Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

OPSR: A Package for Estimating Ordered Probit Switching Regression Models in R

Daniel Heimgartner
ETH Zürich

Xinyi Wang
MIT Boston

Abstract

Selection bias may arise if unobserved factors simultaneously influence the selection process for who gets treated (or not), and the outcome of (not) receiving the treatment. Different methods exist to correct for this bias depending on whether longitudinal or cross-sectional data is available. A possible cure in the latter case (where the counterfactual treatment outcome is never observed) is to explicitly account for the arising error correlation and estimate the covariance matrix of the selection and outcome processes. This is known as endogenous switching regression. The R package **OPSR** introduced in this article provides an easy-to-use, fast and memory efficient interface to ordered probit switching regression, accounting for self-selection into an ordinal treatment. It handles log-transformed outcomes which need special consideration when computing conditional expectations and thus treatment effects. Besides the usual R modeling methods (`update()`, `summary()`, `predict()`, etc.) post-estimation routines to compute and visualize (weighted) treatment effects are available. Convergence diagnostics and exclusion-restriction tests assist the analyst in verifying that the model is empirically identified, and a built-in simulation study benchmarks estimator performance under varying identification conditions.

Keywords: ordered probit switching regression, endogenous switching regression, Heckman selection, selection bias, treatment effect, R.

1. Introduction

The goal of the program evaluation literature is to estimate the effect of a treatment program (e.g., a new policy, technology, medical treatment, or agricultural practice) on an outcome. To evaluate such a program, the "treated" are compared to the "untreated". In an experimental setting, the treatment can be (randomly) assigned by the researcher. However, in an observational setting, the treatment is not always exogenously prescribed but rather self-selected.

This gives rise to a selection bias when factors (either observed or unobserved) influencing the treatment adoption also influence the outcome (also known as selection on observables and unobservables). Simple group comparisons no longer yield an unbiased estimate of the treatment effect. In more technical terms, the counterfactual outcome of the treated ("if they had not been treated") does not necessarily correspond to the factual outcome of the untreated. For example, cyclists riding without a helmet (the "untreated") might be young and have a risk-seeking tendency. We therefore potentially overestimate the benefit of wearing a helmet if we compare the accident rate and/or crash severity rate between those who wear and do not wear helmets directly. Even if we control for age in the comparison, variables such as risk-seeking are not readily measured, remain part of the error term in applied research, and are thus a leading cause of selection bias.

To properly account for the selection bias, various techniques exist, both for longitudinal and cross-sectional data. In the former case, difference in differences is a widely adopted measure. In the latter case, instrumental variables, propensity score matching, regression-discontinuity design, and the endogenous switching regression model have been applied (Wang and Mokhtarian 2024). The endogenous switching regression model, an extension of Heckman's classic sample selection model, is particularly well-suited to correct for both selection on observables and unobservables (unlike other methods which only address and correct for selection on observables).

The seminal work by Heckman (1979) proposed a two-part model to address the selection bias that often occurs when modeling a continuous outcome which is only observable for a subpopulation. A very nice exposition of this model is given in Cameron and Trivedi (2005, Chapter 16). The classical Heckman model consists of a probit selection equation and a continuous outcome equation. A natural extension is then switching regression, where the population is partitioned into different groups (regimes) and separate parameters are estimated for the continuous outcome process of each group. This model is originally known as the Roy model (Cameron and Trivedi 2005) or Tobit-5 model (Amemiya 1985). These classical models (the Tobit models for truncated, censored or interval data and their extensions) are implemented in various environments for statistical computing and in R's (R Core Team 2017) **sampleSelection** package (Toomet and Henningsen 2008).

Many different variants can then be derived by placing different distributional assumptions on the errors and/or by varying how the latent process manifests into observed outcomes (i.e., the dependent variables can be binary, ordinal, censored, or continuous). These variants are more generally known as conditional mixed-process (CMP) models: a broad family involving two or more equations featuring a joint error distribution assumed to be multivariate normal. The Stata (StataCorp 2023) command `cmp` (Roodman 2011) can fit such models. The variant at the heart of this paper is an ordered probit switching regression (OPSR) model, with ordered treatments and continuous outcome. Throughout the text we use the convention that OPSR refers to the general methodology, while **OPSR** refers specifically to the package.

OPSR is available as a Stata command, `oheckman` (Chiburis and Lokshin 2007), which, however, does not allow distinct specifications for the continuous outcome processes (i.e., the same explanatory variables must be used for all treatment groups). The relatively new R package **switchSelection** (Potanin 2024) can estimate multivariate and multinomial sample selection and endogenous switching models with multiple outcomes. These models are systems of ordinal, continuous and multinomial equations and thus nest OPSR as a special case.

OPSR is tailored to one particular method, easy to use (understand, extend and maintain), fast and memory efficient. Unlike the implementations mentioned, this approach accommodates log-transformed continuous outcomes. Log transformation is a widely used technique in real-world applications to enhance data normality and meet model assumptions. In multi-layer models like OPSR, special consideration is required for computing conditional expectations on the original scale (i.e., back-transform from the log scale) to ensure meaningful real-world interpretations. **OPSR** follows R's implicit modeling conventions (by providing a formula interface to a fitter function and by extending the established generics such as `summary()`, `predict()`, `update()`, `anova()`, `plot()` among others) and produces production-grade output tables. Meanwhile, it is easy to compute and visualize treatment effects. This work generalizes the learnings from Wang and Mokhtarian (2024) and makes the OPSR methodology readily available. The mathematical notation presented here translates to code almost verbatim which hopefully serves a pedagogical purpose for the curious reader.

The accommodation of log-transformed outcomes in addition to distinct specifications for the continuous outcome processes makes **OPSR** more powerful than Stata's `oheckman` command. Compared to **switchSelection**, **OPSR** is tailored to one form of switching selection and supports the extended **Formula** syntax. Its methods provide more detailed insights for this particular model (inspired by `oheckman`) and provide tailored post-estimation routines such as the computation and visualization of factual estimates under the observed treatment status, counterfactual estimates under hypothetical treatment status and treatment effects. Further, tools to diagnose convergence and identification tests are implemented, helping the modeler to assess optimality conditions as well as exclusion restrictions. We therefore believe that **OPSR** is the most powerful and easiest to use implementation if modelers specifically wish to account for selection bias and calculate treatment effects for interventions with an ordinal nature.

The remainder of this paper is organized as follows: Section 2 outlines the ordered probit switching regression model, lists all the key formulas underlying the software implementation, discusses identification, and details **OPSR**'s architecture. In Section 3 the key functionality is demonstrated both on simulated data and the data from Wang and Mokhtarian (2024) which we use to reproduce their core model. Section 4 presents a simulation study replicating Chiburis and Lokshin (2007) for the three-regime switching case, comparing estimates and diagnostics performance under baseline conditions and several identification failure modes. The case study in Section 5 leverages tracking data from the TimeUse+ study (Winkler, Meister, and Axhausen 2024) investigating telework treatment effects on weekly distance traveled. There, we also compare the OPSR model to a model not accounting for error correlation and discuss the implications for treatment effects. The summary in Section 6 concludes.

2. Model and software

In the following, we outline the ordered probit switching regression model as well as list all the key formulas underlying the software implementation. **OPSR** follows the R-typical formula interface to a workhorse fitter function. Its architecture is detailed after the mathematical part.

2.1. The OPSR model

Likelihood and conditional expectation

As alluded to above, OPSR contains two layers: one process governs the ordinal outcome and separate processes (for each ordinal outcome) govern the continuous outcomes. The ordinal outcome can also be thought of as a regime or treatment. In the subsequent exposition, we will refer to the two processes as *selection* and *outcome* process.

We borrow the notation from Wang and Mokhtarian (2024) where also all the derivations are detailed. For a similar exhibition, Chiburis and Lokshin (2007) can be consulted. Individual subscripts are suppressed throughout, for simplicity.

Let $\mathcal{Z}$ be a latent propensity governing the selection outcome

$$\mathcal{Z} = \boldsymbol{W}\boldsymbol{\gamma} + \epsilon, \tag{1}$$

where $\boldsymbol{W}$ represents the vector of attributes of an individual, $\boldsymbol{\gamma}$ is the corresponding vector of parameters and $\epsilon \sim \mathcal{N}(0, 1)$ a normally distributed error term.

As $\mathcal{Z}$ increases and passes some unknown but estimable thresholds, we move up from one ordinal treatment to the next higher level

$$Z = j \quad \text{if } \kappa_{j-1} < \mathcal{Z} \leq \kappa_j, \tag{2}$$

where Z is the observed ordinal selection variable, $j = 1, \ldots, J$ indexes the ordinal levels of Z, and κ_j are the thresholds (with $\kappa_0 = -\infty$ and $\kappa_J = \infty$). Hence, there are $J-1$ thresholds to be estimated. The probability that an individual self-selects into treatment group j is

$$\begin{aligned} \mathsf{P}[Z = j] &= \mathsf{P}[\kappa_{j-1} < \mathcal{Z} \leq \kappa_j] \\ &= \mathsf{P}[\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma} < \epsilon \leq \kappa_j - \boldsymbol{W}\boldsymbol{\gamma}] \\ &= \Phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \Phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma}), \end{aligned} \tag{3}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

The outcome model for the j^{th} treatment group is expressed as

$$y_j = \boldsymbol{X_j}\boldsymbol{\beta_j} + \eta_j, \tag{4}$$

where y_j is the observed continuous outcome, $\boldsymbol{X_j}$ the vector of observed explanatory variables associated with the j^{th} outcome model, $\boldsymbol{\beta_j}$ is the vector of associated parameters, and $\eta_j \sim \mathcal{N}(0, \sigma_j^2)$ is a normally distributed error term. At this point it should be noted that $\boldsymbol{X_j}$ and $\boldsymbol{W}$ may share some explanatory variables but not all, due to identification problems otherwise (Chiburis and Lokshin 2007).

The key assumption of OPSR is now that the errors of the selection and outcome models are jointly multivariate normally distributed

$$\begin{pmatrix} \epsilon \\ \eta_1 \\ \vdots \\ \eta_j \\ \vdots \\ \eta_J \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho_1\sigma_1 & \cdots & \rho_j\sigma_j & \cdots & \rho_J\sigma_J \\ \rho_1\sigma_1 & \sigma_2^2 & & & & \\ \vdots & & \ddots & & & \\ \rho_j\sigma_j & & & \sigma_j^2 & & \\ \vdots & & & & \ddots & \\ \rho_J\sigma_J & & & & & \sigma_J^2 \end{pmatrix} \right), \tag{5}$$

where ρ_j represents the correlation between the errors of the selection model (ϵ) and the j^{th} outcome model (η_j). If the covariance matrix should be diagonal (i.e., no error correlation), no selection-bias exists and the selection and outcome models can be estimated separately.

As shown in Wang and Mokhtarian (2024), the log-likelihood of observing all individuals self-selecting into treatment j and choosing continuous outcome y_j can be expressed as

$$\ell(\theta \mid \boldsymbol{W}, \boldsymbol{X_j}) = \sum_{j=1}^{J} \sum_{\{j\}} \left\{ \ln \left[\frac{1}{\sigma_j} \phi \left(\frac{y_j - \boldsymbol{X_j}\boldsymbol{\beta_j}}{\sigma_j} \right) \right] \quad + \right.$$
$$\left. \ln \left[\Phi \left(\frac{\sigma_j(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \rho_j(y_j - \boldsymbol{X_j}\boldsymbol{\beta_j})}{\sigma_j\sqrt{1-\rho_j^2}} \right) - \Phi \left(\frac{\sigma_j(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma}) - \rho_j(y_j - \boldsymbol{X_j}\boldsymbol{\beta_j})}{\sigma_j\sqrt{1-\rho_j^2}} \right) \right] \right\} \tag{6}$$

where $\sum_{\{j\}}$ means the summation of all the cases belonging to the j^{th} selection outcome, $\phi(\cdot)$ and $\Phi(\cdot)$ are the density and cumulative distribution function of the standard normal distribution.

