Package vignette

Maximum Entropy Bootstrap for Time Series
The `meboot` R-package

Hrishikesh D. Vinod [*] — Fordham University

Javier López-de-Lacalle [†] — Universidad del País Vasco

This version: 26th January 2007

Abstract

The maximum entropy bootstrap is an algorithm that creates an ensemble for time series inference. Stationarity is not required and the ensemble satisfies the ergodic theorem and the central limit theorem. The `meboot` R-package implements such algorithm. This document introduces the procedure and illustrates its scope by means of several guided applications.

Keywords: Non-stationary series, Dependent Data bootstrap, Maximum Entropy, Confidence Intervals.
JEL Classification: C10, C22, C32.

1 Introduction

This paper illustrates the use of `meboot` R-package based on Vinod (2004, 2006). In traditional theory, an ensemble Ω represents the population behind the observed time series. The maximum entropy (ME) bootstrap constructs a large number of replicates (J =999, say) as elements of Ω for inference using a seven-step algorithm designed to satisfy the ergodic theorem (the grand mean of all ensembles is close to the sample mean). The algorithm's practical appeal is that it avoids all structural

[*]Department of Economics, Institute of Ethics and Economic Policy, Fordham University, Bronx, NY 10458, USA. http://www.fordham.edu/economics/vinod/.
E-mail Vinod@forham.edu.

[†]Departamento de Economía Aplicada III. Facultad de Ciencias Económicas y Empresariales del País Vasco. Bilbao, Spain. http://www.bl.ehu.es/~jedlobej/.
E-mail javlacalle@yahoo.es.

change and unit root type testing involving complicated asymptotics and all shape-destroying transformations like de-trending or differencing to achieve stationarity. The constructed ensemble elements retain the basic shape and dependence structure of autocorrelation function (acf) and partial autocorrelation function (pacf) of the original time series.

This discussion collects relevant portions of Vinod (2004, 2006) as templates for users of the meboot package. Let us begin with some motivation. In social sciences the underlying system is often dynamic, complex and adaptive leading to irreversible non-stationary and short time series. Economists often use detrending and differencing to convert such series to stationarity. We avoid the 'non-standard' Dickey-Fuller type sampling distribution of regression coefficients with severe inference problems for panel data. Wiener, Kolmogorov and Khintchine (WKK), among others, developed the stationary model in 1930's where the data x_t arise from a collection of an infinite set of time series, called the ensemble.

Vinod (2004, 2006) offers a computer intensive construction of a plausible ensemble created from a density satisfying the maximum entropy principle. The meboot algorithm uses quantiles $x_{j,t}$ for j=1,.., J (=999, say), of the ME density as members of the ensemble from the inverse of its 'empirical' cumulative distribution function (CDF). The ergodic theorem (grand mean of all $x_{j,t}$ representing the ensemble average equals the time average of x_t) is guaranteed as is the central limit theorem.

Stationary times series are integrated of order zero, I(0). Many real world applications involve a mixture of I(0) and nonstationary I(d) series, where the order of integration d can be different for different series and even fractional, and where the stationarity assumptions are difficult to verify. The WKK theory mostly needs the zero memory I(0) white noise type processes, where some results are true only for circular processes, implying that we can go back in history, (e.g., undo the SEC, FCC, or go back to horse and buggy, pre 9/11 days, etc.) and is quite unrealistic.

One can bring realism by testing and allowing for finite 'structural changes', often with *ad hoc* tools. It is hard to accept the notion of infinite memory of the random walk I(1) when the very definitions of economic series (e.g., quality and content of the GDP, names of stocks in the Dow Jones average) change over finite time intervals. This is often not a problem in natural sciences. For example, the definition of water or the height of an ocean wave is unchanged over time.

2 Maximum Entropy Bootstrap

This section describes the ME bootstrap procedure and indicates the similarities and differences with the traditional bootstrap.

2.1 The algorithm

The set of steps entailed in the Vinod's ME bootstrap algorithm to create a random realization of x_t is as follows.

1. Sort the original data in increasing order and store the ordering index vector.

2. Compute intermediate points on the sorted series.

3. Compute lower limit for left tail and upper limit for right tail . This is done by extrapolating beyond sample limits using the trimmed mean of deviations among all consecutive observations.

