RationalExp: Tests of and Deviations from Rational Expectations

Xavier D'Haultfœuille
CREST

Christophe Gaillac
CREST and TSE

Arnaud Maurel
Duke University and NBER

Abstract

This vignette presents the R package **RationalExp** associated to D'Haultfœuille, Gaillac, and Maurel (2018) (DGM hereafter). This package implements a test of the rational expectations hypothesis based on the marginal distributions of realizations and subjective beliefs. This test can be used in cases where realizations and subjective beliefs are observed in two different datasets that cannot be matched, or when they are observed in the same dataset. The test can be implemented with covariates and survey weights. The package also computes the estimator of the minimal deviations from rational expectations than can be rationalized by the data. R and the package **RationalExp** are open-source software projects and can be freely downloaded from CRAN: `http://cran.r-project.org`.

Keywords: Rational expectations, test, minimal deviations, R.

1. How to get started

R is an open source software project and can be freely downloaded from the CRAN website. The R package **RationalExp** can be downloaded from `cran.r-project.org`. To install the **RationalExp** package from R use

```
install.packages("RationalExp")
```

Provided that your machine has a proper internet connection and you have write permission in the appropriate system directories, the installation of the package should proceed automatically. Once the **RationalExp** package is installed, it can be loaded to the current R session by the command

```
library(RationalExp)
```

Online help is available in two ways: either help(package="RationalExp") or ?test. The first gives an overview over the available commands in the package. The second gives detailed information about a specific command. A valuable feature of R help files is that the examples used to illustrate commands are executable, so they can be pasted into an R session or run as a group with a command like example(test).

The R package **RationalExp** can be also downloaded from Github `https://github.com/cgaillac/RationalExp`. To install the **RationalExp** package from Github, the devtools library is required. Then, use the command

```
library("devtools")
install_github('RationalExp','cgaillac')
```

2. Theory

2.1. Testing rational expectations

We first explain the test procedure proposed in DGM for the test with covariates:

$$\begin{aligned}\mathrm{H}_{0X} : &\text{ there exists a pair of random variables } (Y', \psi') \text{ and a sigma-algebra } \mathcal{I}' \\ &\text{such that } \sigma(\psi', X) \subset \mathcal{I}',\ Y'|X \sim Y|X,\ \psi'|X \sim \psi|X \text{ and } \mathbb{E}\left[Y'|\mathcal{I}'\right] = \psi'.\end{aligned}$$

To simplify notation, we use as in DGM a potential outcome framework to describe our data combination problem. Specifically, instead of observing (Y, ψ), we suppose to observe only the covariates X, $\widetilde{Y} = DY + (1-D)\psi$ and D, where $D = 1$ (resp. $D = 0$) if the unit belongs to the dataset of Y (resp. ψ). We assume that the two samples are drawn from the same population, which amounts to supposing that $D \perp\!\!\!\perp (X, Y, \psi)$. In order to build DGM's test, DGM use the following characterization of H_{0X}:

$$\mathbb{E}\left[(y-Y)^+ - (y-\psi)^+ \middle| X\right] \geq 0 \quad \forall y \in \mathbb{R} \text{ and } \quad \mathbb{E}\left[Y - \psi | X\right] = 0,$$

where $u^+ = \max(0, u)$. Equivalently but written with $\widetilde{Y}$ only,

$$\mathbb{E}\left[W\left(y - \widetilde{Y}\right)^+ \middle| X\right] \geq 0 \quad \forall y \in \mathbb{R} \text{ and } \quad \mathbb{E}\left[W\widetilde{Y} \middle| X\right] = 0,$$

where $W = D/\mathbb{E}(D) - (1-D)/\mathbb{E}(1-D)$. This formulation of the null hypothesis allows one to apply the instrumental functions approach of Andrews and Shi (2017) (AS hereafter), who consider the issue of testing many conditional moment inequalities and equalities. The initial step is to transform the conditional moments into the following unconditional moments conditions:

$$\mathsf{E}\left[W\left(y - \widetilde{Y}\right)^+ h(X)\right] \geq 0, \quad \mathsf{E}\left[(Y-\psi)\, h(X)\right] = 0.$$

for all $y \in \mathbb{R}$ and $h \geq 0$ belonging to a suitable class of functions.