The conditional expectation can be expressed as

$$\begin{aligned} \mathsf{E}[y_j \mid Z = j] &= \boldsymbol{X_j}\boldsymbol{\beta_j} + \mathsf{E}[\eta_j \mid \kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma} < \epsilon \leq \kappa_j - \boldsymbol{W}\boldsymbol{\gamma}] \\ &= \boldsymbol{X_j}\boldsymbol{\beta_j} - \rho_j\sigma_j \frac{\phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma})}{\Phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \Phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma})}, \end{aligned} \tag{7}$$

where the negative fraction $(-\frac{\phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma})}{\Phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \Phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma})})$ is the ordered probit switching regression model counterpart to the inverse Mills ratio (IMR) term of a binary switching regression model (because of its resemblance, we will also refer to this fraction as inverse Mills ratio in the OPSR case). We immediately see that regressing $\boldsymbol{X_j}$ on y_j leads to an omitted variable bias if $\rho_j \neq 0$, which is the root cause of the selection bias. However, the IMR can be pre-computed based on an ordered probit model and then included in the second stage regression, which describes the Heckman correction (Heckman 1979). Note, however, that since the Heckman two-step procedure includes an estimated quantity in the second step regression, the resulting OLS standard errors and heteroskedasticity-robust standard errors are incorrect (Greene 2002).

To obtain unbiased treatment effects, we must further evaluate the "counterfactual outcome", which reflects the expected outcome under a counterfactual treatment (i.e., for $j' \neq j$)

$$\begin{aligned} \mathsf{E}[y_{j'} \mid Z = j] &= \boldsymbol{X_{j'}}\boldsymbol{\beta_{j'}} + \mathsf{E}[\eta_{j'} \mid \kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma} < \epsilon \leq \kappa_j - \boldsymbol{W}\boldsymbol{\gamma}] \\ &= \boldsymbol{X_{j'}}\boldsymbol{\beta_{j'}} - \rho_{j'}\sigma_{j'} \frac{\phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma})}{\Phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \Phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma})}. \end{aligned} \tag{8}$$

Let us now assume that $y_j = \ln(Y_j + \delta)$ in the previous equations, i.e., the continuous outcome was log-transformed, as is common in regression analysis. In such cases, Equations 7–8 provide the conditional expectation of the *log-transformed* outcome. Therefore we need to back-transform $Y_j = \exp(y_j) - \delta$, which yields

$$\mathsf{E}[Y_j \mid Z = j] = \exp\left(\boldsymbol{X_j}\boldsymbol{\beta_j} + \frac{\sigma_j^2}{2} \right) \left[\frac{\Phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma} - \rho_j\sigma_j) - \Phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma} - \rho_j\sigma_j)}{\Phi(\kappa_j - \boldsymbol{W}\boldsymbol{\gamma}) - \Phi(\kappa_{j-1} - \boldsymbol{W}\boldsymbol{\gamma})} \right] - \delta \tag{9}$$

for the factual case, and

$$\mathsf{E}[Y_{j'} \mid Z = j] = \exp\left(\boldsymbol{X_{j'}}\boldsymbol{\beta_{j'}} + \frac{\sigma_{j'}^2}{2}\right)\left[\frac{\Phi(\kappa_j - \boldsymbol{W\gamma} - \rho_{j'}\sigma_{j'}) - \Phi(\kappa_{j-1} - \boldsymbol{W\gamma} - \rho_{j'}\sigma_{j'})}{\Phi(\kappa_j - \boldsymbol{W\gamma}) - \Phi(\kappa_{j-1} - \boldsymbol{W\gamma})}\right] - \delta \tag{10}$$

for the counterfactual case (Wang and Mokhtarian 2024).

Identification

The identification of ρ_j is a critical practical concern. From Equation 7 we see that if the inverse Mills ratio term is collinear with $\boldsymbol{X_j}$, the outcome regression cannot separately identify $\boldsymbol{\beta_j}$ and $\rho_j\sigma_j$. This collinearity arises whenever $\boldsymbol{W}$ and all $\boldsymbol{X_j}$ share exactly the same variables: the correction term becomes a near-linear function of $\boldsymbol{X_j\beta_j}$, so ρ_j is identified only through the curvature of $\Phi(\cdot)$ — a distributional assumption rather than genuine variation in the data (Chiburis and Lokshin 2007). This motivates the standard requirement that at least one variable in $\boldsymbol{W}$ must be excluded from all $\boldsymbol{X_j}$: the *exclusion restriction* or *instrument*.

The severity of functional-form-only identification depends on the regime structure. Chiburis and Lokshin (2007) show that for *interior* regimes (two finite thresholds κ_{j-1} and κ_j), the generalized inverse Mills ratio is nearly linear in $\boldsymbol{W\gamma}$, rendering functional-form identification practically useless at empirically relevant sample sizes. For *exterior* regimes (one infinite threshold), the nonlinearity is stronger, but still fragile. The consequence is that a weak or absent exclusion restriction leads to inflated standard errors for ρ_j, biased estimates of $\boldsymbol{\beta_j}$, and unreliable treatment effects — even if the optimizer converges without numerical warning.

Three failure modes arise in practice, with different consequences (foreshadowing the simulation results from Section 4):

1. **No exclusion restriction**: all variables in $\boldsymbol{W}$ also appear in all $\boldsymbol{X_j}$. Identification relies entirely on functional form. Full information maximum likelihood (FIML) estimates are downward biased, with coverage collapsing well below the nominal 95%, while the two-step estimator has enormous variance.

2. **Weak instrument**: the excluded variable has negligible first-stage predictive power for Z. This case is numerically similar to the no-exclusion case: the condition number of the Hessian inflates and the profile likelihood for ρ_j is near-flat.

3. **Invalid exclusion restriction**: the excluded variable directly affects y_j and should appear in $\boldsymbol{X_j}$. Both FIML and the two-step estimator converge confidently to the wrong answer (bias can exceed 80% and coverage collapses to 0%). No model-internal diagnostic can detect this failure mode — it requires substantive subject-matter reasoning.

OPSR provides two complementary tools for failure modes (1) and (2): a likelihood-ratio test for instrument relevance, `opsr_lr_instrument()`, and a per-regime profile log-likelihood for ρ_j, `opsr_profile_rho()`, combined in `opsr_test_exclusion_restriction()`. These are demonstrated in Section 3.2 and Section 5.2. The simulation evidence for all three failure modes is presented in Section 4.

2.2. Software

This concludes the mathematical treatment and we briefly outline **OPSR**'s architecture which can be conceptualized as follows:

- We provide the usual formula interface to specify a model. To allow for multiple parts and multiple responses, we rely on the **Formula** package (Zeileis and Croissant 2010).
- After parsing the formula object, checking the user inputs and computing the model matrices, the Heckman two-step estimator is called in `opsr_2step()` to generate reasonable starting values.
- These are then passed together with the data to the basic computation engine `opsr.fit()`. The main estimates are retrieved using maximum likelihood estimation by passing the log-likelihood function `loglik_cpp()` (Equation 6) to `maxLik()` from the **maxLik** package (Henningsen and Toomet 2011). No analytical gradient or Hessian is implemented, so `maxLik()` uses numerical finite-difference approximations by default. Reported standard errors and Wald statistics are derived from the negative inverse of this numerical Hessian. Near identification failure the Hessian may be numerically imprecise, which is flagged by the condition number diagnostic in `opsr_diagnostics()`.
- All the above calls are nested in the main interface `opsr()` which returns an object of class '`opsr`'. Several methods then exist to post-process this object as illustrated below.

The likelihood function `loglik_cpp()` is implemented in C++ using **Rcpp** (Eddelbuettel and Balamuta 2018) and relying on the data types provided by **RcppArmadillo** (Eddelbuettel and Sanderson 2014). Parallelization is available using OpenMP. This makes **OPSR** both fast and memory efficient (as data matrices are passed by reference).

3. Illustrations

We first illustrate how to specify a model using **Formula**'s extended syntax and simulated data and demonstrate the main functionality of the package (Section 3.1). We then demonstrate the convergence diagnostics and identification tests (Section 3.2), followed by a replica of Wang and Mokhtarian (2024) illustrating nuances of the interface (Section 3.3).

3.1. OPSR core

Let us simulate data from an OPSR process with three ordinal outcomes and distinct design matrices $\boldsymbol{W}$ and $\boldsymbol{X}$ (where $\boldsymbol{X} = \boldsymbol{X_j}\ \forall j$) by

```
R> sim_dat <- opsr_simulate(seed = 0)
R> dat <- sim_dat$data
R> head(dat)
```

```
  ys    yo      xs1    xs2    xo1     xo2
1  2 -1.26  0.44435 -0.538  1.263 -0.2869
2  2  3.80  0.01193  0.497 -0.326  1.8411
3  1  3.95 -0.00928 -1.442  1.330 -0.1568
4  2 -1.68 -0.30238 -1.113  1.272 -1.3898
```

```
5  1  1.50  0.49236 -1.015  0.415 -1.4731
6  2  2.20 -0.60272  0.567 -1.540 -0.0695
```

where `ys` is the selection dependent variable (or treatment group), `yo` the outcome dependent variable and `xs` respectively `xo` the corresponding explanatory variables.

Models are specified symbolically. A typical model has the form `ys | yo ~ terms_s | terms_o1 | terms_o2 | ...` where the `|` separates the two responses and process specifications. If the user wants to specify the same process for all continuous outcomes, a single outcome specification is enough (`ys | yo ~ terms_s | terms_o`). Hence the minimal `opsr()` interface call reads

```
R> fit <- opsr(ys | yo ~ xs1 + xs2 | xo1 + xo2, data = dat,
+    printLevel = 0)
```

where `printLevel = 0` omits working information during maximum likelihood iterations.

As usual, the fitter function does the bare minimum model estimation while inference is performed in a separate call to

```
R> summary(fit)

Call:
opsr(formula = ys | yo ~ xs1 + xs2 | xo1 + xo2, data = dat, printLevel = 0)

BFGS maximization, 104 iterations
Return code 0: successful convergence
Runtime: 0.339 secs
Max. absolute gradient: 2.75e-05
Hessian negative definite: TRUE
Condition number (Hessian): 31.9
Number of regimes: 3
Number of observations: 1000 (152, 507, 341)
Estimated parameters: 19

Log-Likelihood: -2016
AIC: 4071
BIC: 4164
Pseudo R-squared (EL): 0.506
Pseudo R-squared (MS): 0.456
Multiple R-squared: 0.815 (0.836, 0.762, 0.847)

Estimates:
               Estimate Std. error t value Pr(> t)
kappa1          -1.9618     0.0933  -21.02 < 2e-16 ***
kappa2           0.8826     0.0615   14.34 < 2e-16 ***
s_xs1            0.9373     0.0570   16.44 < 2e-16 ***
s_xs2            1.4964     0.0713   21.00 < 2e-16 ***
o1_(Intercept)   0.9877     0.1457    6.78 1.2e-11 ***
```

```
o1_xo1           2.0512     0.0931   22.02 < 2e-16 ***
o1_xo2           1.0133     0.0711   14.25 < 2e-16 ***
o2_(Intercept)   0.9574     0.0463   20.67 < 2e-16 ***
o2_xo1          -0.9884     0.0435  -22.70 < 2e-16 ***
o2_xo2           1.5545     0.0412   37.72 < 2e-16 ***
o3_(Intercept)   1.0028     0.0911   11.01 < 2e-16 ***
o3_xo1           1.5766     0.0561   28.13 < 2e-16 ***
o3_xo2          -1.9227     0.0527  -36.48 < 2e-16 ***
sigma1           1.0442     0.0534   19.57 < 2e-16 ***
sigma2           1.0478     0.0316   33.11 < 2e-16 ***
sigma3           1.1717     0.0430   27.28 < 2e-16 ***
rho1             0.1696     0.1380    1.23 0.21912
rho2             0.3482     0.0639    5.45 5.2e-08 ***
rho3             0.3699     0.1076    3.44 0.00059 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Wald chi2 (null): 5060 on 8 DF, p-value: < 0
Wald chi2 (rho): 42.6 on 3 DF, p-value: < 0
```

The presentation of the model results is fairly standard and should not warrant further explanation, with the following exceptions:

1. The number of regimes is reported along with the absolute counts per regime.

2. Pseudo R-squared (EL) is determined by comparing the log-likelihood of the specified model to that of the "equally likely" model, while Pseudo R-squared (MS) is obtained by comparing the log-likelihood of the specified model to that of the "market-share" model. These indicators reflect the goodness of fit for the selection process. The multiple R-squared is reported for all continuous outcomes collectively and for the regimes separately in brackets (i.e., only considering the continuous observations belonging to the respective treatment regime). These indicators reflect the goodness of fit for the outcome processes.

