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

Daniel Heimgartner
ETH Zürich

Xinyi Wang
MIT Boston

Abstract

This introduction to the R package **OPSR** is a (slightly) modified version of a submission to the *Journal of Statistical Software*. 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.

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 comparison 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 may control age for the comparison, variables such as risk-seeking are not readily measured, and it may still be part of the error in applied research and thus leading cause of a selection bias.

To properly account for the selection bias, various techniques exist, both for longitudinal and cross-sectional data. In the first case, difference in differences is a widely adopted measure. In the latter case, instrumental variables, matching propensity scores, 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 modelling 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 equation and 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 either placing different distributional assumptions on the errors and/or how the latent process manifests into observed outcomes (i.e., the dependent variables can be of various types, such as binary, ordinal, censored, or continuous) more generally known as conditional mixed-process (CMP) models. CMP models comprise 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) allows to 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** obeys to 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 make **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. 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 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. Further, it is shown, that **OPSR** can be used to estimate the well-known Tobit-5 model and yields the same parameters as the implementation in **sampleSelection**. The case study in Section 4 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 5 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.

As alluded, 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

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

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

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

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

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

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

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

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

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

where the negative fraction $(-\frac{\phi(\kappa_j - \boldsymbol{W\gamma}) - \phi(\kappa_{j-1} - \boldsymbol{W\gamma})}{\Phi(\kappa_j - \boldsymbol{W\gamma}) - \Phi(\kappa_{j-1} - \boldsymbol{W\gamma})})$ 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 precomputed based on an ordered probit model and then included in the second stage regression, which describes the Heckman correction (Heckman 1979). It should be warned, that since the Heckman two-step procedure includes an estimate 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 $j' \neq j$)

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

Let's assume that $y_j = \ln(Y_j + \delta)$ in the previous equations. I.e., the continuous outcome was log-transformed as is usual in regression analysis. We have to note, that in such cases the 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

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

for the factual case, and

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

for the counterfactual case (Wang and Mokhtarian 2024).

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).
- 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. Then the main functionality of the package is demonstrated. We conclude this section by demonstrating some nuances, reproducing the core model of Wang and Mokhtarian (2024). Finally, we show that **OPSR** can also estimate the classic Tobit-5 model and matches the results obtained with the implementation from **sampleSelection**.

3.1. OPSR core

Let us simulate date 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()
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, two processes are 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, 102 iterations
Return code 0: successful convergence
Runtime: 0.578 secs
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     0.0932  -21.05 < 2e-16 ***
kappa2           0.8826     0.0615   14.36 < 2e-16 ***
s_xs1            0.9373     0.0570   16.44 < 2e-16 ***
s_xs2            1.4964     0.0712   21.01 < 2e-16 ***
o1_(Intercept)   0.9877     0.1440    6.86   7e-12 ***
o1_xo1           2.0512     0.0930   22.06 < 2e-16 ***
o1_xo2           1.0133     0.0712   14.23 < 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   37.73 < 2e-16 ***
o3_(Intercept)   1.0028     0.0909   11.03 < 2e-16 ***
o3_xo1           1.5766     0.0560   28.15 < 2e-16 ***
o3_xo2          -1.9227     0.0528  -36.44 < 2e-16 ***
sigma1           1.0442     0.0534   19.55 < 2e-16 ***
sigma2           1.0478     0.0316   33.12 < 2e-16 ***
sigma3           1.1717     0.0430   27.27 < 2e-16 ***
rho1             0.1696     0.1361    1.25 0.21279
rho2             0.3482     0.0639    5.45   5e-08 ***
rho3             0.3699     0.1077    3.44 0.00059 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Wald chi2 (null): 5071 on 8 DF, p-value: < 0
Wald chi2 (rho): 42.7 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 along absolute counts are reported.
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 (0.14)          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 (0.38) *        0.99 (0.14) ***
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 $\rho_j \sigma_j \text{IMR}$ respectively $\rho_{j'} \sigma_{j'} \text{IMR}$ (if `counterfact = j'` was specified) from Equation 7 or 8.
- `type = "Xb"`: Returns $\boldsymbol{X}_j \boldsymbol{\beta}_j$ respectively $\boldsymbol{X}_{j'} \boldsymbol{\beta}_{j'}$ (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 4, Figure 3.