4. Compute the mean of the maximum entropy density within each interval in such a way that the *mean preserving constraint* is satisfied. (Denoted as m_t in the reference paper.) The first and last interval means have different formulas.

5. Generate random numbers from the [0,1] uniform interval and compute sample quantiles at those points.

6. Apply to the sample quantiles the correct order to keep the dependence relationships of the observed data using ordering index of step 1.

7. Repeat steps 2 to 6 several times (e.g. 999).

2.2 A toy example

The procedure described above is illustrated with a small example. Let the sequence $x_t = 4, 12, 36, 20, 8$ be the series of data observed from the period $t = 1$ to $t = 5$ as indicated in the first two columns of Table 1. We start by sorting the observed data in increasing order and store the ordering index vector (Table 1 columns 3 and 4).

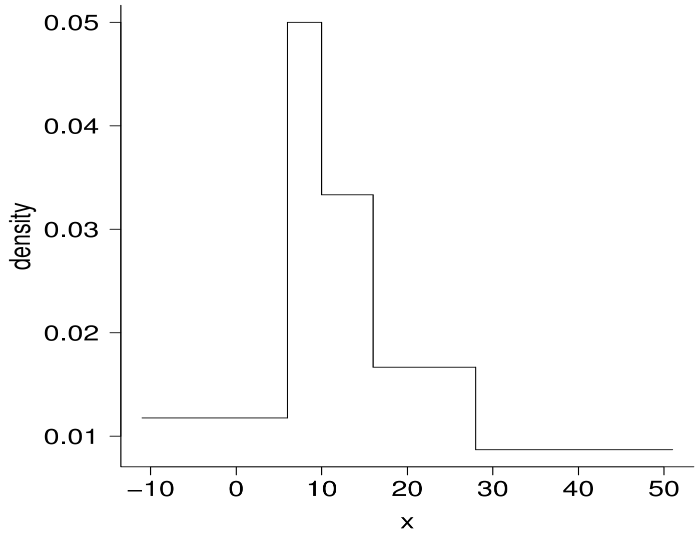
Next, intermediate points between two consecutive ordered observations are obtained (column 5, simple averages), being the limits used to build half open intervals. The maximum entropy density in the ME bootsrap is defined as the combination of T uniform densities where the lower and upper limits are the intermediate points shown in column 5. For the first and last intervals, the lower limit and the upper limit, respectively, (i.e. the smallest and the largest allowable values) are obtained by substracting to the minimum and maximum observed values the trimmed mean of the absolute deviations among all consecutive observations. Hence the tails are uniform distributed as well. For a trimming proportion equal to 10%, these values are -11 and 51, respectively.

Table 1: Example of the ME bootsrap procedure

Time	x_t	Ordering vector	Sorted x_t	Intermediate points	Desired means	Uniform draws	Preliminary values	Final replicate
1	4	1	4	6	5	0.12	5.85	5.85
2	12	5	8	10	8	0.83	6.70	8.90
3	36	2	12	16	13	0.53	8.90	23.95
4	20	4	20	28	22	0.59	10.70	10.70
5	8	3	36		32	0.11	23.95	6.70

In the example, the ME density is the combination of the following uniform densities: $U(-11, 6] \times U(6, 10] \times U(10, 16] \times U(16, 28] \times U(28, 51]$ as shown in Figure 1.

Figure 1: Maximum entropy density for the $x_t = 4, 12, 36, 20, 8$ example



Then, half open intervals are defined between two consecutive ordered data and their intermediate points are obtained (column 5). In order to satisfy the mean preserving constraint of the ME density, the interval means, m_t, are obtained as follows:

$$
\begin{aligned}
m_1 &= 0.75xx_1 + 0.25xx_2\,, \quad \text{for the lowest interval,} \\
m_k &= 0.25xx_{k-1} + 0.50xx_k + 0.25xx_{k+1}\,, \quad \text{for } k = 2, ..., T-1 \\
m_T &= 0.25xx_{T-1} + 0.75xx_T\,,
\end{aligned}
$$