3. Coefficient names are based on the variable names as passed to the formula specification, except that `"s_"` is prepended to the selection coefficients, `"o[0-9]_"` to the outcome coefficients and the structural components `"kappa", "sigma", "rho"` (aligning with the letters used in Equation 6) are hard-coded (but can be over-written).

4. The coefficients table reports robust standard errors based on the sandwich covariance matrix as computed with help of the **sandwich** package (Zeileis 2006). `rob = FALSE` reports conventional standard errors.

5. Two Wald-tests are conducted. One, testing the null that all coefficients of explanatory variables are zero and two, testing the null that all error correlation coefficients (`rho`) are zero. The latter being rejected indicates that selection bias is an issue.

A useful benchmark is always the null model with structural parameters only. The null model can be derived from an '`opsr`' model fit as follows

```
R> fit_null <- opsr_null_model(fit, printLevel = 0)
```

A model can be updated as usual

```
R> fit_intercept <- update(fit, . ~ . | 1)
```

where we have removed all the explanatory variables from the outcome processes.
Several models can be compared with a likelihood-ratio test using

```
R> anova(fit_null, fit_intercept, fit)

Likelihood Ratio Test

Model 1: ~Nullmodel
Model 2: ys | yo ~ xs1 + xs2 | 1
Model 3: ys | yo ~ xs1 + xs2 | xo1 + xo2
  logLik    Df  Test Restrictions Pr(>Chi)
1  -3293     8
2  -2835    13   917            5   <2e-16 ***
3  -2016    19  1636            6   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If only a single object is passed, then the model is compared to the null model. If more than one object is specified a likelihood ratio test is conducted for each pair of neighboring models. As expected, both tests reject the null hypothesis.
Models can be compared side-by-side using the **texreg** package (Leifeld 2013), which also allows the user to build production-grade tables as illustrated later.

```
R> texreg::screenreg(list(fit_null, fit_intercept, fit),
+    include.pseudoR2 = TRUE, include.R2 = TRUE, single.row = TRUE)

================================================================================
                 Model 1              Model 2              Model 3
--------------------------------------------------------------------------------
kappa1              -1.03 (0.05) ***     -1.96 (0.09) ***     -1.96 (0.09) ***
kappa2               0.41 (0.04) ***      0.88 (0.06) ***      0.88 (0.06) ***
sigma1               2.56 (0.13) ***      2.56 (0.13) ***      1.04 (0.05) ***
sigma2               2.07 (0.06) ***      2.08 (0.06) ***      1.05 (0.03) ***
sigma3               2.91 (0.11) ***      2.91 (0.11) ***      1.17 (0.04) ***
rho1                                      0.03 (0.13)          0.17 (0.14)
rho2                                      0.17 (0.07) *        0.35 (0.06) ***
rho3                                      0.15 (0.11)          0.37 (0.11) ***
s_xs1                                     0.94 (0.06) ***      0.94 (0.06) ***
s_xs2                                     1.49 (0.07) ***      1.50 (0.07) ***
o1_(Intercept)       0.78 (0.21) ***      0.85 (0.34) *        0.99 (0.15) ***
o1_xo1                                                         2.05 (0.09) ***
```

```
o1_xo2                                                          1.01 (0.07) ***
o2_(Intercept)       0.82 (0.09) ***      0.85 (0.09) ***       0.96 (0.05) ***
o2_xo1                                                         -0.99 (0.04) ***
o2_xo2                                                          1.55 (0.04) ***
o3_(Intercept)       1.16 (0.16) ***      0.94 (0.22) ***       1.00 (0.09) ***
o3_xo1                                                          1.58 (0.06) ***
o3_xo2                                                         -1.92 (0.05) ***
-------------------------------------------------------------------------------
AIC               6601.95              5695.26              4070.83
BIC               6641.21              5759.06              4164.08
Log Likelihood   -3292.97             -2834.63             -2016.41
Pseudo R^2 (EL)      0.09                 0.51                 0.51
Pseudo R^2 (MS)     -0.00                 0.46                 0.46
R^2 (total)          0.00                 0.01                 0.82
R^2 (1)             -0.00                 0.00                 0.84
R^2 (2)             -0.00                 0.01                 0.76
R^2 (3)             -0.00                 0.01                 0.85
Num. obs.         1000                 1000                 1000
===============================================================================
*** p < 0.001; ** p < 0.01; * p < 0.05
```

Finally, the key interest of an OPSR study almost certainly is the estimation of treatment effects which relies on (counterfactual) conditional expectations as already noted in the mathematical exposition.

```
R> p1 <- predict(fit, group = 1, type = "response")
R> p2 <- predict(fit, group = 1, counterfact = 2, type = "response")
```

where `p1` is the result of applying Equation 7 and `p2` is the counterfactual outcome resulting from Equation 8. The following `type` arguments are available:

- `type = "response"`: Predicts the continuous outcome according to the Equations referenced above.
- `type = "unlog-response"`: Predicts the back-transformed response according to Equations 9-10 if the continuous outcome was log-transformed (either in the `formula` or during data pre-processing). The smoothing constant used during the continuity correction (i.e., the δ in $y_j = \ln(Y_j + \delta)$) can be specified via the `delta` argument and defaults to 1.
- `type = "prob"`: Returns the probability vector of belonging to `group`.
- `type = "mills"`: Returns the "inverse Mills ratio".
- `type = "correction"`: Returns $\rho_j\sigma_j$IMR respectively $\rho_{j'}\sigma_{j'}$IMR (if `counterfact = j'` was specified) from Equation 7 or 8.
- `type = "Xb"`: Returns $\boldsymbol{X_j\beta_j}$ respectively $\boldsymbol{X_{j'}\beta_{j'}}$ (if `counterfact = j'` was specified) from Equation 7 or 8.

Elements are `NA_real_` if the `group` does not correspond to the observed regime. This ensures consistent output length.

The function `opsr_te()` wraps the required `predict()` calls and prepares the inputs for treatment effect computations returning an object of class '`opsr.te`'. A subsequent call to `summary()` actually computes the treatment effects (TE) and average treatment effects (ATE). An associated `print()` method presents the final computations. `print.opsr.te()` internally calls `summary()` and therefore the explicit call to `summary()` can be omitted if the analyst does not require the underlying computed objects

```
R> print(opsr_te(fit, type = "response"))

Treatment Effects

TE

              G1         G2         G3
T1->T2  0.00112   -0.23522 .  0.36207 *
T1->T3 -0.32352    0.04925    0.35061 *
T2->T3 -0.32464    0.28447   -0.01146


ATE

   T1->T2    T1->T3    T2->T3
1 0.00438   0.09535   0.09097
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

where (weighted) pairwise t tests indicate whether the treatment effects are significantly different from zero (not accounting for uncertainty in the treatment effect estimates themselves). For TE, the columns reflect the factual regime or group, whereas the rows reflect all possible pairs of treatment combinations. For ATE, the columns reflect the treatment combinations.

Last, there is a `plot()` method for model fits of class '`opsr`'. The method internally calls `opsr_te()` and then `pairs.opsr.te()` which visualizes the treatment effects in a matrix of scatterplots. The plot method is demonstrated later in Section 5, Figure 4.

3.2. Convergence diagnostics and identification tests

The second-order optimality conditions at the MLE estimate can be inspected with `opsr_diagnostics()`

```
R> opsr_diagnostics(fit)

Convergence diagnostics

Max. absolute gradient: 2.75e-05
Hessian negative definite: TRUE
Condition number (Hessian): 31.9
```

The output reports (i) the maximum absolute gradient element (`max_abs_gradient`), (ii) whether the Hessian is negative definite (`hessian_neg_def`) — a necessary condition for a proper local maximum rather than a saddle point — and (iii) the condition number (ratio of largest to smallest absolute eigenvalue). All eigenvalues are stored in the returned object and can be printed via `print.eigenvalues = TRUE`. A large condition number — typically above 10^3–10^4 — signals near-collinearity in the parameter space and suggests that some parameter combination is weakly identified.

These second-order conditions are *necessary but not sufficient* to confirm empirical identification. A model with a weak or absent exclusion restriction can produce a gradient close to zero and a negative-definite Hessian while ρ_j is essentially unidentified: the log-likelihood surface is flat in the ρ_j direction, not ill-shaped. The condition number provides a soft signal, but cannot alone distinguish identification failure from benign covariate collinearity.

Two dedicated diagnostics address this. `opsr_lr_instrument()` performs a likelihood-ratio test that drops specified instrument(s) from the selection equation, where a small LR statistic signals a weak instrument. `opsr_profile_rho()` fixes each ρ_j on a grid and re-maximises over all other parameters, producing a profile log-likelihood curve per regime: a flat profile means the data carry almost no information about ρ_j beyond the normality assumption. Both diagnostics are combined in `opsr_test_exclusion_restriction()`

```
R> test_sim <- opsr_test_exclusion_restriction(fit, instrument = "xs2",
+    grid = seq(-0.99, 0.99, length.out = 11), printLevel = 0)
R> print(test_sim)


Exclusion restriction diagnostics
Instrument(s): xs2

Likelihood Ratio Test

Model 1: ys | yo ~ xs1 + xs2 | xo1 + xo2  [xs2 restricted to 0 in selection]
Model 2: ys | yo ~ xs1 + xs2 | xo1 + xo2
  logLik    Df  Test Restrictions Pr(>Chi)
1  -2401    18
2  -2016    19   769            1   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Profile log-likelihood for rho

Regime 1  (rho_hat = 0.17):
  Profile LL range: [-2199, -2016]  |  drop from MLE: 183
Regime 2  (rho_hat = 0.348):
  Profile LL range: [-2160, -2017]  |  drop from MLE: 143
Regime 3  (rho_hat = 0.37):
  Profile LL range: [-2094, -2016]  |  drop from MLE: 77.3

Use plot() to visualise the profile.
```

Because the simulated data uses `xs2` as a genuine excluded instrument, the LR test is strongly significant and the profile is peaked, confirming that ρ_j is well identified by the data. A plot of the profile can be produced with `plot(test_sim$profile_rho)`. The contrast with the weak-instrument and no-exclusion conditions is shown in Section 4, and a real-data application is in Section 5.2.

