Discussion Papers in Economics

No. 05/2017

partialCI: An R package for the analysis of partially cointegrated time series

Matthew Clegg
Independent

Christopher Krauss
University of Erlangen-Nürnberg

Jonas Rende
University of Erlangen-Nürnberg

ISSN 1867-6707

Friedrich-Alexander-Universität Erlangen-Nürnberg
Institute for Economics
https://www.iwf.rw.fau.de/research/iwf-discussion-paper-series/

partialCI: An R package for the analysis of partially cointegrated time series

Matthew Clegg[a,1], Christopher Krauss[b,1], Jonas Rende[c,1]

[a] *Independent*

[b] *University of Erlangen-Nürnberg, Department of Statistics and Econometrics, Lange Gasse 20, 90403 Nürnberg, Germany*

[c] *University of Erlangen-Nürnberg, Department of Statistics and Econometrics, Lange Gasse 20, 90403 Nürnberg, Germany*

Friday 10th February, 2017

Abstract

Partial cointegration is a weakening of cointegration, allowing for the residual series to contain a mean-reverting and a random walk component. Analytically, the residual series is described by a partially autoregressive process. The **partialCI** package provides estimation, testing, and simulation routines for PCI models in state space. We illustrate the functionality with two examples: A financial application in the context of pairs trading and a macroeconomic application, i.e., the relationship between GDP and consumption. For both examples, we show that the variables are not cointegated in the classic sense, but can be modeled with partial cointegration.

Keywords: R software, cointegration, partial cointegration, pairs trading, permanent components, transient components.

Email addresses: `matthewcleggphd@gmail.com` (Matthew Clegg), `christopher.krauss@fau.de` (Christopher Krauss), `jonas.rende@fau.de` (Jonas Rende)

[1] The authors have benefited from many helpful discussions with Ingo Klein.

1. Introduction

The **partialCI** package (Clegg, 2016) fits a partial cointegration model[2] to describe a time series. Partial cointegration (PCI) is a weakening of cointegration, allowing for the residual series to contain a mean-reverting and a random walk component. Analytically, this residual series is described by a partially autoregressive process (PAR – see Summers (1986), Poterba and Summers (1988), and Clegg (2015a))[3], consisting of a stationary AR-process and a random walk. Related is the short-term / long-term model introduced by Schwartz and Smith (2000), which models a security price as the sum of a Brownian motion and an Ornstein-Uhlenbeck process. Whereas classic cointegration in the sense of Engle and Granger (1987) requires all shocks to be transient, PCI is more flexible and allows for permanent shocks as well – a realistic assumption across many (macro)economic applications. Even though neither the residual series, nor its mean-reverting and permanent component are directly observable, estimation is still possible in state space – see Brockwell and Davis (2010) and Durbin and Koopman (2012). The **partialCI** package encloses suitable estimation, testing, and simulation routines for such PCI models.

Partial cointegration enhances several existing cointegration concepts in the literature – namely classic cointegration, fractional cointegration and threshold cointegration.

In their seminal paper, Engle and Granger (1987) introduce the concept of classic cointegration. Loosely speaking, if a collection of time series is cointegrated, they share a long-run equilibrium. Shocks to the cointegration process are not persistent, i.e., the process adjusts exponentially towards the long-run equilibrium value after exhibiting a shock (Pfaff, 2008). Thus, if the cointegration process is subject to permanent shocks, the partial cointegration model may be more appropriate. Test procedures for classic cointegration are implemented in the R packages **urca** (Pfaff, 2008) and **egcm** (Clegg, 2015c).

[2] Please note that we use the term partial cointegration according to Clegg and Krauss (2016).

[3] Partially autoregressive processes are already implemented in the corresponding R package **partialAR** (Clegg, 2015b).

In a fractional cointegration model the residual series is assumed to follow a fractionally integrated process. Such a process incorporates weighted higher-order lags to model long-term effects (Baillie, 1996). In terms of shock persistence, fractionally integrated processes are between classic cointegrating processes (short-run persistence) and random walks (infinite persistence). The ability to account for long-term persistence makes fractionally integrated processes especially useful to analyze long-memory time series data (Baillie, 1996). The benefit of PCI compared to fractional cointegration is twofold: First, with PCI it is possible to disentangle the transient and permanent component, allowing to separately investigate the dynamics associated with the transient component (Clegg and Krauss, 2016). Second, within a PCI framework the proportion of variance attributable to mean-reversion (PVMR) can be computed (Clegg and Krauss, 2016). The PVMR allows to assess the degree of noise in the time series.

In their seminal paper, Balke and Fomby (1997) introduce the concept of threshold cointegration. In the cointegration models introduced so far, every shock, independent of its magnitude, induces an instant adjustment process towards the long-run equilibrium value. Balke and Fomby (1997) flexiblize this assumption of linear adjustment. The process is assumed to solely consist of a permanent component, if it does not exceed a certain threshold level. By contrast, if the time series exceeds the threshold level the process is modeled as a classic cointegration process and adjustment towards the corresponding long-run equilibrium occurs as long as the process exceeds the threshold value in absolute terms. The advantage of the partial cointegration model is the ability to model the impact of permanent shocks *globally* and not just *locally* as in a thresold cointegration model. Threshold cointegration models are implemented in the R package **tsDyn** (Stigler, 2010).

Potential fields of application for the PCI model in a financial context are: term structures, stock indices and tracking portfolios, stock pairs, spot and future prices, commodities, spread options, international stock indices, as well as foreign exchange (Alexander, 2011). In addition, the PCI framework could be used to revisit macroeconomic theories, e.g., monetary policy, fiscal policy or business cycle models. An initial show case for PCI can be found in

Clegg and Krauss (2016). They apply the **partialCI** package to detect partially cointegrated pairs of stocks on the S&P 500 from January 1990 to October 2015. The authors extract the mean-reverting component of the price spread time series of the partially cointegrated pairs of stocks as baseline for a relative-value arbitrage strategy.

The remainder of this paper is organized as follows. In section 2, we outline the methodological details of the PCI model. In section 3, we explain how to use the key functions of the **partialCI** package. In section 4, we provide a finance as well as a macroeconomic example. Finally, section 5 provides concluding thoughts.

2. The partial cointegration framework

2.1. Model definition

Based on Engle and Granger (1987), Clegg and Krauss (2016, p. 4) define the concept of partial cointegration as follows:

Definition: "The components of the vector X_t are said to be partially cointegrated of order d, b, denoted $X_t \sim PCI(d, b)$, if (i) all components of X_t are $I(d)$[4]; (ii) there exists a vector α so that $Z_t = \alpha\prime X_t$ and Z_t can be decomposed as a sum $Z_t = R_t + M_t$, where $R_t \sim I(d)$ and $M_t \sim I(d-b)$."

