# Box 1 - Example DFA workflow

## Model specification

For this example, we will assume that the observations at time $t$ ($\mathbf{y}_t$) arise from a multivariate Gaussian distribution wherein the mean vector is the sum of a vector of intercepts ($\boldsymbol{\alpha}$) and latent factors ($\mathbf{L}\boldsymbol{\gamma}_t$), and the covariance matrix $\boldsymbol{\Phi}$ is diagonal and unequal. Eqn (1) then becomes:

$$\mathbf{y}_t \sim \text{MVN}(\boldsymbol{\alpha} + \mathbf{L}\boldsymbol{\gamma}_t, \boldsymbol{\Phi}).$$

The latent factors are themselves a multivariate random walk with Gaussian errors. Here we assume $\mathbf{Q}$ equals the identity matrix $\mathbf{I}$, such that Eqn (3) is

$$\boldsymbol{\gamma}_t \sim \text{MVN}(\boldsymbol{\gamma}_{t-1}, \mathbf{Q}).$$

## Fitting the model

For this example, we make use of the **MARSS** package (Holmes et al. 2012) for **R** to estimate the parameters in the DFA model. The notation of Holmes et al. is somewhat different than ours, however, so we must write out our DFA model in a form that the `MARSS` function will understand based on the following specification of a state-space model:

$$\begin{gathered}\mathbf{y}_t \sim \text{MVN}(\mathbf{a} + \mathbf{Z}\mathbf{x}_t, \mathbf{R}) \\ \mathbf{x}_t \sim \text{MVN}(\mathbf{u} + \mathbf{B}\mathbf{x}_{t-1}, \mathbf{Q})\end{gathered}.$$

Thus, we will need to define $\mathbf{a} = \boldsymbol{\alpha}$, $\boldsymbol{\gamma}_t = \mathbf{x}_t$, $\boldsymbol{\Phi} = \mathbf{R}$, $\mathbf{u} = \mathbf{0}$, and $\mathbf{B} = \mathbf{I}$.

The `MARSS` function makes use of list matrices, which allow one to mix character and numeric classes within the same matrix. Any values of class `character` are interpreted as parameters to be estimated; anything `numeric` is fixed at its value.

```r
library(MARSS)
## matrix dims
nn <- 5
mm <- 3
## model list
mod_list <- list(
  ## observation eqn
  Z = matrix(list(0),nn,mm), # all 0's for now
  A = matrix(list(0),nn,1),  # all 0's for now
  R = matrix(list(0),nn,nn), # all 0's for now
  ## state eqn
  B = diag(mm),              # identity matrix
  U = matrix(0,mm,1),        # zero vector
  Q = diag(mm)               # identity matrix
)
## specify observation parameters
## last n-m values of A are non-zero
mod_list$A[(mm+1):nn] <- as.character(seq(mm+1,nn))
## upper right corner of Z stays zero
for(cc in 1:mm) {
  for(rr in cc:nn) {
    mod_list$Z[rr,cc] <- paste0(rr,cc)
```

```r
  }
}
## R is diagonal and unequal
diag(mod_list$R) <- paste0(seq(nn),seq(nn))
mod_list
```

```
## $Z
##      [,1] [,2] [,3]
## [1,] "11" 0    0
## [2,] "21" "22" 0
## [3,] "31" "32" "33"
## [4,] "41" "42" "43"
## [5,] "51" "52" "53"
##
## $A
##      [,1]
## [1,] 0
## [2,] 0
## [3,] 0
## [4,] "4"
## [5,] "5"
##
## $R
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "11" 0    0    0    0
## [2,] 0    "22" 0    0    0
## [3,] 0    0    "33" 0    0
## [4,] 0    0    0    "44" 0
## [5,] 0    0    0    0    "55"
##
## $B
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
##
## $U
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
##
## $Q
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

```r
#mod1 <- MARSS(rbind(x1, x2), model=mod_list, inits=list(x0=matrix(0,2,1)))
```