3.3. Replication exercise

Now that the user understands the basic workflow, we illustrate some nuances by reproducing a key output of Wang and Mokhtarian (2024) where they investigate the treatment effect of telework (TW) on weekly vehicle miles driven. The data is attached, documented (`?telework_data`) and can be loaded by

```
R> data("telework_data", package = "OPSR")
```

The final model specification reads

```
R> f <-
+    twing_status | vmd_ln ~
+    edu_2 + edu_3 + hhincome_2 + hhincome_3 + flex_work + work_fulltime +
+    twing_feasibility + att_proactivemode + att_procarowning + att_wif +
+    att_proteamwork + att_tw_effective_teamwork + att_tw_enthusiasm +
+    att_tw_location_flex |
+    female + age_mean + age_mean_sq + race_black + race_other + vehicle +
+    suburban + smalltown + rural + work_fulltime + att_prolargehouse +
+    att_procarowning + region_waa |
+    edu_2 + edu_3 + suburban + smalltown + rural + work_fulltime +
+    att_prolargehouse + att_proactivemode + att_procarowning |
+    female + hhincome_2 + hhincome_3 + child + suburban + smalltown +
+    rural + att_procarowning + region_waa
```

and the model can be estimated by

```
R> start_default <- opsr(f, telework_data, .get2step = TRUE)
R> fit <- opsr(f, telework_data, start = start, method = "NM", iterlim = 50e3,
+    printLevel = 0)
```

where we demonstrate that:

1. Default starting values as computed by the Heckman two-step procedure can be retrieved (`.get2step = TRUE`).
2. `start` values can be overridden (we have hidden the `start` vector here for brevity). If the user wishes to pass start values manually, some minimal conventions have to be followed as documented in `?opsr_check_start`.
3. Alternative maximization methods (here "Nelder-Mead" via `method = "NM"`) can be used (as in the original paper).

With help of the **texreg** package, production-grade tables (in various output formats) can be generated with ease.

```
R> texreg::texreg(
+    fit, beside = TRUE, include.R2 = TRUE, include.pseudoR2 = TRUE,
+    custom.model.names = custom.model.names, custom.coef.names = custom.coef.names,
+    groups = groups, scalebox = 0.76, booktabs = TRUE, dcolumn = TRUE,
+    no.margin = TRUE, use.packages = FALSE, float.pos = "htbp", single.row = TRUE,
+    caption = "Replica of \\citet{Wang+Mokhtarian:2024}, Table 3.",
+    label = "tab:wang-replica",
+    custom.note = custom.note
+  )
```

Dot arguments (`...`) passed to `texreg()` (or similar functions) are forwarded to a S4 method `extract()` which extracts the variables of interest from a model fit (see also `?extract.opsr`). We demonstrate here that:

1. The model components can be printed side-by-side (`beside = TRUE`).
2. Additional goodness-of-fit indicators can be included (`include.R2 = TRUE` and `include.pseudoR2 = TRUE`).
3. The output formatting can be controlled flexibly, by reordering, renaming and grouping coefficients (the fiddly but trivial details are hidden here for brevity).

Weighted treatment effects in the original (log-backtransformed scale) can be obtained as follows

```
R> te <- opsr_te(fit, type = "unlog-response", weights = telework_data$weight)
R> print(te)

Treatment Effects

TE

            G1          G2          G3
T1->T2  -32.74 ***   -7.54       25.42 ***
T1->T3 -111.38 ***  -96.13 ***  -92.87 ***
T2->T3  -78.64 ***  -88.60 *** -118.29 ***


ATE

  T1->T2     T1->T3     T2->T3
1  -12.4 *** -103.8 ***  -91.4 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

	Structural	Selection	NTWer (535)	NUTWer (322)	UTWer (727)
Kappa 1	1.23 (0.17)***				
Kappa 2	2.46 (0.17)***				
Sigma 1	1.18 (0.05)***				
Sigma 2	1.23 (0.07)***				
Sigma 3	1.43 (0.04)***				
Rho 1	0.05 (0.10)				
Rho 2	0.13 (0.07)				
Rho 3	0.30 (0.07)***				
Education (ref: high school or less)					
Some college		0.32 (0.14)*		0.15 (0.33)	
Bachelor's degree or higher		0.47 (0.13)***		0.62 (0.31)*	
Household income (ref: less than $50,000)					
$50,000 to $99,999		0.06 (0.12)			0.47 (0.23)*
$100,000 or more		0.25 (0.11)*			0.31 (0.23)
Flexible work schedule		0.31 (0.10)**			
Full time worker		0.33 (0.10)***	0.45 (0.13)***	0.69 (0.17)***	
Teleworking feasibility		0.13 (0.01)***			
Attitudes					
Pro-active-mode		0.08 (0.04)*		−0.18 (0.08)*	
Pro-car-owning		−0.08 (0.04)*	0.14 (0.07)*	0.16 (0.09)	0.25 (0.06)***
Work interferes with family		0.11 (0.04)**			
Pro-teamwork		0.09 (0.04)*			
TW effective teamwork		0.32 (0.04)***			
TW enthusiasm		0.09 (0.04)*			
TW location flexibility		0.08 (0.04)*			
Intercept			3.64 (0.27)***	2.49 (0.37)***	2.38 (0.26)***
Female			−0.21 (0.10)*		−0.36 (0.11)***
Age			0.01 (0.00)*		
Age squared			−0.00 (0.00)		
Race (ref: white)					
Black			−0.40 (0.24)		
Other races			−0.06 (0.18)		
Number of vehicles			0.12 (0.05)*		
Residential location (ref: urban)					
Suburban			0.07 (0.15)	0.45 (0.17)**	0.28 (0.14)*
Small town			0.47 (0.18)**	0.19 (0.29)	0.29 (0.28)
Rural			0.60 (0.23)**	0.81 (0.31)**	0.88 (0.34)**
Pro-large-house			0.18 (0.05)***	0.18 (0.08)*	
Region indicator (WAA)			−0.25 (0.11)*		−0.27 (0.11)*
Number of children					0.18 (0.06)**
AIC	7191.35				
BIC	7491.94				
Log Likelihood	−3539.67				
Pseudo R^2 (EL)	0.49				
Pseudo R^2 (MS)	0.46				
R^2 (total)	0.24				
R^2 (1)	0.18				
R^2 (2)	0.18				
R^2 (3)	0.12				
Num. obs.	1584				

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$. We used robust standard errors in this replica, which may result in slight differences from the original standard errors.

Table 1: Replica of Wang and Mokhtarian (2024), Table 3.

where all ATEs are negative. Only `G3` (the current UTWers) would increase weekly VMD when switching from NTWing to NUTWing (25.416 miles). All treatment effects are significantly different from zero, except `G2 T1->T2`, i.e., the NUTWers switching from NTWing to NUTWing.

4. Simulation study

To benchmark estimator performance and illustrate how the identification diagnostics behave under known conditions, we replicate and extend the Monte Carlo study of Chiburis and Lokshin (2007) for the three-regime OPSR case. Pre-computed results are available as the package data object `simulation_study`

```
R> data("simulation_study", package = "OPSR")
R> monte_carlo <- simulation_study$monte_carlo
R> diagnostics_sim <- simulation_study$diagnostics
```

4.1. Design

The data-generating process (DGP) mirrors Chiburis and Lokshin (2007). It comprises two standard-normal regressors x_1, x_2, bivariate-normal errors with correlation $\rho = 0.5$ (except run 2), an ordered-probit selection rule with thresholds $\kappa = (-1, 1)$ yielding three regimes, and outcome $y = \beta x_1 + \varepsilon$ (true $\beta = 1$). Variable x_2 enters the selection equation with coefficient $\alpha_2 = 1$ and is excluded from the outcome equation — the exclusion restriction. Six conditions are run with $R = 1000$ replications at $N = 1000$ (the baseline also varies $N \in \{50, 100, 200, 300, 500\}$), summarised in Table 2.

Run	Condition	Description
1	Baseline	Textbook DGP, valid exclusion restriction, six sample sizes
2	High $\rho = 0.99$	Severe selection bias
4	Non-normal shocks	Squared standard-normal errors (mean-centred)
5	No exclusion restriction	x_2 absent from the selection equation ($\alpha_2 = 0$) and the outcome equation, leaving identification via functional form only
6	Violated exclusion restriction	x_2 in the selection equation ($\alpha_2 = 1$) but also enters the outcome directly ($\beta_{x_2} = 1$), making the instrument invalid
7	Weak instrument	x_2 in the selection equation with $\alpha_2 = 0$ (no predictive power) but correctly excluded from outcomes

Table 2: Simulation conditions.

Results are stored for both the FIML estimator (default `opsr()`) and the Heckman two-step (`opsr_2step()`). Run 3 of the original study is not replicated; we keep the original run numbering to ease comparison with Chiburis and Lokshin (2007).

4.2. Parameter recovery

Table 3 reports FIML mean $\hat{\beta}$ and 95% CI coverage across the three regimes at $N = 1000$. The baseline and high-ρ conditions achieve near-nominal coverage for all three regimes with

Run	Condition	Regime 1	Regime 2	Regime 3
1	Baseline	1.001 (95%)	0.998 (95%)	0.995 (94%)
2	High $\rho = 0.99$	0.999 (95%)	0.998 (95%)	0.998 (95%)
4	Non-normal shocks	0.939 (86%)	1.009 (94%)	1.049 (92%)
5	No excl. restriction	0.896 (81%)	0.675 (67%)	0.885 (84%)
6	Excl. restriction violated	0.154 (0%)	0.047 (0%)	0.160 (0%)
7	Weak instrument	0.859 (82%)	0.673 (63%)	0.891 (79%)

Table 3: FIML mean $\hat{\beta}$ and 95% CI coverage at $N = 1000$ across three regimes. True $\beta = 1$.

negligible bias, confirming that FIML performs well when a valid exclusion restriction is in place. Non-normal shocks primarily hurt the exterior regimes (1 and 3), while the interior regime 2 remains close to nominal because the near-linearity of its Mills-ratio correction makes FIML approximately equivalent to a linear correction there.

The identification conditions reveal sharply different failure modes. Runs 5 and 7 both set $\alpha_2 = 0$, but differ in what the analyst does: run 5 omits x_2 from the model entirely (no exclusion restriction is attempted), while run 7 includes x_2 in the selection equation as a nominal instrument that carries no first-stage signal — the more dangerous scenario because standard output gives no immediate warning. With no exclusion restriction or a weak instrument (runs 5 and 7), coverage collapses substantially for the interior regime 2 — exactly the regime predicted by Chiburis and Lokshin (2007) to suffer most from functional-form-only identification. Under these two conditions FIML also fails to converge in a non-negligible share of replications (about 25% in run 5 and 7% in run 7), and the reported summaries are based on the converged replications only. The two-step estimator (not shown in the table, but included in the stored object) remains nearly unbiased on average under runs 5 and 7 but with enormous dispersion (e.g., a standard deviation of 2.7 for regime 2 in run 5, against 0.07 in the baseline). With a violated exclusion restriction (run 6), all estimates are tightly concentrated but far from the truth (coverage $\approx 0\%$ for all regimes): the model converges confidently to the wrong answer.

4.3. Diagnostic performance

Table 4 summarises the output of `opsr_test_exclusion_restriction()` for one representative dataset per condition. The diagnostics correctly signal identification problems for runs 5 and 7: the condition number is more than an order of magnitude larger than the baseline (1434 and 649 vs. ~ 32), the profile log-likelihood for regime 2 is near-flat (range < 1 LL unit, compared to ~ 160 for the baseline), and the LR test is either non-significant ($p = 0.97$ in run 7) or not applicable at all (in run 5 the instrument is absent from the DGP, leaving nothing to drop). For a valid but strong instrument (run 1) or high error correlation (run 2) these quantities show healthy values.

Critically, the violated exclusion restriction (run 6) is *not* flagged: the LR test is strongly significant, the profile is curved, and the condition number is similar to the baseline. The model

Run	Condition	Cond. number	Profile range	LR p-value
1	Baseline	32	160.1	<0.001
2	High $\rho = 0.99$	32	143.5	<0.001
4	Non-normal shocks	61	85.2	<0.001
5	No excl. restriction	1434	0.6	`NA`
6	Excl. restriction violated	49	211.6	<0.001
7	Weak instrument	649	0.5	0.973

Table 4: Diagnostics for one representative dataset per condition ($N = 1000$). Profile range is the log-likelihood drop across the ρ grid for the interior regime 2, where smaller values indicate a flatter (weaker) profile.

estimates ρ_j with apparent precision, but the estimates are wrong because the normality-driven correction has absorbed the misspecified direct effect of x_2 on the outcome. This is the key practical limitation: instrument validity requires subject-matter justification, not just a significant LR test. Practitioners should therefore apply `opsr_test_exclusion_restriction()` as a *necessary* check, while treating a clean result as evidence of instrument *relevance*, not validity.

5. Case study

Now that the reader is familiar with the main functionality of **OPSR**, this section demonstrates how to employ it in a real-world example. The emphasis, therefore, lies not on what each function does but on guiding the reader through the modeling and post-estimation steps. We investigate telework treatment effects on weekly distance traveled (aggregated over all modes of transport). This contrasts with Wang and Mokhtarian (2024) who used vehicle miles driven (i.e., car only).

We first discuss the model building strategy to arrive at an appropriately specified OPSR model. The OPSR model is then compared to a model not accounting for error correlation and implications for treatment effects are shown. The case study concludes with a discussion on unit treatment effects investigating to what degree teleworking influence total travel demand across all modes.

5.1. Estimation

We use the TimeUse+ dataset (Winkler *et al.* 2024), a smartphone-based diary, recording travel, time use, and expenditure data. Our analytical sample comprises employed individuals and is based on what Winkler and Axhausen (2024) identified as valid days. A valid day has at least 20 hours of information where 70% of the events were validated by the user. Users who did not have at least 14 valid days were excluded. For the remaining 824 participants mobility indicators for a typical week were constructed. The telework status is based on tracked (and labelled) work activities and three regimes are differentiated: Non-teleworkers (NTWers), Non-usual teleworkers (NUTWers, <3 days/week) and Usual teleworkers (UTWers, 3+ days/week).

The data, underlying this analysis, is attached, documented (`?timeuse_data`) and can be

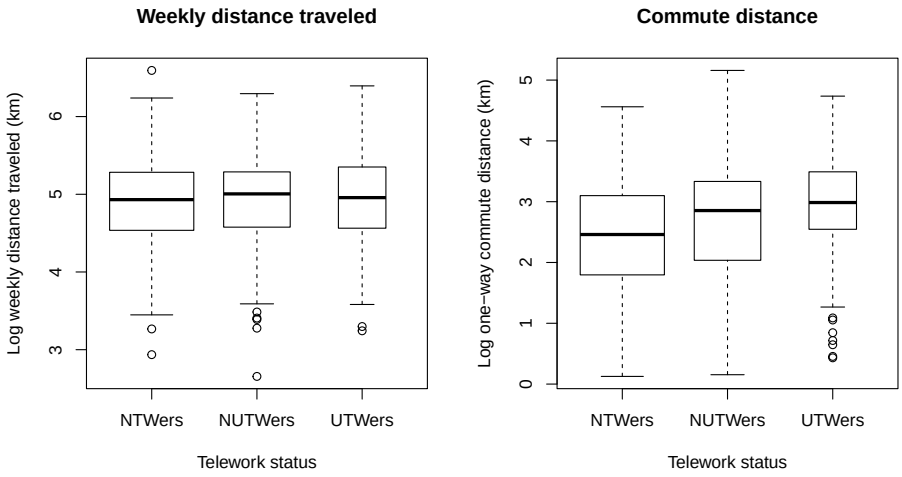

Figure 1: Log weekly distance traveled and log one-way commute distance for different telework statuses.

loaded by