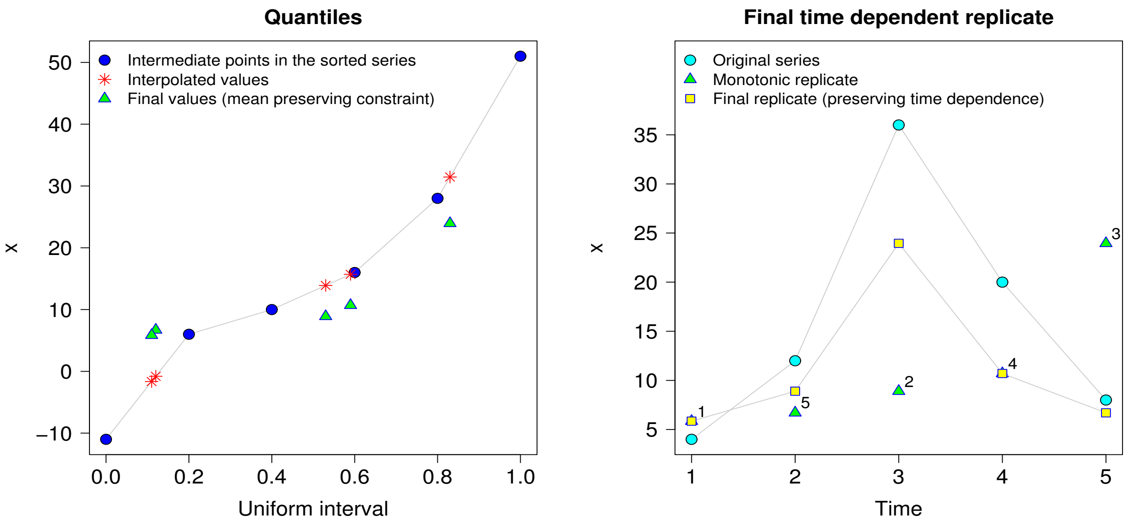
where xx_t stands for the data sorted in increasing order. The desired means for this case are reported in column 6.

Then random numbers from the $[0, 1]$ uniform intervals are drawn and used as reference to compute quantiles of the ME density. (See left side plot in Figure 2.) The ME density quantiles obtained in this way provide a monotonic series. The final replicate is obtained by recovering the original order. (See right side plot in Figure 2.)

2.3 Contrast with traditional iid bootstrap

Vinod (2004, 2006) mentions three properties of traditional iid bootstrap which are worth avoiding for the purpose of constructing ensembles viewed as reincarnations of original time series.

Figure 2: Example of the ME bootstrap procedure



- The traditional bootstrap sample repeats some x_t values while not using as many others. We are considering applications where there is no reason to believe that values near the observed x_t are impossible. For example, let $x_t = 49.2$. Since 49.19 or 49.24, both of which round up to $x_t = 49.2$, there is no justification for excluding at least such values.

- The traditional bootstrap resamples must lie in the closed interval $[\min(x_t), \max(x_t)]$. In most real world series this is an artificial restriction with no justification. Since the observed range is random, we cannot rule out somewhat smaller or larger x_t and wider ranges.

- The traditional bootstrap resample shuffles x_t such that any dependence information in the time series sequence $(x_1, ..., x_t, x_{t+1}, ..., x_T)$ is lost. If we try to restore the original order to the shuffled resample of the traditional bootstrap, we end up with essentially the original set x_t, except that some dropped x_t values are replaced by the repeats of adjacent values. Hence, it is impossible to generate a large number J of sensibly distinct resamples with the traditional bootstrap.

2.4 Shape retention

In addition to strong dependence arguments offering a justification for perfect rank matching, we now show that pseudo utilities associated with x_t and $x_{j,t}$ share comparable ordinal utilities. Economists familiar with the ordinal utility theory know that economists do not like to make interpersonal comparisons of utility, since no two persons can 'feel' exactly the same satisfactions. Yet economists must compare utilities to make policy recommendations.

The problem was solved in classical economics by using the concept of ordinal utility, which says that utilities experienced by individuals are comparable to each other, provided the utility bundles satisfy common partial ordering (what we call perfect rank matching).

Imagine that x_t represents the evolving time path for income of one individual (or collection of individuals in a country) sensitive to initial resources at birth and intellectual endowments. Our aim is to make reincarnations comparable to each other without assuming more than necessary knowledge. Our double sorting in the ME boot algorithm retains just enough of the basic shape of x_t, without pretending to compare underlying 'feelings' across reincarnations. Formally each j obeys the partial ordering of x_t and retains the broad shape of its ups and downs.