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"; `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 ***
```

	Structural	Selection	NTWer (535)	NUTWer (322)	UTWer (727)
Kappa 1	1.23 (0.17)***				
Kappa 2	2.46 (0.18)***				
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 (0.32)*	
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.

```
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
```

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`, e.g., the NUTWers switching from NTWing to NUTWing.

3.2. 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 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, 67 iterations
```

```
Return code 0: successful convergence
Runtime: 0.0944 secs
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     0.1692    0.16    0.87
o1_xo1           0.8396     0.1453    5.78 7.6e-09 ***
o2_(Intercept)   0.1583     0.2129    0.74    0.46
o2_xo2           0.8375     0.1669    5.02 5.2e-07 ***
sigma1           0.9319     0.0949    9.82 < 2e-16 ***
sigma2           0.9070     0.0472   19.20 < 2e-16 ***
rho1             0.8899     0.0515   17.27 < 2e-16 ***
rho2             0.1768     0.3848    0.46    0.65
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Wald chi2 (null): 100 on 3 DF, p-value: < 0
Wald chi2 (rho): 298 on 2 DF, p-value: < 0
```

4. 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 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.

We use the TimeUse+ dataset (Winkler *et al.* 2024), a smartphone-based diary, recording

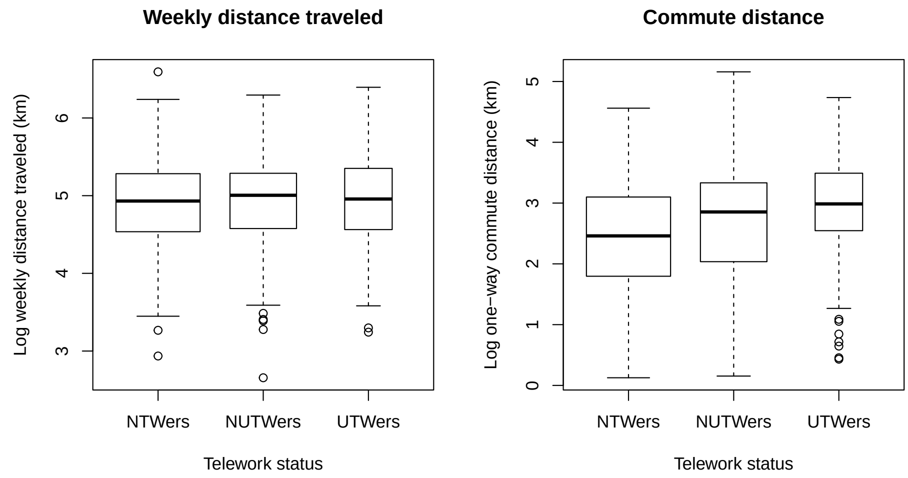

Figure 1: Log weekly distance traveled and log one-way commute distance for different telework statuses.

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 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 conterfactuals 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 mentioned in Section 2, the analyst needs to think of an identification restriction: In our application, we reserve the international standard classification of occupations (ISCO-08) variables for the selection process. To simplify model specification, we first estimate the ordered probit model separately, using `polr()` from the **MASS** package (Venables and Ripley 2002). It should be noted here, that the resulting parameter estimates of the selection process are unbiased.

```
R> drop <- c("id", "weekly_km", "log_weekly_km", "commute_km", "log_commute_km",
+    "wfh_days")
R> dat_polr <- subset(timeuse_data, select = !(names(timeuse_data) %in% drop))
R> dat_polr$wfh <- factor(dat_polr$wfh)
R> fit_polr <- MASS::polr(wfh ~ ., dat_polr, method = "probit")
```