```
R> data("timeuse_data", package = "OPSR")
```

A basic boxplot of the response variable against the three telework statuses is displayed in Figure 1. By simply looking at the data descriptively, we might prematurely conclude that telework does not impact weekly distance traveled. However, the whole value proposition of OPSR (and switching regression models in general) lies in estimating treatment effects by generating counterfactuals that are otherwise unobservable in cross-sectional datasets. If the teleworkers self-select, the counterfactual is not simply the group average of the non-teleworkers. More prosaically, if UTWers stopped teleworking, they might travel more or less than the actual NTWers. And as discussed, this might stem from both observable as well as unobservable factors. Meanwhile, UTWers have the highest average commute distance, followed by NUTWers and NTWers.

As discussed in Section 2.1.2, the analyst must designate valid exclusion restrictions: variables that enter the selection equation but are excluded from all outcome equations. In our application, we use task-based teleworkability shifters as instruments: shift-work status and occupation type (ISCO-08 categories). Shift workers have to be physically present at fixed times and therefore cannot telework, site-bound manual occupations (the ISCO-08 major groups 6–9, comprising craft, plant operation, elementary, and agricultural work) involve tasks that cannot be performed remotely, while clerical work is highly teleworkable. These variables determine the *opportunity* to telework but — conditional on workload, income, education, household structure, and workplace fixity — have no direct channel to weekly distance traveled. The one credible threat to validity, namely that manual workers may travel to changing job sites, is blocked by retaining `fixed_workplace` in every outcome equation.

Sparse instrument categories deserve special attention. Only four respondents work in elementary occupations and all of them are non-teleworkers, so entering `isco_elementary` as a separate regressor perfectly predicts the lowest regime (quasi-complete separation): its se-

lection coefficient diverges to $-\infty$, the likelihood has no interior maximum, and the Hessian cannot be negative definite at the reported estimate (compare Section 3.2). We therefore group the site-bound manual occupations into a single indicator

```
R> timeuse_data$isco_manual <- with(timeuse_data,
+    pmin(1, isco_craft + isco_elementary + isco_plant + isco_agri))
```

The full model is then specified according to two rules: the designated instruments (`shift_work`, `isco_manual`, `isco_tech` and `isco_clerical`) enter the selection equation only, while every other selection covariate also enters each outcome equation. Travel-specific variables (e.g., `car_access`, `parking_work`, `driverlicense`, or residential location) enter the outcome equations only, because vehicle accessibility and the residential context directly predict travel distance but are not expected to primarily determine telework status after conditioning on occupation type. The formula specification of the full model is hidden here for brevity.

Starting from the full model, we exclude variables that do not produce significant estimates (at the 10% level). The function `opsr_step()` can help in this iterative process, as it excludes all coefficients whose p values exceed some threshold from the model specification and refits (see `?opsr_step` for further details). A caveat is warranted here: significance-based pruning can inadvertently create invalid exclusion restrictions. If a covariate is retained in the selection equation but pruned from all outcome equations, it silently becomes an instrument — one whose validity must be verified by subject-matter reasoning. We therefore never remove a non-instrument selection covariate from all outcome equations: `educ_higher` and `hh_income`, in particular, are retained in every outcome equation despite individually insignificant estimates, since income and education plausibly affect travel directly and would be theoretically invalid instruments. Conversely, `isco_tech` is dropped altogether in the reduced model because of its weak first stage (it neither predicts selection nor belongs in the outcomes). Additionally, the reported p values of the retained variables are optimistic due to post-selection bias.

```
R> fit_full <- opsr(f_full, timeuse_data, printLevel = 0)
R> f_red <- wfh | log_weekly_km ~
+    educ_higher + hh_income + young_kids + workload + fixed_workplace +
+    shift_work + isco_manual + isco_clerical |
+    educ_higher + hh_income + young_kids + workload + fixed_workplace +
+    permanent_employed + parking_work |
+    educ_higher + hh_income + young_kids + workload + fixed_workplace +
+    swiss + res_loc + parking_work |
+    educ_higher + hh_income + young_kids + workload + fixed_workplace +
+    permanent_employed + sex_male + swiss + parking_work
R> fit_red <- opsr(f_red, timeuse_data, printLevel = 0)
R> fit_red <- opsr(f_red, timeuse_data, start = coef(fit_red), method = "NR",
+    printLevel = 0)
R> print(anova(fit_red, fit_full), print.formula = FALSE)

Likelihood Ratio Test

   logLik      Df    Test Restrictions Pr(>Chi)
1 -1343.9    61.0
2 -1322.1    95.0    43.7           34     0.12
```

```
R> summary(fit_red)

Call:
opsr(formula = f_red, data = timeuse_data, start = coef(fit_red),
    method = "NR", printLevel = 0)

Newton-Raphson maximisation, 1 iterations
Return code 8: successive function values within relative tolerance limit (reltol)
Runtime: 4.05 secs
Max. absolute gradient: 2.23e-04
Hessian negative definite: TRUE
Condition number (Hessian): 63039
Number of regimes: 3
Number of observations: 824 (424, 265, 135)
Estimated parameters: 61

Log-Likelihood: -1344
AIC: 2810
BIC: 3097
Pseudo R-squared (EL): 0.187
Pseudo R-squared (MS): 0.109
Multiple R-squared: 0.221 (0.198, 0.201, 0.32)

Estimates:
                       Estimate Std. error t value Pr(> t)
kappa1                -0.31125    0.29435   -1.06 0.29033
kappa2                 0.77684    0.29205    2.66 0.00782 **
s_educ_higher          0.46539    0.09296    5.01 5.6e-07 ***
s_hh_income4001_8000  -1.01104    0.25061   -4.03 5.5e-05 ***
s_hh_income8001_12000 -0.73639    0.25068   -2.94 0.00331 **
s_hh_income12001_16000 -0.54998   0.25679   -2.14 0.03221 *
s_hh_income16001+     -0.58609    0.27980   -2.09 0.03620 *
s_hh_incomeNA         -0.50144    0.34849   -1.44 0.15018
s_young_kids           0.26607    0.09352    2.85 0.00444 **
s_workload             0.06699    0.02224    3.01 0.00259 **
s_fixed_workplace     -0.56987    0.14350   -3.97 7.2e-05 ***
s_shift_work          -0.75936    0.21157   -3.59 0.00033 ***
s_isco_manual         -0.59936    0.19413   -3.09 0.00202 **
s_isco_clerical        0.49356    0.08997    5.49 4.1e-08 ***
o1_(Intercept)         4.36766    0.26434   16.52 < 2e-16 ***
o1_educ_higher         0.03043    0.07076    0.43 0.66720
o1_hh_income4001_8000 -0.22951    0.19389   -1.18 0.23652
o1_hh_income8001_12000 -0.01352   0.17560   -0.08 0.93864
o1_hh_income12001_16000 -0.07277  0.17860   -0.41 0.68370
o1_hh_income16001+    -0.10366    0.20333   -0.51 0.61020
o1_hh_incomeNA        -0.18649    0.22823   -0.82 0.41385
o1_young_kids         -0.07311    0.06004   -1.22 0.22338
```