3 The `meboot` `R`-package

The package `meboot` implements the maximum bootstrap bootstrap procedure for time series described in Vinod (2004, 2006). The package can be obtained from the Comprehensive R Archive Network at `http://www.cran.r-project.org` and can be easily installed by typing `install.packages("meboot")` or `install.packages("meboot", lib.loc="R-library-no-default-path"` in the `R` console.

Once the package is installed, the functions can be made available in the workspace loading the package by means of `library(meboot)` or `library(meboot, lib.loc="R-library-no-default-path")`.

Reference information about the package (`help(package=meboot)`) or help pages for the functions implemented in it (for instance `help(meboot)`) can also be obtained.

4 Applications

4.1 Consumption function

This example describes how to carry out inference through the ME ensemble in the following regression:

$$c_t = \beta_1 + \beta_2\, c_{t-1} + \beta_3\, y_{t-1} + u_t\,, \tag{1}$$

for the null hypothesis $\beta_3 = 0$.

We use the data set employed in Murray, M.P. (2006, pp. 799-801) to discuss the Keynesian consumption function on the basis of the Friedman's permanent income hypothesis and Robert Hall's model. The data are the logarithm of the US consumption, c_t, and disposable income, y_t, in the period 1948-1998.

```
> library(meboot)
> library(car)
> library(lmtest)
```

```
> data(USconsum)
> attach(USconsum)

> lc <- log(consum)
> ly <- log(dispinc)
> lmcf <- lm(lc[2:51] ~ lc[1:50] + ly[1:50])
> coeftest(lmcf)

t test of coefficients:

            Estimate Std. Error t value  Pr(>|t|)
(Intercept) 0.026932   0.026101  1.0318    0.3074
lc[1:50]    0.969680   0.142612  6.7994 1.647e-08 ***
ly[1:50]    0.026953   0.143889  0.1873    0.8522
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> durbin.watson(model = lmcf, max.lag = 4)

 lag Autocorrelation D-W Statistic p-value
   1      0.14597875      1.690194   0.180
   2     -0.03521029      2.017553   0.894
   3     -0.08825760      2.082973   0.768
   4     -0.08849678      2.078104   0.604
 Alternative hypothesis: rho[lag] != 0
```

The estimated coefficient of lagged income, $\beta_3 = 0.027$, is statistically insignificant. The 95% confidence interval is $(-0.263, 0.316)$, with zero inside this interval. The residuals are serially uncorrelated since the p-values of the generalized Durbin-Watson statistics up to order 4 are larger than the significance level 0.05.

This result is traditionally interpreted as supporting the Friedman permanent income hypothesis. However, with the advent of unit root literature we know that the sampling distribution of β_3 is nonstandard and that traditional inference based on the Student's t or Normal distributions may lead to spurious results.

The literature suggests differencing or de-trending C_t and Y_t and/or using a non-standard density to assess statistical significance. Instead of unit root tests on C_t and Y_t to decide whether differencing is appropriate, we use the ME bootstrap to create J=999 replicates of C_t and Y_t. The ensemble can be used to construct confidence intervals and decide whether the estimated coefficient is statistically significantly different from zero.

By means of the `meboot` function it is straightforward to create a set of replicates that approximate the population for the logarithm of the disposable income. We compute `bigJ`=999 replicates and make sure that the ensemble elements satisfy the desirable statistical properties of central limit theorem and ergodic theorem.[1]

```
> semx <- meboot(x = ly, reps = 999, reachbnd = TRUE, expand.sd = TRUE,
+     force.clt = TRUE)$ensemble
```

[1] For example, the grand mean of the 999 realizations equals the sample mean.

Creation of a similar population for the dependent variable C_t needs further care since we must abide by the expression (1) under the null hypothesis $\beta_3 = 0$. For it we generate a first order autoregressive model where the coefficients are obtained from the regression restricted to the null hypothesis $\beta_3 = 0$, `lmrcf`.

Since the C_t is subject to error, u_t, we need to create a set of innovations. We do this by creating an ensemble for the residuals of the restricted regression, `lmrcf`. As one observation is lost due to the presence of a lagged variable, we take the first value for consumption, observation in year 1948, as the first observation.

```
> lmrcf <- lm(lc[2:51] ~ lc[1:50])
> bb <- coef(lmrcf)
> resi <- resid(lmrcf)
> bigJ <- 999
> et <- meboot(x = resi, reps = 1)$ensemble
> AC <- arima.sim(n = length(lc), model = list(order = c(1, 0,
+     0), ar = bb[2]), n.start = 1, innov = c(lc[1], et + bb[1]),
+     start.innov = 0)
> semy <- meboot(x = AC, reps = bigJ, reachbnd = TRUE, expand.cd = TRUE,
+     force.clt = TRUE)$ensemble
```

Thus we have all the needed data to fit `bigJ` additional regressions as in (1). By repeating this $k = 1, ...,$`bigK`$=100$ times we have 99900 regression coefficient estimates of β_3, the focus of our inference. Next, confidence limits for β_3 can be obtained, rejecting the null hypothesis if zero lies outside these confidence interval limits.

Now run all commands in the appendix (allow enough time). In each iteration from `k`$=1$ to `bigK`, two-sided confidence interval around zero based on the 1000 coefficients, `all.artif.b`, are stored in `all.LO` and `all.UP` respectively for the lower and upper limit for the purpose of averaging.

```
> zci <- zero.ci(all.artif.b)
> all.LO[k] <- zci$lolim
> all.UP[k] <- zci$uplim
> avLO <- mean(all.LO)
> avUP <- mean(all.UP)
```

The averaging lower and upper limits are `avLO` $= -0.083$ and `avUP` $= 0.333$.

Size corrected lower and upper limits (confidence interval around zero as the true value as computed by the function `zero.ci`) can be obtained for the grand set of slopes (`allK.artif.b` in the appendix).

```
> cc <- rbind(as.matrix(all.LO), as.matrix(all.UP))
> zcc <- zero.ci(cc)
> scLO <- zcc$lolim
> scUP <- zcc$uplim
```

The refined 95% ME bootstrap confidence interval (`scLO`, `scUP`) for the statistic of interest estimated as $\hat{\beta}_3 = 0.027$ is $[-0.110, 0.396]$. Since zero lies within this null interval, the results provide statistical evidence in favour of the Friedman's hypothesis.

4.2 Assessment of the Fed effect on stock prices using panel data

This example shows how the ME bootstrap can be employed for panel data analysis. Our example is from Vinod (2002) where the effect of monetary policy (interest rates) on prices and their *turning points* in the stock market is evaluated.

The *Fed effect* discussed in the financial press refers to a rally in the S&P 500 prices few days before the Fed meeting and a price decline after the meeting. This example focuses on the longer term than daily price fluctuations by using the monthly data (May 1993 to November 1998) for stocks with ticker symbols: ABT, AEG, ATI, ALD, ALL, AOL, and AXP and regard this as a representative sample. Thus the data set consists of $67 \times 7 = 469$ observations.

First, we regress the stock price (P) on the natural log of market capitalization (LMV), as a control variable for the size of the firm and the interest on 3-month Treasury bills (TB3). Note that TB3 is the key interest rate influenced by monetary policy of the Federal Reserve Bank (Fed). The model is:

$$P_{it} = \beta_0 + \beta_1 \, LMV_{it} + \beta_3 \, TB3_{it} + \varepsilon_{it} \,, \tag{2}$$

where the subscript it refers to i-th individual at time t. The Fed effect is present, if the variable TB3 in equation (2) is statistically significant.

4.2.1 Pooled effects

As reported below, the t-value for TB3 in the pooled model is highly significant and the corresponding p-value suggests that the Fed does have a statistically significant effect on the level of stock prices in a pooled regression. The multiple regression R^2 is 0.4972, if adjusted for the degrees of freedom, it becomes 0.495 based on T=468 observations.

```
> library(meboot)
> library(plm)
> data(ullwan)
> attach(ullwan)

> LMV <- log(MktVal)
> summary(lm(Price ~ LMV + Tb3))

Call:
lm(formula = Price ~ LMV + Tb3)

Residuals:
    Min      1Q  Median      3Q     Max
-35.415 -11.474  -2.923   4.822  70.909

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) -125.5963    11.2089 -11.205  < 2e-16 ***
LMV           18.8479     0.8864  21.264  < 2e-16 ***
Tb3           -4.8046     1.4795  -3.247  0.00125 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.95 on 466 degrees of freedom
Multiple R-Squared: 0.4972,       Adjusted R-squared: 0.495
F-statistic: 230.4 on 2 and 466 DF,  p-value: < 2.2e-16
```

We use the `R` package `plm` (Croissant, 2005) and t-value for the coefficient of TB3 from the pooled model suggests that the Fed does have a statistically significant effect on the level of stock prices in a pooled regression. The high value of the F-statistic in the following table suggests that pooling may not be appropriate.

```
> gi <- plm(Price ~ LMV + Tb3, data = ullwan)
> gicoef <- gi$pooling[[1]]
> n1 <- length(gicoef)
> summary(gi)

----------------------------------------------------------------------
_________________________ Model Description __________________________
Oneway (individual) effect

Model Formula        : Price ~ LMV + Tb3
----------------------------------------------------------------------
__________________________ Panel Dimensions __________________________
Balanced Panel
Number of Individuals        :  7
Number of Time Obserbations  :  67
Total Number of Observations :  469
----------------------------------------------------------------------
____________________________ Coefficients ____________________________
                within        wse     random      rse
(intercept)          .          . -184.92025 10.4036
LMV           25.14412    0.78547   25.00087  0.7842
Tb3           -4.96342    1.01881   -4.95981  1.0217
----------------------------------------------------------------------
_______________________________ Tests ________________________________
Hausman Test                 : chi2(2) = 10.09357 (p.value=0.00642998)
F Test                       : F(6,460) = 87.14226 (p.value=0)
Lagrange Multiplier Test     : chi2(1) = 3253.871 (p.value=0)
----------------------------------------------------------------------
```

We need to slightly modify the ME bootstrap algorithm for panel data to carefully create J replicates over time, separately for N individuals (N=7 stock symbols here). The extended ME boot algorithm is used to create 999 ensembles for the 67 time series points for the 7 stocks separately. Collecting them together, we have

1000 sets of $67 \times 7 = 469$ data points upon including the original data as the first column and 999 additional columns. Below we use `panel.boot` to create ensembles of the panel data set for the logarithm of the market capitalization, stock prices and interest on 3-month Treasury bills.

```
> jboot <- 999
> ullwan[, 3] <- log(ullwan[, 3])
> set.seed(567)
> LMV.ens <- panel.meboot(x = ullwan, reps = jboot, colsubj = 1,
+     coldata = 3, expand.sd = TRUE, force.clt = TRUE)
> Price.ens <- panel.meboot(x = ullwan, reps = jboot, colsubj = 1,
+     coldata = 4, expand.sd = TRUE, force.clt = TRUE)
> Tb3.ens <- panel.meboot(x = ullwan, reps = jboot, colsubj = 1,
+     coldata = 6, expand.sd = TRUE, force.clt = TRUE)
```

The purpose of the ME boot here is to assess if we continue to have significant Fed effect for pooled and other models described below. Results based on ME bootstrap can be computed as follows.

```
> slopeTb3 <- slopeLMV <- rep(0, jboot)
> for (j in 1:jboot) {
+     frm <- data.frame(Subj = ullwan[, 1], Time = ullwan[, 2],
+         Price = Price.ens[, j], Tb32 = Tb3.ens[, j],
+         LMV2 = LMV.ens[,j])
+     frm <- pdata.frame(frm, 7)
+     gip <- plm(Price ~ LMV + Tb3, data = frm)$pooling[[1]]
+     slopeTb3[j] <- as.numeric(gip[3])
+     slopeLMV[j] <- as.numeric(gip[2])
+ }
```

The 95% ME boot percentile confidence interval for TB3 obtained by:

```
> n25 <- ceiling(0.025 * jboot)
> n975 <- floor(0.975 * jboot)
> sortb <- sort(as.numeric(slopeTb3[1:jboot]))
> c(sortb[n25], sortb[n975])
```

is (-5.597, -3.381). A refined asymmetric interval obtained by calling the function `zero.ci` provided in the `meboot` package as follows: `zero.ci(as.numeric(slopeTb3))`, is (-5.746 -3.931). Both intervals do not contain zero implying that the effect of TB3 on stock prices is statistically significantly negative (different from zero) in a pooled model.

In the same way, the percentile intervals for the regressor LMV is (17.622, 20.845), with the refined one as (18.482, 21.171) suggesting a significantly positive regressor.

