

# 1 dsepadi Functions

In R, the functions in this package are made available with

```
> library("dsepadi")
```

and most of the examples also need

```
> require("padi")
```

This section describes utilities for retrieving data from an online database. This has been implemented using the TS PADI interface. The examples use series names which are specific to the Bank of Canada.

Data is retrieved with a description which gives an indication of where the data comes from, which series are model inputs and which are model outputs, any transformations which should be applied to the data, and some padding information indicating whether the series should be padded with NAs to the length of the longest available series or truncated to the subset where all data is available for all series. Data is retrieved by using the generic function `freeze()` on the description. When `freeze()` is applied to an object which is already time series data then the data is simply returned. When applied to a data description object the data is retrieved from the data base. Most of the functions in the DSE library use the function `freeze()` on data, so data descriptions can be used interchangeably with data. For model estimation purposes it is usually desirable to retrieve the data and work with a fixed data set, but once a model is established and is routinely used with newly available data then the data description is more convenient.

The following simple example specifies the series *I37005* from the *ets* server as the single output series, and gives it a more descriptive name. No data transformations are performed.

```
> eg2.DSE.data.names <- TSPADIdata(output = "I37005", output.names = "manuf.prod.",  
  server = "ets")
```

Setting `output.names` is optional. If they are set then they will be used in many printing and plotting routines. The following line then returns the data.

```
> if (require("padi") && checkPADIServer("ets")) eg2.DSE.data <- freeze(eg2.DSE.data.names)
```

The following example specifies one input and one output series.

```
> eg3.DSE.data.names <- TSPADIdata(input = "lfsa455", input.transforms = "percentChange",  
  input.names = "manuf.emp.", output = "i37005", output.names = "manuf.prod.",  
  output.transforms = "percentChange", pad.start = F, pad.end = T,  
  server = "ets")  
> if (require("padi") && checkPADIServer("ets")) eg3.DSE.data <- freeze(eg3.DSE.data.names)
```

Here is a multivariate example:

```
> JofF.VAR.data.names <- TSPADIdata(input = "B14017", input.transforms = "diff",
  input.names = "R90", output = c("P100000", "I37026", "b1627",
    "b14013", "b4237", "D767608", "b3400", "M.BCPI", "M.JQIND",
    "M.CUSA0"), output.transforms = c("percentChange", "percentChange",
    "percentChange", "diff", "diff", "percentChange", "percentChange",
    "percentChange", "percentChange", "percentChange"), output.names = c("CPI",
    "GDP", "M1", "RL", "TSE300", "employment", "PFX", "com. price ind.",
    "US ind. prod.", "US CPI"), server = "ets")
> if (require("padi") && checkPADIServer("ets")) JofF.VAR.data <- freeze(JofF.VAR.data.names)
```

This is the example used in Gilbert (1995), except that `freeze()` will retrieve that latest version of the data, and P484549 has been discontinued is replaced with P100000 (CPI, 1992=100).

The variables `pad`, `pad.start`, and `pad.end` control what happens at the beginning and end of multivariate data when all series are not available for the same periods. If `pad.start` is TRUE then NAs are placed at the beginning of series if data is not available, so the multivariate series starts with the first available data. If `pad.start` is FALSE then the beginning is truncated so that the first multivariate data point contains values for all variables. Similarly, `pad.end` works with the last periods of the series. `pad` can be used in place of `pad.start` and `pad.end`.

Most estimation routines require a complete data set for all variables (`pad=F`), but for many purposes it is useful to have all the data. The function `trimNA` takes a complete data set and removes padding at both ends. This is a convenient way to remove NAs from the beginning and end before estimation. The function `tfwindow` can also be used to truncate series to a desired sample period.

An alternate form for specifying the data names can be given using the function `TSPADIdata2`:

```
> alt.JofF.VAR.data.names <- TSPADIdata2(input = list(c("ets",
  "", "B14017", "diff", "R90")), output = list(c("ets", "",
  "P100000", "percentChange", "CPI"), c("ets", "", "I37026",
  "percentChange", "GDP"), c("ets", "", "b1627", "percentChange",
  "M1"), c("ets", "", "b14013", "diff", "RL"), c("ets", "",
  "b4237", "diff", "TSE300"), c("ets", "", "D767608", "percentChange",
  "employment"), c("ets", "", "b3400", "percentChange", "PFX"),
  c("ets", "", "M.BCPI", "percentChange", "com. price ind."),
  c("ets", "", "M.JQIND", "percentChange", "US ind. prod."),
  c("ets", "", "M.CUSA0", "percentChange", "US CPI")))
```

The result is the same but this form may be more convenient in some circumstances. For each series the character strings indicate the server, additional server information, the series identifier, any transformation, and finally a series description. The order of these strings is important. The additional server information may be empty, as above, but cannot be omitted. For some servers it may be used to pass information such as a source database. If no data transformation is to be done then the third string should be empty ("").

A smaller example, also used in Gilbert (1995), is given by:

```
> eg4.DSE.data.names <- TSPADIdata(input = "B14017", input.transforms = "diff",  
  input.names = "R90", output = c("P100000", "I37026", "D767608",  
    "b3400"), output.transforms = c("percentChange", "percentChange",  
    "percentChange", "percentChange"), output.names = c("CPI",  
    "GDP", "employment", "PFX"), server = "ets")  
> if (require("padi") && checkPADIservers("ets")) eg4.DSE.data <- freeze(eg4.DSE.data.names)
```