```
o1_workload              0.07352    0.01451    5.07 4.1e-07 ***
o1_fixed_workplace      -0.14937    0.10495   -1.42 0.15465
o1_permanent_employed    0.29138    0.11033    2.64 0.00826 **
o1_parking_work          0.28472    0.05116    5.57 2.6e-08 ***
o2_(Intercept)           3.86530    0.25422   15.20 < 2e-16 ***
o2_educ_higher          -0.00796    0.08509   -0.09 0.92545
o2_hh_income4001_8000    0.26603    0.22045    1.21 0.22751
o2_hh_income8001_12000   0.25735    0.21104    1.22 0.22268
o2_hh_income12001_16000  0.31683    0.21296    1.49 0.13682
o2_hh_income16001+       0.28364    0.24271    1.17 0.24255
o2_hh_incomeNA           0.25705    0.29992    0.86 0.39141
o2_young_kids           -0.17745    0.07156   -2.48 0.01315 *
o2_workload              0.06511    0.01658    3.93 8.6e-05 ***
o2_fixed_workplace      -0.17166    0.14446   -1.19 0.23472
o2_swiss                 0.20064    0.11431    1.76 0.07923 .
o2_res_locrural          0.41706    0.15317    2.72 0.00647 **
o2_res_locsuburban       0.22751    0.14813    1.54 0.12456
o2_res_locurban          0.15476    0.16545    0.94 0.34959
o2_parking_work          0.15660    0.06980    2.24 0.02486 *
o3_(Intercept)           3.52974    0.40810    8.65 < 2e-16 ***
o3_educ_higher           0.07976    0.08874    0.90 0.36872
o3_hh_income4001_8000   -0.11205    0.20569   -0.54 0.58593
o3_hh_income8001_12000  -0.13268    0.17502   -0.76 0.44840
o3_hh_income12001_16000  0.00816    0.18318    0.04 0.96447
o3_hh_income16001+      -0.12320    0.21414   -0.58 0.56507
o3_hh_incomeNA           0.30643    0.32416    0.95 0.34450
o3_young_kids           -0.03849    0.09219   -0.42 0.67627
o3_workload              0.04100    0.03103    1.32 0.18646
o3_fixed_workplace      -0.34022    0.12296   -2.77 0.00566 **
o3_permanent_employed    0.49146    0.22641    2.17 0.02996 *
o3_sex_male              0.18593    0.09782    1.90 0.05734 .
o3_swiss                 0.45751    0.12472    3.67 0.00024 ***
o3_parking_work          0.29685    0.09408    3.16 0.00160 **
sigma1                   0.56729    0.05151   11.01 < 2e-16 ***
sigma2                   0.53493    0.03213   16.65 < 2e-16 ***
sigma3                   0.52554    0.05002   10.51 < 2e-16 ***
rho1                     0.64526    0.18974    3.40 0.00067 ***
rho2                    -0.17397    0.20473   -0.85 0.39546
rho3                     0.43624    0.17760    2.46 0.01404 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Wald chi2 (null): 423 on 50 DF, p-value: < 0
Wald chi2 (rho): 18.6 on 3 DF, p-value: < 0
```

The second call to `opsr()` polishes the BFGS solution with a Newton-Raphson step (starting from the BFGS estimate), which tightens the gradient at the reported maximum without

changing the log-likelihood. The reduced model specification (`fit_red`) is not rejected in the likelihood ratio test. Further, the Wald-test of joint significance of the `rho` coefficients can be assessed from the summary output. Rejection of the null (`rho1` = `rho2` = `rho3` = 0) indicates that the selection correction is beneficial given our model assumptions. The estimates suggest positive selection on unobservables into the boundary regimes (`rho1` = 0.65, `rho3` = 0.44): unobserved factors that make telework more likely are associated with longer weekly distances among NTWers and UTWers.

5.2. Exclusion restriction and identification

Before interpreting treatment effects, we validate the exclusion restriction of the fitted model. The instruments are the shift-work and occupation-teleworkability variables that appear in the selection equation but not in the outcome equations. We test their joint relevance with `opsr_test_exclusion_restriction()` and inspect the per-regime profile likelihood for ρ_j

```
R> instruments_cs <- c("shift_work", "isco_manual", "isco_clerical")
R> test_cs <- opsr_test_exclusion_restriction(fit_red,
+    instrument = instruments_cs,
+    grid = seq(-0.99, 0.99, length.out = 7), printLevel = 0)
R> print(test_cs)

Exclusion restriction diagnostics
Instrument(s): shift_work, isco_manual, isco_clerical

Likelihood Ratio Test

Model 1: wfh | log_weekly_km ~ educ_higher + hh_income + young_kids +
    workload + fixed_workplace + shift_work + isco_manual + isco_clerical |
    educ_higher + hh_income + young_kids + workload + fixed_workplace +
        permanent_employed + parking_work | educ_higher + hh_income +
    young_kids + workload + fixed_workplace + swiss + res_loc +
    parking_work | educ_higher + hh_income + young_kids + workload +
    fixed_workplace + permanent_employed + sex_male + swiss +
    parking_work  [shift_work, isco_manual, isco_clerical restricted to 0 in selection]
Model 2: wfh | log_weekly_km ~ educ_higher + hh_income + young_kids +
    workload + fixed_workplace + shift_work + isco_manual + isco_clerical |
    educ_higher + hh_income + young_kids + workload + fixed_workplace +
        permanent_employed + parking_work | educ_higher + hh_income +
    young_kids + workload + fixed_workplace + swiss + res_loc +
    parking_work | educ_higher + hh_income + young_kids + workload +
    fixed_workplace + permanent_employed + sex_male + swiss +
    parking_work
   logLik      Df    Test Restrictions Pr(>Chi)
1 -1384.2    58.0
2 -1343.9    61.0    80.7            3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
R> plot(test_cs$profile_rho)
```

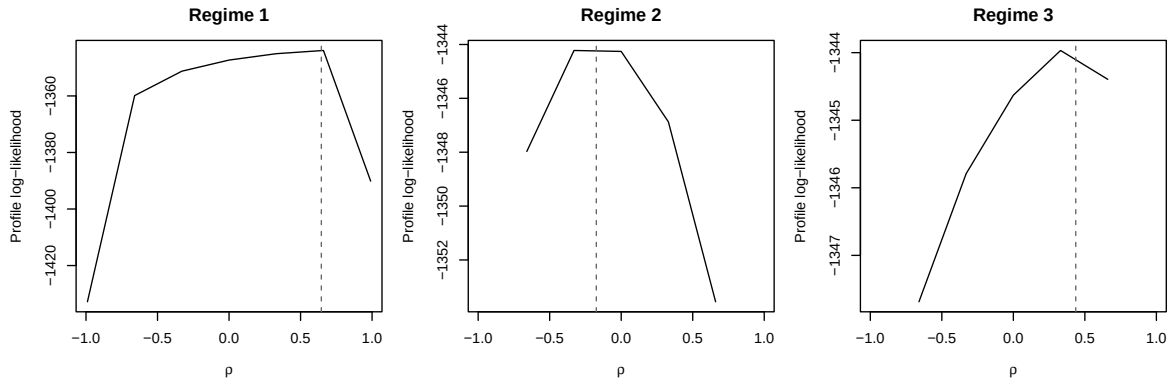

Figure 2: Profile log-likelihood for ρ_j per regime. Each panel fixes ρ_j at the grid value and re-maximises over all other parameters. Peaked profiles indicate that ρ_j is identified by the data. Near-boundary grid values (ρ_j close to ± 1) can fail to converge and are omitted, which is why some panels appear truncated before the grid limits.

```
Profile log-likelihood for rho

Regime 1  (rho_hat = 0.645):
  Profile LL range: [-1433, -1344]  |  drop from MLE: 88.9
Regime 2  (rho_hat = -0.174):
  Profile LL range: [-1354, -1344]  |  drop from MLE: 9.64
Regime 3  (rho_hat = 0.436):
  Profile LL range: [-1348, -1344]  |  drop from MLE: 3.78

Use plot() to visualise the profile.
```

The LR test is strongly significant (the instruments jointly improve the selection fit), confirming instrument relevance. The profile log-likelihood plots per regime (Figure 2) are produced by `plot(test_cs$profile_rho)`. Peaked profiles for all three regimes confirm that the error correlations are empirically identified, not merely pinned by the bivariate normality assumption. This stands in contrast to the flat profiles produced when no exclusion restriction is present (Section 4). `opsr_diagnostics()` adds the second-order optimality checks

```
R> opsr_diagnostics(fit_red)

Convergence diagnostics

Max. absolute gradient: 2.23e-04
Hessian negative definite: TRUE
Condition number (Hessian): 63039
```

The gradient is near zero and the Hessian is negative definite, confirming a proper local maximum. The condition number, however, is large by the rule of thumb of Section 3.2. Inspect-

ing the eigenvector associated with the smallest eigenvalue attributes it to near-collinearity between the regime-specific intercepts and the household-income dummy block — benign covariate collinearity rather than weak identification of ρ_j. This is exactly the ambiguity the condition number cannot resolve on its own (compare Section 3.2); the peaked profile likelihoods and the significant LR test are the decisive evidence here.

5.3. Treatment effects

We first define a helper function (wrapping `opsr_te()`) that provides more intuitive labels for the treatment effects, simplifying the discussion that follows. Unless otherwise mentioned, we use the `fit_red` model in the remainder.

```
R> te <- function(fit) {
+    te <- summary(opsr_te(fit, type = "unlog-response"))$te
+    colnames(te) <- c("NTWers", "NUTWers", "UTWers")
+    rownames(te) <- c("NTWing->NUTWing", "NTWing->UTWing", "NUTWing->UTWing")
+    te
+  }
R> te(fit_red)

                  NTWers     NUTWers      UTWers
NTWing->NUTWing   33.80 ***  -59.63 *** -180.92 ***
NTWing->UTWing   -57.03 ***  -98.51 *** -171.04 ***
NUTWing->UTWing  -90.84 ***  -38.88 ***    9.88 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Telework reduces weekly kilometers traveled across all groups, with the exception of NTWers who would be more mobile when switching from NTWing to NUTWing (33.8 km; column `NTWer`, row `NTWing -> NUTWing`) and UTWers who would travel slightly more when further adopting telework from NUTWing (9.88 km). The treatment effects when switching from NTWing to NUTWing are strongest for UTWers (-180.92 km) compared to NTWers (33.8 km) and NUTWers (-59.63 km). The treatment effect over the full range (NTWing to UTWing) increases in magnitude with the factual telework intensity of the group: -57.03 km for NTWers, -98.51 km for NUTWers and -171.04 km for UTWers. This ordering is exactly what self-selection on unobservables (the positive `rho1` and `rho3` estimates) implies: the individuals who stand to reduce their travel the most by teleworking are also the ones who telework the most. Interestingly, NTWers show a non-linear pattern, first increasing weekly kilometers when adopting some telework (33.8 km, NTWing to NUTWing) but then substantially decreasing weekly kilometers with more telework (-90.84 km, NUTWing to UTWing). An explanation could be that these individuals (living closer to their workplace) initially do not adjust activity chains and location choices when only occasionally teleworking. For example, an individual might stay subscribed to the gym close to the workplace and visit that facility even on a home office day. On the other hand, UTWers show exactly an inverse pattern, first (NTWing to NUTWing) strongly reducing weekly kilometers (-180.92 km) but upon further telework adoption (NUTWing to UTWing) only minimally adjusting weekly kilometers (9.88 km). A similar argument could be made that these individuals (living further from their workplace)

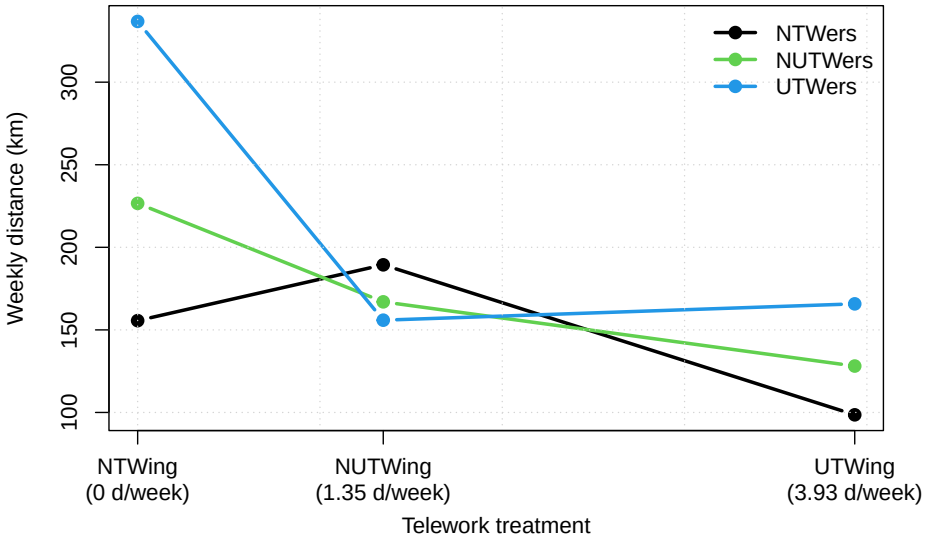

Figure 3: Treatment effects.

adjust activity chains and location choices from the start. One can therefore conclude that the main travel reduction happens at different treatment intensities for the different groups. Figure 3 visualizes these treatment effects and shows the roughly linear pattern for NUTWers and the (mirrored) hockey stick pattern for NTWers and UTWers.