While Clegg and Krauss (2016) focus on the special case of two partially cointegrated time series, we extend the model to the case of $(k+1)$ partially cointegrated time series. Let Y_t denote the target time series and $X_{j,t}$ the j^{th} factor time series at time t, where $j = \{1, 2, \ldots, k\}$. The target time series and the k factor time series are partially cointegrated, if a parameter vector $\iota = \{\beta_1, \beta_2, \ldots, \beta_k, \rho, \sigma_M, \sigma_R, M_0, M_R\}$ exists such that the subsequent model

[4] If a time series exhibits d unit roots, it is said to be integrated of order d ($I(d)$) (Lütkepohl, 2007, p. 238-242).

equations are satisfied (Clegg and Krauss, 2016)[5]:

$$\begin{aligned}
Y_t &= \beta_1 X_{1,t} + \beta_2 X_{2,t} + \ldots + \beta_k X_{k,t} + W_t \\
W_t &= M_t + R_t \\
M_t &= \rho M_{t-1} + \varepsilon_{M,t} \\
R_t &= R_{t-1} + \varepsilon_{R,t} \\
\varepsilon_{M,t} &\sim \mathcal{N}\left(0, \sigma_M^2\right) \\
\varepsilon_{R,t} &\sim \mathcal{N}\left(0, \sigma_R^2\right) \\
\beta_j \in \mathbb{R}; \rho &\in (-1,1)\,; \sigma_M^2, \sigma_R^2 \in \mathbb{R}_0^+.
\end{aligned} \tag{1}$$

Thereby, W_t denotes the partially autoregressive process, R_t the permanent component, M_t the transient component and $\beta = \{\beta_1, \beta_2, \ldots, \beta_k\}$ is the partially cointegrating vector.[6] The permanent component is modeled as a random walk and the transient component as an AR(1)-process with AR(1)-coefficient ρ. The corresponding error terms $\varepsilon_{M,t}$ and $\varepsilon_{R,t}$ are assumed to follow mutually independent, normally distributed white noise processes with mean zero and variances σ_M^2 and σ_R^2. For the sake of simplicity, we set $M_0 = 0$ and $R_0 = Y_0 - \beta_1 X_{1,0} - \beta_2 X_{2,0} - \ldots - \beta_k X_{k,0}$. A key advantage of modeling the cointegrating process as a partially autoregressive process is that we are able to calculate the PVMR, defined as (Clegg and Krauss, 2016),

$$R_{MR}^2 = \frac{VAR\left[(1-B)\, M_t\right]}{VAR\left[(1-B)\, W_t\right]} = \frac{2\sigma_M^2}{2\sigma_M^2 + (1+\rho)\,\sigma_R^2}, \quad R_{MR}^2 \in [0,1]\,, \tag{2}$$

where B denotes the backshift operator. The statistic R_{MR}^2 is useful to assess how close the cointegration process is to either a pure random walk ($R_{MR}^2 = 0$) or a pure AR(1)-process ($R_{MR}^2 = 1$).

2.2. State space representation

The applied state space transformation is in line with Clegg and Krauss (2016). Given that the PAR process W_t is not observable, we convert the PCI model into the following

[5]It is possible to include an intercept within the **partialCI** package.

[6]Note that in the implemented estimation routine the estimated partially cointegrating vector is a linear combination of all existing partially cointegrating vectors in the sense of Verbeek (2010, p. 324).

state space model, consisting of an observation (3) and a state equation (4):

$$\boldsymbol{X_t} = \boldsymbol{H}\boldsymbol{Z_t} \tag{3}$$

$$\boldsymbol{Z_t} = \boldsymbol{F}\boldsymbol{Z_{t-1}} + \boldsymbol{W_t}. \tag{4}$$

Thereby, $\boldsymbol{Z_t}$ (4) denotes the state which is assumed to be influenced linearly by the state in the last period and a noise term $\boldsymbol{W_t}$. The matrix $\boldsymbol{F}$ is assumed to be time invariant. The observable part is denoted by $\boldsymbol{X_t}$ (3). By assumption, there is a linear dependence between $\boldsymbol{X_t}$ and $\boldsymbol{Z_t}$, captured in the time invariant matrix $\boldsymbol{H}$.

The PCI framework presented in equation (1) consists of the observable target as well as factor time series and the two hidden state variables M_t and R_t. Following the approach of Clegg and Krauss (2016), the k factor variables are declared as additional hidden state variables. As a consequence $X_{1,t}, X_{2,t}, ..., X_{k,t}$ are part of both, the observation and the state equation. Applying the state space transformation yields the following observation equation:

$$\boldsymbol{X_t} = \begin{bmatrix} Y_t \\ X_{1,t} \\ X_{2,t} \\ \vdots \\ X_{k,t} \end{bmatrix} = \boldsymbol{H}\boldsymbol{Z_t} = \begin{bmatrix} \beta_1 & \beta_1 & \cdots & \beta_k & 1 & 1 \\ 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_{1,t} \\ X_{2,t} \\ \vdots \\ X_{k,t} \\ M_t \\ R_t \end{bmatrix}. \tag{5}$$

The state equation is given as follows:

$$\boldsymbol{Z_t} = \begin{bmatrix} X_{1,t} \\ X_{2,t} \\ \vdots \\ X_{k,t} \\ M_t \\ R_t \end{bmatrix} = \boldsymbol{F}\boldsymbol{Z_{t-1}} + \boldsymbol{W_t} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & \rho & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \\ \vdots \\ X_{k,t-1} \\ M_{t-1} \\ R_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{X_{1,t}} \\ \varepsilon_{X_{2,t}} \\ \vdots \\ \varepsilon_{X_{k,t}} \\ \varepsilon_{M,t} \\ \varepsilon_{R,t} \end{bmatrix}, \tag{6}$$

with $\varepsilon_{X_{j,t}}$ denoting the innovation of process $X_{j,t}$. By assumption, $\varepsilon_{X_{j,t}}$ is normally distributed with zero mean and variance $\sigma^2_{x_j} \in \mathbb{R}^+_0$ and is independent of $\varepsilon_{M,t}$ and $\varepsilon_{R,t}$.

2.3. Estimation of a partial cointegration model

Parameters are estimated via the maximum likelihood (ML) method. Using a quasi-Newton algorithm, the ML method searches for the parameters ρ, σ_M^2, σ_R^2 and the parameter vector β which maximizes the likelihood function of the associated Kalman filter.[7] The following likelihood score is maximized (Clegg and Krauss, 2016):

$$\mathcal{L}_{MR}\left(\beta, \rho, \sigma_M^2, \sigma_R^2\right) = \prod_{k=2}^{n} \phi\left(\varepsilon_{M,k} + \varepsilon_{R,k}; 0, \sigma_M^2 + \sigma_R^2\right), \tag{7}$$

where $\phi(\cdot)$ denotes the probability density function of the normal distribution. Clegg and Krauss (2016) provide (i) a derivation of the likelihood function (7), (ii) a proof that the partial cointegration model is identifiable, and (iii) a comprehensive discussion about the consistency of the ML estimation routine.[8]

2.4. A likelihood ratio test routine for partial cointegration

The likelihood ratio test (LRT) implemented in the **partialCI** package adopts the LRT routine for PAR models proposed by Clegg (2015a). In a PCI scenario the null hypothesis consists of two conditions – namely the hypothesis that the residual series is a pure random walk ($\mathcal{H}_0^R$) or a pure AR(1)-process ($\mathcal{H}_0^M$). The two conditions are separately tested. Only if both, $\mathcal{H}_0^R$ and $\mathcal{H}_0^M$ are individually rejected, the null hypothesis of no partial cointegration is rejected. On the first stage the LRT for partial cointegration tests the null hypothesis of a pure random walk versus the alternative hypothesis of a pure AR(1)-process or PCI $\left(\mathcal{H}_1^{AR(1),PCI}\right)$. To construct the first stage of the LRT for partial cointegration it is necessary to estimate the likelihood scores of an unrestricted and a restricted model. The likelihood score of the unrestricted model, i.e., the largest likelihood score found by the Kalman filter optimization routine, is denoted by

$$\mathcal{L}_{MR}^{*} = \max_{\beta,\rho,\sigma_M^2,\sigma_R^2} \mathcal{L}_{MR}\left(\beta, \rho, \sigma_M^2, \sigma_R^2\right) \tag{8}$$

[7] The complete algorithm as well as the determination of the starting values are available in the R package **partialCI**.

[8] The **partialCI** package also provides a two-step estimation method, which often produces results that are inferior to the joint-penalty method, and so the joint-penalty method is to be preferred.

The restricted model is obtained by setting ρ and σ_M to zero which is in line with the null hypothesis of a pure random walk. The restricted model is given by

$$\mathcal{L}^*_{RW} = \underset{\beta,\sigma^2_R}{max} \mathcal{L}_{MR}\left(\beta, \rho = 0, \sigma^2_M = 0, \sigma^2_R\right). \tag{9}$$

The test statistic for the pure radom walk hypothesis is given as

$$\Lambda_R = \log\left(\frac{\mathcal{L}^*_{RW}}{\mathcal{L}^*_{MR}}\right). \tag{10}$$

Let $C_R(\alpha)\,(C_M(\alpha))$ denote the critical value associated with $\Lambda_R\,(\Lambda_M)$ dependent on the significance level α. If $\mathcal{H}^R_0$ cannot be rejected, i.e., $\Lambda_R < C_R(\alpha)$, the tested time series is classified as a pure random walk. On the other hand, if the test rejects $\mathcal{H}^R_0$, the routine continues, testing the conditional null hypothesis $\mathcal{H}^M_0|\Lambda_R < C_R(\alpha)$ against $\mathcal{H}^{PCI}_1$. Setting $\sigma^2_R = 0$ yields the likelihood score of the restriced model:

$$\mathcal{L}^*_M = \underset{\beta,\rho,\sigma^2_M}{max} \mathcal{L}_{MR}\left(\beta, \rho, \sigma^2_M, \sigma^2_R = 0\right). \tag{11}$$

The test statistic for the second stage is given as,

$$\Lambda_M = \log\left(\frac{\mathcal{L}^*_M}{\mathcal{L}^*_{MR}}\right). \tag{12}$$

If the conditional null hypothesis $\mathcal{H}^M_0|\Lambda_R < C_R(\alpha)$ cannot be rejected, i.e., $\Lambda_M < C_M(\alpha)$, the tested time series follows a pure AR(1)-process. Vice versa, if $\Lambda_M > C_M(\alpha)$ holds, the time series is classified as partially cointegrated. Note that the critical values for both test statistics Λ_R as well as Λ_M need to be simulated because the test statistics do not follow a standard distribution. They are embedded in the package **partialCI**.

3. Using the PCI package

In this section, we outline the four key functions of the **partialCI** package in detail – namely fit.pci(), test.pci(), statehistory.pci(), and hedge.pci().

3.1. fit.pci()

The function **fit.pci()** fits a partial cointegration model to a given collection of time series.

```
fit.pci(Y, X, pci_opt_method = c("jp", "twostep"), par_model = c("par",
"ar1", "rw"), lambda = 0, robust = FALSE, nu = 5, include_alpha=FALSE)
```

- Y: Denotes the target time series and X is a matrix containing the k factors used to model Y.[9]

- `pci_opt_method`: Specifies, whether the joint-penalty method (`"jp"`) or the twostep (`"twostep"`) method is applied to obtain the model with the best fit. If `pci_opt_method` is specified as `"twostep"`, a two-step procedure similar to the method introduced by Engle and Granger (1987) is performed. The residuals of the first stage regression are extracted and a prespecified model is fitted to the residual series. Which model is fitted to the residual series, depends on the specification for the argument `par_model`. In case of `"par"`, a partial autoregressive model is used, in case of `"ar1"`, an AR(1)-process and in case of `"rw"` a random walk (default: `par_model = "par"`). On the other hand, if the `pci_opt_method` is specified as `"jp"`, the joint-penalty method is applied, to estimate β, ρ, σ_M^2 and σ_R^2 jointly via ML. The likelihood score of the associated Kalman filter is extended by a penalty value $\lambda\sigma_R^2$, where $\lambda \in \mathbb{R}_0^+$. Larger values for λ favor solutions with a larger transient component and vice versa (default: `lambda = 0`). To reach a higher chance of finding the global minimum, the procedure uses several different starting points. One of these starting points are the parameter estimates of an ex-ante two-step procedure, ensuring that the likelihood score obtained under `"jp"` is at least as good as under `"twostep"` (default: `pci_opt_method = "jp"`).

- `robust`: Determines whether the residuals are assumed to be normally (`FALSE`) or t-distributed (`TRUE`) (default: `robust = TRUE`). If `robust` is set to `TRUE` the degrees of freedom can be specified, using the argument `nu` (default: `nu = 5`). If `pci_opt_method` matches `"twostep"`, a robust linear model (rlm()) included in the R package **MASS** (Ripley and Venables, 2002) is applied, i.e., a Huber (1981) M-estimator is calculated.[10]

- `include_alpha`: If `TRUE`, an intercept α is added to the PCI relationship (default:

[9] Both, X and Y are plain or **zoo** (Grothendieck and Zeileis, 2005) objects. If $k = 1$, X is a vector.

[10] For a discussion about robust parameter estimation in a PAR context, see Clegg (2015a).

include_alpha = FALSE).

- key return values: The proportion of variance attributable to mean-reversion ($pvmr), the partially cointegrating vector ($beta), the AR(1)-coefficient ($rho) and the negative log likelihood ($negloglik).

3.2. test.pci()

The **test.pci()** function tests the goodness of fit of a PCI model.

```
test.pci(Y, X, alpha = 0.05, null_hyp = c("rw", "ar1"), robust = FALSE,
           pci_opt_method = c("jp", "twostep"))
```

- alpha: Determines at which significance level the null hypothesis is rejected (default: alpha = 0.05).

- null_hyp: Specifies whether the null hypothesis is a random walk ("rw"), an AR(1)-process ("ar1") or a union of both hypotheses (c("rw", "ar1")) (default: null_hyp = c("rw", "ar1")).

- key return values: The test statistic ($statistic) and p-values ($p.value) for the selected null hypothesis.

3.3. statehistory.pci()

To estimate the sequence of hidden states the **statehistory.pci()** function can be applied.

```
statehistory.pci(A, data = A$data, basis = A$basis)
```

- A: Denotes a fit.pci() object.

- data: Is a matrix consisting of the target time series and the k factor time series (default: data = A$data).

- basis: Captures the coefficients of the factor time series (default: basis = A$basis).

- key return values: The two estimated hidden states M_t ($M) and R_t ($R).

3.4. hedge.pci()

The function hedge.pci() finds those k factors from a predefined set of factors which yield the best fit to the target time series.

```
hedge.pci(Y, X, maxfact = 10, lambda = 0, use.multicore = TRUE,
minimum.stepsize = 0, verbose = TRUE, exclude.cols = c(), search_type =
c("lasso", "full", "limited"), pci_opt_method=c("jp", "twostep"))
```

- `maxfact`: Denotes the maximum number of considered factors (default: `maxfact = 10`).
- `use.multicore`: If `TRUE`, parallel processing is activated (default: `use.multicore = TRUE`).
- `verbose`: Controls whether detailed information are printed (default: `verbose = TRUE`).
- `exclude.cols`: Defines a set of factors which should be excluded from the search routine (default: `exclude.cols = c()`).
- `search_type`: Determines the search algorithm applied to find the model that fits best to the target time series. The likelihood ratio score (LRT score) is used to compare the model fits, whereby lower scores are associated with better fits. If the option `"lasso"` is specified the lasso algorithm as implemented in the R package **glmnet** (Friedman et al., 2010) is deployed to search for the portfolio of factors that yields the best linear fit to the target time series. If the option `"full"` is specified, then at each step, all possible additions to the portfolio are considered and the one which yields the highest likelihood score improvement is chosen. If the option `"limited"` is specified, then at each step, the correlation of the residuals of the current portfolio is computed with respect to each of the candidate series in the input set X, and the top B series are chosen for further consideration. Among these top B candidates, the one which improves the likelihood score by the greatest amount is chosen. The parameter B can be controled via `maxfact` (default: `search_type = "lasso"`).

- key return values: The best fit (`$pci`), the column indices (`$indexes`), and the names of the factors included in the best fit (`$index_names`).

4. Examples

4.1. Finance

As an introductory example, we explore the relationship between Royal Dutch Shell plc A (RDS-A) and Royal Dutch Shell plc B (RDS-B), using daily (closing) price data from 1 January 2006 to 1 December 2016.[11] To download the price data we use the getYahooData() function, implemented in the R package **TTR** (Ulrich, 2016). The subsequent R code is used to obtain the data.

```
library(partialCI)
library(TTR)

RDSA<-getYahooData("RDS-A", 20060101, 20161201)$Close
RDSB<-getYahooData("RDS-B", 20060101, 20161201)$Close
```

A classic cointegration analysis yields that the two time series are not cointegrated. In particular, we apply the two-step approach of Engle and Granger (1987) implemented in the R package **egcm**. By default, the **egcm** package uses the unit root test of Phillips and Perron (1988)[12] (specification: with constant, no linear time trend) to investigate the residuals obtained from an Ordinary Least Squares (OLS) regression. The R code,

```
library(egcm)

egcm_finance <- egcm(RDSA,RDSB,include.const = FALSE),
```

results in the following output:

```
Y[i] =   0.9732 X[i] +   0.0000 + R[i], R[i] =   0.9941 R[i-1] + eps[i],
        (0.0005)        (0.0000)                (0.0025)
```

[11]RDS-A (Royal Dutch Shell plc - A, 2016) and RDS-B (Royal Dutch Shell plc - B, 2016) data are downloaded from Yahoo Finance.

[12]The test of Phillips and Perron (1988) corrects for heteroscedasticity, a well-known stylized fact of financial price time series (Krauss and Herrmann, 2017).

```
eps ~ N(0, 0.1679^2)


R[2016-12-01] = -1.8991 (t = -1.477)

WARNING: X and Y do not appear to be cointegrated.
```

The residual plot in figure 1 (code: `plot(egcm_finance$residuals,type = "l")` suggests that the residual series is not purely mean-reverting, but rather shows a stochastical trend as well as a mean-reverting behavior. Hence, it is not suprising that RDS-A and RDS-B are

Figure 1: Residual plot classic cointegration: RDS-A and RDS-B (1.01.2006 - 1.12.2016, daily)

not cointegrated. Using the PCI framework, we are able to fit a PCI model to RDS-A and RDS-B with the following R code:

```
PCI_RDSA_RDSB<-fit.pci(RDSA, RDSB, pci_opt_method = c("jp"), par_model
=c("par"), lambda = 0, robust = FALSE, nu = 5, include_alpha = FALSE)).
```

The R output is given as,

```
Fitted values for PCI model
  Y[t] = X[t] %*% beta + M[t] + R[t]
  M[t] = rho * M[t-1] + eps_M [t], eps_M[t] ~ N(0, sigma_M^2)
  R[t] = R[t-1] + eps_R [t], eps_R[t] ~ N(0, sigma_R^2)
```

```
                    Estimate Std. Err
beta_Close            0.9274   0.0038
rho                   0.3959   0.0965
sigma_M               0.1081   0.0083
sigma_R               0.1195   0.0076

-LL = -1117.29, R^2[MR] = 0.540,
```

where `beta_Close` denotes the partially cointegrating coefficient. Thereby, the coefficient of 0.9274 indicates a positive relationship between RDS-A and RDS-B, and the PVMR of 0.54 suggests that the spread time series also exhibits a clear mean-reverting behavior.

In the subsequent step, we utilize the test.pci() function to check whether RDS-A and RDS-B are partially cointegrated. The R code

```
test.pci(RDSA, RDSB, alpha = 0.05, null_hyp = c("rw", "ar1"), robust =
                    FALSE, pci_opt_method = c("jp")),
```

leads to the following output:

```
Likelihood ratio test of [Random Walk or CI(1)] vs Almost PCI(1)
        (joint penalty method)

data:  StockA

Hypothesis             Statistic    p-value
Random Walk               -55.09      0.010
AR(1)                     -52.88      0.010
Combined                              0.010.
```

Recall that a time series is classified as partially cointegrated, if and only if the random walk as well as the AR(1)-hypotheses are rejected. The p-value of 0.010 for the combined null hypothesis indicates that RDS-A and RDS-B are partially cointegrated in the considered period of time.

Next, we demonstrate the use of the statehistory.pci() function which allows to estimate and extract the hidden states. The R code,

`statehistory.pci(PCI_RDSA_RDSB)`, results in the R output:

```
                   Y     Yhat         Z           M         R       eps_M       eps_R
2006-01-03 35.87002 35.26781 0.6022031  0.00000000 0.6022031  0.00000000  0.00000000
2006-01-04 36.23993 35.57175 0.6681755  0.02030490 0.6478706  0.02030490  0.04566752
2006-01-05 35.80276 35.24161 0.5611509 -0.02112621 0.5822771 -0.02916450 -0.06559352
2006-01-06 36.48653 35.83377 0.6527591  0.01590352 0.6368556  0.02426695  0.05457850
...
2016-11-25 50.18000 49.52231 0.6576906 -0.08762384 0.7453144 -0.07643882 -0.17191764
2016-11-28 49.20000 48.22397 0.9760311  0.04699758 0.9290335  0.08168603  0.18371909
2016-11-29 49.06000 48.02922 1.0307808  0.04419468 0.9865862  0.02558931  0.05755262
2016-11-30 51.10000 50.23639 0.8636066 -0.02573955 0.8893462 -0.04323530 -0.09724000
2016-12-01 51.78000 51.15450 0.6254956 -0.08826115 0.7137567 -0.07807140 -0.17558945.
```

The latter table covers the estimates of the hidden states M and R as well as the corresponding error terms `eps_M` and `eps_R`. `Z` is equal to the sum of M and R. The estimate of the target time series is denoted by `Yhat`. Figure 2 illustrates a plot of the extracted mean-reverting component of the spread associated with the RDS-A and RDS-B price time series (`plot(statehistory.pci(PCI_RDSA_RDSB)[,4] ,type = "l",ylab = "", xlab = "")`). The horizontal blue lines are equal to two times

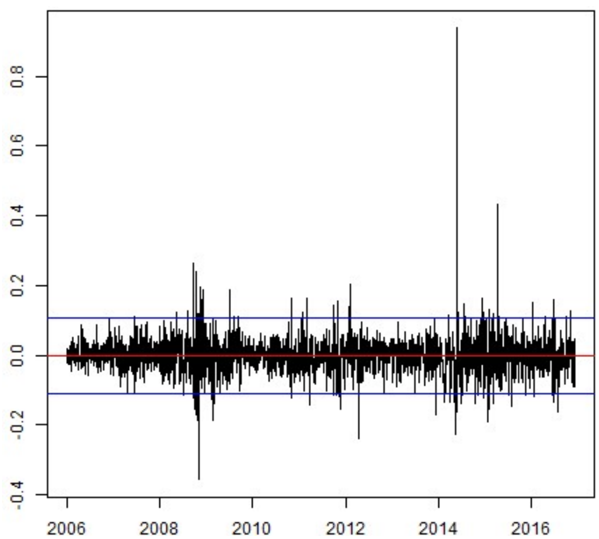

Figure 2: Mean-reverting component RDS-A and RDS-B (1.01.2006 - 1.12.2016, daily)

the historical standard deviation in absolute terms of the mean-reverting component. A pairs trading strategy could exploit the mean-reverting behavior of M_t. Note that this example is in-sample; for a true out-of-sample application see Clegg and Krauss (2016).

We continue with using hedge.pci() to find the set of sector ETFs forming the best hedging portfolio for the SPY index (S&P500 index). Thereby, the R code,

```
sectorETFS <- c("XLB", "XLE", "XLF", "XLI", "XLK", "XLP", "XLU", "XLV", "XLY")
prices <- multigetYahooPrices(c("SPY", sectorETFS), start=20060101)

hedge.pci(prices[,"SPY"], prices),
```

results in the subsequent output:

```
   -LL     LR[rw]    p[rw]    p[mr]     rho    R^2[MR] Factor |  Factor coefficients
 2320.00 -23.3743  0.0100  0.0100  0.5759  0.4526    XLI | 3.1106
 1765.50 -46.5925  0.0100  0.0100  0.3170  0.4713    XLY | 1.8951  1.1989
 1494.95 -53.7256  0.0100  0.0100  0.3244  0.5038    XLV | 1.6999  0.9106  0.6619
  972.58 -65.9058  0.0100  0.0100  0.4060  0.5904    XLK | 1.3089  0.4933  0.5320 1.5182.
```

The table summarizes information about the best hedging portfolio, where each row corresponds to an increasing number of factors. Row 1: The best single-factor hedging portfolio comprises XLI (industrials) as only factor. Row 2: The best two-factor hedging portfolio consists of XLI and XLY (consumer discretionary). As such, XLY leads to the best improvement of the LRT score among all remaining factors. Row 3 includes XLV (health care) for the three-factor portfolio and row 4 XLK (technology) for the best four-factor portfolio. The last row corresponds to the overall best fit out of the nine potential sector ETFs, based on the LRT score. Note that for all rows, the union of random walk and AR(1)-null hypothesis is rejected at the 5 percent significant level, so we find a PCI model at each step.

4.2. Macroeconomics

As a second example, we revisit the relationship between GDP and personal consumption expenditures for the United States (among others see Cochrane (1994), Gonzalo et al. (2008) and Guisan (2008)), using quarterly seasonally adjusted annual rates in billion US-Dollar from January 1976 to July 2016.[13] The following R code triggers the data download:

[13] We utilize the R package **Quandl** (Daroczi et al., 2016) to download the GDP (US. Bureau of Economic Analysis, 2016a) as well as personal consumption expenditures data (US. Bureau of Economic Analysis, 2016b). Thereby, the time series data are directly converted into **xts** (Ryan and Ulrich, 2014) objects.

```
library(xts)
library(Quandl)
library(partialCI)

GDP = Quandl("FRED/GDP", start_date = "1976-01-01",
        end_date = "2016-04-01", type = "xts")
Consumption = Quandl("FRED/PCEC", start_date = "1976-01-01",
        end_date = "2016-04-01",type = "xts").
```

Applying the unit root test of Phillips and Perron (1988) as implemented in the R package **egcm** yields that GDP and personal consumption are not cointegrated in the classic sense, within the considered time frame.[14] The residual plot in figure 3 (code: `plot(egcm_macro$residuals,type = "l"`) obtained from standard cointegration analysis shows that the residuals exhibit both, mean-reverting and stochastic trending behavior.[15] To account for the

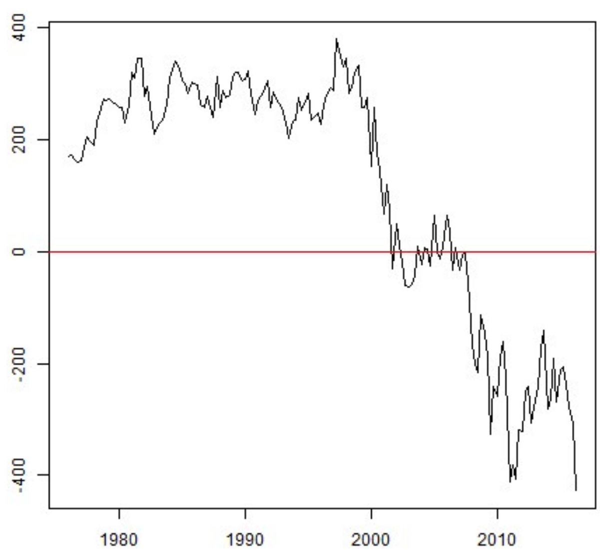

Figure 3: Residual plot classic cointegration: GDP and consumption (1976-2016, quarters)

stochastic trending behavior we apply the following PCI model:

[14]The R code is given as `egcm_macro <- egcm(Consumption,GDP,include.const = FALSE)`. For the sake of brevity, we do not show the R output.

[15]We are aware of the structural break in the residual series around the second quarter of the year 2000. The function breakpoints() implemented in the R package **strucchange** (Hornik et al., 2003) is used to obtain the estimate of the structural break.

```
PCI_GDP_Consumption<-fit.pci(GDP, Consumption, pci_opt_method = c("jp"),
par_model =c("par"), lambda = 0, robust = FALSE, nu = 5, include_alpha =
FALSE)).
```

The latter function yields the following R output:

```
Fitted values for PCI model
  Y[t] = X[t] %*% beta + M[t] + R[t]
  M[t] = rho * M[t-1] + eps_M [t], eps_M[t] ~ N(0, sigma_M^2)
  R[t] = R[t-1] + eps_R [t], eps_R[t] ~ N(0, sigma_R^2)

            Estimate Std. Err
beta_        1.3963   0.0358
rho          0.2812   0.3357
sigma_M     27.1132   8.7402
sigma_R     35.3842   6.8836

-LL = 845.02, R^2[MR] = 0.478.
```

Thereby, the coefficient of 1.396 is associated with a positive relationship between GDP and personal consumption. From a policy makers point of view the existence of such a partial equilibrium relationship is crucial for designing appropriate economic stimulus packages. The mean-reverting component accounts for 47.8 percent of the total variance, i.e., political authorities could utilize this partly predicitive behavior for anti-cyclical fiscal policy interventions.

Next, we use test.pci() to test, if GDP and personal consumption are indeed partially cointegrated. The R code is given by,

```
test.pci(GDP, Consumption, alpha = 0.05, null_hyp =c("rw", "ar1"), robust
= FALSE, pci_opt_method =c("jp")),
```

leading to the subsequent output:

```
Likelihood ratio test of [Random Walk or CI(1)] vs Almost PCI(1)
        (joint penalty method)


data:  GDP
```

```
Hypothesis              Statistic    p-value
Random Walk                -12.76      0.010
AR(1)                       -2.47      0.010
Combined                               0.010.
```

Folllowing the p-value for the combined null hypothesis, GDP and personal consumption in the United Stated are indeed partially cointegrated within the considered time frame.

To estimate the hidden states we use the statehistory.pci() function:

```
statehistory.pci(PCI_GDP_Consumption).
```

The latter code yields to the following output:

```
                Y       Yhat        Z           M         R       eps_M       eps_R
 1976 Q1 1824.5 1553.123 271.3768  0.0000000 271.3768  0.00000000  0.0000000
 1976 Q2 1856.9 1580.631 276.2693  1.2902076 274.9791  1.29020760  3.6023508
 1976 Q3 1890.5 1621.543 268.9573 -1.3209016 270.2782 -1.68368735 -4.7009741
 1976 Q4 1938.4 1668.738 269.6618 -0.4360234 270.0978 -0.06460694 -0.1803871
 1977 Q1 1992.5 1718.307 274.1925  0.9895420 273.2030  1.11214485  3.1051870
 ...
 2015 Q1 17783.6 16893.90 889.7023  12.3240495 877.3782  14.279077  39.868192
 2015 Q2 17998.3 17091.20 907.1027  10.3900797 896.7126   6.924754  19.334401
 2015 Q3 18141.9 17254.15 887.7525  -0.2117553 887.9643  -3.133280  -8.748339
 2015 Q4 18222.8 17368.51 854.2942  -8.9229218 863.2171  -8.863380 -24.747182
 2016 Q1 18281.6 17451.17 830.4322 -10.4929841 840.9252  -7.984001 -22.291894
 2016 Q2 18450.1 17723.03 727.0693 -32.1971225 759.2665 -29.246663 -81.658749.
```

Thereby, `M` denotes the mean-reverting component and `R` the random walk component, respectively. To illustrate a possible application of the statehistory.pci() function in a macroeconomic context we extract and plot the mean-reverting component. To reduce the noise and smooth the mean-reverting component series, we use a moving average, i.e., observation i is replaced by the mean of the observations $i, i-1, i-2$ and $i-3$, where $i \geq 4$. In particular, the rollmean() function from the **zoo** package is applied:

```
MRC_GDP<-statehistory.pci(PCI_GDP_Consumption)[,4]
RollingMean<-as.zoo(coredata(rollmean(MRC_GDP,4)),index(MRC_GDP)[-c(1:3)])
plot(RollingMean, type = "l").
```

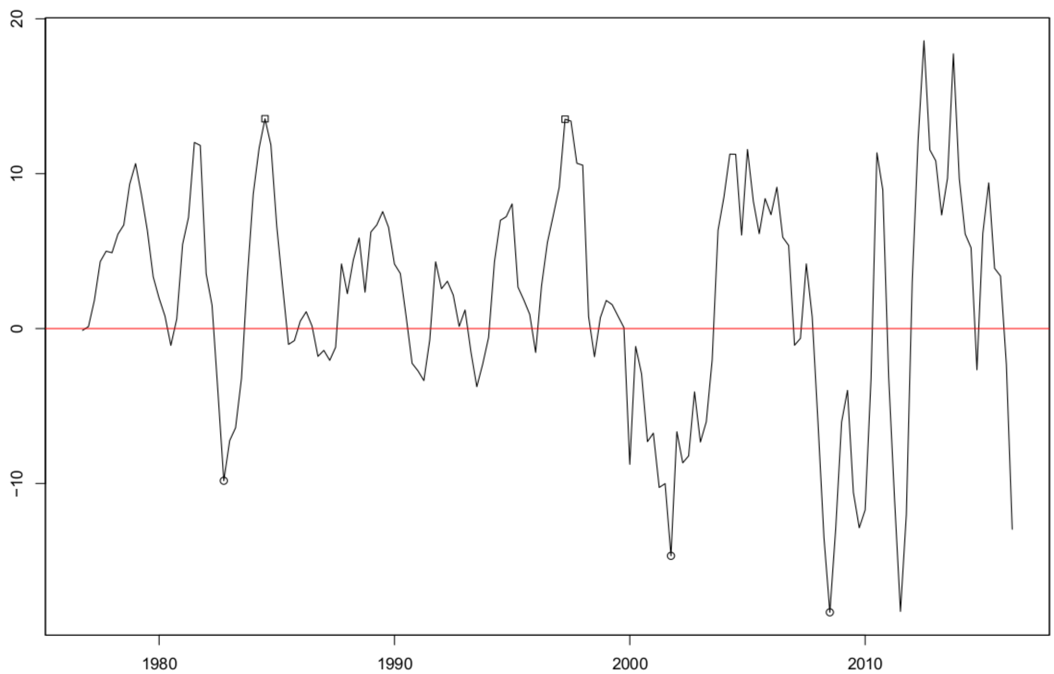

Figure 4: Mean-reverting component (running-mean (k = 4)): GDP and consumption (1976-2016, quarters); circles = troughs, squares = peaks

A close investigation of figure 4 shows that the mean-reverting component identifies peaks and troughs of major macroeconomic expansions and recessions. The circles denote troughs during severe U.S. recessions, whereas the squares represent peaks of important economic U.S. expansions. From left to right, the first circle corresponds to the early 1980's crisis, mainly caused by the 1979 energy crisis and the contractionary policy of the U.S. central bank (FED). The next circle identifies the early 2000's crisis which can to some extent be attributed to the bust of the dot-com bubble and the September 11 attacks. The third circle is associated with the global financial crisis. The first square is associated with the economic expansion during the Reagan era. The second square covers the emergence of the dot-com bubble. To evaluate the accuracy of event identification associated with the mean-reverting component, we contrast the mean-reverting component with a Hodrick-Prescott filter (HP filter) – the standard tool in macroeconomics (Hodrick and Prescott (1997), Guay and St.-Amant (2005), Harvey and Trimbur (2008), Choudhary et al. (2014)).[16] The basic idea of

[16]To deal with the well-known drawbacks of the HP filter (among others see King and Rebelo (1993) and Canova (1998)) we apply the approximate band-pass filter of Baxter and King (1999), but the general pattern does not change.

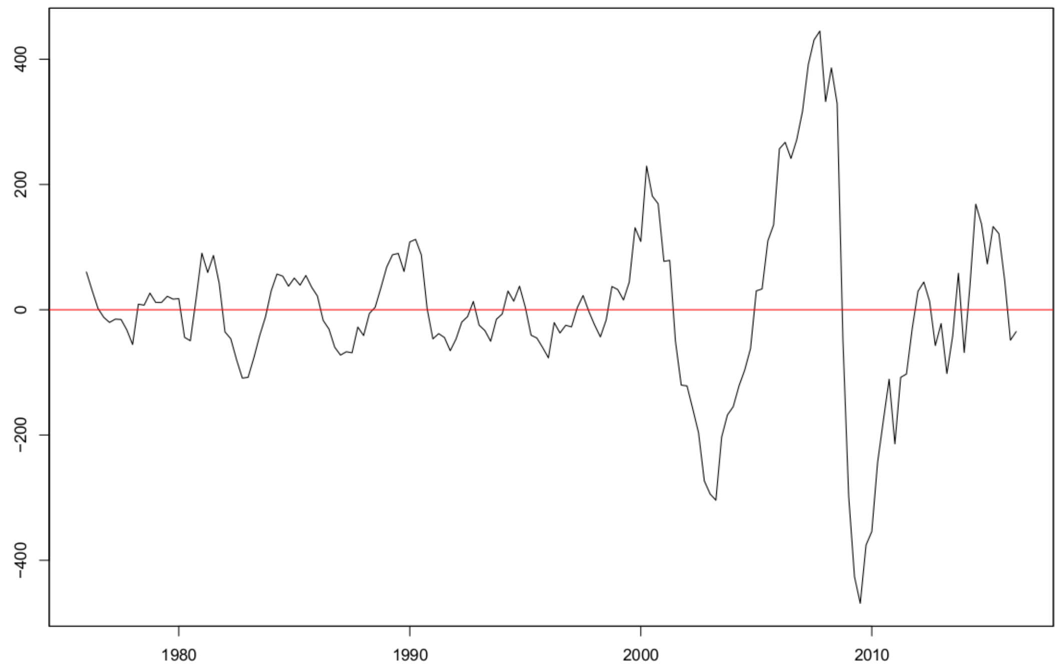

Figure 5: Hodrick-Prescott filter ($\lambda = 1400$): GDP (1976-2016, quarters)

Hodrick and Prescott (1997) is to seperate a given time series in a trend and a stationary component. The HP filter is already implemented in the R package **mFilter** (Balcilar, 2007), and we can apply it with:

```
library(mFilter)

HPF_GDP <- mFilter::hpfilter(GDP, freq=1600, type=c("lambda"), drift=TRUE),
```

where `lambda` denotes the smoothing parameter. In the business cycle literature it is common to choose $\lambda = 1600$ (`freq`) when analyzing quarterly data (Hodrick and Prescott, 1997; Ravn and Uhlig, 2002). Figure 5 (code: `plot(HPF_GDP,type = "l"`) shows the plot of the cyclical GDP component. A comparison of figures 4 and 5 reveal that many of the peaks and troughs identified by the mean-reverting component are similar to those identified by the HP filter.

The GDP consists of four major components – namely personal consumption expenditures, investment[17], government expenditures and net exports (Hodrick and Prescott, 1997). Given

[17]In line with Hodrick and Prescott (1997) we consider total fixed investment.

these four possible factors, we utilize the hedge.pci() function to identify the optimal hedging portfolio for GDP.[18] The R code is given as,

```
GS = Quandl("FRED/GCE", start_date = "1976-01-01",
        end_date = "2016-04-01", type = "xts")
Investment = Quandl("FRED/FPI", start_date = "1976-01-01",
        end_date = "2016-04-01", type = "xts")
Export = Quandl("FRED/EXPGS", start_date = "1976-01-01",
        end_date = "2016-04-01", type = "xts")
Import = Quandl("FRED/IMPGS", start_date = "1976-01-01",
        end_date = "2016-04-01", type = "xts")
NetExport <- Export - Import.
```

Next, we run the hedge.pci() function with the search algorithm `"full"`.

```
FactorMatrix <- cbind(Consumption,Investment,GS,NetExport)

HedgeGDP<-hedge.pci(GDP, FactorMatrix,
                            maxfact = 4,
                            lambda = 0 ,
                            use.multicore = TRUE,
                            minimum.stepsize = 0,
                            verbose = TRUE,
                            exclude.cols = c(),
                            search_type = c("full"),
                            pci_opt_method=c("jp")).
```

The corresponding R output is given as,

```
-LL      LR[rw]    p[rw]    p[mr]      rho    R^2[MR]  Factor | Factor coefficients
845.02 -12.7580   0.0100   0.0100   0.2812   0.4782       ..1 | 1.3963
829.04 -14.5563   0.0100   0.0100   0.2532   0.6465       ..2 | 1.2622 0.4907.
```

[18] As a preliminary step we download quarterly investment (US. Bureau of Economic Analysis, 2016c), government expenditures (US. Bureau of Economic Analysis, 2016d), export (US. Bureau of Economic Analysis, 2016e) and import data (US. Bureau of Economic Analysis, 2016f) for the time span of interest, using **Quandl**. Net exports are derived as exports minus imports.

At the first stage, the best single-factor hedging portfolio contains personal consumption expenditures. At the second stage, the best two-factor hedging portfolio consists of personal consumption expenditures and investment, i.e., investment leads to the highest LRT score improvement compared to government expenditures and net exports. Out of the four potential components of GDP, the overall best hedging portfolio consists of personal consumption expenditures and investment. Note that GDP, investment and personal consumption expenditures are partially cointegrated, i.e., they share a partial equilibrium relationship. Thus, for policy makers investment is a second possible channel to stimulate the economy.

5. Conclusion

In this article, we introduce the partial cointegration model and discuss differences to other cointegration concepts. Thereby, we contribute to the literature by extending the partial cointegration model from the special case of two partially cointegrated time series (see Clegg and Krauss (2016)) to the general case of $k+1$ partially cointegrated time series. Next, we outline the estimation procedure and the likelihood ratio test routine for partial cointegration. Furthermore, we explain in detail how to use the most important functions implemented in the **partialCI** package – our second contribution to the literature. The functionality is illustrated with a financial application in the context of pairs trading and a macroeconomic application, revisiting the relationship between GDP and consumption. For both examples, we demonstrate that the variables are not cointegated in the classic sense, but can be modeled with partial cointegration.

Bibliography

Alexander, C., 2011. Practical financial econometrics, reprinted with corr Edition. Vol. / Carol Alexander ; Vol. 2 of Market risk analysis. Wiley, Chichester [u.a.].

Baillie, R. T., 1996. Long memory processes and fractional integration in econometrics. Journal of Econometrics 73 (1), 5–59.

Balcilar, M., 2007. mFilter: Miscellaneous time series filters. URL https://CRAN.R-project.org/package=mFilter

Balke, N. S., Fomby, T. B., 1997. Threshold cointegration. International Economic Review 38 (3), 627.

Baxter, M., King, R. G., 1999. Measuring business cycles: Approximate band-pass filters for economic time series. Review of Economics and Statistics 81 (4), 575–593.

Brockwell, P. J., Davis, R. A., 2010. Introduction to time series and forecasting, 2nd Edition. Springer texts in statistics. Springer, New York [u.a.].

Canova, F., 1998. Detrending and business cycle facts: A user's guide. Journal of Monetary Economics 41 (3), 533–540.

Choudhary, M. A., Hanif, M. N., Iqbal, J., 2014. On smoothing macroeconomic time series using the modified HP filter. Applied Economics 46 (19), 2205–2214.

Clegg, M., 2015a. Modeling time series with both permanent and transient components using the partially autoregressive model. SSRN Electronic Journal.
URL http://dx.doi.org/10.2139/ssrn.2556957

Clegg, M., 2015b. partialAR: Partial autoregression.
URL https://CRAN.R-project.org/package=partialAR

Clegg, M., 2015c. egcm: Engle-Granger cointegration models.
URL https://CRAN.R-project.org/package=egcm

Clegg, M., 2016. partialCI: Partial cointegration.
URL https://github.com/matthewclegg/partialCI

Clegg, M., Krauss, C., 2016. Pairs trading with partial cointegration. FAU Discussion Papers in Economics, University of Erlangen-Nürnberg.

Cochrane, J. H., 1994. Permanent and transitory components of GNP and stock prices. The Quarterly Journal of Economics 109 (1), 241–265.

Daroczi, G., Leung, C., McTaggart, R., 2016. Quandl: API wrapper for quandl.com.
URL https://CRAN.R-project.org/package=Quandl

Durbin, J., Koopman, S. J., 2012. Time series analysis by state space methods, 2nd Edition. Vol. 38 of Oxford statistical science series. Oxford University Press, Oxford.

Engle, R. F., Granger, C. W. J., 1987. Co-Integration and error correction: Representation, estimation, and testing. Econometrica 55 (2), 251.

Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33 (1), 1–22.
URL https://CRAN.R-project.org/package=glmnet

Gonzalo, J., Lee, T.-H., Yang, W., 2008. Permanent and transitory components of GDP and stock prices: Further analysis. Macroeconomics and Finance in Emerging Market Economies 1 (1), 105–120.

Grothendieck, G., Zeileis, A., 2005. zoo: S3 infrastructure for regular and irregular time series. Journal of Statistical Software 14 (6), 1–27.
URL https://CRAN.R-project.org/package=zoo

Guay, A., St.-Amant, P., 2005. Do the Hodrick-Prescott and Baxter-King filters provide a good approximation of business cycles? Annales d'Économie et de Statistique (77), 133–135.

Guisan, M.-C., 2008. Causality and cointegration between consumption and GDP in 25 OECD countries: Limitations of the cointegration approach.

Harvey, A., Trimbur, T., 2008. Trend estimation and the Hodrick-Prescott filter. Journal of the Japan Statistical Society 38 (1), 41–49.

Hodrick, R., Prescott, E., 1997. Postwar U.S. business cycles: An empirical investigation. Journal of Money, Credit and Banking 29 (1), 1–16.

Hornik, K., Kleiber, C., Kraemer, W., Zeileis, A., 2003. Testing and dating of structural changes in practice. Computational Statistics & Data Analysis 44, 109–123.
URL https://CRAN.R-project.org/package=strucchange

Huber, P. J., 1981. Robust statistics. John Wiley & Sons, Inc.

King, R. G., Rebelo, S. T., 1993. Low frequency filtering and real business cycles. Journal of Economic Dynamics and Control 17 (1-2), 207–231.

Krauss, C., Herrmann, K., 2017. On the power and size properties of cointegration tests in the light of high-frequency stylized facts. Journal of Risk and Financial Management 10 (1), 7.

Lütkepohl, H., 2007. New introduction to multiple time series analysis, 1st Edition. Springer, Berlin.

Pfaff, B., 2008. Analysis of integrated and cointegrated time series with R, 2nd Edition. Springer, New York.

Phillips, P. C. B., Perron, P., 1988. Testing for a unit root in time series regression. Biometrika 75 (2), 335–346.

Poterba, J. M., Summers, L. H., 1988. Mean reversion in stock prices. Journal of Financial Economics 22 (1), 27–59.

Ravn, M. O., Uhlig, H., 2002. On adjusting the Hodrick-Prescott filter for the frequency of observations. Review of Economics and Statistics 84 (2), 371–376.

Ripley, B. D., Venables, W. N., 2002. Modern applied statistics with S, 4th Edition. Springer, New York.

Royal Dutch Shell plc - A, 2016. Historical data.
URL https://finance.yahoo.com/quote/RDS-A/history?p=RDS-A

Royal Dutch Shell plc - B, 2016. Historical data.
URL https://finance.yahoo.com/quote/RDS-B/history?p=RDS-B

Ryan, J. A., Ulrich, J. M., 2014. xts: eXtensible time series.
URL https://CRAN.R-project.org/package=xts

Schwartz, E., Smith, J. E., 2000. Short-term variations and long-term dynamics in commodity prices. Management Science 46 (7), 893–911.

Stigler, M., 2010. tsDyn: Threshold cointegration: Overview and implementation in R. URL https://CRAN.R-project.org/package=tsDyn

Summers, L. H., 1986. Does the stock market rationally reflect fundamental values? The Journal of Finance 41 (3), 591.

Ulrich, J., 2016. TTR: Technical Trading Rules. URL https://CRAN.R-project.org/package=TTR

US. Bureau of Economic Analysis, 2016a. Gross domestic product [GDP]. Federal Reserve Bank of St. Louis. URL https://fred.stlouisfed.org/series/GDP

US. Bureau of Economic Analysis, 2016b. Personal consumption expenditures [PCEC]. Federal Reserve Bank of St. Louis. URL https://fred.stlouisfed.org/series/PCEC

US. Bureau of Economic Analysis, 2016c. Fixed private investment [FPI]. Federal Reserve Bank of St. Louis. URL https://fred.stlouisfed.org/series/FPI

US. Bureau of Economic Analysis, 2016d. Government consumption expenditures and gross investment [GCE]. Federal Reserve Bank of St. Louis. URL https://fred.stlouisfed.org/series/GCE

US. Bureau of Economic Analysis, 2016e. Exports of goods and services [EXPGS]. Federal Reserve Bank of St. Louis. URL https://fred.stlouisfed.org/series/EXPGS

US. Bureau of Economic Analysis, 2016f. Imports of goods and services [IMPGS]. Federal Reserve Bank of St. Louis. URL https://fred.stlouisfed.org/series/IMPGS

Verbeek, M., 2010. A guide to modern econometrics, 3rd Edition. Wiley, Chichester.