The `stepAIC()` function chooses a selection model specification by AIC in a stepwise algorithm.

```
R> fit_step <- MASS::stepAIC(fit_polr, trace = FALSE)
R> fit_step$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
wfh ~ start_tracking + age + car_access + dogs + driverlicense +
    educ_higher + fixed_workplace + grocery_shopper + hh_income +
    hh_size + isco_clerical + isco_craft + isco_elementary +
    isco_managers + isco_plant + isco_professionals + isco_service +
    isco_agri + isco_tech + married + n_children + freq_onl_order +
    parking_home + parking_work + permanent_employed + rents_home +
    res_loc + sex_male + shift_work + swiss + vacation + workload +
    young_kids

Final Model:
wfh ~ age + car_access + educ_higher + fixed_workplace + grocery_shopper +
    hh_income + isco_clerical + isco_craft + isco_elementary +
    isco_tech + freq_onl_order + parking_home + permanent_employed +
    shift_work + workload + young_kids


                  Step Df Deviance Resid. Df Resid. Dev  AIC
1                                        778       1429 1521
2      - start_tracking  6   2.0260       784       1431 1511
3             - res_loc  3   2.8697       787       1434 1508
4       - isco_managers  1   0.0133       788       1434 1506
5           - isco_agri  1   0.0415       789       1434 1504
6            - vacation  1   0.1212       790       1434 1502
7       - driverlicense  1   0.2959       791       1434 1500
8            - sex_male  1   0.5118       792       1435 1499
9          - n_children  1   0.4769       793       1435 1497
10            - hh_size  1   0.4137       794       1436 1496
11            - married  1   0.3909       795       1436 1494
12       - isco_service  1   0.4773       796       1437 1493
```

```
13          - isco_plant  1   0.6176       797       1437 1491
14          - rents_home  1   1.2327       798       1438 1490
15        - parking_work  1   1.1424       799       1440 1490
16               - swiss  1   1.5091       800       1441 1489
17                - dogs  1   1.8158       801       1443 1489
18 - isco_professionals  1   1.8728       802       1445 1489
```

The resulting selection process specification can then be passed to `opsr()`, along with a common (or separate) process specification for the outcome processes. **OPSR** recognizes potential identification problems (e.g., colinear variables or missing factor levels in one of the groups), raises a warning if such problems arise and fixes the causing coefficients at 0. Through this process, we have identified two singularity issues for the UTWers: First, `shift_work` is a constant and second, `parking_home` is colinear with `car_access`.

We then follow the conventional (somewhat heuristic) model building strategy to specify the full identified model and then exclude all 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 from the model specification with p values below some threshold (see `?opsr_step` for further details). The formula specification of the full model is hidden here for brevity.

```
R> fit_full <- opsr(f_full, timeuse_data, printLevel = 0)
R> f_red <- wfh | log_weekly_km ~
+    age + educ_higher + hh_income + young_kids + workload + fixed_workplace +
+    shift_work + permanent_employed + isco_craft + isco_tech + isco_clerical +
+    isco_elementary + car_access + parking_home + freq_onl_order +
+    grocery_shopper |
+    sex_male + res_loc + workload + permanent_employed + parking_work |
+    swiss + res_loc + young_kids + workload + parking_work |
+    sex_male + swiss + fixed_workplace + permanent_employed + parking_work
R> fit_red <- opsr(f_red, timeuse_data, printLevel = 0)
R> print(anova(fit_red, fit_full), print.formula = FALSE)

Likelihood Ratio Test

   logLik      Df    Test Restrictions Pr(>Chi)
1 -1337.0    50.0
2 -1316.8    99.0    40.4           49      0.8

R> summary(fit_red)

Call:
opsr(formula = f_red, data = timeuse_data, printLevel = 0)

BFGS maximization, 234 iterations
Return code 0: successful convergence
Runtime: 3.46 secs
```

```
Number of regimes: 3
Number of observations: 824 (424, 265, 135)
Estimated parameters: 50

Log-Likelihood: -1337
AIC: 2774
BIC: 3010
Pseudo R-squared (EL): 0.202
Pseudo R-squared (MS): 0.126
Multiple R-squared: 0.214 (0.201, 0.189, 0.289)

Estimates:
                      Estimate Std. error t value Pr(> t)
kappa1                 0.13345    0.40919    0.33 0.74433
kappa2                 1.25047    0.40781    3.07 0.00217 **
s_age                  0.00725    0.00403    1.80 0.07219 .
s_educ_higher          0.44929    0.09295    4.83 1.3e-06 ***
s_hh_income4001_8000  -1.06428    0.25627   -4.15 3.3e-05 ***
s_hh_income8001_12000 -0.89366    0.25137   -3.56 0.00038 ***
s_hh_income12001_16000 -0.72192    0.26184   -2.76 0.00583 **
s_hh_income16001+     -0.69387    0.28776   -2.41 0.01590 *
s_hh_incomeNA         -0.63145    0.34501   -1.83 0.06722 .
s_young_kids           0.29617    0.10095    2.93 0.00335 **
s_workload             0.05353    0.02404    2.23 0.02598 *
s_fixed_workplace     -0.55419    0.14298   -3.88 0.00011 ***
s_shift_work          -0.82518    0.16677   -4.95 7.5e-07 ***
s_permanent_employed   0.33270    0.18560    1.79 0.07305 .
s_isco_craft          -0.67913    0.22364   -3.04 0.00239 **
s_isco_tech            0.21921    0.13246    1.65 0.09794 .
s_isco_clerical        0.55330    0.09817    5.64 1.7e-08 ***
s_isco_elementary     -4.46545    1.29525   -3.45 0.00057 ***
s_car_access          -0.71446    0.26447   -2.70 0.00690 **
s_parking_home         0.64134    0.25170    2.55 0.01083 *
s_freq_onl_order       0.20944    0.08812    2.38 0.01747 *
s_grocery_shopper     -0.13267    0.08788   -1.51 0.13116
o1_(Intercept)         3.90114    0.17240   22.63 < 2e-16 ***
o1_sex_male            0.09334    0.05623    1.66 0.09691 .
o1_res_locrural        0.21702    0.09467    2.29 0.02188 *
o1_res_locsuburban     0.10923    0.09818    1.11 0.26593
o1_res_locurban       -0.01088    0.10899   -0.10 0.92049
o1_workload            0.06058    0.01314    4.61 4.0e-06 ***
o1_permanent_employed  0.29905    0.11690    2.56 0.01052 *
o1_parking_work        0.23222    0.05204    4.46 8.1e-06 ***
o2_(Intercept)         3.88702    0.21570   18.02 < 2e-16 ***
o2_swiss               0.18517    0.10900    1.70 0.08935 .
o2_res_locrural        0.43405    0.14799    2.93 0.00336 **
o2_res_locsuburban     0.22649    0.14363    1.58 0.11482
```

```
o2_res_locurban          0.17387    0.16409    1.06 0.28933
o2_young_kids           -0.15630    0.06958   -2.25 0.02469 *
o2_workload              0.07455    0.01515    4.92 8.7e-07 ***
o2_parking_work          0.16130    0.07033    2.29 0.02182 *
o3_(Intercept)           3.85223    0.33710   11.43 < 2e-16 ***
o3_sex_male              0.23879    0.08908    2.68 0.00735 **
o3_swiss                 0.39109    0.12054    3.24 0.00118 **
o3_fixed_workplace      -0.36832    0.12432   -2.96 0.00305 **
o3_permanent_employed    0.54238    0.25706    2.11 0.03487 *
o3_parking_work          0.28905    0.09202    3.14 0.00168 **
sigma1                   0.51246    0.02225   23.04 < 2e-16 ***
sigma2                   0.54030    0.02954   18.29 < 2e-16 ***
sigma3                   0.54263    0.05799    9.36 < 2e-16 ***
rho1                     0.28712    0.19141    1.50 0.13360
rho2                    -0.18889    0.14349   -1.32 0.18804
rho3                     0.46193    0.22531    2.05 0.04034 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Wald chi2 (null): 2584 on 39 DF, p-value: < 0
Wald chi2 (rho): 8.93 on 3 DF, p-value: < 0.03
```

The reduced model specification (`fit_red`) is not rejected in the likelihood ratio test. Further, there is significant error correlation between the selection process and the outcome process for the UTWers (`rho3`). The Wald-test suggests that the null hypothesis (`rho1` = `rho2` = `rho3` = 0) can be rejected at the 5% level, suggesting that OPSR is beneficial given our model assumptions.

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   37.7 ***   -10.6 ***  -54.2 ***
NTWing->UTWing   -54.7 ***   -47.0 ***  -44.3 ***
NUTWing->UTWing  -92.4 ***   -36.4 ***    9.9 **
---
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 (37.74 km; column

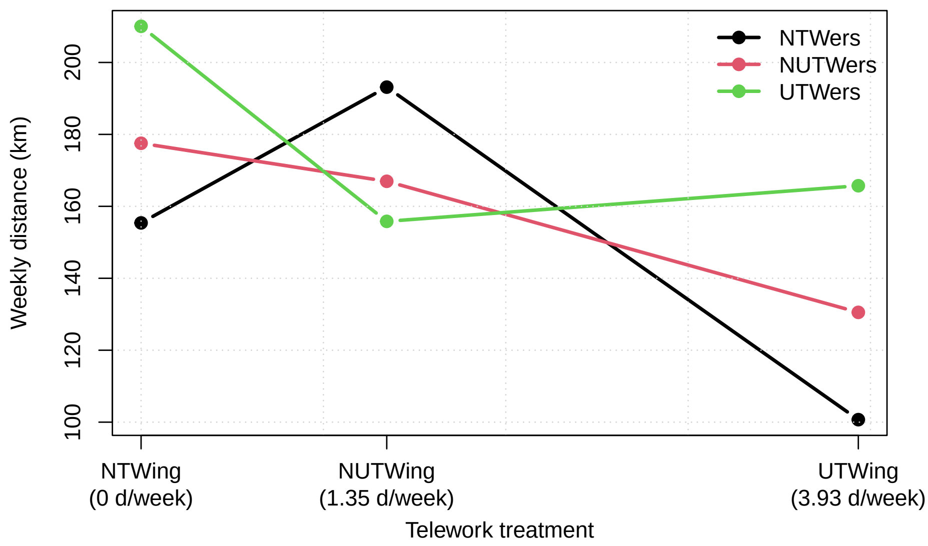

Figure 2: Treatment effects.

`NTWer`, row `NTWing -> NUTWing`) and UTWers who would travel more when further adopting telework from NUTWing (9.9 km). The treatment effects when switching from NTWing to NUTWing are strongest for UTWers (-54.22 km) compared to NTWers (37.74 km) and NUTWers (-10.58 km). Treatment effects for NTWing to UTWing are similar across all three groups, slightly stronger for NTWers (-54.7 km). Interestingly, NTWers show a non-linear pattern, first increasing weekly kilometers when adopting some telework (37.74 km; NTWing to NUTWing) but then substantially decreasing weekly kilometers with more telework (-92.44 km; NUTWing to UTWing). An explanation could be, that these individuals (living closer to their workplace) do initially 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 (-54.22 km) but upon further telework adoption (NUTWing to UTWing) only minimally adjusting weekly kilometers (9.9 km). A similar argument could be made, that these individuals (living further from their workplace) already from the start adjust activity chains and location choices. One can therefore conclude, that the treatment effect over the full range (NTWing to UTWing) is similar across all groups but the main travel reduction happens at different treatment intensities. Figure 2 visualizes these treatment effects and shows the 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 3 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,

```
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)
```

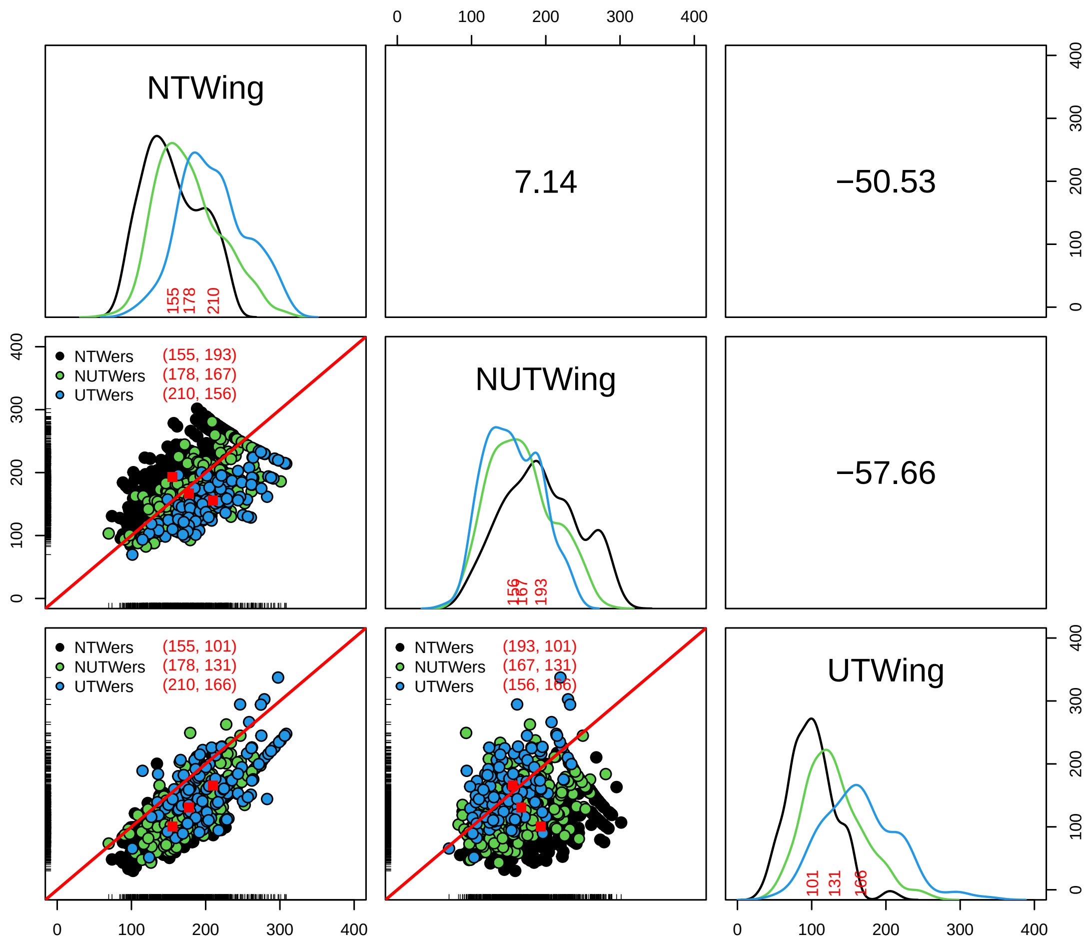

Figure 3: Pairs plot for model fits of class '`opsr`'.

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.

Model	Parent	Error correlation	Description
`fit_full`		•	Full identified model, including all variables as linear effects
`fit_red`	`fit_full`	•	Excluding all variables not significant at the 10% level
`fit_nocor`	`fit_red`	◦	Fixing the `rho` coefficients at 0

Table 2: Model overview. The model is based on *Parent* as elaborated under *Description*.

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 setting the `rho` coefficients to 0. I.e., 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> 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    37.7 ***   -10.6 ***  -54.2 ***
NTWing->UTWing    -54.7 ***   -47.0 ***  -44.3 ***
NUTWing->UTWing   -92.4 ***   -36.4 ***    9.9 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R> te(fit_nocor)

                  NTWers     NUTWers     UTWers
NTWing->NUTWing   17.72 ***   14.14 ***  14.55 ***
NTWing->UTWing     8.80 ***   12.31 ***   7.89 *
NUTWing->UTWing   -8.93 ***   -1.83      -6.66 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we see, `fit_nocor` yields completely different insights, in particular, that telework generally increases weekly distance traveled, 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).

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)
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      28.0   -2.69 -20.97
NTWing->UTWing      -40.6  -11.95 -17.14
NUTWing->UTWing     -68.6   -9.26   3.83
```

Generally, telework reduces weekly distance traveled by less than the foregone commute distance, which indicates, that a rebound effect (compensating leisure travel) exists. For example, the NUTWers could save 43.33 km in commute travel but only reduce -2.69 km per marginal teleworking day when switching from NTWing to NUTWing. This compensating travel exists for all TW groups except the NTWers (NTWing to UTWing and NUTWing to UTWing), where we observe diminished travel activity beyond foregone commutes. 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.

5. 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 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 selection biases for ordinal treatments (where the well-known Tobit-5 model is a special case of OPSR, i.e., 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 here presented R implementation in package **OPSR** re-uses 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. 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 appropriate model and check for error correlation, and second, in how far computed treatment effects differ if the error correlation is not accounted for. We find that, overall, telework reduces travel. Non-teleworkers tend to have shorter commutes and adjust mobility patterns mainly when switching from non-usual telework to usual telework. On the other hand, weekly distance traveled slightly increases when initially adopting some telework. Contrary, 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 telework generally reduces weekly distance traveled and it does so by less than the foregone commute. Therefore, some compensating travel (rebound effects) exists for most of the teleworker groups.

Computational details

The results in this paper were obtained using R 4.4.0 with the packages **OPSR** 0.2.0.9001, **MASS** 7.3.60, **texreg** 1.39.4, **sampleSelection** 1.2.12, **mvtnorm** 1.2.5, **gridExtra** 2.3 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/`.

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:10.1177/1536867X1101100202.

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.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S.* 4th edition. Springer-Verlag, New York. doi:10.1007/978-0-387-21706-2.

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.

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