While the discussion above was based on averaged group-level treatment effects, Figure 4 shows the distributions of predicted weekly distance traveled by teleworker group and treatment regime. Following matrix terminology, each figure on the diagonal depicts predicted outcome distributions (i.e., weekly kilometers traveled here) for a given state (NTWing, NUTWing and UTWing) and separate by the current (factual) teleworker group (NTWers, NUTWers and UTWers). The weighted mean predicted outcomes by state are shown as red numbers. The lower triangular panels compare the model-implied (predicted) outcomes of two treatment states (including NTWing) again separated by observed teleworker groups. The red reference line marks the instances where weekly distance traveled is equal for both of the paired (un)treated telework statuses. The two red numbers to the right of the current teleworking status report the weighted mean predicted outcomes by state (and hence align with the numbers shown in the figures on the diagonal). I.e., those are the coordinate values of the group averages, visualized as the red squares. The upper triangular panels show average treatment effects.

We now demonstrate that not controlling for error correlation leads to different and most likely wrong conclusions, since parameter estimates might be biased. We derive a model (`fit_nocor`) without error correlation by fixing the `rho` coefficients at 0. This is the same as separately estimating an ordered probit model and three linear regression models.

```
R> start <- coef(fit_red)
R> fixed <- c("rho1", "rho2", "rho3")
```

```
R> plot(fit_red, type = "unlog-response", col = c(1, 3, 4),
+    labels.diag = c("NTWing", "NUTWing", "UTWing"),
+    labels.reg = c("NTWers", "NUTWers", "UTWers"),
+    xlim = c(0, 400), ylim = c(0, 400), cex = 1.5)
```

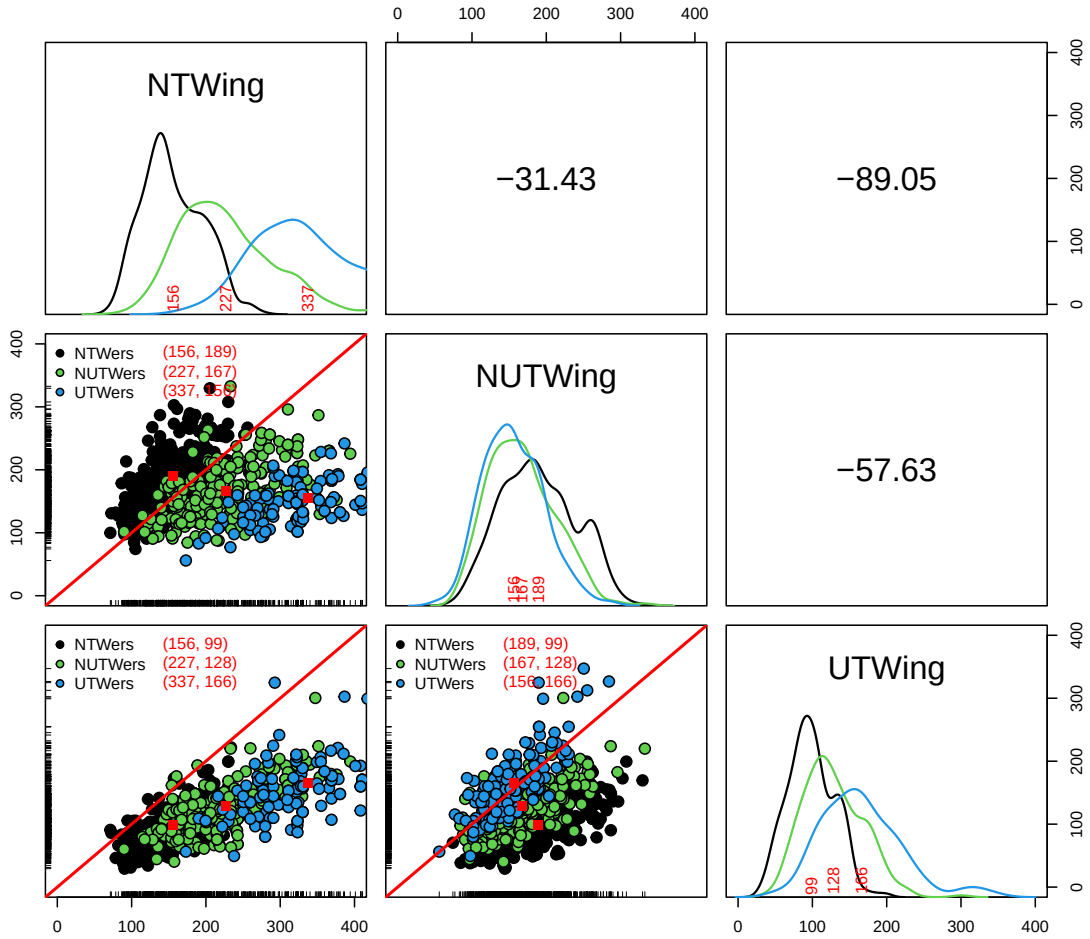

Figure 4: Pairs plot for model fits of class '`opsr`'.

```
R> start[fixed] <- 0
R> fit_nocor <- opsr(f_red, timeuse_data, start = start, fixed = fixed,
+    printLevel = 0)
```

The treatment effects are

```
R> te(fit_red)
```

```
                  NTWers      NUTWers       UTWers
NTWing->NUTWing   33.80 ***  -59.63 *** -180.92 ***
NTWing->UTWing   -57.03 ***  -98.51 *** -171.04 ***
```

Model	Parent	Error correlation	Description
`fit_full`		•	Full identified model: instruments in the selection equation only, all other selection covariates in every outcome equation
`fit_red`	`fit_full`	•	Excluding variables not significant at the 10% level, while retaining the shared selection covariates in all outcome equations
`fit_nocor`	`fit_red`	○	Fixing the `rho` coefficients at 0

Table 5: Model overview. The model is based on *Parent* as elaborated under *Description*.

```
NUTWing->UTWing  -90.84 ***  -38.88 ***    9.88 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R> te(fit_nocor)

                NTWers     NUTWers     UTWers
NTWing->NUTWing  13.05 ***   17.42 ***  20.94 ***
NTWing->UTWing    8.24 ***   13.82 ***  13.95 ***
NUTWing->UTWing  -4.80 **    -3.60 .    -6.99 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we see, `fit_nocor` yields completely different insights — in particular, that adopting telework increases weekly distance traveled for all groups (only the switch from NUTWing to UTWing shows a small reduction), consistent with previous cross-sectional studies that did not account for self-selection bias (for studies indicating that telework increases travel demand, see Zhu and Mason 2014; He and Hu 2015; Kim, Choo, and Mokhtarian 2015). The sign reversal relative to `fit_red` is a direct consequence of the positive error correlations: ignoring them attributes the longer distances of the (self-selected) teleworkers to the treatment itself.

Lastly (using `fit_red`), we compute unit treatment effects and compare them to the average two-way commute distance for each group. The unit treatment effect is calculated by dividing the total treatment effect by the corresponding average teleworking frequency difference (`twdiff1` to `twdiff3` below). I.e., the treatment effect is standardized and therefore also comparable for different regime switching (e.g., NTWing to NUTWing vs. NUTWing to UTWing).

```
R> dat_ute <- subset(timeuse_data, select = c(commute_km, wfh, wfh_days))
R> dat_ute <- aggregate(cbind(wfh_days, 2 * commute_km) ~ wfh, data = dat_ute,
+    FUN = mean)
R> top <- t(dat_ute[2:3])
R> colnames(top) <- c("NTWers", "NUTWers", "UTWers")
R> rownames(top) <- c("WFH (days)", "2-way commute (km)")
R> i <- "WFH (days)"
```

```
R> twdiff1 <- top[i, "NUTWers"] - top[i, "NTWers"]
R> twdiff2 <- top[i, "UTWers"] - top[i, "NTWers"]
R> twdiff3 <- top[i, "UTWers"] - top[i, "NUTWers"]
R> twdiff <- matrix(c(rep(twdiff1, 3), rep(twdiff2, 3), rep(twdiff3, 3)),
+    nrow = 3, byrow = TRUE)
R> bottom <- te(fit_red) / twdiff
R> ute <- rbind(top, bottom)
R> ute