We suppose to observe here a sample $(D_i, X_i, \widetilde{Y}_i)_{i=1\ldots n}$ of n i.i.d. copies of $(D, X, \widetilde{Y})$. For notational convenience, we let $\widetilde{X}_i$ denote the nontransformed vector of covariates and redefine X_i as the transformed vector in the following way:

$$X_i = \Phi_0\left(\widehat{\Sigma}_{\widetilde{X},n}^{-1/2}\left(\widetilde{X}_i - \overline{\widetilde{X}}_i\right)\right),$$

where, for any $x = (x_1, \ldots, x_{d_X})$, we let $\Phi_0(x) = (\Phi(x_1), \ldots, \Phi(x_{d_X}))^\top$. Here Φ denotes the standard normal cdf, $\widehat{\Sigma}_{\widetilde{X},n}$ is the sample covariance matrix of $\left(\widetilde{X}_i\right)_{i=1\ldots n}$ and $\overline{\widetilde{X}}_n$ its sample mean.

Now that $X_i \in [0,1]^{d_X}$, we consider h functions that are indicators of belonging to specific hypercubes within $[0,1]^{d_X}$. Namely, we consider the class of functions $\mathcal{H}_r = \{h_{a,r},\ a \in A_r\}$,

with $A_r = \{1, 2, \ldots, 2r\}^{d_X}$ $(r \geq 1)$, $h_{a,r}(x) = \mathbb{1}\{x \in C_{a,r}\}$ and, for any $a = (a_1, ..., a_{d_X})^\top \in A_r$,

$$C_{a,r} = \prod_{u=1}^{d_X} \left(\frac{a_u - 1}{2r}, \frac{a_u}{2r} \right].$$

To define the test statistic T, we need to introduce additional notation. First, we define, for any given y,

$$m\left(D_i, \widetilde{Y}_i, X_i, h, y\right) = \begin{pmatrix} m_1\left(D_i, \widetilde{Y}_i, X_i, h, y\right) \\ m_2\left(D_i, \widetilde{Y}_i, X_i, h, y\right) \end{pmatrix} = \begin{pmatrix} w_i \left(y - \widetilde{Y}_i\right)^+ h\left(X_i\right) \\ w_i \widetilde{Y}_i h\left(X_i\right) \end{pmatrix}, \tag{1}$$

where $w_i = nD_i / \sum_{j=1}^n D_j - n(1-D_i) / \sum_{j=1}^n (1-D_j)$. Let $\overline{m}_n(h, y) = \sum_{i=1}^n m\left(D_i, \widetilde{Y}_i, X_i, h, y\right)/n$ and define similarly $\overline{m}_{n,j}$ for $j = 1, 2$. For any function h and any $y \in \mathbb{R}$, let us also define, for some $\epsilon > 0$,

$$\overline{\Sigma}_n(h, y) = \widehat{\Sigma}_n(h, y) + \epsilon \text{Diag}\left(\widehat{\mathbb{V}}\left(\widetilde{Y}\right), \widehat{\mathbb{V}}\left(\widetilde{Y}\right)\right),$$

where $\widehat{\Sigma}_n(h, y)$ is the sample covariance matrix of $\sqrt{n}\overline{m}_n(h, y)$ and $\widehat{\mathbb{V}}\left(\widetilde{Y}\right)$ is the empirical variance of $\widetilde{Y}$. We then denote by $\overline{\Sigma}_{n,jj}(h, y)(j = 1, 2)$ the j-th diagonal term of $\overline{\Sigma}_n(h, y)$.

Then the (Cramér-von-Mises) test statistic T is defined by:

$$T = \sup_{y \in \widehat{\mathcal{Y}}} \sum_{r=1}^{r_n} \frac{(2r)^{-d_X}}{(r^2 + 100)} \sum_{a \in A_r} \left((1-p) \left(-\frac{\sqrt{n}\overline{m}_{n,1}(h_{a,r}, y)}{\overline{\Sigma}_{n,11}(h_{a,r}, y)^{1/2}} \right)^{+2} + p \left(\frac{\sqrt{n}\overline{m}_{n,2}(h_{a,r}, y)}{\overline{\Sigma}_{n,22}(h_{a,r}, y)^{1/2}} \right)^2 \right),$$

where $\widehat{\mathcal{Y}} = \left[\min_{i=1,\ldots,n} \widetilde{Y}_i, \max_{i=1,\ldots,n} \widetilde{Y}_i\right]$, p is a parameter that weights the moments inequalities versus equalities and $(r_n)_{n \in \mathbb{N}}$ is a deterministic sequence tending to infinity.