4.2.2 Random effects

The random effects model results are obtained below. To save space and focus on the results based on ME replicates we do not display here the results from the traditional procedure.

```
> summgi = summary(gi)
> sr <- summgi$Coef[, 3:4]
> gicoef <- sr[, 1]
> n2 <- length(gicoef)
> se <- sr[, 2]
> print(gicoef)
```

Results based on the ME bootstrap are computed as follows:

```
> slopeTb3 <- slopeLMV <- rep(0, jboot)
> for (j in 1:jboot) {
+     frm <- data.frame(Subj = ullwan[, 1], Tim = ullwan[, 2],
+         Price = Price.ens[, j], Tb32 = Tb3.ens[, j],
+         LMV2 = LMV.ens[,j])
+     frm <- pdata.frame(frm, 7)
+     gip <- plm(Price ~ LMV + Tb3, data = frm)$random[[1]]
+     slopeTb3[j] <- as.numeric(gip[3])
+     slopeLMV[j] <- as.numeric(gip[2])
+ }
```

The random effects 95% ME boot confidence interval using the 999 replicates of data yields (22.614, 28.779) as the refined interval and (23.864, 29.542) as the percentile interval for LMV. More important, it yields (-5.782, -3.497) as the refined interval and (-5.951, -4.083) as the percentile interval for TB3. Since the latter intervals do not cover zero, we can conclude that Fed effect is significant for the random effects panel data model.

5 Concluding remarks

We illustrated the performance and usage of the Vinod's maximum entropy bootstrap for dependent data by means of several examples of relevance for time series inference.

A Appendix

A.1 Consumption function

```
> olsHALL.b <- function(x, y) {
+     n <- length(x)
+     x <- cbind(y[-n], x[-n])
```

```
+     y <- y[-1]
+     tol <- 1e-07
+     p <- ncol(x)
+     ny <- NCOL(y)
+     lmh <- lm(y ~ x)
+     coef(lmh)[3]
+ }
> bigJ <- 999
> bigK <- 100
> allK.artif.b <- rep(0, (bigJ * bigK +
+     1))
> all.LO <- rep(0, bigK)
> all.UP <- all.LO
> set.seed(321)
> for (k in 1:bigK) {
+     et <- meboot(x = resi, reps = 1)$ensemble
+     AC <- arima.sim(n = length(lc), model = list(order = c(1,
+         0, 0), ar = bb[2]), n.start = 1,
+         innov = c(lc[1], et + bb[1]),
+         start.innov = 0)
+     semy <- meboot(x = AC, reps = bigJ,
+         reachbnd = TRUE, expand.sd = TRUE,
+         force.clt = TRUE)$ensemble
+     semy <- cbind(AC, semy)
+     semx <- meboot(x = ly, reps = bigJ,
+         reachbnd = TRUE, expand.sd = TRUE,
+         force.clt = TRUE)$ensemble
+     semx <- cbind(ly, semx)
+     all.artif.b <- rep(NA, ncol(semy))
+     for (h in seq(along = all.artif.b))
+        all.artif.b[h] <- olsHALL.b(x = semx[,h], y = semy[, h])
+     if (k == 1)
+         allK.artif.b[1] <- all.artif.b[1]
+     k1 <- (k - 1) * bigJ + 2
+     k2 <- k * bigJ + 1
+     allK.artif.b[k1:k2] <- all.artif.b[2:(bigJ + 1)]
+     zci <- zero.ci(all.artif.b)
+     all.LO[k] <- zci$lolim
+     all.UP[k] <- zci$uplim
+     if ((k * 100/bigK)%%5 == 0)
+         cat(paste((k * 100/bigK), "%",
+             sep = ""), "complete.\n")
+ }
```

References

Croissant, Y. (2005), *plm: Linear models for panel data.* R package version 0.1-2.
URL: *http://www.r-project.org*

Murray, M.P. (2006), *Econometrics. A modern introduction*, Pearson Addison Wesley, New York.

R Development Core Team (2006), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
URL: *http://www.R-project.org*

Vinod, H. (2002), *Econometric Applications of Generalized Estimating Equations for Panel Data and Extensions to Inference*, in Handbook of Applied Econometrics, Marcel Dekker, New York, chapter 26, pp. 553–574. Editors: A.T. Aman Ullah. and K. Wan and A. Chaturvedi.

Vinod, H. D. (2004), 'Ranking mutual funds using unconventional utility theory and stochastic dominance', *Journal of Empirical Finance* **11**(3), 353–377.

Vinod, H. D. (2006), 'Maximum entropy ensembles for time series inference in economics', *Journal of Asian Economics* **17**(6), 955–978.