                    NTWers NUTWers  UTWers
WFH (days)             0.0    1.35    3.93
2-way commute (km)    30.1   43.33   51.07
NTWing->NUTWing       25.1  -44.27 -134.30
NTWing->UTWing       -14.5  -25.05  -43.48
NUTWing->UTWing      -35.1  -15.03    3.82
```

Over the full switching range (NTWing to UTWing), telework reduces weekly distance traveled by less than the foregone commute distance for all three groups (14.5 vs. 30.08 km for NTWers, 25.05 vs. 43.33 km for NUTWers and 43.48 vs. 51.07 km for UTWers), which indicates that a rebound effect (compensating leisure travel) exists. Individual switching margins deviate from this pattern, however. The NUTWers reduce travel roughly one-for-one with the foregone commute when initially adopting telework (44.27 vs. 43.33 km). The NTWers (NUTWing to UTWing, 35.12 km) and especially the UTWers (NTWing to NUTWing, 134.3 km) reduce travel beyond their foregone commutes, consistent with the restructuring of activity chains and location choices discussed above. The insights from the previous discussion on treatment effects carry over: Adjustments in weekly distance traveled are very different both across the three teleworker groups but also across the regime switching.

6. Summary and discussion

In a real-world setting, the treatment is usually not exogenously prescribed but self-selected. Various methods in various statistical environments exist to account for the selection bias which arises if unobserved factors simultaneously influence both the selection and outcome process. OPSR is introduced as a special case of endogenous switching regression to account for selection bias with ordinal treatments (the well-known Tobit-5 model being a special case of OPSR with only two treatment regimes). The model frame for such Heckman-type models as well as their implementation in the R system for statistical computing is reviewed. The R implementation presented in package **OPSR** reuses design and functionality of the corresponding R software. Hence, the new function `opsr()` is straightforward to apply for model fitting and diagnostics. Further, it is fast and memory efficient thanks to the C++ implementation of the log-likelihood function which can also be parallelized. **OPSR** handles log-transformed outcomes which need special consideration when computing conditional expectations and thus treatment effects. Post-estimation functions to compute and visualize (weighted) treatment effects are included in the package.

The package provides two layers of model-quality diagnostics. `opsr_diagnostics()` inspects the second-order optimality conditions at the MLE (gradient magnitude, Hessian negative-

definiteness, condition number), addressing whether the optimizer found a well-defined maximum. `opsr_test_exclusion_restriction()` goes further by testing instrument relevance via a likelihood-ratio test and examining whether ρ_j is empirically identified beyond distributional assumptions via the profile log-likelihood. Together these diagnostics make identification failure modes — absent instruments, weak instruments — visible before treatment effects are computed. The key limitation, shared with all Heckman-type models, is that an *invalid* exclusion restriction (instrument that belongs in the outcome equation) cannot be detected from within the model and requires substantive judgment.

A Monte Carlo study replicating Chiburis and Lokshin (2007) for the three-regime case confirms three findings. FIML achieves near-nominal coverage when a valid instrument is in place. Coverage collapses (especially for interior regimes) when identification relies on functional form alone. A violated exclusion restriction produces confident but catastrophically wrong estimates with 0% coverage.

In the case study, the OPSR method is applied to a tracking and activity diary dataset collected in Switzerland, investigating the telework treatment effects on weekly distance traveled across all modes. We demonstrate, first, how to specify an appropriately identified model (using shift-work status and grouped occupation-teleworkability indicators as exclusion restrictions, where grouping the sparse site-bound occupation categories avoids a quasi-complete separation that would otherwise leave the likelihood without an interior maximum) and how to validate the exclusion restriction using the diagnostic tools; and second, to what extent computed treatment effects differ if the error correlation is not accounted for. We find that, overall, telework reduces travel and that the reduction over the full switching range grows with the factual telework intensity of a group — the pattern implied by positive selection on unobservables. Non-teleworkers tend to have shorter commutes and adjust mobility patterns mainly when switching from non-usual telework to usual telework; their weekly distance traveled even increases when initially adopting some telework. Conversely, usual teleworkers (had they not been teleworking) adjust mobility patterns strongly when adopting some telework but then only marginally adjust distance traveled when further adopting telework. Comparing the unit treatment effects to the two-way commute distance indicates that, over the full switching range, telework reduces weekly distance traveled by less than the foregone commute, so some compensating travel (rebound effects) exists; only for the sharpest adjustments (usual teleworkers initially adopting telework, and non-teleworkers moving from non-usual to usual telework) does the reduction exceed the foregone commute.

Computational details

The results in this paper were obtained using R 4.6.1 with the packages **OPSR** 1.0.1.9001, **MASS** 7.3.66, **texreg** 1.39.5, **sampleSelection** 1.2.14, **mvtnorm** 1.4.2, **gridExtra** 2.3.1 and **gridGraphics** 0.5.1. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Please note that slight differences in the results might occur depending on the platform. The results presented here were generated on x86_64-pc-linux-gnu.

Acknowledgments

Xinyi Wang and Daniel Heimgartner conceived the presented idea to formalize the findings of Wang and Mokhtarian (2024) into an R package. The theory presented in Section 2 stems from that work. Daniel Heimgartner implemented the functionality and R package architecture based on Xinyi Wang's original scripts, as well as drafted the paper. All authors discussed the results and contributed to the final manuscript.

We would like to thank Kay W. Axhausen and Patricia L. Mokhtarian for their support during this and previous work.

References

Amemiya T (1985). *Advanced Econometrics*. Harvard University Press, Cambridge, Massachusetts.

Cameron AC, Trivedi PK (2005). *Microeconometrics: Methods and Applications*. Cambridge University Press, Cambridge.

Chiburis R, Lokshin M (2007). "Maximum Likelihood and Two-Step Estimation of Ordered-Probit Selection Model." *The Stata Journal*, **7**(2), 167–182. doi:10.1177/1536867X0700700202.

Eddelbuettel D, Balamuta JJ (2018). "Extending R with C++: A Brief Introduction to **Rcpp**." *The American Statistician*, **72**(1), 28–36. doi:10.1080/00031305.2017.1375990.

Eddelbuettel D, Sanderson C (2014). "**RcppArmadillo**: Accelerating R with high-performance C++ linear algebra." *Computational Statistics and Data Analysis*, **71**, 1054–1063. doi:10.1016/j.csda.2013.02.005.

Greene WH (2002). *LIMDEP Version 8.0 Econometric Modeling Guide, vol. 2*. Econometric Software, Plainview, New York.

He SY, Hu L (2015). "Telecommuting, Income, and Out-of-Home Activities." *Travel Behaviour and Society*, **2**(3), 131–147. doi:10.1016/j.tbs.2014.12.003.

Heckman J (1979). "Sample Selection Bias as a Specification Error." *Econometrica*, **47**(1), 153–161. doi:10.2307/1912352.

Henningsen A, Toomet O (2011). "**maxLik**: A package for maximum likelihood estimation in R." *Computational Statistics*, **26**(3), 443–458. doi:10.1007/s00180-010-0217-1.

Kim SN, Choo S, Mokhtarian PL (2015). "Home-Based Telecommuting and Intra-Household Interactions in Work and Non-Work Travel: A Seemingly Unrelated Censored Regression Approach." *Transportation Research Part A: Policy and Practice*, **80**, 197–214. doi:10.1016/j.tra.2015.07.018.

Leifeld P (2013). "**texreg**: Conversion of Statistical Model Output in R to LaTeX and HTML Tables." *Journal of Statistical Software*, **55**(8), 1–24. doi:10.18637/jss.v055.i08.

Potanin B (2024). ***switchSelection**: Endogenous Switching and Sample Selection Regression Models*. R package version 2.0.0, URL https://CRAN.R-project.org/package=switchSelection.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Roodman D (2011). "Fitting Fully Observed Recursive Mixed-Process Models with **cmp**." *The Stata Journal*, **11**(2), 159–206. doi:10.1177/1536867X110110020.

StataCorp (2023). *Stata Statistical Software*. StataCorp LLC, College Station, Texas. URL https://www.stata.com/.

Toomet O, Henningsen A (2008). "Sample Selection Models in R: Package **sampleSelection**." *Journal of Statistical Software*, **27**(7), 1–23. doi:10.18637/jss.v027.i07.

Toomet O, Henningsen A (2020). ***sampleSelection**: Sample Selection Models*. R package version 1.2-12, URL https://cran.r-project.org/package=sampleSelection.

Wang X, Mokhtarian PL (2024). "Examining the Treatment Effect of Teleworking on Vehicle-Miles Driven: Applying an Ordered Probit Selection Model and Incorporating the Role of Travel Stress." *Transportation Research Part A*, **186**, 104072. doi:10.1016/j.tra.2024.104072.

Winkler C, Axhausen KW (2024). "How Do the Swiss Spend Their Time?" *Findings*. doi:10.32866/001c.108600.

Winkler C, Meister A, Axhausen KW (2024). "The TimeUse+ Data Set: 4 Weeks of Time Use and Expenditure Data Based on GPS Tracks." *Transportation*, pp. 1–27. doi:10.1007/s11116-024-10517-1.

Zeileis A (2006). "Object-Oriented Computation of Sandwich Estimators." *Journal of Statistical Software*, **16**(9), 1–16. doi:10.18637/jss.v016.i09.

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.

Zhu P, Mason SG (2014). "The Impact of Telecommuting on Personal Vehicle Usage and Environmental Sustainability." *International Journal of Environmental Science and Technology*, **11**, 2185–2200. doi:10.1007/s13762-014-0556-5.

A. Tobit-5 model and comparison to sampleSelection

As noted in Section 1, the Tobit-5 model can be seen as a form of OPSR with only two selection outcomes and can be fitted with the R-package **sampleSelection**. In this section, we illustrate that **OPSR** can estimate Tobit-5 models (as all the other examples involve three regimes) and that the results match the ones obtained with **sampleSelection**. The example, using simulated data, is directly taken from the vignette of Toomet and Henningsen (2020, Section 4.2) (`vignette("selection", package = "sampleSelection")`).

We create the following switching regression problem

```
R> set.seed(0)
R> vc <- diag(3)
R> vc[lower.tri(vc)] <- c(0.9, 0.5, 0.1)
R> vc[upper.tri(vc)] <- vc[lower.tri(vc)]
R> eps <- rmvnorm(500, c(0, 0, 0), vc)
R> xs <- runif(500)
R> ys <- xs + eps[, 1] > 0
R> xo1 <- runif(500)
R> yo1 <- xo1 + eps[, 2]
R> xo2 <- runif(500)
R> yo2 <- xo2 + eps[, 3]
R> yo <- ifelse(ys, yo2, yo1)
R> ys <- as.numeric(ys) + 1
R> dat <- data.frame(ys, yo, yo1, yo2, xs, xo1, xo2)
R> head(dat)
```

```
  ys       yo      yo1     yo2    xs    xo1    xo2
1  2  2.34301  0.99101  2.3430 0.531 0.5724 0.6716
2  2 -0.89646  1.75863 -0.8965 0.802 0.5999 0.1878
3  1  0.00931  0.00931  0.2238 0.479 0.7945 0.5048
4  2 -0.01742  2.57710 -0.0174 0.177 0.5046 0.0273
5  1 -0.35597 -0.35597 -0.1317 0.397 0.5402 0.4963
6  2 -0.03943  0.02728 -0.0394 0.814 0.0241 0.9474
```

Using **sampleSelection**, the estimation call reads

```
R> tobit5_s <- selection(ys ~ xs, list(yo1 ~ xo1, yo2 ~ xo2), data = dat)
R> summary(tobit5_s)
```

```
--------------------------------------------
Tobit 5 model (switching regression model)
Maximum Likelihood estimation
Newton-Raphson maximisation, 11 iterations
Return code 1: gradient close to zero (gradtol)
Log-Likelihood: -896
500 observations: 172 selection 1 (1) and 328 selection 2 (2)
10 free parameters (df = 490)
```

```
Probit selection equation:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.155      0.105   -1.47     0.14
xs             1.141      0.179    6.39  3.9e-10 ***
Outcome equation 1:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.0271     0.1640    0.17     0.87
xo1           0.8396     0.1497    5.61  3.4e-08 ***
Outcome equation 2:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.158      0.188    0.84      0.4
xo2            0.838      0.171    4.91  1.3e-06 ***
   Error terms:
       Estimate Std. Error t value Pr(>|t|)
sigma1   0.9319     0.0921   10.12   <2e-16 ***
sigma2   0.9070     0.0443   20.45   <2e-16 ***
rho1     0.8899     0.0535   16.62   <2e-16 ***
rho2     0.1770     0.3314    0.53     0.59
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
--------------------------------------------
```

which is equivalent to **OPSR**

```
R> tobit5_o <- opsr(ys | yo ~ xs | xo1 | xo2, data = dat, printLevel = 0)
R> summary(tobit5_o)

Call:
opsr(formula = ys | yo ~ xs | xo1 | xo2, data = dat, printLevel = 0)

BFGS maximization, 68 iterations
Return code 0: successful convergence
Runtime: 0.0649 secs
Max. absolute gradient: 5.17e-03
Hessian negative definite: TRUE
Condition number (Hessian): 350
Number of regimes: 2
Number of observations: 500 (172, 328)
Estimated parameters: 10

Log-Likelihood: -896
AIC: 1812
BIC: 1854
Pseudo R-squared (EL): 0.122
Pseudo R-squared (MS): 0.054
Multiple R-squared: 0.336 (0.247, 0.069)
```

```
Estimates:
               Estimate Std. error t value Pr(> t)
kappa1           0.1550     0.1047    1.48    0.14
s_xs             1.1408     0.1792    6.37 1.9e-10 ***
o1_(Intercept)   0.0271     0.1688    0.16    0.87
o1_xo1           0.8396     0.1453    5.78 7.6e-09 ***
o2_(Intercept)   0.1583     0.2095    0.76    0.45
o2_xo2           0.8375     0.1672    5.01 5.5e-07 ***
sigma1           0.9319     0.0948    9.83 < 2e-16 ***
sigma2           0.9070     0.0466   19.45 < 2e-16 ***
rho1             0.8899     0.0515   17.29 < 2e-16 ***
rho2             0.1768     0.3766    0.47    0.64
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Wald chi2 (null): 101 on 3 DF, p-value: < 0
Wald chi2 (rho): 299 on 2 DF, p-value: < 0
```

Affiliation:

Daniel Heimgartner
Institute for Transport Planning and Systems
Eidgenössische Technische Hochschule Zürich
IFW C 46.1
Haldeneggsteig 4
8092 Zürich, Switzerland
E-mail: daniel.heimgartner@ivt.baug.ethz.ch

Xinyi Wang
Department of Urban Studies and Planning
Massachusetts Institute of Technology
105 Massachusetts Avenue
Cambridge, MA 02139
E-mail: xinyi174@mit.edu

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd