets** specializes in estimating market shortages, providing functionality through which shortage estimates are measured in absolute, normalized, and probability terms.

The common interface of **markets** exposes different estimation methods and optimization tools for the market models it implements. The implemented models follow commonly used structural assumptions found in empirical applications of economics and finance. The most used structural assumption is market clearing, which postulates that prices are infinitely responsive to demand and supply changes and swiftly adjust so that demand and supply perpetually equalize. Alternative structural assumptions are more useful in cases where market shortages or surpluses are observed, and the market clearing assumption constitutes a rather poor approximation. Unemployment in labor markets and financial constraints in credit markets are prominent examples of market surpluses and shortages correspondingly. Shortages and surpluses can also manifest due to exogenous effects. The semiconductor markets offer contemporary examples of unexpected shortages reported after the Coronavirus pandemic in the late 2019. Market models describe such market states using the short-side rule, which presumes that the price adjustment mechanism is imperfect and temporary market imbalances can lead to disparities between demanded and supplied quantities.

Estimating models under the market clearing condition is straightforward because the system resulting from this identification rule is linear, and many software solutions offer appropriate methods. In contrast, the estimation of short side rule systems is comparatively more involved. Because the short-side rule introduces non-linearities to the market system, computational difficulties related to estimating models with these systems hindered the development of standard tools, despite the well-understood methods for their estimation. The package **markets** closes this persisting gap by providing a common framework for estimating, visualizing, and analyzing models relying on both market clearing and short side rules.

The common framework comprises five market models; a model based on market clearing and four models based on the short side rule. All market models of the package are estimated by maximizing their full information likelihoods by default. The market clearing model can additionally be estimated using two-stage least squares. Furthermore, the estimation interface of **markets** allows choosing among the available optimization methods in `optim` of R package **stats** (R Core Team 2023). Lastly, access to a native optimization procedure from the GNU Scientific Library (henceforth **GSL**, Galassi and Gough 2009) is provided for the market clearing model.

More importantly, **markets** uses analytic expressions for calculating the gradients of all likelihoods by default, which greatly reduces the computation times of maximum likelihood optimizations. The article presents statistical evidence documenting the overperformance of the usage of analytic gradients, the expressions of which are package exclusive. These statistics are obtained from large-scale benchmarking estimations using simulated data in the high-performance cluster of Goethe University's center for scientific computing (CSC, https://csc.uni-frankfurt.de/wiki/doku.php?id=public:start). The article presents benchmarking measurements for estimating all models using Broyden-Fletcher-Goldfarb-Shanno (hereafter BFGS) with analytically calculated gradients, BFGS with numerically approximated gradients, and Nelder-Mead.

The package additionally incorporates analytic Hessian expressions for two of the implemented short side rule models. These expressions are used to calculate standard errors by default. In addition, **markets** exports functionality options allowing to estimate heteroscedastic or clustered standard errors for all implemented market models.

Last but not least, **markets** offers a set of comprehensive analysis and visualization post-estimation methods. The visualization methods offer an intuitive way of examining the estimated price elasticities against the quantity-price points observed in the data. The post-estimation analysis methods fall into three categories. First, **markets** offers methods for obtaining fitted demand and supply values, as well as their aggregates. Second, it provides tools for measuring shortages and surpluses from fitted market models. Lastly, the package exposes methods for assessing the impact of changes in the market system's variables on shortages and surpluses.

The **markets** package (Karapanagiotis 2024) is available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=markets and developed on GitHub at https://github.com/pi-kappa-devel/markets.

The remaining article gives an overview of the design and examples of analyses that can be performed with **markets**. Section 2 is a short introduction to equilibrium and disequilibrium econometrics. It presents the stochastic systems of the five market models of **markets** and discusses their likelihoods. Section 3 documents the design approach followed by the package, the scope of its functionality, and compares it to its closest alternatives. Section 4 demonstrates the functionality provided by **markets**. The presentation uses an empirical example based on the classic dataset of Fair and Jaffee (1972), which is also shipped with the package. Finally, Section 5 documents the performance benefits of the package by presenting the results of the large-scale estimation benchmarking exercises comparing various optimization tools. The last section concludes.

2. Econometric background

Market models have been used in econometrics to concisely describe the trading interactions of multiple independently acting agents. They comprise at least two aggregate forces, one stemming from the agents that ask for a commodity or service (the demand side) and one originating from those that offer this commodity or service (the supply side). Despite abstracting from particular characteristics of agents' behavior, such top-down market models still have notable interpretability and predictability capacities.

To illustrate this point, consider a normal and linear in parameters system of demand and supply forces

$$D_{n,t} = \alpha^d P_{n,t} + \beta_0^d + \sum_{j=1}^{k_d} \beta_j^d X_{n,t}^{j,d} + \sum_{j=1}^{k} \eta_j^d X_{n,t}^{j} + u_{n,t}^d \tag{1}$$

$$S_{n,t} = \alpha^s P_{n,t} + \beta_0^s + \sum_{j=1}^{k_s} \beta_j^s X_{n,t}^{j,s} + \sum_{j=1}^{k} \eta_j^s X_{n,t}^{j} + u_{n,t}^s, \tag{2}$$

where D is the demanded quantity, S the supplied quantity, P the market price, $X^{j,d}$ and $X^{j,s}$ are equation specific control variables, X^j are market wide control variables, and u^d and u^s are jointly, normally distributed shocks. The parameter a^d governs how elastic is demand, i.e., how sensitive it is with regard to price changes. Estimates of a^d offer valuable insight to firm owners or managers because, based on this information, they can evaluate how different pricing strategies available to them will affect demand and, thereby, firm profits. Analogously, a^s controls the elasticity of supply, estimates of which can be very informative to have when designing output taxation strategies as a policymaker.

In spite of the above modeling approach's elegance, Equations 1 and 2 are not enough to obtain parameter estimates because demanded and supplied quantities are not typically observed in market data. Instead, the observed quantities in market data are the traded quantities. Surveys can be (and have been) used to gather more information on how much of a commodity is asked or offered for each potential price point. However, the cost of accumulating survey data and, on some occasions, the unwillingness of market participants to truthfully reveal such information due to strategic reasons limits the applicability of the approach.

On the other hand, estimating the parameters of Equations 1 and 2 using market data requires specifying how the observed traded quantity is related to the unobserved demanded and supplied quantities. The easiest and most prominent way of establishing this relationship is via the *market clearing condition*

$$Q_{n,t} = D_{n,t} = S_{n,t}. \tag{3}$$

The market clearing condition postulates that demanded and supplied quantities are equal for all entities and time points. The stochastic, linear system combining Equations 1 to 3 is commonly referred to as the *equilibrium model*[1].

The equilibrium model can be estimated using either two-stage least squares or full information maximum likelihood (see Zellner and Theil 1962; Balestra and Varadharajan-Krishnakumar 1987), with the two methods being asymptotically equivalent. Substituting

[1]This is perhaps an unfortunate effect of institutional inertia that reinforces the usage of the less accurate 'equilibrium model' naming convention in favor of the more descriptive 'market clearing model'.

Equation 3 into Equations 1 and 2 gives a system of two stochastic equations on traded quantities and prices, which can be used to derive the model's likelihood. To write the likelihood's expression, suppose that φ denotes the (normal) joint density of u^d and u^s, and let $\theta = \left(\alpha^d, \theta_d, \alpha^s, \theta_s\right)$ and $Y = (Y_d, Y_s)$, where

$$\theta_m = \left(\beta_0^m, \left\{\beta_j^m\right\}_j, \left\{\eta_j^m\right\}_j\right)^\top \quad \text{and} \quad Y_m = \left(1, \left\{X_j^{j,m}\right\}_j, \left\{X^j\right\}_j\right)^\top \quad \text{for} \quad m = d, s.$$

Then, the equilibrium model's likelihood is given by

$$\mathrm{L}\left(\theta; q, p, Y\right) = \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q, P\right)}\left(q, p \mid Y; \theta\right) = \varphi\left(q - \alpha^d p - \theta_d^\top Y_d,\ q - \alpha^s p - \theta_s^\top Y_s\right).$$

Estimating the parameters of vector θ allows one to also calculate fitted values for the demanded and supplied quantities (say $\hat{D}_{n,t}$ and $\hat{S}_{n,t}$). Because of the error terms $u_{n,t}^d$ and $u_{n,t}^s$, the fitted values of $\hat{D}_{n,t}$ and $\hat{S}_{n,t}$ for individual observations can defer, allowing entity n at time t to have an expected shortage given by $\hat{G}_{n,t} = \hat{D}_{n,t} - \hat{S}_{n,t}$[2]. Nevertheless, the market clearing condition imposes that the shortages of some observations are canceled out by the surpluses of other observations on average. Therefore, on aggregate shortages tend to vanish, i.e., $\sum_{n,t} \hat{G}_{n,t}$ is close to zero.

Thus, market clearing is not the most conducive identifying condition for describing all markets for all periods. Market shortages and surpluses can occur either due to frictions (e.g., unemployment in labor markets or financial constraints in loan markets), strategic behaviors (e.g., the 1973-74 petroleum shortages resulted from the embargo of the organization of the petroleum exporting countries), or unexpected crises (for example the 2021-2022 semiconductor shortages following the 2019's Coronavirus pandemic).

An alternative identifying condition allowing for aggregate shortages is the *short side rule*, i.e.,

$$Q_{nt} = \min\left\{D_{n,t}, S_{n,t}\right\}. \tag{4}$$

Models that replace the market clearing condition with the short side rule are known as disequilibrium models. This modification makes the systems of the disequilibrium models non-linear, which prevents using least square methods for estimating them. Full information maximum likelihood still offers a viable method (Maddala 1986), albeit accompanied by computation complexities (see also Section 5).

Instead of equating demanded and supplied quantities, the short side rule postulates that the minimum from these two quantities is observed in the data. This identifying condition allows shortages both for particular observations and on aggregate. For a particular observation, the probability that the traded quantity equals the supplied quantity is calculated by

$$\pi_S = \mathbb{P}\left(D > S \mid p, Y; \theta\right) = \mathbb{P}\left(u^d - u^s > \alpha^s p + \theta_s^\top Y_s - \alpha^d p - \theta_d^\top Y_d \mid p, Y; \theta\right), \tag{5}$$

where the random variable $u^d - u^s$ is normally distributed as the difference of normally distributed random variables. The probability that the traded quantity equals the demanded quantity, denoted by π_D, is defined analogously.

[2]There is no need to introduce a separate symbol for surpluses as they can be thought as negative shortages.

The *basic (disequilibrium) model* is defined by Equations 1, 2 and 4 (Maddala and Nelson 1974). Using the total probability theorem, its likelihood can be represented by a two-parts expression, namely

$$\begin{aligned}\mathrm{L}\left(\theta; q, p, Y\right) &= \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}Q}\left(q \mid S > D, p, Y; \theta\right)\pi_D + \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}Q}\left(q \mid D \geq S, p, Y; \theta\right)\pi_S \\ &= \int_q^\infty \varphi\left(q - \alpha^d p - \theta_d^\top Y_d,\ S\right)\mathrm{d}S + \int_q^\infty \varphi\left(D,\ q - \alpha^s p - \theta_s^\top Y_s\right)\mathrm{d}D.\end{aligned}$$

The first part is the density of the demanded quantities, conditional on that they are equal to the observed traded quantity, multiplied by the probability that this market state is observed. The second part is the analogous expression for cases when supplied quantities are observed.

One limitation of the basic model is that it ignores the role of prices in market systems. If the market is in an excess demand state, the producers can increase their profits by raising prices. Analogously, if the market is in an excess supply state, producers can eliminate the surpluses in their inventories by offering lower prices. Abstracting from such a core feature of the market can lead to underfitting and increase the bias error of the basic model.

The *directional model* mitigates this issue by introducing a separation rule based on price movements (Fair and Jaffee 1972). In addition to Equations 1, 2 and 4, the directional model separates the sample according to

$$\Delta P_{n,t} \geq 0 \implies D_{n,t} \geq S_{n,t}. \tag{6}$$

With its observations belonging either to an excess demand or to a excess supply state, the directional model's likelihood is

$$\mathrm{L}\left(\theta; q, p, Y\right) = \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}Q}\left(q \mid S > D, p, Y; \theta\right)\pi_D\right)^{1-\mathbb{I}_{\Delta P \geq 0}} \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}Q}\left(q \mid D \geq S, p, Y; \theta\right)\pi_S\right)^{\mathbb{I}_{\Delta P \geq 0}},$$

where $\mathbb{I}_{\Delta P \geq 0}$ is an indicator function taking the value one if condition (6) is satisfied. Because prices are used to separate the sample, they cannot be included in both the demand and supply specifications (see Maddala and Nelson 1974, p.1021), making the model applicable only for markets with fully inelastic demand or supply.

The *deterministic adjustment model* maintains sample separation based on price movements; however, it allows the estimation of price coefficients for both sides of the market by including an additional parameter in the price equation. The extra price adjustment parameter increases the model's flexibility. To be concrete, the model comprises Equations 1, 2 and 4, and the deterministic price dynamics

$$\Delta P_{n,t} = \frac{1}{\gamma}\left(D_{n,t} - S_{n,t}\right), \tag{7}$$

which also serve as the separation rule. Given the classification of an observation based on this rule, one of the variables D and S can be eliminated. As a result, the remaining variable becomes equal to the traded quantity. This, by slightly abusing the notation so that the vector θ also contains parameter γ, results in the likelihood

$$\begin{aligned}&\mathrm{L} := \mathrm{L}\left(\theta; q, p, p_{-1}, Y\right) \\ &= \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q, P\right)}\left(q, p \mid S > D, p_{-1}, Y; \theta\right)\pi_D\right)^{1-\mathbb{I}_{\Delta P \geq 0}} \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q, P\right)}\left(q, p \mid D \geq S, p_{-1}, Y; \theta\right)\pi_S\right)^{\mathbb{I}_{\Delta P \geq 0}} \\ &= \varphi\left(q - \alpha^d p - \theta_d^\top Y_d,\ q - \gamma\Delta p - \alpha^s p - \theta_s^\top Y_s\right)^{1-\mathbb{I}_{\Delta P \geq 0}} \varphi\left(q - \gamma\Delta p - \alpha^d p - \theta_d^\top Y_d,\ q - \alpha^s p - \theta_s^\top Y_s\right)^{\mathbb{I}_{\Delta P \geq 0}}.\end{aligned}$$

The deterministic adjustment model is conceptually close to the equilibrium model in that if one lets γ approach zero, prices become infinitely flexible, which essentially dictates that demanded and supplied quantities tend to be equal.

The last model of **markets** is the *stochastic adjustment model*. Its system is determined by Equations 1, 2 and 4, and the stochastic price dynamics

$$\Delta P_{n,t} = \frac{1}{\gamma}\left(D_{n,t} - S_{n,t}\right) + \beta_0^p + \sum_{j=1}^{k_p} \beta_j^p X_{n,t}^{j,p} + u_{n,t}^p, \tag{8}$$

where $X^{j,p}$ are the control variables and u^p the disturbance term of the price equation. Since Equation 8 is stochastic, it cannot be used to separate the sample and the stochastic adjustment model does not rely on any pre-estimation classification of observations. To write the likelihood of this model, let $\theta_p = \left(\beta_j^p\right)_j^\top$, $Y_p = \left(X^{j,p}\right)_j^\top$. Suppose also that θ contains the parameters γ and θ_p, and that Y contains Y_p. Moreover, redefine φ to denote the joint density of the shocks u^d, u^s, and u^p. Then, the likelihood of this model is given by

$$\begin{aligned}
\mathrm{L}\left(\theta; q, p, p_{-1}, Y\right) = & \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q,P\right)}\left(q, p \mid S > D, p_{-1}, Y; \theta\right)\pi_D + \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q,P\right)}\left(q, p \mid D \geq S, p_{-1}, Y; \theta\right)\pi_S \\
= & \int_q^\infty \varphi\left(q - \alpha^d p - \theta_d^\top Y_d,\ S,\ \Delta p - \frac{1}{\gamma}\left(q - S\right) - \theta_p^\top Y_p\right)\mathrm{d}S + \\
& \int_q^\infty \varphi\left(D,\ q - \alpha^s p - \theta_s^\top Y_s,\ \Delta p - \frac{1}{\gamma}\left(D - q\right) - \theta_p^\top Y_p\right)\mathrm{d}D.
\end{aligned}$$

The stochastic adjustment model has the most flexible system among the five market models reviewed in this section. The model's flexibility reduces the bias error but increases the variance error of its fits. In addition, the increased flexibility intensifies the computation complexity of estimating the model.

3. Scope, design, and alternatives

The estimation functionality of **markets** aims at disentangling demand and supply from market data using various structural identification conditions. In particular, the package implements a model based on market clearing and four models based on the short side rule. The majority of the functionality offered by **markets** does not have any direct implemented alternatives in either R or other statistical software.

The market clearing model is the one for which estimation with alternative software is mostly available. This is mainly due to the availability of the model's two-stage least square estimation method. The method is fast, reliable, and accessible via simple commands in mainstream statistical software, rendering specialized libraries less relevant. However, the situation is entirely different regarding the disequilibrium models. Least square estimation methods are not available, the models have nonlinear systems with comparatively more complicated likelihoods, and their estimation is associated with computational difficulties because their likelihoods have poles and multiple local maxima. Although they have been frequently used in applications (see, e.g., Carbó-Valverde, Rodríguez-Fernández, and Udell 2009; Carbó-Valverde *et al.* 2016; Loberto and Zollino 2018; Baird, Daugherty, and Kumar 2019), each project has been based on distinct (re-)implementations of estimation procedures using different starting

values, tolerance parameters, and optimization tools. Non open-source implementations raise research reproducibility obstacles and make results' comparison difficult, if not impossible.

The package **markets** aims to fill this gap by providing an open-source implementation of market model estimation with simple workflows, providing meaningful defaults and making equilibrium and disequilibrium models equally accessible. In addition, the package delivers a common interface for analyzing all market models irrespective of their identification conditions. This feature greatly ameliorates the cross model comparison of estimation results. The same methods are used to estimate, visualize, and summarize all models implemented in the package. Last but not least, **markets** supports maximum likelihood estimations using analytically calculated expressions of gradients, which are significantly faster than estimations based on numerical gradient approximations or derivative-free methods (see Section 5).

With the single exception of the basic disequilibrium model, which can also be estimated using the R package **disequilibrium** (Latshaw and Guggisberg 2020), the maximum likelihood estimation functionality found in **markets** is not directly offered by alternative statistical software. With the **disequilibrium** package, one can estimate the basic model using the L-BFGS-B implementation of `optim` with the numerically approximated model likelihood's gradient. Instead, **markets** allows the user to choose both the optimization method and whether numerical approximations or analytic gradient and Hessian calculations are used. There are no software alternatives directly estimating the directional, deterministic adjustment, and stochastic adjustment models. By default, the BFGS implementation of `optim` with analytic expressions for the gradients is used in the maximum likelihood estimations of all models.

The estimation methods implemented by **markets** have macroeconomic origins (Fair 1971; Maddala and Nelson 1974), but have found applications in empirical economics and finance research using microdata (see for instance Bulligan, Busetti, Caivano, Cova, Fantino, Locarno, and Rodano 2017; Carbó-Valverde *et al.* 2009). In this respect, demand estimation methods such as the almost ideal demand systems (AIDS) of Deaton and Muellbauer (1980) and the structural estimation of Berry, Levinsohn, and Pakes (1995), typically abbreviated as BLP, partially overlap with the methods described in this article. The AIDS and BLP methodologies are micro-founded, but they focus only on the demand side and do not concern varying structural assumptions such as the market clearing and short side rules. Implementations of these methods can be correspondingly found in the R packages **blpestimator** (Brunner 2022) and **miceconaids** (Henningsen 2022).

The **markets** package is organized based on the object oriented hierarchy depicted in Figure 1. Five front-end model classes, one corresponding to each implemented model, are exposed to the user. These are the (i) `equilibrium_model`, (ii) `diseq_basic`, (iii) `diseq_directional`, (iv) `diseq_deterministic_adjustment`, and (v) `diseq_stochastic_adjustment` classes. The two back-end classes, namely (i) `market_model` and (ii) `disequilibrium_model`, act as base abstract classes and enclose functionality that is common for all market models. The overarching `market_fit` class acts as the common interface through which the functionality of the package is exposed to the user in a uniform fashion. The arrows in Figure 1 indicate how these classes are related with each other. Arrows with filled arrowheads indicate inheritance (is-a relationship), while arrows with empty arrowheads indicate composition (has-a relationship).

The package implements maximum likelihood estimation routines for the four disequilibrium models. Moreover, it implements both maximum likelihood and least squares estimation

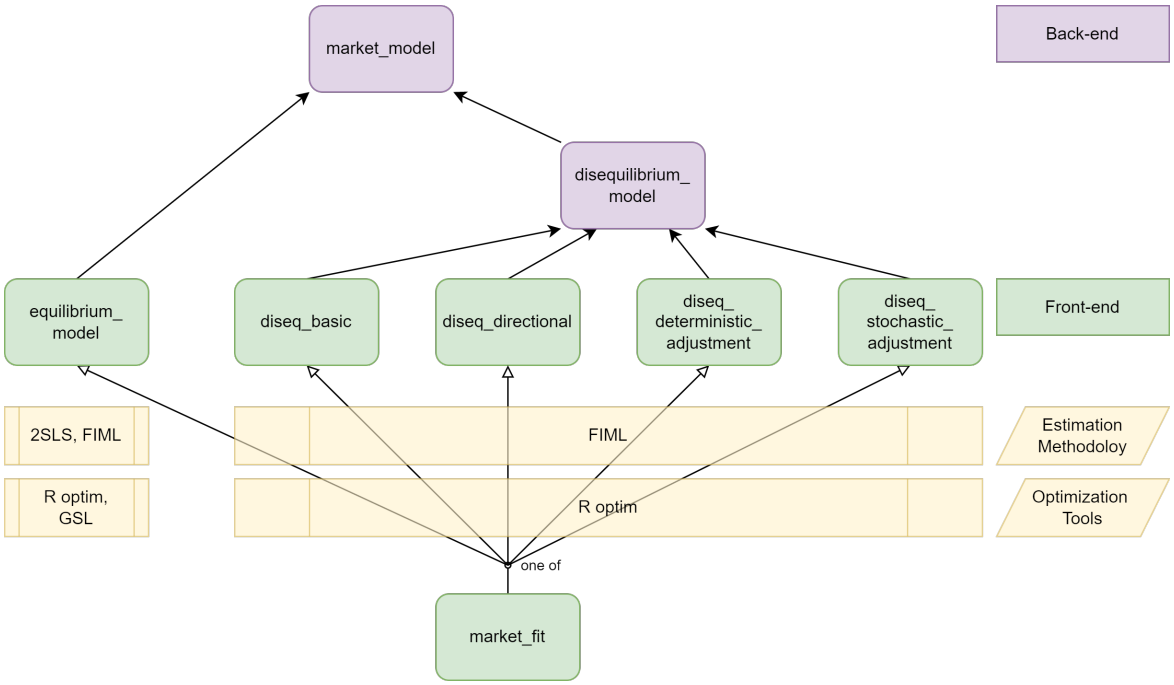


Figure 1: Design overview.

methodologies for the `equilibrium_model`. The least squares estimation is based on function `lm` of the R package **stats**. The maximum likelihood estimation is implemented for all models based on `optim`. In addition, the `equilibrium_model`'s likelihood can be maximized using native **GSL** optimization routines.

When building the package from its sources, if the C++17 version of `execution.h` is available in the target machine during compilation, some parallelization optimizations are enabled in the native estimation routines. Specifically, gradient calculations are parallelized using the `std::execution::par_unseq` execution policy when the likelihood is maximized with **GSL**. Using of native optimization routines does not necessarily result in faster execution times because there is an overhead stemming from the communication between R and **GSL**. Unreported estimation time benchmarks indicate that for small datasets, machines without C++17 support, or machines with few available processors, the communication cost is greater than the benefits of using native routines, which results in slower execution times. Still, estimating the equilibrium model via **GSL** routines is included in the package's exposed functionality to accommodate use cases with access to parallel execution[3]. In addition, native likelihood maximization allows the user to customize the optimization call further by choosing the step size and the gradient tolerance of the BFGS algorithm. The two-stage least squares methodology is, of course, the least computationally intensive estimation method for the equilibrium model as it merely involves linear algebra operations (for the statistical background, see Henningsen and Hamann 2007). For well-behaved samples, all estimation methods and optimization tools available for the `equilibrium_model` result in similar estimates (see Appendix C).

[3]And potential future broader support of the C++17 execution policies by the compilers and systems targeted by R and CRAN.

Two stage least square fits of the equilibrium model are the easiest to obtain with alternative software. One can manually implement the methodology using three linear regressions to obtain an equilibrium model's fit. Firstly, one regresses prices on the collection of instruments, and demand and supply controls. Subsequently, prices are substituted by the first stage's fitted prices in the demand and supply regressions. In Stata, the first and second stage regressions for each market equation are combined in a single call to `ivregress 2sls`. In R, the package **systemfit** (Henningsen and Hamann 2007) simplifies the estimation of the complete market system to a single call and, in addition, calculates the appropriate coefficient and residual covariance matrices. The package provides a comprehensive set of tools estimating linear systems of simultaneous stochastic equations, but full information maximum likelihood is not among them.

Besides using `optim`, likelihood maximization in R can also be performed using the `nlm` of **stats** or the `mle2` function of **bbmle** (Bolker and R Core Team 2023)[4]. To a lesser extent, these approaches constitute alternatives to **markets** since one can use them to obtain estimations of the market models. The functions `nlm` and `mle2` offer more general optimization functionality. Package **markets** specializes in market models and automates many estimation elements. For instance, the simplest way to obtain a maximum likelihood fit of a market model with `nlm` using a derivative-free optimization method requires programming the likelihood of the market model (see Section 2) and providing appropriate initializing values. The implementation requirements become much greater when derivative-based optimization methods are used because the user has to manually program the gradients of market model likelihoods, which involve tedious calculations and complex expressions[5]. Instead, **markets** offers ready-to-use implementations of market model likelihoods and gradients, and automatically calculates appropriate initializing values using linear regressions.

The situation is similar when attempting to estimate market models using maximum likelihood in other software. Stata offers general likelihood maximization functionality via the command `ml` (Gould, Pitblado, and Poi 2010). Additionally, specialized commands such as `logistic` and `poisson` offer pre-programmed likelihoods that facilitate the estimation of the logistic and Poisson models. To estimate the market model of **markets**, users have to resort to `ml` and manually program the likelihoods. The initialization of the estimation process is automatic. With `ml`, the likelihood can be maximized using either derivative-free or derivative-based optimization methods, with programming requirements in each case being similar to those of R for both cases. The function `movestay` (Lokshin and Sajaia 2004) gives an alternative option to estimate the disequilibrium models in Stata. The disequilibrium models are Markov switching models, and their system can be rewritten in terms of latent variables (see, e.g., Maddala 1986). The function `movestay` offers functionality that estimates Markov switching models based on derivative-free optimization methods. Besides lacking the derivative expressions, one disadvantage is that these estimations use a less economically intuitive representation of the market models. The advantage of `movestay` is that the user does not have to program the models' likelihoods.

Internally, all five models of **markets** have similar implementation components, which are depicted in Figure 2. Each model class contains a logging object that handles the exposed methods' output depending on the verbosity chosen during initialization. More importantly, model

[4]Previous versions of **markets** were based on **bbmle** and **systemfit**.

[5]However, derivative based methods have computational advantages when working with large databases or performing an extended number of model estimations.

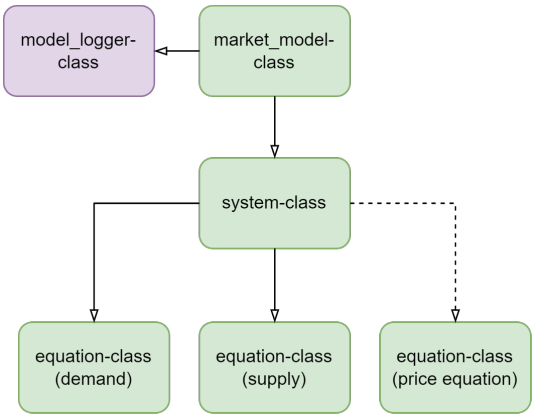


Figure 2: Market class implementation.

classes contain objects describing their corresponding systems. Objects of system classes contain data describing the models' systems of stochastic equations. In this respect, system classes contain equation objects for all stochastic equations of the systems they describe. All system classes contain demand and supply equation objects because these equations are ubiquitous in market models. The `diseq_stochastic_adjustment` system class additionally contains a price equation object. The system and equation classes primarily contain backend functionality used to store and organize intermediate estimation data. The functionality of **markets** is intended to be accessed via the exposed methods of the market models and `market_fit` classes.

4. Functionality via an empirical example

The package is designed following R's model estimation paradigm. Estimation calls expect model formulas evaluatable in accompanied data frames and return fitted market models. Estimation results can then be accessed by applying standard R methods such as `summary`, `coef`, `plot` on the fitted objects. In addition, **markets** offers package specific methods, such as `shortages`, `demanded_quantities`, and `shortage_marginal`, to examine market relevant implications of the obtained fits.

4.1. The houses dataset

The `houses` dataset contains monthly macroeconomic time series for the US housing credit market from January 1958 to December 1969 (144 observations). The housing credit market was initially studied by Fair (1971) for this period. Subsequent work on the estimation and assessment of short side rule market models also uses US housing credit data for this period to illustrate the introduced methodologies (for instance, see Fair and Jaffee 1972; Maddala and Nelson 1974; Hwang 1980).

Table 1 presents the variables of the `houses` dataset and provides short descriptions for them. The dataset was constructed according to the sources provided in Fair (1971). The series of `RM` were directly obtained by Fair (1971, Table A.3). The observations of `HS` were collected from the United States President and Council of Economic Advisers (U.S.) (1959–1978), `W` were

DATE	The date of the record.
HS	Private non-farm housing starts in thousands of units (not seasonally adjusted).
RM	FHA mortgage rate series on new homes in units of 100 (beginning-of-month data).
DSLA	Savings capital (deposits) of savings and loan associations in millions of dollars.
DMSB	Deposits of mutual savings banks in millions of dollars.
DHLB	Advances of the federal home loan bank to savings and loan associations in million of dollars.
W	Number of working days in month.

Table 1: Variables in the houses dataset.

manually compiled, while DSLA, DMSB, and DHLB were collected from the Board of Governors of the Federal Reserve System (U.S.) and Federal Reserve Board (1958–1980).

The examples of this section use the demand and supply equation specifications of Hwang (1980) and Maddala and Nelson (1974) as starting points. In particular, the starting demand equation is given by

$$D_t = \alpha^d \texttt{RM}_t + \beta_0^d + \beta_1^d t + \beta_2^d \texttt{W}_t + \beta_3^d \texttt{CSHS}_t + \beta_4^d \texttt{RM}_{t-1} + \beta_5^d \texttt{RM}_{t-2} + \sum_{i=2}^{12} \beta_{4+i}^d \texttt{MONTH}_{i,t} + u_t^d,$$

where CSHS is the cumulative sum of past housing starts and $\texttt{MONTH}_i$ are monthly indicator variables. The variable CSHS is used as a proxy of the stock of available houses, and the monthly indicators are used to capture seasonal demand effects. The starting supply equation is specified as

$$S_t = \alpha^s \texttt{RM}_t + \beta_0^s + \beta_1^s t + \beta_2^s \texttt{W}_t + \beta_3^s \texttt{RM}_{t-1} + \beta_4^s \texttt{MA6DSF}_t + \beta_5^s \texttt{MA3DHF}_t + \sum_{i=2}^{12} \beta_{4+i}^s \texttt{MONTH}_{i,t} + u_t^s,$$

where MA6DSF is the moving average of order 6 of the flow of deposits in savings associations, loan associations, and mutual savings banks, while MA3DHF is the moving average of order 3 of the flow of advances of the federal home loan bank to savings and loan associations. The stochastic terms $u_{d,t}$ and $u_{s,t}$ are jointly, normally distributed. The moving averages and lagged variables of the analysis are constructed from the variables of the houses dataset. The fair_houses function of **markets** automates the construction.

```
R> house_data <- fair_houses()
```

4.2. Model formulas

The market system is specified in estimation calls by formulas, which contain the quantity and price variables, the subject and time identification variables, and the right-hand sides of the stochastic equations of the system. The back-end implementation of **markets** relies on the functionality of the R package **Formula** (Zeileis and Croissant 2010). Moreover, the format of **Formula** is adopted for the description of market model systems. Pipe operators separate the various elements of the market model formula. The model formulas standardize the input of estimation calls of different market models to a great extent.

As an example, the formula corresponding to the demand and supply equations of Section 4.1 is given in R Code 1. From left to right, the left-hand side of the expression specifies the

quantity, price, subject, and time variables of the `house_data` data frame. The right-hand side of the expression specifies from left to right (the right-hand side of) the demand and supply equations.

R Code 1: An example of a market model formula.

```
HS | RM | ID | TREND ~
  RM + TREND + W + CSHS + L1RM + L2RM + MONTH |
  RM + TREND + W + L1RM + MA6DSF + MA3DHF + MONTH
```

Each market model combines the quantity variable with the demand and supply specifications according to its identification condition (market clearing or short side rule) and, whenever relevant, its sample separation rule and price dynamics. Further, the dynamic market models use the price, subject, and time variables to calculate price differences, as required in their estimation.

4.3. Estimation

Models can be initialized and estimated using either a single function call or two separate calls, one for initialization and one for estimation. The first approach is the most convenient one for most use cases. However, there are some use cases in which the second approach can have some advantages depending on the workflow. For example, maximum likelihood estimations with large datasets can be time consuming; thus, it can be convenient to initialize a market model once and then estimate it in parallel using different optimization methods, starting values, and other estimation parameters. More details about the initialization calls (constructors) can be found in Appendix D. The following presentation focuses on the single call estimation approach. All five market models available in **markets** are estimated using full information maximum likelihood in the following examples.

R Code 2 estimates the equilibrium model. The first argument specifies the market model formula, and the second the used data frame. The first two input arguments are used during the model's initialization. The third argument is a list with options used in the model's estimation. For instance, the call of R Code 2 sets the maximum number of iterations in the control argument of the optimization routine equal to 5000.

R Code 2: Estimating the equilibrium model.

```
R> eq <- equilibrium_model(
+    HS | RM | ID | TREND ~ RM + TREND + W + CSHS + L1RM + L2RM + MONTH |
+      RM + TREND + W + L1RM + MA6DSF + MA3DHF + MONTH,
+    house_data, estimation_options = list(control = list(maxit = 5000)))
```

R Code 3 estimates the deterministic adjustment model without modifying the formula used in R Code 2. By default, operations of **markets** display errors and warnings in the standard output. However, it can be helpful to have more information about the performed operations on some occasions (e.g., during development or testing). The default behavior can be overridden by specifying the verbose argument of the estimation call. For instance, the call of R Code 3 sets the verbosity level to be equal to 2, which besides errors (level 0) and warnings (level 1), displays basic information concerning the performed operations in the standard output. In this case, the `diseq_deterministic_adjustment` call displays three information

messages (preceded by `Info :`) and one warning message (preceded by `Warning :`). The first information message declares the model, the second gives the number of observations dropped due to the calculation of lagged prices, and the third informs the caller about the data subsets created by the model's separation rule.

R Code 3: Estimating the deterministic adjustment model.

```
R> da <- diseq_deterministic_adjustment(
+    HS | RM | ID | TREND ~ RM + TREND + W + CSHS + L1RM + L2RM + MONTH |
+      RM + TREND + W + L1RM + MA6DSF + MA3DHF + MONTH,
+    house_data, verbose = 2,
+    estimation_options = list(control = list(maxit = 5000)))

Info: This is Deterministic Adjustment model.
Warning: Dropping 14 rows due to missing values.
Info: Dropping 1 row to generate 'LAGGED_RM'.
Info: Sample separated with 18 rows in excess supply and 111 rows in
  excess demand states.
```

The default optimization routine used by **markets** is BFGS with analytically calculated gradient expressions, which is the fastest available optimization option (see Section 5). On rare occasions, numerical stability issues might be less prominent in derivative-free optimization algorithms. Access to alternative optimization routines is provided by setting the method in the `estimation_options` list. As an example, R Code 4 estimates the directional model using the Nelder-Mead algorithm.

R Code 4: Estimating the directional model.

```
R> dr <- diseq_directional(
+    HS | RM | ID | TREND ~ TREND + W + CSHS + L1RM + L2RM |
+      RM + TREND + W + MA6DSF + MA3DHF + MONTH,
+    house_data, estimation_options = list(
+      method = "Nelder-Mead", control = list(maxit = 5000)))
```

Maximum likelihood estimations in **markets** are initialized by automatically calculated starting values coming from model-specific, linear regressions by default. For instance, the basic model's starting values are obtained by regressing the (observed) traded quantity on the demand and supply right-hand side specifications. The equilibrium model's starting values are obtained by two stage least square estimates of the system. The user can override the default behavior and provide custom starting values. R Code 5 provides an example of starting value customization. In this case, the estimated coefficients of the equilibrium model are used as starting values for the estimation of the basic model. The underlying optimization routine of `optim` expects that the length of the initializing vector and the names of its entries correspond to the number and names of model coefficients to be estimated. If this is not true, the initialization of the optimizer fails. This behavior has to be taken into account when passing starting values. For example, R Code 5 estimates the basic model using independent shocks, which would fail if the correlation coefficient was not removed from the supplied starting values.

R Code 5: Estimating the basic model.

```
R> start <- coef(eq)
R> start <- start[names(start) != "RHO"]
R> bs <- diseq_basic(
+    HS | RM | ID | TREND ~ RM + TREND + W + CSHS + L1RM + L2RM + MONTH |
+      RM + TREND + W + L1RM + MA6DSF + MA3DHF + MONTH,
+    house_data, verbose = 2, correlated_shocks = FALSE,
+    estimation_options = list(start = start, control = list(maxit = 5000)))

Info: This is Basic model.
Warning: Dropping 14 rows due to missing values.
```

The last estimation example uses the stochastic adjustment model. The stochastic adjustment model is comprised of three stochastic equations. Thus, the passed model formula should also specify the right-hand side of the stochastic price dynamics (see Equation 8). The model constructor automatically accounts for the shortage term $(D_t - S_t)$ appearing in Equation 8 but it is not observed in the data. Thus, the caller needs to provide only the remaining summands. As a concrete example, R Code 6 estimates the stochastic adjustment model with a price dynamics' specification given by

$$\Delta \texttt{RM}_t = \frac{1}{\gamma}\left(D_t - S_t\right) + \beta_0^p + \beta_1^p t + \beta_2^p \texttt{RM}_{t-2} + \beta_3^p \texttt{RM}_{t-3} + u_{n,t}^p + u_t^p.$$

In addition, the call of R Code 6 instructs the estimation routine to calculate clustered standard errors with respect to the levels of variable `W` (number of working days in a month).

R Code 6: Estimating the stochastic adjustment model.

```
R> sa <- diseq_stochastic_adjustment(
+    HS | RM | ID | TREND ~ RM + TREND + W + CSHS + MONTH |
+      RM + TREND + W + L1RM + L2RM + MA6DSF + MA3DHF + MONTH |
+      TREND + L2RM + L3RM,
+    house_data |> dplyr::mutate(L3RM = dplyr::lag(RM, 3)),
+    correlated_shocks = FALSE,
+    estimation_options = list(
+      control = list(maxit = 5000), standard_errors = "W"))
```

Table 2 presents the estimated coefficients for all five models. The coefficients of the monthly indicators are omitted for brevity. Parentheses contain the p values for the estimated coefficients. The *Coefficient* column displays the names of the estimated coefficients used in **markets** by default. Since variables can be simultaneously included in multiple equations, the coefficient names use distinct prefixes for each equation. Demand coefficient names are prefixed by `D_`, supply names are prefixed by `S_`, and the names of the price equation are prefixed by `P_`. The correlation coefficient of the shocks is by default named `RHO`, and the estimate of γ (see Equation 7 or Equation 8) is composed by concatenating the price variable's name with `_DIFF`.

In accordance with the usual economic intuition, estimated demand side price coefficients (`D_RM`) are negative, while estimated supply side price coefficients (`S_RM`) are positive. Both

Coefficient	Equilibrium	Basic	Directional	Deterministic adjustment	Stochastic adjustment
D_RM	−7.4514 (0.00)	−10.5700 (0.00)	–	−4.1059 (0.00)	−0.7750 (0.00)
D_CONST	3.6144 (0.00)	3.7718 (0.00)	−9.4997 (0.00)	42.0735 (0.00)	534.8947 (0.00)
D_TREND	−3.8074 (0.00)	−40.5795 (0.00)	17.6369 (0.00)	−1.6725 (0.00)	−20.8842 (0.00)
D_W	2.4745 (0.18)	−41.0733 (0.00)	4.2710 (0.41)	1.3388 (0.35)	2.1748 (0.00)
D_CSHS	0.0346 (0.00)	0.2211 (0.03)	−0.1512 (0.00)	0.0162 (0.00)	0.1921 (0.00)
D_L1RM	9.9305 (0.00)	19.0681 (0.00)	0.9166 (0.68)	5.9190 (0.00)	–
D_L2RM	−2.4254 (0.00)	−4.0824 (0.00)	−0.9874 (0.66)	−1.7948 (0.00)	–
S_RM	1.0932 (0.00)	0.3609 (0.16)	0.0738 (0.00)	0.2990 (0.49)	0.1298 (0.49)
S_CONST	−50.2872 (0.00)	−71.9648 (0.00)	−40.1998 (0.00)	−35.6559 (0.00)	−142.7089 (0.00)
S_TREND	−0.1809 (0.00)	−0.1403 (0.00)	−0.1799 (0.00)	−0.1324 (0.01)	−0.4197 (0.00)
S_W	2.6866 (0.00)	3.1616 (0.00)	2.0319 (0.00)	2.1560 (0.00)	4.3563 (0.00)
S_L1RM	−1.0396 (0.00)	−0.2892 (0.27)	–	−0.2398 (0.58)	−0.1301 (0.68)
S_MA6DSF	0.0516 (0.00)	0.0510 (0.00)	0.0455 (0.00)	0.0443 (0.00)	0.0479 (0.00)
S_MA3DHF	0.0421 (0.00)	0.0409 (0.00)	0.0440 (0.00)	0.0321 (0.00)	0.0504 (0.00)
S_L2RM	–	–	–	–	0.1907 (0.25)
RM_DIFF	–	–	–	1.7691 (0.02)	35.4703 (0.00)
P_CONST	–	–	–	–	−25.1614 (0.00)
P_TREND	–	–	–	–	−0.0222 (0.09)
P_L2RM	–	–	–	–	0.2299 (0.00)
P_L3RM	–	–	–	–	−0.1869 (0.00)
D_VARIANCE	1241.6599 (0.00)	1228.9326 (0.00)	591.7524 (0.00)	748.3865 (0.00)	47.1873 (0.00)
S_VARIANCE	122.6572 (0.00)	100.2919 (0.00)	89.8593 (0.00)	102.3525 (0.00)	82.8410 (0.00)
P_VARIANCE	–	–	–	–	23.9172 (0.00)
RHO	−0.1180 (0.45)	–	−0.0810 (0.93)	0.1562 (0.44)	–

Table 2: Estimation results.

the deterministic and stochastic adjustment models suggest that price changes are responsive to shortages and surpluses as their estimated shortage response parameters (`RM_DIFF`) are positive and have very small p values.

4.4. Summarizing and visualizing market fits

The estimation examples of R Codes 2 to 6 return `market_fit` objects. These objects can be used to summarize and visualize the fitted models.

Calling `summary` with a `market_fit` object as an input argument prints basic information about the fit of the model to standard output. The output summary comprises four parts separated by empty lines. The first part contains essential information concerning the model. The second part contains information regarding the estimation method of the model. The third part includes the model's estimated coefficients, their standard errors, z values, and p values. Finally, the last part displays information about the model's maximized likelihood.

R Code 7 shows part of the market fit summary for the deterministic adjustment model estimated in R Code 3 (some rows of the output are truncated for brevity). The first part of the summary informs the user about the estimated model in its unindented heading. Then, it describes the specification of the model's equations and gives basic information about the sample separation and the used variables. The second part of the output of `summary` starts with the heading `Maximum likelihood estimation`. It shows the optimization algorithm, convergence status, and used starting values. The third and fourth parts are similar to the estimation summaries of `lm` and `bbmle` objects.

R Code 7: Market model fits' summaries.

```
R> summary(da)

Deterministic Adjustment Model for Markets in Disequilibrium:
  Demand RHS         :   D_RM + D_TREND + D_W + D_CSHS + D_L1RM +
    D_L2RM + D_MONTH
  Supply RHS         :   S_RM + S_TREND + S_W + S_L1RM + S_MA6DSF +
    S_MA3DHF + S_MONTH
  Short Side Rule    : HS = min(D_HS, S_HS)
  Separation Rule    : RM_DIFF analogous to (D_HS - S_HS)
  Shocks             : Correlated
  Nobs               : 129
  Sample Separation  : Demand Obs = 18, Supply Obs = 111
  Quantity Var       : HS
  Price Var          : RM
  Key Var(s)         : ID, TREND
  Time Var           : TREND

Maximum likelihood estimation:
  Method               : BFGS
  Max Iterations       : 5000
  Convergence Status   : success
  Starting Values      :
      D_RM    D_CONST    D_TREND        D_W     D_CSHS     D_L1RM     D_L2RM
 7.528e-02  7.462e+01  2.432e+00  2.395e+00 -1.977e-02  1.512e-01 -3.291e-01
...
 S_MONTH12    RM_DIFF D_VARIANCE S_VARIANCE        RHO
 1.094e+01 -2.154e-15  1.960e+02  1.035e+02  0.000e+00

Coefficients:
               Estimate  Std. Error       z value     Pr(>|z|)
 D_RM        -4.10589655 0.770842666    -5.3265040 1.001211e-07 ***
 D_CONST     42.07353822 0.046772847   899.5291197 0.000000e+00 ***
 D_TREND     -1.67246804 0.166728247   -10.0311019 1.112679e-23 ***
 D_W          1.33881271 1.444887967     0.9265858 3.541416e-01
...
 RHO          0.15624911 0.200804009     0.7781175 4.364997e-01
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

-2 log L: 1724.659
```

An alternative way to inspect the market fit is via the `plot` function. Compared to `summary`, `plot` provides a more concise representation of the fit, primarily focusing on the estimated price coefficients. Figure 3 contains the output of calling `plot` with input arguments the fits of R Codes 2–6 (e.g., `plot(eq)`, `plot(bs)`, etc.).

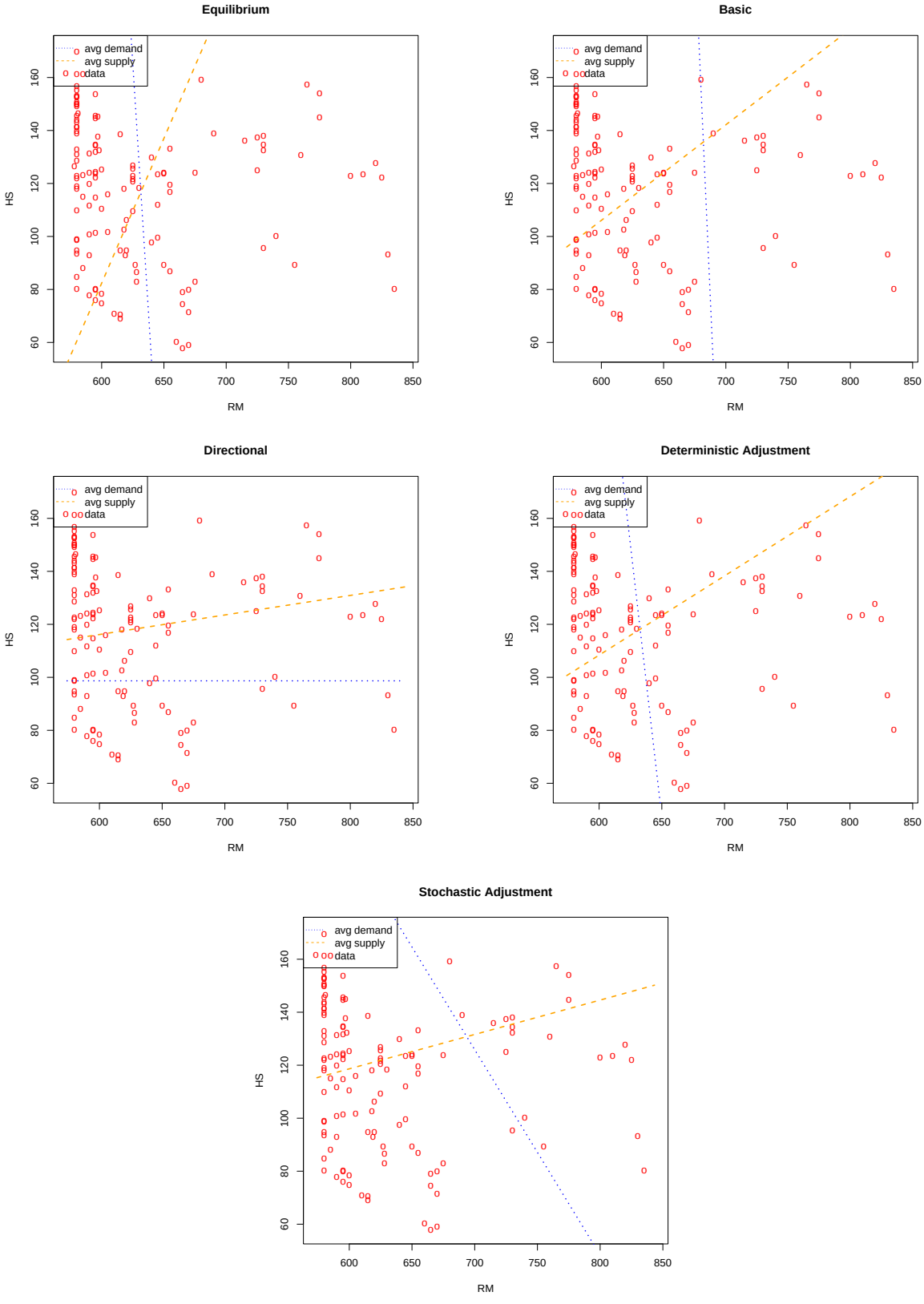


Figure 3: Market fits' visualizations.

The (red) circles in each plot of Figure 3 correspond to the price-quantity points of the data frame used to estimate the model. The (blue) dotted lines represent the average, estimated demand as a function of price. The lines are obtained by calculating the fitted demanded quantity for each price point of the figure's domain using the sample's average values for the remaining control variables. For example, the demand of the equilibrium model of Figure 3 is calculated by

$$D(p) = \hat{\alpha}^d p + \frac{1}{T}\sum_{t=1}^{T}\left[\hat{\beta}_0^d + \hat{\beta}_1^d t + \hat{\beta}_2^d \mathtt{W}_t + \hat{\beta}_3^d \mathtt{CSHS}_t + \hat{\beta}_4^d \mathtt{RM}_{t-1} + \hat{\beta}_5^d \mathtt{RM}_{t-2} + \sum_{i=2}^{12} \hat{\beta}_{4+i}^d \mathtt{MONTH}_{i,t}\right],$$

where hats are used to denote estimated coefficients. The (orange) dashed lines represent average supplies as functions of prices and are analogously calculated.

The estimated price coefficients are relevant both in policymaking and in business applications. Using the `plot` function, the user can quickly assess whether the estimated price coefficients have the expected signs. Microeconomic theory suggests that the relationship between demanded quantities and prices is nonpositive[6], while between supplied quantities and prices is nonnegative.

4.5. Fitted quantities and aggregation

Using the estimated market models, one can obtain fitted values of the (unobserved) demanded and supplied quantities. The package **markets** automates these calculations with the functions `demanded_quantities` and `supplied_quantities`. For example, R Code 8 calculates the fitted values for the stochastic adjustment model estimated in R Code 6. Specifically, it calculates

$$\hat{D}_t = \hat{\alpha}^d \mathtt{RM}_t + \hat{\beta}_0^d + \hat{\beta}_1^d t + \hat{\beta}_2^d \mathtt{W}_t + \hat{\beta}_3^d \mathtt{CSHS}_t + \sum_{i=2}^{12} \hat{\beta}_{4+i}^d \mathtt{MONTH}_{i,t}$$

$$\hat{S}_t = \hat{\alpha}^s \mathtt{RM}_t + \hat{\beta}_0^s + \hat{\beta}_1^s t + \hat{\beta}_2^s \mathtt{W}_t + \hat{\beta}_3^s \mathtt{RM}_{t-1} + \hat{\beta}_4^s \mathtt{RM}_{t-2} + \beta_5^s \mathtt{MA6DSF}_t + \beta_6^s \mathtt{MA3DHF}_t + \sum_{i=2}^{12} \hat{\beta}_{5+i}^s \mathtt{MONTH}_{i,t}$$

for each time point t in the `house_data` data frame.

R Code 8: Fitted demanded and supplied quantities.

```
R> demanded <- demanded_quantities(sa)
R> supplied <- supplied_quantities(sa)
```

The functions `demanded_quantities` and `supplied_quantities` return vectors with a fitted value for each observation in the sample. For example, the fitted values of R Code 8 are plotted against time in Figure 4. In this example, the fitted demanded and supplied quantities are highly seasonal, with the demand side being more volatile than the supply side.

On some occasions, examining the fitted demand and supply at an aggregate market level is more relevant than at an individual observation level. For example, macroeconomic policymakers are primarily interested in aggregate market demand and supply when designing

[6]This relationship is commonly referred to as the law of demand. The law characterization is meant to indicate that it is the predominantly observed relationship in most market settings. However, there are also documented counterexamples, e.g., Giffen goods.

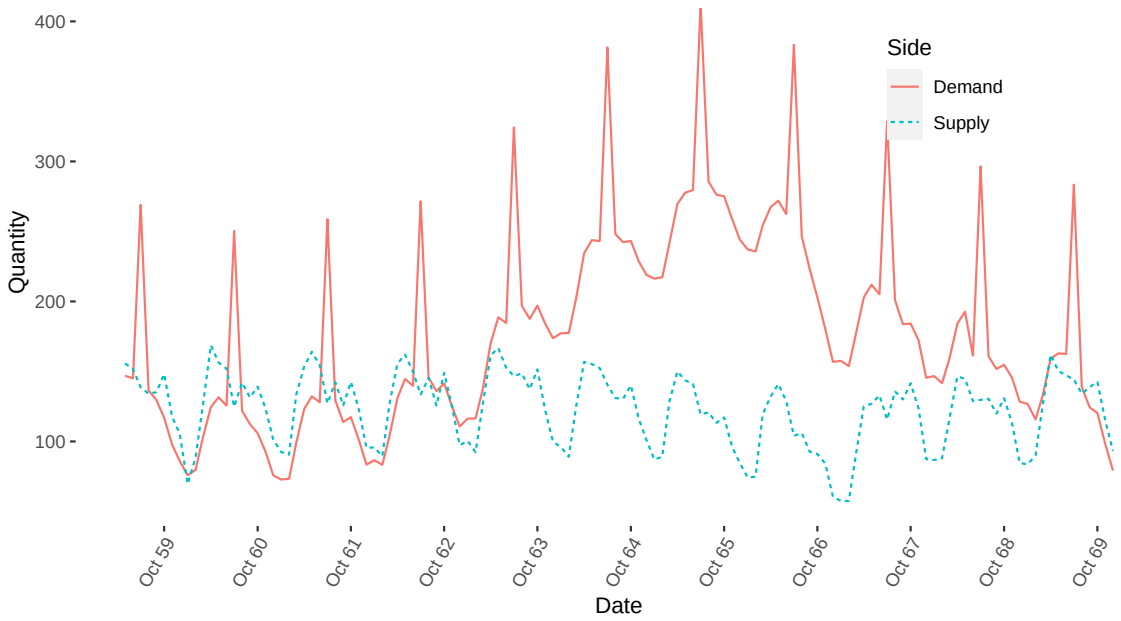


Figure 4: Fitted demanded and supplied quantities.

taxation. The functions `aggregate_demand` and `aggregate_supply` of **markets** calculate aggregated fitted quantities. The functions have two modes depending on the nature of the data frame used to estimate the passed fitted model. For panel data frames, i.e., for data with more than one entity observation per time point, the aggregation functions return a vector of aggregate fitted quantities per time point. For time series data frames, such as the `house_data` data frame of this example, aggregation occurs over all time points.

R Code 9 exemplifies the aggregation functionality using the equilibrium (first command) and stochastic adjustment (second command) models. As the market clearing identification condition implies, aggregate demand and supply are almost equal for the equilibrium model. For the stochastic adjustment model, the fitted aggregate demand is greater than the fitted aggregate supply, which indicates that the market operated on aggregate in a shortage state.

R Code 9: Aggregate fitted quantities.

```
R> c(demand = aggregate_demand(eq), supply = aggregate_supply(eq))

  demand   supply
15180.33 15184.04

R> c(demand = aggregate_demand(sa), supply = aggregate_supply(sa))

  demand   supply
22823.71 15721.84
```

4.6. Analysis of shortages

The package provides an extensive range of options regarding the analysis of shortages and surpluses. The fitted shortages $\hat{G}_{n,t} = \hat{D}_{n,t} - \hat{S}_{n,t}$ are calculated by calling `shortages`.

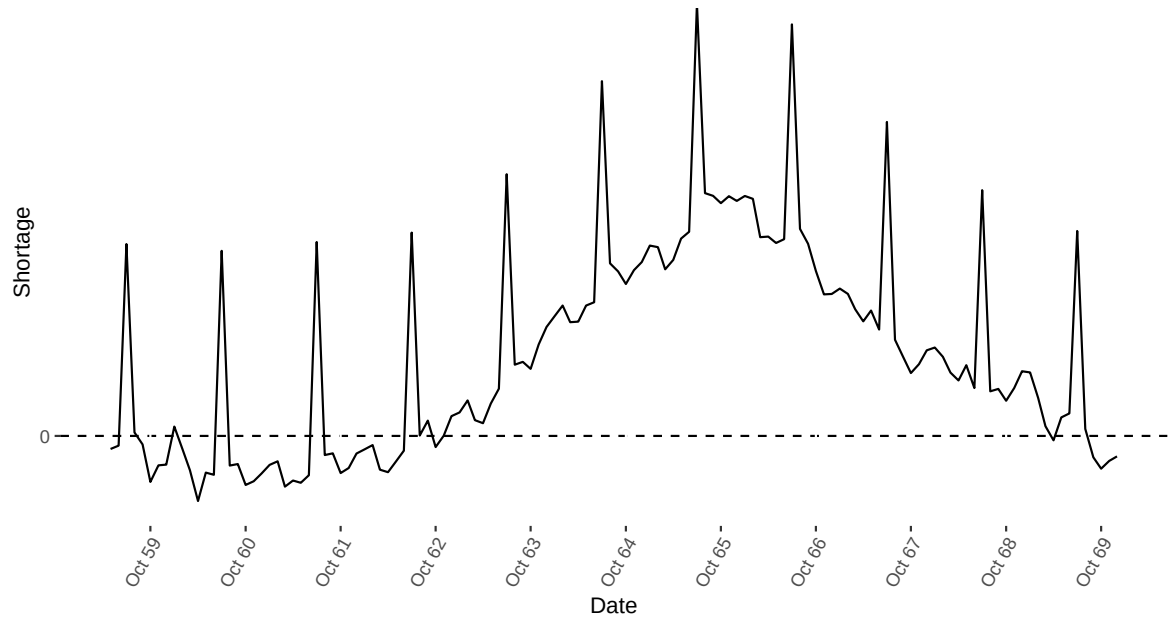


Figure 5: Fitted shortages.

The function `shortages` is a convenience function that calculates the differences between `demanded_quantities` and `supplied_quantities` (for example, compare Figures 4 and 5). It returns a vector of shortages, one for each observation in the sample.

Figure 5 plots the shortages for the stochastic adjustment model fit of R Code 6. Values below the zero horizontal line indicate an estimated market surplus for the given date, while values above indicate estimated market shortages.

Based on the estimated shortages, the sample can be separated (post estimation) into subsets of observations exhibiting excess demand and supply. The sample separation for each predicted state can be easily obtained using the `shortage_indicators` function, which returns a vector of Boolean values indicating whether the corresponding sample observation has a positive fitted shortage. For example, R Code 10 confirms the visual finding of Figure 5, suggesting that most of the sample's observations lie in an excess demand state for the fitted stochastic adjustment model.

R Code 10: Shortage indicators.

```
R> c(no_shortages = sum(shortage_indicators(sa)),
+    no_surpluses = sum(!shortage_indicators(sa)))

no_shortages no_surpluses
          89           39
```

Fitted shortages are measured in quantity units, which prevents their comparison across and within samples due to potential scaling differences. For example, a shortage of 10.000 Euro housing credit for an observation in which supply is 100.000 Euro is 10% relative to the supplied credit, while 100% when supply is 10.000 Euro. The lack of access to trade credit is

more severe in the second case. The **markets** package provides normalization methods that mitigate comparability problems.

One way to normalize shortages is via the shortage standard deviation. For example, an estimate of the standard deviation of the expected shortage in the deterministic adjustment model can be calculated by

$$\hat{\sigma}_G = \sqrt{\hat{\sigma}_d^2 + \hat{\sigma}_s^2 - \hat{\rho}\hat{\sigma}_d\hat{\sigma}_s}.$$

R Code 11 calculates this estimate by applying `shortage_standard_deviation` on the deterministic adjustment fit of R Code 3.

R Code 11: Shortage standard deviation.

```
R> shortage_standard_deviation(da)

shortage_standard_deviation
                   27.64508
```

Then, normalized shortages can be calculated by

$$\hat{N}_{n,t} = \frac{\hat{G}_{n,t}}{\hat{\sigma}_G}.$$

In **markets**, normalized shortaged are obtained by calls to `normalized_shortages`. Figure 6a depicts the histogram of estimated normalized shortages of the deterministic adjustment fit. The normalization performed by `normalized_shortages` is common for all observations in the sample. An alternative normalization is given by

$$\hat{R}_{n,t} = \frac{\hat{G}_{n,t}}{\hat{S}_{n,t}},$$

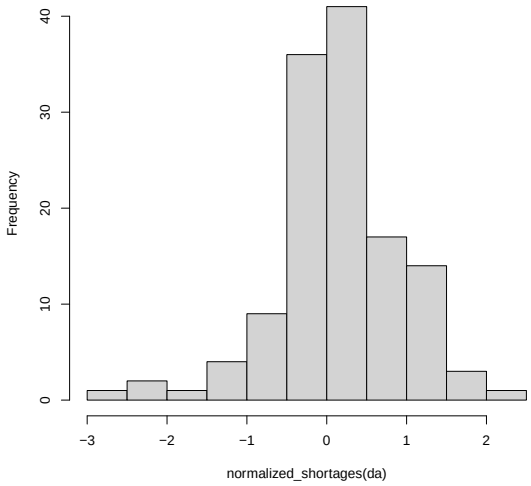
which calculates shortages relative to supply. Since estimated supplied quantities are observation specific, the normalization of $\hat{R}$ is idiosyncratic. Figure 6b depicts the histogram of estimated relative shortages of the deterministic adjustment fit obtained by the function `relative_shortages`.

The last unit-free way to examine estimated shortages in **markets**, albeit indirectly, is via the `shortage_probabilities` function. The function calculates an estimate $\hat{\pi}_S$ of Equation 5 based on the normality assumption of the shocks. The histogram of the estimated shortage probabilities for the deterministic adjustment model is depicted in Figure 6c.

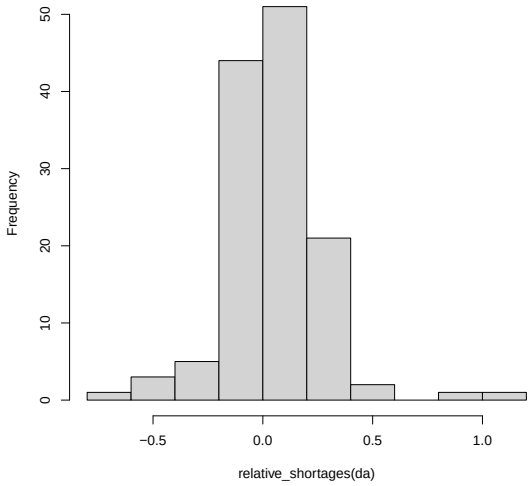
4.7. Marginal effects

Marginal effects give estimates of the impact of changes of control variables on market shortages. This can be helpful for variables that simultaneously affect multiple equations of the market model's system. For example, prices affect both demand and supply in the deterministic adjustment specification of R Code 3. Thus, a price change affects the system via two channels, one from the demand and one from the supply side. The overall effect of the change can be examined by looking at the partial derivative of the normalized shortage with respect to prices, i.e.,

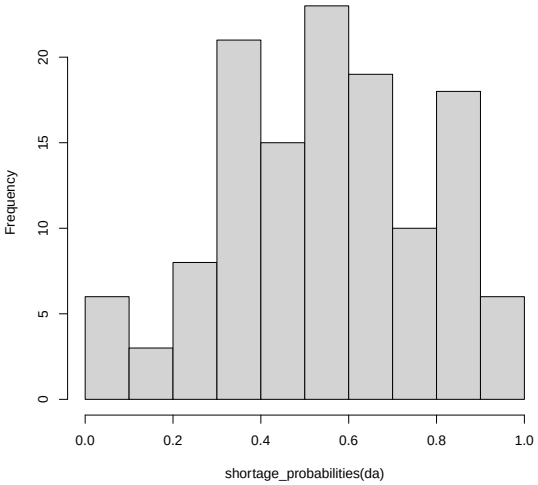
$$M_{\texttt{RM}} = \frac{\partial N}{\partial \texttt{RM}} = \frac{\alpha^d - \alpha^s}{\sqrt{\sigma_d^2 + \sigma_s^2 - 2\rho\sigma_d\sigma_s}}.$$



(a) Histogram of normalized shortages.



(b) Histogram of relative shortages.



(c) Histogram of shortage probabilities.

Figure 6: Normalized shortages, relative shortages, and shortage probabilities.

Had demand or supply been inelastic, only α^s or correspondingly α^d would have been present in the numerator of the marginal effect.

R Code 12 exemplifies the calculation $M_{\texttt{RM}}$ in **markets**. The second command calculates the impact of a marginal price change on the fitted shortages for both the stochastic and deterministic adjustment models using the function `shortage_marginal`. The price variable name `RM` is prefixed by B to indicate that price coefficients are present on both the demand and supply sides. Variables that are present on only one market side are prefixed differently. Supply side variables are prefixed with S, while demand side variables are prefixed with D.

R Code 12: Marginal effects.

```
R> fits <- c(sa = sa, da = da)
R> sapply(fits, function(m) shortage_marginal(m, "RM"))
```

```
    sa.B_RM     da.B_RM
-0.07934694 -0.15933786
```

The marginal effects of normalized shortages are state-independent because their partial derivative expression involves constant model parameters for all market states. Hence, the marginals of observations with large and small shortages are equal irrespective of the state of the market. In contrast, marginal effects on shortage probabilities are state-dependent. As a result, the marginal effects of observations with large shortages differ from those of observations with small shortages.

There are many ways to evaluate the market impact of a control change on shortage probabilities with state dependencies[7]. One of them is to calculate the average marginal effect, i.e.,

$$\mathbb{E}\frac{\partial\,\Phi(N)}{\partial \mathtt{RM}} = M_{\mathtt{RM}}\,\mathbb{E}\varphi(N),$$

where φ denotes the normal density of the difference of the shocks $(u^d - u^s)$ and Φ the normal distribution. An alternative evaluation involves calculating the marginal effect at the average shortage, namely

$$\frac{\partial\,\Phi(\mathbb{E}N)}{\partial \mathtt{RM}} = M_{\mathtt{RM}}\,\varphi(\mathbb{E}N).$$

Both marginal effects have the same sign with $M_{\mathtt{RM}}$ because the standard normal density is positive.

R Code 13 calculates probability marginals for `MA3DHF` and `CSHS` for the fitted stochastic and deterministic adjustment models. The first command of R Code 13 calculates the mean marginal effects of `MA3DHF`, while the second command calculates marginal effects of `CSHS` at the mean.

R Code 13: Marginal effects.

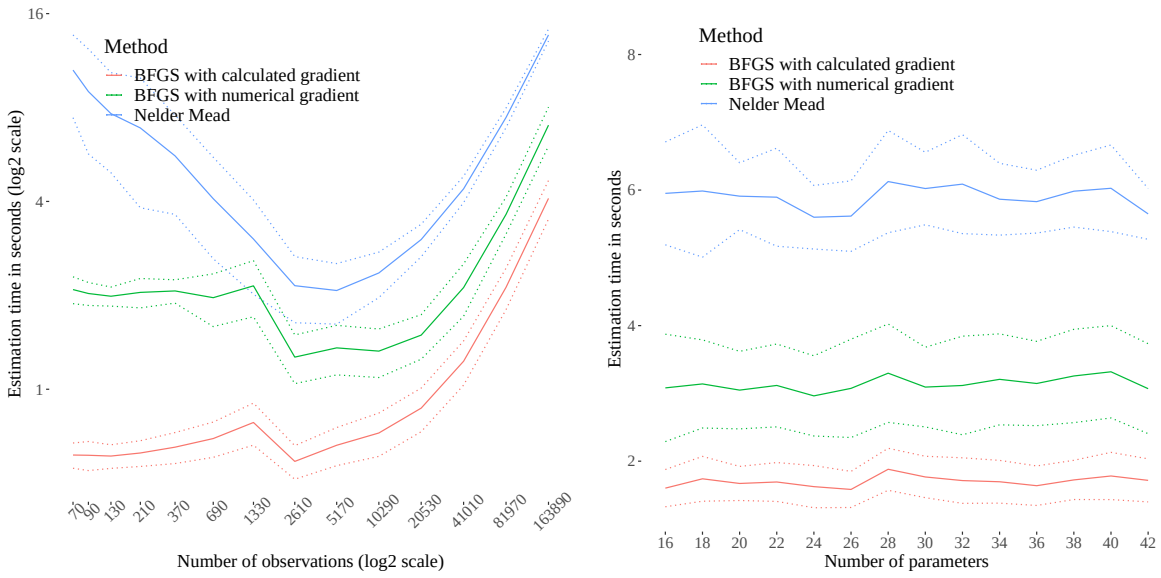
```
R> sapply(fits, function(m) shortage_probability_marginal(m, "MA3DHF"))

  sa.S_MA3DHF   da.S_MA3DHF
-0.0003189794 -0.0003719712

R> sapply(fits, function(m) {
+    shortage_probability_marginal(m, "CSHS", aggregate = "at_the_mean")
+  })

   sa.D_CSHS    da.D_CSHS
4.857798e-08 2.323340e-04
```

[7] In this respect, the discussion here parallels the evaluation of marginal effects of binary response probability models.

5. Estimation benchmarks

A major difficulty in estimating models for markets in disequilibrium comes from their computational complexity. Dorsey and Mayer (1995) classify the estimation of disequilibrium models among the most demanding econometric estimation problems, as the likelihoods of these models have poles and non-unique local maxima. The authors propose a genetic algorithm optimization method for estimating the basic model and compare its computational performance with Nelder-Mead. Instead, the classic estimation approach proposed by Maddala (1986) obtains maximum likelihood estimates using a global, iterative Newton method. Zilinskas and Bogle (2006) use random interval arithmetic optimization for locating global maxima. They apply the technique to the basic model with independent shocks using the dataset of Fair and Jaffee (1972) to assess its performance experimentally. Bowden (1978) considers the deterministic and stochastic adjustment models and proposes a re-parametrization that allows their estimation using more straightforward procedures. Instead, Quandt and Ramsey (1978) estimate the stochastic adjustment model with a methodology based on the moment generating function of the likelihood.

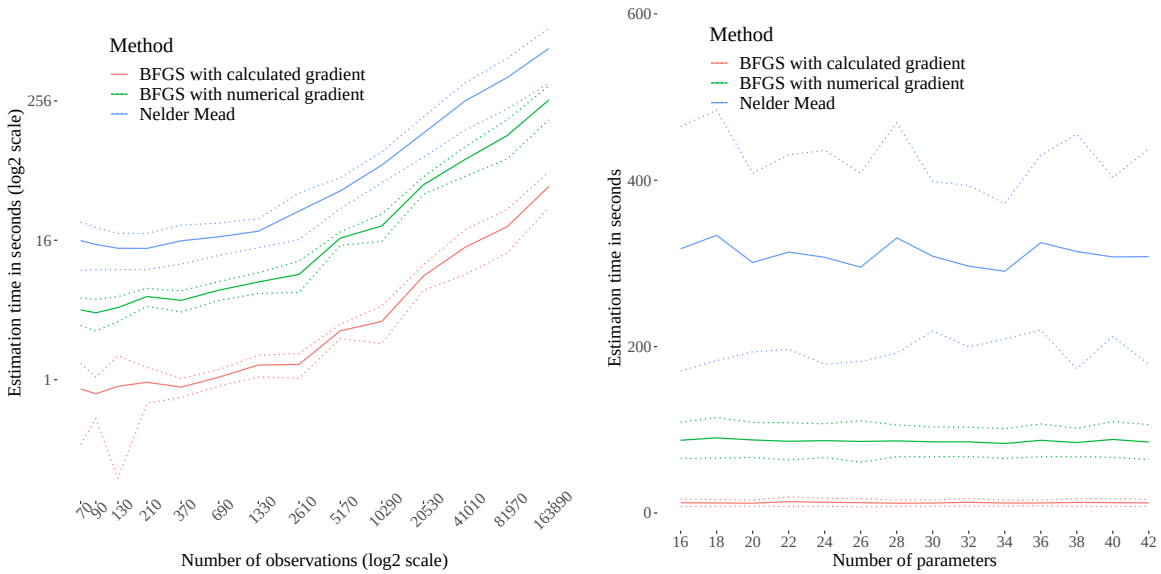
The benchmarks of this section compare the computational performance of the maximum likelihood estimation procedure for three of the optimization options that **markets** provides. Specifically, the analysis compares the mean estimation time of likelihood maximizations via BFGS with analytically calculated gradients, BFGS with numerically approximated gradients, and Nelder-Mead's simplex method. The execution time statistics are calculated using collected measurements from a series of benchmarking simulations performed in the CSC cluster of Goethe University. The benchmark data are collected from one water-cooled computing node with 2 Intel Xeon E5-2670 v2 (Ivy Bridge) CPUs, 10 cores per CPU, and hyper-threading (in total 40 logical processors). One logical processor is left unused to make space for other operations and minimize the scheduling competition between benchmark and system tasks.



(a) Over observations.

(b) Over parameters.

Figure 7: Equilibrium model estimation time benchmarks.



(a) Over observations.

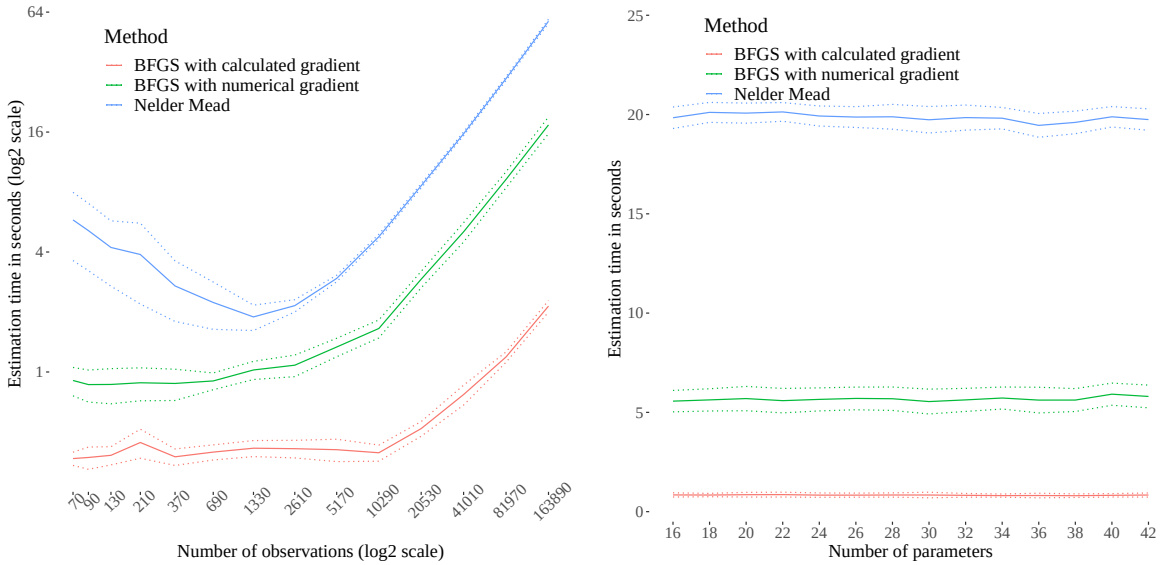
(b) Over parameters.

Figure 8: Basic model estimation time benchmarks.

The models are simulated using both random coefficients and samples drawn from normal distributions. The sample data are generated using the structural assumptions of each model and the randomly drawn coefficients. Estimating the models using BFGS with numerically approximated gradients is the most error-prone procedure among the three examined algorithms because numerical differentiation can fail nearby likelihood poles. For this reason, an untimed estimation using BFGS with numerically approximated gradients is executed for each simulation to ensure that the collected statistics accurately measure execution times and are not biased by estimation failures. If the estimation succeeds, the simulated data are used in timed executions. If the untimed estimation fails, new coefficients and data are regenerated.

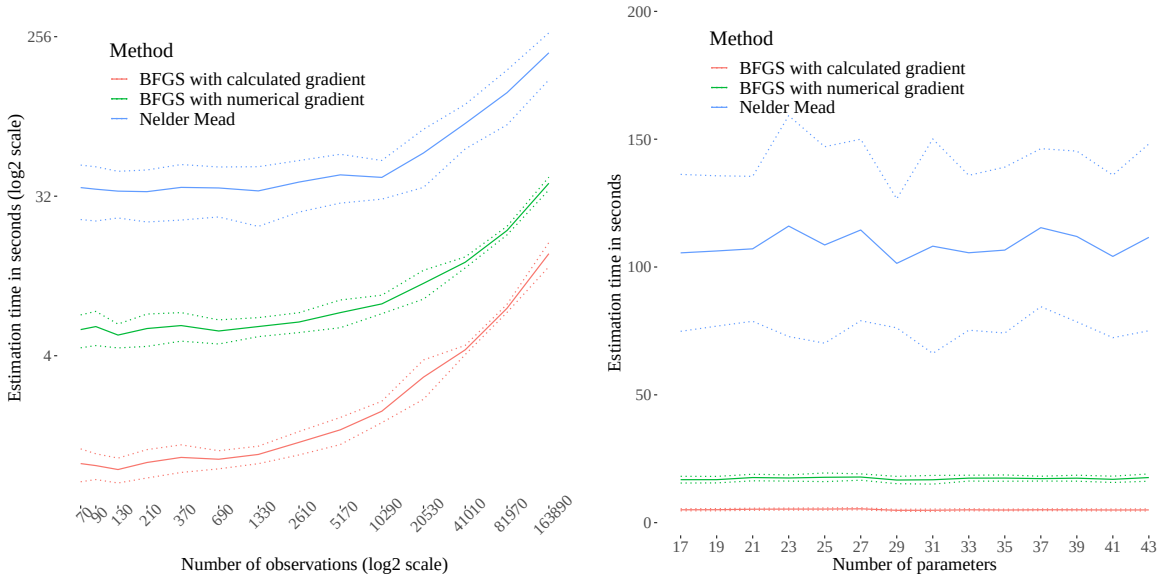
Each model is simulated 100 times for 14 different sample sizes and 14 different numbers of model parameters. The sample sizes grow exponentially according to the mapping $p \mapsto 5(2^{p+1}+10)$ for $p = 1, \ldots, 14$. The number of parameter grow linearly by adding one coefficient in the demand and one in the supply equations. It is ensured that the simulated data are economically well behaved in all simulation cases. If shortages or surpluses represent more than 90% of the sample, the simulated data are discarded, and a new simulation is initiated. Each well-behaved simulated dataset is used to estimate the model with all three optimization options to allow comparing of the resulting statistics. The execution time is saved at the end of each round. The saved time concerns only the estimation of the models and not their simulation or the calculation of standard errors. The estimation tolerance is kept constant for all optimization methods. The processors are warmed up using 2 untimed estimations performed at the beginning of the process.

Appendix B details the data generating process of each model. The **markets** package exposes this simulation functionality via the functions `simulate_data` and `simulate_model`. The results of the benchmarking simulations are depicted in Figures 7 (for the `equilibrium_model`), 8 (`diseq_basic`), 9 (`diseq_directional`), 10 (`diseq_deterministic_adjustment`), and 11 (`diseq_stochastic_adjustment`). The vertical axes of the figures measure the estimation



(a) Over observations. (b) Over parameters.

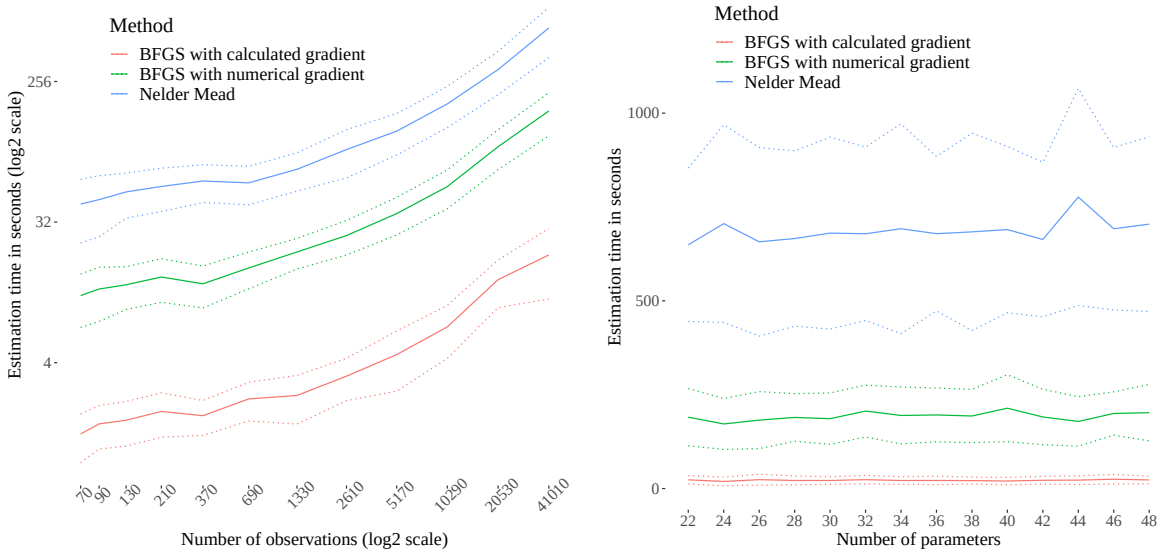
Figure 9: Directional model estimation time benchmarks.



(a) Over observations. (b) Over parameters.

Figure 10: Deterministic adjustment model estimation time benchmarks.

times of each optimization option. The horizontal axes of Figures 7a, 8a, 9a, 10a and 11a measure the number of observations of the simulated sample for a constant number of simulated



(a) Over observations.

(b) Over parameters.

Figure 11: Stochastic adjustment model estimation time benchmarks.

parameters[8]. The horizontal axes of Figures 7b, 8b, 9b, 10b and 11b measure the number of simulated parameters for a constant sample size of 41,010 observations. The points of solid lines represent mean estimation times over 100 estimations, and dotted lines depict one standard deviation intervals from the measured means.

The parameter benchmarks exhibit a similar pattern for all five models. Small changes in the number of estimated parameters do not significantly affect estimation times. Out of the three compared methods, Nelder-Mead results on average to the lengthiest estimation times and the greatest estimation time variability. BFGS with calculated gradients has the shortest estimation time and the least variability. Compared to BFGS with numerically approximated gradients, which offers the fastest alternative, BFGS with calculated gradients executes 1.84 times faster for the equilibrium, 7.02 the basic, 6.82 the directional, 3.41 the deterministic adjustment, and 8.7 for the stochastic adjustment model.

The ordering of the methods in terms of estimation time is the same for the benchmarks over a growing number of observations. In all cases, BFGS with analytically calculated gradients is the most efficient estimation option among those compared. The estimation times of all methods grow exponentially in the sample size for the basic, deterministic adjustment, and stochastic adjustment models. A different pattern is observed for the equilibrium and directional models and small sample sizes. For sample sizes below 5,170 observations, the estimation times for BFGS optimizations mostly remain constant, while the Nelder-mead estimation times reduce. Eventually, exponential growth is observed for larger sample sizes in the equilibrium and directional models.

[8]The number of parameters depends on the simulated model. The equilibrium and basic models use 14 parameters, namely 6 demand, 5 supply, 2 variances, and 1 correlation parameters. The directional model uses 13 parameters because prices cannot be used on both sides of the market. The deterministic adjustment model uses 15 parameters since it introduces an additional parameter in the price dynamics. Lastly, the stochastic adjustment model uses 20 parameters, 3 of which are introduced in the price dynamics, 1 additional variance, and 2 correlations.

6. Conclusion

This article introduces the R package **markets**. The package provides a common interface that unifies the estimation and harmonizes the post-estimation analysis of a diverse set of market models with various structural assumptions. Its methods are used for estimating, simulating, and analyzing five market models; a market clearing model and four short side rule models. In addition, the package provides methods to aggregate, summarize, and visualize the fitted models. Special emphasis is given to the analysis of market shortages.

The article begins by reviewing the equilibrium and disequilibrium econometrics upon which it relies. Then, it dwells into the details of its object-oriented design of the common interface implementation and compares the package's content with its closest software alternatives. Next, the core functionality of the package is exemplified via an empirical example using the classic dataset of Fair and Jaffee (1972). Based on this dataset, examples of estimating, visualizing, and summarizing the five market models of the package are presented. Finally, the obtained market fits are further used to illustrate the post estimation functionality capabilities of the package concerning fitted (unobserved) demanded and supplied quantities, as well as predicted shortages.

The estimation functionality of **markets** is based on analytic gradient and Hessian expressions for the likelihoods of the implemented market models. These expressions are not available in other statistical software. This implementation feature attributes to **markets** a computational edge in terms of estimation time efficiency. The article documents the computational benefits of the package's estimation functionality by presenting a series of estimation time statistics based on data collected from large-scale benchmarking simulations. Compared to the fastest estimation alternative not using the expressions employed by default in **markets**, the benchmark statistics indicate that one can estimate the models of the package from 1.84 (equilibrium model) to 8.7 (stochastic adjustment model) times faster.

Acknowledgments

The package was initially named **diseq** (see https://CRAN.R-project.org/package=diseq), but eventually grew to include general market estimation functionality following many suggestions of the R community. Additionally, I had a series of motivating discussions regarding the package's functionality with Oivind Nilsen. Alexander Ludwig has given me access to the CSC cluster of Goethe University, which I have used for the computationally intensive parts of the development process. I have also received helpful comments while presenting various applications and results using the functionality of **markets** from participants of the conference "Modelling with Big Data and Machine Learning: Measuring Economic Instability", the Goethe University's Management and Microeconomics Colloquium of 2020, and the Brown Bag seminar of the Leibniz Institute SAFE (https://safe-frankfurt.de/). Last but not least, various members of the CRAN project helped me to make the package available and subsequently improve it. An earlier version of the package was presented at the useR! 2021 conference.

References

Baird M, Daugherty L, Kumar K (2019). "Improving Estimation of Labor Market Disequilibrium Using Shortage Indicators, with an Application to the Market for Anesthesiologists." *IZA Discussion Paper 12129*, Institute of Labor Economics (IZA), Bonn. doi:10.2139/ssrn.3390116.

Balestra P, Varadharajan-Krishnakumar J (1987). "Full Information Estimations of a System of Simultaneous Equations with Error Component Structure." *Econometric Theory*, **3**(2), 223–246. doi:10.1017/s0266466600010318.

Berry S, Levinsohn J, Pakes A (1995). "Automobile Prices in Market Equilibrium." *Econometrica*, **63**(4), 841. doi:10.2307/2171802.

Board of Governors of the Federal Reserve System (US), Federal Reserve Board (1958–1980). "Federal Reserve Bulletin." URL https://fraser.stlouisfed.org/title/62.

Bolker B, R Core Team (2023). ***bbmle**: Tools for General Maximum Likelihood Estimation*. R package version 1.0.25.1, URL https://CRAN.R-project.org/package=bbmle.

Bowden RJ (1978). "Specification, Estimation and Inference for Models of Markets in Disequilibrium." *International Economic Review*, **19**(3), 711. doi:10.2307/2526335.

Broyden CG (1970). "The Convergence of a Class of Double-Rank Minimization Algorithms 1. General Considerations." *IMA Journal of Applied Mathematics*, **6**(1), 76–90. doi:10.1093/imamat/6.1.76.

Brunner D (2022). ***BLPestimatoR**: Performs a BLP Demand Estimation*. R package version 0.3.4, URL https://CRAN.R-project.org/package=BLPestimatoR.

Bulligan G, Busetti F, Caivano M, Cova P, Fantino D, Locarno A, Rodano ML (2017). "The Bank of Italy Econometric Model: An Update of the Main Equations and Model Elasticities." *Working Paper 1130*, Bank of Italy Temi di Discussione. doi:10.2139/ssrn.3040651.

Carbó-Valverde S, Rodríguez-Fernández F, Udell GF (2009). "Bank Market Power and SME Financing Constraints." *Review of Finance*, **13**(2), 309–340. doi:10.1093/rof/rfp003.

Carbó-Valverde S, Rodríguez-Fernández F, Udell GF (2016). "Trade Credit, the Financial Crisis, and SME Access to Finance." *Journal of Money, Credit and Banking*, **48**(1), 113–143. doi:10.1111/jmcb.12292.

Deaton A, Muellbauer J (1980). "An Almost Ideal Demand System." *The American Economic Review*, **70**(3), 312–326. doi:10.1017/cbo9780511805653.

Dorsey RE, Mayer WJ (1995). "Genetic Algorithms for Estimation Problems with Multiple Optima, Nondifferentiabiiity, and Other Irregular Features." *Journal of Business and Economic Statistics*, **13**(1), 53–66. doi:10.1080/07350015.1995.10524579.

Eddelbuettel D, François R (2023). ***RcppGSL**: **Rcpp** Integration for GNU **GSL** Vectors and Matrices*. R package version 0.3.13, URL https://CRAN.R-project.org/package=RcppGSL.

Fair RC (1971). *A Short-Run Forecasting Model of the United States Economy*. Heath Lexington Books. URL https://fairmodel.econ.yale.edu/RAYFAIR/pdf/1971EI.PDF.

Fair RC, Jaffee DM (1972). "Methods of Estimation for Markets in Disequilibrium." *Econometrica*, **40**(3), 497. doi:10.2307/1913181.

Fletcher R (1970). "A New Approach to Variable Metric Algorithms." *The Computer Journal*, **13**(3), 317–322. doi:10.1093/comjnl/13.3.317.

Galassi M, Gough B (2009). *GNU Scientific Library: Reference Manual*. 3rd edition. Network Theory. URL https://www.gnu.org/software/gsl.

Goldfarb D (1970). "A Family of Variable-Metric Methods Derived by Variational Means." *Mathematics of Computation*, **24**(109), 23–26. doi:10.1090/s0025-5718-1970-0258249-6.

Gould W, Pitblado J, Poi B (2010). *Maximum Likelihood Estimation with Stata*. 4th edition. Stata Press.

Henningsen A (2022). ***micEconAids**: Demand Analysis with the Almost Ideal Demand System (AIDS)*. R package version 0.6-20, URL https://CRAN.R-project.org/package=micEconAids.

Henningsen A, Hamann JD (2007). "**systemfit**: A Package for Estimating Systems of Simultaneous Equations in R." *Journal of Statistical Software*, **23**(4), 1–40. doi:10.18637/jss.v023.i04.

Hwang HS (1980). "A Test of a Disequilibrium Model." *Journal of Econometrics*, **12**(3), 319–333. doi:10.1016/0304-4076(80)90059-7.

Karapanagiotis P (2024). ***markets**: Estimation Methods for Markets in Equilibrium and Disequilibrium*. R package version 1.1.5, URL https://CRAN.R-project.org/package=markets.

Latshaw N, Guggisberg M (2020). ***Disequilibrium**: Disequilibrium Models*. R package version 1.1, URL https://CRAN.R-project.org/package=Disequilibrium.

Loberto M, Zollino F (2018). "Housing and Credit Markets in Italy in Times of Crisis." *Working Paper 1087*, Bank of Italy Temi di Discussione. doi:10.2139/ssrn.2914244.

Lokshin M, Sajaia Z (2004). "Maximum Likelihood Estimation of Endogenous Switching Regression Models." *Stata Journal*, **4**(3), 282–289(8). doi:10.1177/1536867x0400400306.

Maddala GS (1986). *Handbook of Econometrics*, volume 3, chapter Disequilibrium, Self-Selection, and Switching Models, pp. 1633–1688. Elsevier. doi:10.1016/S1573-4412(86)03008-8.

Maddala GS, Nelson FD (1974). "Maximum Likelihood Methods for Models of Markets in Disequilibrium." *Econometrica*, **42**(6), 1013. doi:10.2307/1914215.

Quandt RE, Ramsey JB (1978). "Estimating Mixtures of Normal Distributions and Switching Regressions." *Journal of the American Statistical Association*, **73**(364), 730–738. doi:10.1080/01621459.1978.10480085.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Shanno D (1970). "Conditioning of Quasi-Newton Methods for Function Minimization." *Mathematics of Computation*, **24**(111), 647–656. doi:10.1090/s0025-5718-1970-0274029-x.

United States President, Council of Economic Advisers (US) (1959–1978). "Economic Report of the President." URL https://fraser.stlouisfed.org/title/45.

Zeileis A, Croissant Y (2010). "Extended Model Formulas in R: Multiple Parts and Multiple Responses." *Journal of Statistical Software*, **34**(1), 1–13. doi:10.18637/jss.v034.i01.

Zellner A, Theil H (1962). "Three-Stage Least Squares: Simultaneous Estimation of Simultaneous Equations." *Econometrica*, **30**(1), 54. doi:10.2307/1911287.

Zilinskas J, Bogle IDL (2006). "Balanced Random Interval Arithmetic in Market Model Estimation." *European Journal of Operational Research*, **175**(3), 1367–1378. doi:10.1016/j.ejor.2005.02.013.

A. Installation

When building **markets** from its source code, parts of the package's native functionality are enabled or disabled based on the availability of two shared libraries in the target machine. Specifically, the package attempts to locate the `GSL` and the threading building blocks library (`tbb`), which are used in the native likelihood maximization routine for the equilibrium model. Native sources are compiled with the macros `_MARKETS_HAS_GSL_` and `_MARKETS_HAS_EXECUTION_POLICIES_` defined, if correspondingly `gsl` and `tbb` are located.

The function `maximize_log_likelihood` relies on the `gsl_multimin.h` header. If `gsl` is not located in the target system, `maximize_log_likelihood` becomes vacuous. The function is still exported by the package, but calling it results to void execution. This scenario cannot occur in builds for which the **RcppGSL** package (Eddelbuettel and François 2023) is installed. When installing from CRAN using `install.packages`, **RcppGSL** is also installed among other dependencies, and the **GSL** functionality is enabled by default.

The dependency on `tbb` comes from the `GCC` implementation of the `par_unseq` execution policy of the C++17 standard. At the moment of writing, the `par_unseq` feature is supported by two C++ compilers, namely `GCC` (version 9 and above) and `MSVC` (version 19.28 and above). The `GCC` implementation of `par_unseq` requires linking to the `tbb` library. Thus, **markets** examines whether `par_unseq` is available in the system during configuration. If this is the case, it enables the `-std=c++17` compilation flag, links to `tbb` when building with `GCC`, and enables some parallelization optimizations in the gradient calculations of `maximize_log_likelihood`. If `tbb` is not located, versions of the native sources with serialized gradient calculations are compiled.

B. Simulation details

Two simulation options are available for each market model provided by **markets**. The user can generate a dataset based on the stochastic process implied by a market model (*i*) with and (*ii*) without initializing a model object. Option (*i*) accommodates situations when the data is intended to be used with a single setup, while (*ii*) when used with multiple setups. These two options are accessed by correspondingly calling the methods `simulate_model` and `simulate_data`. The first method is a wrapper of the second method combined with a constructor call. The simulation functionality is employed by the unit tests of **markets**, some of its documentation examples and vignettes, and the benchmarking exercises of this article.

All simulation functions follow the baseline specifications for demand and supply equations given by Equations 1 and 2. In equilibrium model simulations, prices are not simulated. Instead, they are calculated so that the market clears, i.e.,

$$P_{n,t} = \frac{\sum_{j=1}^{k} \left(\eta_j^s - \eta_j^d\right) X_{n,t}^j + \beta_0^s - \beta_0^d + \sum_{j=1}^{k_s} \beta_j^s X_{n,t}^{j,s} - \sum_{j=1}^{k_d} \beta_j^d X_{n,t}^{j,d} + u_{n,t}^s - u_{n,t}^d}{\alpha^d - \alpha^s}.$$

In basic model's simulations, prices are simulated similarly to the remaining control variables. The demanded and supplied quantities are subsequently calculated, and the observed quantity is determined by the short side rule (Equation 4). In the directional model, prices are also simulated similarly to the remaining controls. Then, price differences are calculated, and the quantities are set according to the separation rule of the sample. If the condition $\Delta P_{n,t} \geq$

$0 \implies D_{n,t} \geq S_{n,t}$ is satisfied, the traded quantity is calculated using the supply equation. If not, it is calculated by the demand equation. An out-of-sample initial price value is drawn for the deterministic adjustment model, and the remaining prices are then sequentially generated by the rule

$$P_{n,t} = \frac{\sum_{j=1}^{k} \left(\eta_j^s - \eta_j^d\right) X_{n,t}^j + \beta_0^s - \beta_0^d + \sum_{j=1}^{k_s} \beta_j^s X_{n,t}^{j,s} - \sum_{j=1}^{k_d} \beta_j^d X_{n,t}^{j,d} - \gamma P_{n,t-1} + u_{n,t}^s - u_{n,t}^d}{\alpha^d - \alpha^s - \gamma}.$$

The procedure for the stochastic adjustment is similar to that of the deterministic adjustment model. The price generation rule with stochastic dynamics is given by

$$P_{n,t} = \frac{\left(\eta_j^s - \eta_j^d\right) X_{n,t}^j + \beta_0^s - \beta_0^d + \beta_j^s X_{n,t}^{j,s} - \beta_j^d X_{n,t}^{j,d} - \gamma P_{n,t-1} + \gamma\beta_0^p + \gamma\beta_j^p X_{n,t}^{j,p} + u_{n,t}^s - u_{n,t}^d + \gamma u_{n,t}^p}{\alpha^d - \alpha^s - \gamma},$$

where Einstein summation notation over j is used to save some space. The simulation methods perform various validity checks in the generated data and instruct the user to reparametrize the model if any of them fails. For instance, it is ensured that simulated samples of disequilibrium models do not exclusively contain demand or supply observations.

A call to `simulate_data` requires specifying the model that is to be simulated by passing the corresponding model string used in initialization calls as the first argument. The number of observations is set by the product of the arguments `nobs` (number of subjects) and `tobs` (number of time points). The model parameters are specified by `alpha_d`, `beta_d0`, `beta_d`, `eta_d`, `alpha_s`, `beta_s0`, `beta_s`, `eta_s`, `gamma`, `beta_p0`, `beta_p`, `sigma_d`, `sigma_s`, `sigma_p`, `rho_ds`, `rho_dp`, `rho_sp`. The argument names follow the notation of this article, and correspond to the symbols of Equations 1, 2, 4, 7 and 8. The default value of all variances is one, and the correlations is zero. The caller can optionally pass values for the `seed`, `verbose`, `price_generator`, and `control_generator` arguments. The last two options expect a function callback that is given an integer n and returns n randomly generated values. The default generators return standard normally distributed values.

The `simulate_model` call extends the calling convention of `simulate_data`. The given parameter and generator arguments passed down to `simulate_data` are specified as a list through the `simulation_parameters` input argument. Values for the optional arguments `seed` and the `verbose` can be specified separately from the `simulation_parameters` list. Any additional arguments given to `simulate_model` are passed down to the specified model's construction call.

C. Comparison of equilibrium estimates

The `equilibrium_model` can be estimated using different methods and tools having distinct characteristics. The two stage least square method is recommended because it is the least computationally intensive method. Maximum likelihood estimation can still be helpful when comparing equilibrium and disequilibrium models via information criteria. In most systems, there are either no, or negligible gains from resorting to the native optimization option that the package offers. For estimation exercises with large datasets, native optimization functionality can be faster in systems with many processors, such as computing clusters. Nevertheless, all approaches give similar results.

The exercise of this appendix exemplifies the above claim by comparing the estimates obtained by fitting the equilibrium model using different tools and methods for a simulated dataset with 20,000 observations. The simulated equilibrium model has, besides prices and a constant, two demand covariates (X_1^d and X_2^d), one supply covariate (X_1^s), and two market-wide covariates (X_1 and X_2). Moreover, it allows for temporal correlation between demand and supply shocks (ρ). R Code 14 uses `simulate_model` to simulate the model (see Appendix B for details on simulation functionality).

R Code 14: Equilibrium model simulation.

```
R> seed <- 25
R> parameters <- list(nobs = 4000, tobs = 10, alpha_d = -1.7,
+    beta_d0 = 14.9, beta_d = c(2.3, -1.2), eta_d = c(-1.3, -1.1),
+    alpha_s = 1.6, beta_s0 = 10.2, beta_s = c(-1.3), eta_s = c(2.5, 2.2),
+    sigma_d = 2.1, sigma_s = 2.5, rho_ds = -0.1)
R> mdl <- simulate_model("equilibrium_model", parameters, seed, verbose = 2)

Info: This is Equilibrium model.
```

R Code 15 uses the available options in **markets** to estimate the simulated model. The function `maximize_log_likelihood` wraps **GSL** calls to estimate the equilibrium model using `gsl_multimin_fdfminimizer_vector_bfgs2`. The arguments `objective_tolerance` and `gradient_tolerance` control the accuracy of the optimization. The `step` argument sets the first trial step size that the minimizer uses. See the **GSL** documentation (Galassi and Gough 2009) for more information about the used multidimensional minimization routines.

R Code 15: Equilibrium model estimation.

```
R> optim_fit <- estimate(mdl)
R> gsl_fit <- estimate(mdl, optimizer = "gsl", control = list(
+    step = 1e-0, maxit = 1e+4, objective_tolerance = 1e-2,
+    gradient_tolerance = 1e-2))
R> ls_fit <- estimate(mdl, method = "2SLS")
```

Table 3 summarizes the results. Parentheses report absolute differences between the estimated and simulated parameters. The last row calculates the average mean absolute error for each estimation option.

D. Model initialization

Section 4 uses the `equilibrium_model`, `dise_basic`, `dise_deterministic_adjustment`, `dise_directional`, and , `dise_stochastic_adjustment` to initialize and estimate the corresponding market models in a single call. Package **markets** provides methods to separate these two steps, which can be convenient in some workflows. Market models can be constructed without being estimated using the `new` function. Subsequently, the `market_fit` objects can be obtained by calling `estimate` with a previously constructed model object as an argument.

Coefficient	Simulated	ls	gsl	optim
D_P	−1.70	−1.7421 (0.0421)	−1.7421 (0.0421)	−1.7421 (0.0421)
D_CONST	14.90	14.9590 (0.0590)	14.9590 (0.0590)	14.9590 (0.0590)
D_Xd1	2.30	2.3250 (0.0250)	2.3250 (0.0250)	2.3250 (0.0250)
D_Xd2	−1.20	−1.2130 (0.0130)	−1.2130 (0.0130)	−1.2130 (0.0130)
D_X1	−1.30	−1.3582 (0.0582)	−1.3582 (0.0582)	−1.3582 (0.0582)
D_X2	−1.10	−1.1556 (0.0556)	−1.1556 (0.0556)	−1.1556 (0.0556)
S_P	1.60	1.5990 (0.0010)	1.5990 (0.0010)	1.5990 (0.0010)
S_CONST	10.20	10.2091 (0.0091)	10.2091 (0.0091)	10.2091 (0.0091)
S_Xs1	−1.30	−1.2922 (0.0078)	−1.2922 (0.0078)	−1.2922 (0.0078)
S_X1	2.50	2.5055 (0.0055)	2.5055 (0.0055)	2.5055 (0.0055)
S_X2	2.20	2.2197 (0.0197)	2.2197 (0.0197)	2.2197 (0.0197)
D_VARIANCE	4.41	4.5676 (0.1576)	4.5675 (0.1575)	4.5676 (0.1576)
S_VARIANCE	6.25	6.1995 (0.0505)	6.1993 (0.0507)	6.1995 (0.0505)
RHO	−0.10	−0.1090 (0.0090)	−0.1090 (0.0090)	−0.1090 (0.0090)
Mean abs. error	–	0.0366	0.0367	0.0366

Table 3: Comparison of equilibrium estimation methods and tools.

D.1. Initialization

The initialization arguments of the constructors for all models mostly coincide. Each model initialization requires specifying the model class, the used data frame, the identifiers of the dataset, the quantity and price variables, and the demand and supply right-hand side specifications. The construction operation for the stochastic adjustment model, which involves nontrivial price dynamics, additionally requires specifying the price equation. Furthermore, one can choose whether the initialized model should allow the shocks of the stochastic equations to be correlated and the verbosity level with which the operations of the constructed object should emit messages to the user.

R Code 16 constructs objects for each model using the corresponding initialization options of R Codes 2–6 of Section 4. The constructors create model formulas (see Section 4.2) using the `quantity`, `price`, `subject`, `time` (left-hand side of formula), `demand`, `supply`, and, if present, `price_dynamics` arguments (right-hand side of formula). These variables `quantity`, `price`, `subject`, and `time` are expected to be of type language and the variable `demand`, `supply`, and `price_dynamics` are expected to follow the syntax of formula. Indicator variables for factor type columns included in the market equations are automatically created by the constructors.

R Code 16: Model initialization.

```
R> eq <- new("equilibrium_model",
+    quantity = HS, price = RM, subject = ID, time = TREND,
+    demand = RM + TREND + W + CSHS + L1RM + L2RM + MONTH,
+    supply = RM + TREND + W + L1RM + MA6DSF + MA3DHF + MONTH,
+    house_data)
R> da <- new("diseq_deterministic_adjustment",
+    quantity = HS, price = RM, subject = ID, time = TREND,
```

```
+    demand = RM + TREND + W + CSHS + L1RM + L2RM + MONTH,
+    supply = RM + TREND + W + L1RM + MA6DSF + MA3DHF + MONTH,
+    house_data, verbose = 2)

Info: This is Deterministic Adjustment model.
Warning: Dropping 14 rows due to missing values.
Info: Dropping 1 row to generate 'LAGGED_RM'.
Info: Sample separated with 18 rows in excess supply and 111 rows in
  excess demand states.

R> dr <- new("diseq_directional",
+    quantity = HS, price = RM, subject = ID, time = TREND,
+    demand = TREND + W + CSHS + L1RM + L2RM,
+    supply = RM + TREND + W + MA6DSF + MA3DHF + MONTH,
+    house_data
+  )
R> bs <- new("diseq_basic",
+    quantity = HS, price = RM, subject = ID, time = TREND,
+    demand = RM + TREND + W + CSHS + L1RM + L2RM + MONTH,
+    supply = RM + TREND + W + L1RM + MA6DSF + MA3DHF + MONTH,
+    house_data, verbose = 2, correlated_shocks = FALSE
+  )

Info: This is Basic model.
Warning: Dropping 14 rows due to missing values.

R> sa <- new("diseq_stochastic_adjustment",
+    quantity = HS, price = RM, subject = ID, time = TREND,
+    demand = RM + TREND + W + CSHS + MONTH,
+    supply = RM + TREND + W + L1RM + L2RM + MA6DSF + MA3DHF + MONTH,
+    price_dynamics = TREND + L2RM + L3RM,
+    house_data |> dplyr::mutate(L3RM = dplyr::lag(RM, 3)),
+    correlated_shocks = FALSE
+  )
```

D.2. Model summaries

The `show` and `summary` functions display short and more extended information about constructed model objects in the standard output. R Code 17 gives an example of the displayed information. The output of these two commands is also displayed at the beginning of `market_fit` object summaries (see R Code 7).

R Code 17: Model objects' output operations.

```
R> show(dr)

Directional Model for Markets in Disequilibrium:
  Demand RHS        :   D_TREND + D_W + D_CSHS + D_L1RM + D_L2RM
```

```
  Supply RHS         :   S_RM + S_TREND + S_W + S_MA6DSF + S_MA3DHF +
    S_MONTH
  Short Side Rule    : HS = min(D_HS, S_HS)
  Separation Rule    : RM_DIFF >= 0 then D_HS >= S_HS
  Shocks             : Correlated
```

```
R> summary(sa)
```

```
Stochastic Adjustment Model for Markets in Disequilibrium:
  Demand RHS         :   D_RM + D_TREND + D_W + D_CSHS + D_MONTH
  Supply RHS         :   S_RM + S_TREND + S_W + S_L1RM + S_L2RM +
    S_MA6DSF + S_MA3DHF +  S_MONTH
  Price Dynamics RHS:   I(D_HS - S_HS) + TREND + L2RM + L3RM
  Short Side Rule    : HS = min(D_HS, S_HS)
  Shocks             : Independent
  Nobs               : 128
  Sample Separation  : Not Separated
  Quantity Var       : HS
  Price Var          : RM
  Key Var(s)         : ID, TREND
  Time Var           : TREND
```

D.3. Estimation

Model objects are estimated by calling `estimate`. For completeness, R Code 18 replicates the estimations of Section 4.3. Essentially, the elements of the list `estimation_options` in the calls of R Codes 2–6 are directly passed as input arguments when calling `estimate`.

R Code 18: Model estimation.

```
R> eq <- estimate(eq, control = list(maxit = 5e3))
R> da <- estimate(da, control = list(maxit = 5e3))
R> dr <- estimate(dr, method = "Nelder-Mead", control = list(maxit = 5e3))
R> start <- coef(eq)
R> start <- start[names(start) != "RHO"]
R> bs <- estimate(bs, start = start, control = list(maxit = 5e3))
R> sa <- estimate(sa, control = list(maxit = 5e3), standard_errors = c("W"))
```

The optimization method is selected by the `method` argument of the `estimate` function. When the passed value is among `"Nelder-Mead"`, `"BFGS"`, `"CG"`, `"L-BFGS-B"`, `"SANN"`, and `"Brent"`, the model is estimated using full information maximum likelihood. When the method is set equal `"2SLS"` for the equilibrium model the model is estimated using two-stage least squares.

The default value of the `gradient` argument is `"calculated"`, which instructs the estimate call to use analytically calculated gradient expressions whenever compatible with the optimization method. Alternatively, numerical gradient approximation can be used by setting `gradient` equal to `"numeric"`.

The argument `hessian` accepts one of the values `"skip"`, `"numerical"`, and `"calculated"`. The default is to use the `"calculated"` Hessian for the models that expressions are available

and the "numerical" Hessian in other cases. Calculated Hessian expressions are only available for the basic and directional models. The "skip" option abstains from calculating the Hessian.

If the Hessian calculation is not skipped, "homoscedastic", "heteroscedastic", or clustered standards errors can be calculated by setting the input argument standard_errors. The default value is "homoscedastic". If the option "heteroscedastic" is passed, the variance-covariance matrix is calculated using heteroscedasticity adjusted standard errors by the sandwich estimator. Clustered standard errors are calculated when a vector with variable names is supplied (see, e.g., the stochastic adjustment model estimation in R Code 18). In this case, the variance-covariance matrix is calculated by grouping the score matrix based on the passed variables.

Affiliation:

Pantelis Karapanagiotis
Group of Economics and Philosophy
EBS University
Rheingaustrasse 1
65375 Oestrich-Winkel, Germany
and
Leibniz Institute for Financial Research SAFE
E-mail: karapanagiotis@ebs.edu
URL: https://www.pikappa.eu/

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
February 2024, Volume 108, Issue 2 *Submitted:* 2022-06-15
doi:10.18637/jss.v108.i02 *Accepted:* 2023-06-21