To test for rational expectations in the absence of covariates, we simply restrict ourselves to the constant function $h(X) = 1$, and the test statistic is simply

$$T = \sup_{y \in \widehat{\mathcal{Y}}} (1-p) \left(-\frac{\sqrt{n}\overline{m}_{n,1}(y)}{\overline{\Sigma}_{n,11}(y)^{1/2}} \right)^{+2} + p \left(\frac{\sqrt{n}\overline{m}_{n,2}(y)}{\overline{\Sigma}_{n,22}(y)^{1/2}} \right)^2,$$

where $\overline{m}_{n,j}(y)$ and $\overline{\Sigma}_{n,jj}(y)$ $(j = 1, 2)$ are defined as above but with $h(x) = 1$.

Whether or not covariates are included, the resulting test is of the form $\varphi_{n,\alpha} = \mathbb{1}\left\{T > c^*_{n,\alpha}\right\}$ where the estimated critical value $c^*_{n,\alpha}$ is obtained by bootstrap using as in AS the Generalized Moment Selection method. Specifically, we follow these three steps:

1. Compute the function $\overline{\varphi}_n(y, h) = \left(\overline{\varphi}_{n,1}(y, h), 0\right)^\top$ for (y, h) in $\widehat{\mathcal{Y}} \times \cup_{r=1}^{r_n} \mathcal{H}_r$, with

$$\overline{\varphi}_{n,1}(y, h) = \overline{\Sigma}_{n,11}^{1/2} B_n \mathbb{1}\left\{ \frac{n^{1/2}}{\kappa_n} \overline{\Sigma}_{n,11}^{-1/2} \overline{m}_{n,1}(y, h) > 1 \right\},$$

and where $B_n = (b_0 \ln(n) / \ln(\ln(n)))^{1/2}$, $b_0 > 0$, $\kappa_n = (\kappa \ln(n))^{1/2}$, and $\kappa > 0$. To compute $\overline{\Sigma}_{n,11}$, we fix ϵ to 0.05, as in AS.

2. Let $\left(D_i^*, \widetilde{Y}_i^*, X_i^*\right)_{i=1,...,n}$ denote a bootstrap sample, i.e., an i.i.d. sample from the empirical cdf of $\left(D, \widetilde{Y}, X\right)$, and compute from this sample $\overline{m}_n^*$ and $\overline{\Sigma}_n^*$. Then compute T^* like T, replacing $\overline{\Sigma}_n(y, h_{a,r})$ and $\sqrt{n}\overline{m}_n(y, h_{a,r})$ by $\overline{\Sigma}_n^*(y, h_{a,r})$ and

$$\sqrt{n}\left(\overline{m}_n^* - \overline{m}_n\right)(y, h_{a,r}) + \overline{\varphi}_n(y, h_{a,r}).$$

3. The threshold $c_{n,\alpha}^*$ is the (conditional) quantile of order $1 - \alpha + \eta$ of $T^* + \eta$ for some $\eta > 0$. Following AS, we set η to 10^{-6}.

2.2. Minimal Deviations from Rational Expectations

In DGM, we also introduce the minimal deviations from rational expectations, as the unique function g^* satisfying:

$$\mathbb{E}[\rho(|\psi - g^*(\psi)|)] = \inf_{(Y',\psi',\psi'')\in\mathbf{\Psi}} \mathbb{E}[\rho(|\psi' - \psi''|)].$$

We refer the user to DGM for the motivation behind the computation of g^* and the formal existence and unicity result. Though g^* does not have a simple form in general, DGM propose in the following a simple procedure to construct a consistent estimator of it, based on i.i.d. copies $(Y_i)_{i=1...L}$ and $(\psi_i)_{i=1...L}$ of Y and ψ. For simplicity, we suppose hereafter that the two samples have equal size.[1]

To define the DGM's estimator, note first that we have

$$g^* = \arg\min_{g\in\mathcal{G}_0} \mathbb{E}\left[(\psi - g(\psi))^2\right], \tag{2}$$

where the set $\mathcal{G}_0$ is defined by

$$\mathcal{G}_0 = \left\{g \text{ non-decreasing} : \mathbb{E}\left[(y-Y)^+ - (y - g(\psi))^+\right] \geq 0 \ \forall y \in \mathbb{R},\ \mathbb{E}[g(\psi)] = \mathbb{E}[Y]\right\}.$$

In other words, g^* is the (increasing) function such that (i) $g^*(\psi)$ is closest to ψ for the L^2 norm; (ii) g^* belongs to $\mathcal{G}_0$, which means that we can rationalize $\mathbb{E}(Y|g^*(\psi)) = g^*(\psi)$.

To estimate g^*, we replace expectations and cdfs by their empirical counterpart. Letting $\psi_{(1)} < ... < \psi_{(L)}$ denote the ordered statistic, our estimator of $\left(g^*(\psi_{(1)}), ..., g^*(\psi_{(L)})\right)$ is the solution of:

$$\left(\widehat{g}^*(\psi_{(1)}), ..., \widehat{g}^*(\psi_{(L)})\right) = \arg\min_{\widetilde{\psi}_{(1)}<\cdots<\widetilde{\psi}_{(L)}} \sum_{i=1}^{L}\left(\psi_{(i)} - \widetilde{\psi}_{(i)}\right)^2 \text{ s.t.} \sum_{i=j}^{L} Y_{(i)} - \widetilde{\psi}_{(i)} \geq 0,\ j = 2...L,$$
$$\sum_{i=1}^{L} Y_{(i)} - \widetilde{\psi}_{(i)} = 0. \tag{3}$$

[1] If both samples do not have equal size, one can first apply our analysis after taking a random subsample of the larger one, with the same size as the smaller one. Then we can compute the average of the estimates over a large number of such random subsamples.

Then, for any $t \in \mathbb{R}$, we let

$$\widehat{g}^*(t) = \widehat{g}^*\left(\min\{(\psi_i)_{i=1\ldots L} : \psi_i \geq \min\{t, \psi_{(L)}\}\right).$$

We solve the convex quadratic programming problem (3) using the algorithm proposed in Suehiro, Hatano, Kijima, Takimoto, and Nagano (2012). We refer to DGM for more details.

3. The functions in the RationalExp package

3.1. The test function

This function implements the RE tests proposed in DGM. The code of the `test()` function is based on the Stata code cmi_test from Andrews, Kim, and Shi (2017). The syntax of the function `test()` is as follows:

```
test(Y_tilde,D,X,weights,generalized,nbCores,tuningParam)
```

Y_tilde	a vector of size $n_Y + n_\psi$ stacking first the $(Y_i)_{i=1,\ldots,n_Y}$, then the $(\psi_i)_{i=1,\ldots,n_\psi}$.
D	a vector stacking the $(D_i)_{i=1\ldots n}$: n_Y ones, then n_ψ zeros.
X	the matrix of covariates. Equal to a vector of ones by default (in which case the test without covariates is performed).
weights	the vector of survey weights. Uniform by default.
generalized	whether a generalized test should be performed or not: "Add" for additive shocks (default), "Mult" for multiplicative shocks. Set by default to "No" (no generalized test).
nbCores	the number of cores used by the program. To reduce the computational time, this function can use several cores, in which case the library snowfall should be loaded first. By default, nbCores is set to 1.
tuningParam	a dictionnary, including the parameters p, epsil, B, c, kap, and y_grid. The first four corresponds to the parameters p, ϵ, B, c, and κ above (with default values equal to 0.05, 0.05, 500, 0.3 and 0.001 respectively). Following AS, the interval $\widehat{\mathcal{Y}} = \left[\min_{i=1\ldots n} \widetilde{Y}_i, \max_{i=1,\ldots,n} \widetilde{Y}_i\right]$ is approximated by a grid denoted by y_grid. By default, y_grid is equal to the empirical quantiles of $\widetilde{Y}$ of order 0, 1/29 2/29,..., and 1.

We refer to Section 4.1 below for an example of the use of several cores, and to Section 4.2 for how to modify tuningParam. We also refer to the reference manuel or help file for additional details.

3.2. The estimDev function

This function estimates the minimal deviations from RE. The estimDev() function has the following syntax:

```
estimDev(psi, y)
```

psi vector of subjective expectations
y vector of realizations of an individual outcome.

Both vectors should have the same length. If not, one can randomly select a subset of the longer vector with length equal to that of the shorter one. The function returns a function via the approxfun of the package stats. This function can then be evaluated directly on a desired grid. We refer the user to Section 4.3 for an example.

4. Examples

4.1. Test without covariates

We consider the same DGP as in DGM (Section 5), namely we suppose that the outcome Y is given by

$$Y = \rho\psi + \epsilon,$$

with $\rho \in [0, 1]$, $\psi \sim \mathcal{N}(0, 1)$ and

$$\epsilon = \zeta \left(-1\{U \leq 0.1\} + 1\{U \geq 0.9\}\right),$$

where ζ, U and ψ are mutually independent, $\zeta \sim \mathcal{N}(2, 0.1)$ and $U \sim \mathcal{U}[0, 1]$. We consider 1,200 observations and $\rho = 0.29$.

```
rm(list=ls())
### load packages
library(RationalExp)
library(snowfall)
set.seed(1829384)

### Data generating process
n_p=1200 # number of observations
n_y=n_p
N <- n_y + n_p
rho <-0.29 # parameter rho
sig=0.1 # parameter sigma
u=1
b=0.10
a=2
psi <-rnorm(n_p,0,u) ## vector of psi's
pp_y <- runif(n_y,0,1)
zeta <- rnorm(n_y,a,sig)
zeta1 <- rnorm(n_y,-a,sig)
pp1_y <- 1*(pp_y <b)
pp2_y <- 1*(pp_y >1-b)
pp3_y <- 1*(pp_y <=(1-b) & pp_y >=b)
psi_y <-rnorm(n_y,0,u)
y =  rho*psi_y+ pp1_y*zeta + pp2_y*zeta1 ## vector of y's
```

We then concatenate the two datasets:

```
y_tilde <- rbind(matrix(y,n_y,1),matrix(psi,n_p,1))  ## concatenation of y then psi
D <- rbind(matrix(1,n_y,1),matrix(0,n_p,1)) ## vector of D's
```

By default, the function test runs the test without covariates, where system.time() is used to compute the elapsed time:

```
system.time(res <- test(y_tilde ,D))
#> Conditional Moment Inequalities Test   Number of obs :  2400
#> Test Statistic :  4.697479
#>  Critical Value (1%)  2.51336
#> Critical Value (5%)  0.9971287
#> Critical Value (10%)  0.7172106
#> p-value  :  0
#>    user  system elapsed
#>   27.27    0.09   28.74
```

The test prints the total number of observations, the test statistic, the different critical values, and the p-value. It returns a list containing all these informations and the vector of bootstraped test statistics (see the reference manual) and should be obtained using:

```
N <- res[[1]]
cv01 <- res[[2]]
cv05 <- res[[3]]
cv10 <- res[[4]]
T_n<- res[[5]]
p_value <- res[[7]]
T_reps <- res[[8]]
```

We now show how to modify the tuning parameter of the number of cores nbClust (in tuningParam) to 3 and run the test of the parallelized version of the test:

```
system.time(res <- test(y_tilde ,D,NULL,NULL,NULL,3,NULL))
#> Warning in searchCommandline(parallel, cpus = cpus, type = type,
#> socketHosts = socketHosts, : Unknown option on commandline:
#> rmarkdown::render('C:/Users/gaillac/Dropbox/Package_R/RationalExp/
#> vignettes/how-to-use-RationalExp-package.Rmd', encoding
#> R Version:  R version 3.4.4 (2018-03-15)
#> snowfall 1.84-6.1 initialized (using snow 0.4-3): parallel execution on 3 CPUs.
#>
#> Stopping cluster
#>Conditional Moment Inequalities Test   Number of obs :  2400
#>Test Statistic :  4.697479
#> Critical Value (1%)  2.532613
#>Critical Value (5%)  1.28576
#>Critical Value (10%)  0.8261835
#>p-value  :  0
```

```
#> user system elapsed
#> 1.12   0.36  15.34
```

Note that elapsed time has been divided by 1.63.

4.2. Test with covariates

We now present an example of the test with covariates. We consider the same DGP as in DGM (Appendix D), namely we suppose that the outcome Y is given by the following DGP:

$$Y = \rho\psi + \sqrt{X}\epsilon,$$

with $\rho \in [0,1]$, $\psi \sim \mathcal{N}(0,1)$, $X \sim \text{Beta}(0.1, 10)$ and

$$\epsilon = \zeta\left(-1\{U \leq 0.1\} + 1\{U \geq 0.9\}\right),$$

where $\zeta \sim \mathcal{N}(2, 0.1)$ and $U \sim \mathcal{U}[0,1]$. (ψ, ζ, U, X) are supposed to be mutually independent. Again we start with the data generating process:

```
n_p=1200
n_y=n_p
N <- n_y + n_p
sig=0.1
u=1
b=0.10
a=2
alp = 0.1
bet = 10

# Data Generating process
X_p = rbeta(n_p,alp ,  bet)+1
X_y =  rbeta(n_y,alp ,  bet)+1
transf <- function(X_y,   f0){
  res <-f0*sqrt(X_y)
  return(res)
}
psi <-rnorm(n_p,0,u)
pp_y <- runif(n_y,0,1)
zeta <- rnorm(n_y,a,sig)
zeta1 <- rnorm(n_y,-a,sig)
pp1_y <- 1*(pp_y <b)
pp2_y <- 1*(pp_y >1-b)
pp3_y <- 1*(pp_y <=(1-b) & pp_y >=b)
psi_y <-rnorm(n_y,0,u)
y =  rho*psi_y+ transf(X_y, 1)*(pp1_y*zeta + pp2_y*zeta1)
```

We concatenate the two datasets as above:

```
y_tilde <- rbind(matrix(y,n_y,1),matrix(psi,n_p,1))## concatenation of y then psi
D <- rbind(matrix(1,n_y,1),matrix(0,n_p,1)) ## vector of D's
```

Then we concatenate the covariates by rows, those associated with y coming first.

```
X_tilde <- rbind(matrix(X_y,n_y,1),matrix(X_p,n_p,1))
```

Then run the test, after modifying the X parameter, the number of core (from 1 to 3), and the number of grid points (from 30 to 5):

```
tuningParam<- vector(mode="list", length=6)
tuningParam[["p"]] <- 0.05 # the parameter p in  Section 3 in DGM
tuningParam[["epsilon"]] <- 0.05 # the parameter c in  Section 3 in DGM
tuningParam[["B"]] <-500 #the number of bootstrap samples
tuningParam[["y_grid"]] <- quantile(y_tilde,seq(0,1,length.out=5))
tuningParam[["c"]] <- 0.3 # the parameter c in  Section 3 in DGM
tuningParam[["kappa"]] <- 0.001  #the parameter kappa  in  Section 3 in DGM

system.time(res <- test(y_tilde ,D,X,NULL,NULL,NULL,3,tuningParam ))
#> Warning in searchCommandline(parallel, cpus = cpus, type = type,
#> socketHosts = socketHosts, : Unknown option on commandline:
#> rmarkdown::render('C:/Users/gaillac/Dropbox/Package_R/RationalExp/
#> vignettes/how-to-use-RationalExp-package.Rmd', encoding
#> snowfall 1.84-6.1 initialized (using snow 0.4-3): parallel execution on 3 CPUs.
#>
#> Stopping cluster
#> Conditional Moment Inequalities Test   Number of obs :  2400
#>Test Statistic :  2.929523
#> Critical Value (1%)  1.807603
#>Critical Value (5%)  1.162568
#>Critical Value (10%)  0.9045701
#>p-value  :  0
#>user  system  elapsed
#> 0.87  0.36    38.50
#>
#>T_n<-  res[[5]]
#>p_value <- res[[7]]
```

4.3. Estimation of minimal deviations

The data generating process is the same as for the test without covariates in Section 4.1.

```
sig=0.1
u=1
b=0.10
a=2
rho= 0.4
psi <- rnorm(n_p,0,u)
pp_y <- runif(n_y,0,1)
zeta <- rnorm(n_y,a,sig)
```

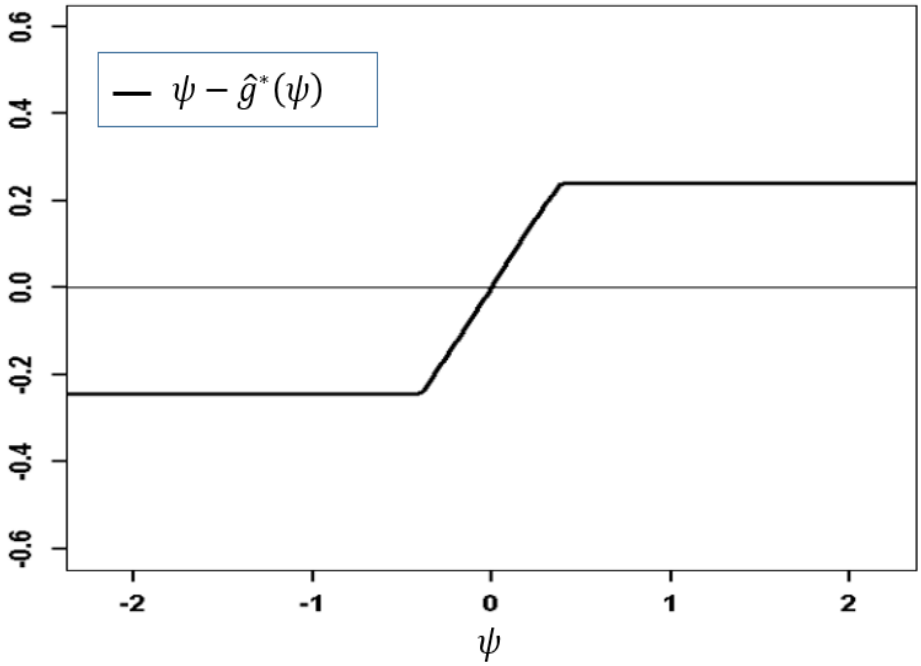
```
zeta1 <- rnorm(n_y,-a,sig)
pp1_y <- 1*(pp_y <b)
pp2_y <- 1*(pp_y >1-b)
pp3_y <- 1*(pp_y <=(1-b) & pp_y >=b)
psi_y <-rnorm(n_p,0,u)
y =  rho*psi_y+ pp1_y*zeta + pp2_y*zeta1
```

Then we estimate g^* using the `estimDev()` function, and the two vectors `psi` and `y` with same length

```
system.time(g_star <- estimDev(psi,y))
#>    user  system elapsed
#>    4.32    0.03    4.62
```

We plot the result on a grid (we refer to D'Haultfœuille *et al.* (2018) for a detailled analysis and an enhanced plot):

```
t<- seq(-2.2,2.2, length.out=300)
plot( t, t- g_star(t),type="l",col=1 , lwd=2, xlim=c(-2.2,2.2),
ylim=c(min(t- g_star(t))-0.1,max(t- g_star(t))+0.1))
abline(h=0)
```



Note: The plain black curve corresponds to the average of $\psi - \widehat{g}^*(\psi)$ over 1, 000 simulations (with $n = 800$)

Figure 1: Estimation of $\psi - g^*(\psi)$.

References

Andrews D, Shi X (2017). "Inference Based on Many Conditional Moment Inequalities." *Journal of Econometrics*, **196**, 119–157.

Andrews DW, Kim W, Shi X (2017). "Commands for testing conditional moment inequalities and equalities." *The Stata journal*, **17**(1).

D'Haultfœuille X, Gaillac C, Maurel A (2018). "Rationalizing Rational Expectations? Tests and Deviations." *NBER Working paper*.

Suehiro D, Hatano K, Kijima S, Takimoto E, Nagano K (2012). "Online prediction under submodular constraints." *International Conference on Algorithmic Learning Theory*, pp. 260–274.

Affiliation:

Xavier D'Haultfœuille
CREST
5, avenue Henry Le Chatelier
91 120 Palaiseau, France
E-mail: xavier.dhaultfoeuille@ensae.fr

Christophe Gaillac
CREST
5, avenue Henry Le Chatelier
91 120 Palaiseau, France
and TSE
E-mail: christophe.gaillac@ensae.fr

Arnaud Maurel
Department of Economics
Duke University
213 Social Sciences
Durham, NC 27708-0097
and NBER
E-mail: arnaud.maurel@duke.edu