title: “Ch 5 PFA in dynr” author: “Sy-Miin Chow” output: pdf_document —

```
rm(list=ls())
# Load packages
require(dynr)
```

```
## Warning: package 'dynr' was built under R version 3.4.1
```

A demo for how to specify dynr recipes for a process factor analysis model

```
#Prepare dynr recipes
#Define the dynamic model
 dynamics <- prep.matrixDynamics(
    values.dyn=matrix(c(.5, 0.1, .3, .5), ncol=2,byrow=TRUE),
    params.dyn=matrix(c('phi11', 'phi12', 'phi21', 'phi22'), ncol=2,byrow=TRUE),
    isContinuousTime=FALSE)

meas <- prep.measurement(
    values.load=matrix(c(1,0,
                         2,0,
                         1,0,
                         0,1,
                         0,2,
                         0,1),ncol=2,byrow=TRUE), #Starting values for entries in Lambda
    params.load=matrix(c('fixed',0,
                         'lambda21',0,
                         'lambda31',0,
                         0,'fixed',
                         0,'lambda52',
                         0,'lambda62'),
                       ncol=2,byrow=TRUE), #Labels for fixed and freed parameters
    values.int = rep(0.1,6),
    params.int = paste0('int',1:6),
    state.names=c("eta1","eta2"), #Labels for latent variables in eta(t)
    obs.names=paste0('V',1:6) #Labels for observed variables in y(t)
  )

#Note that in dynr, prep.initial sets the structure of E(eta(1|0)) and Cov(eta(1|0))
#Here, initial condition covariance matrix is fixed to a diagonal matrix of 2s.
#Could also be freely estimated with #multiple-subject data.
#Iinitial means are fixed to a vector of zeros.
  initial <- prep.initial(
    values.inistate=c(0, 0),
    params.inistate=c('fixed', 'fixed'),
    values.inicov=matrix(c(2,0,0,2),ncol=2),
    params.inicov=matrix(c('fixed','fixed','fixed','fixed'),ncol=2))


  #Process and measurement noise covariance matrices
    mdcov <- prep.noise(
    values.latent=matrix(c(2,.5,
                           .5,6),ncol=2,byrow=TRUE),
    params.latent=matrix(c('psi_11','psi_12',
                           'psi_12','psi_22'),ncol=2,byrow=TRUE),
    values.observed=diag(rep(.5,6),6),
    params.observed=diag(paste0('var_e',1:6),6)
    )
```

Read in data and set up data structure in dynr

```
ch5 = read.table('./Data/ch5_data.csv',header=TRUE,sep=",")
ch5$ID = rep(1,dim(ch5)[1]) #Add subject ID to the data set
# Data
ch52 <- dynr.data(ch5, id="ID", time="Time", observed=paste0("V",1:6))
```

Cook it!

```
#Put recipes and data together to prepare the full model
  model <- dynr.model(dynamics=dynamics, measurement=meas,
                      noise=mdcov, initial=initial, data=ch52,
                      outfile="PFA3.c")

#Use the `$' sign to set upper and lower boundaries for the parameters
#For parameters that are subjected to user-specified (e.g., via
#"prep.tfun) transformations or system transformations (e.g., variance-
#covariance parameters in the process and measurement error cov matrices),
#it may be easier to use the `@' sign to set upper and lower boundaries
#on the unconstrained (untransformed) scales - e.g., for the log of a variance
#parameter as opposed to the variance.

  model@ub[!model$param.names %in% c('psi_11','psi_12','psi_22')] =
    c(rep(2,3), rep(5,4), rep(log(10),6))
```

```
## Warning in model@ub[!model$param.names %in% c("psi_11", "psi_12",
## "psi_22")] = c(rep(2, : number of items to replace is not a multiple of
## replacement length
```

```
  model@lb[!model$param.names %in% c('psi_11','psi_12','psi_22')] =
    c(rep(-2,3), rep(-5,4), rep(log(1e-10),6))
```

```
## Warning in model@lb[!model$param.names %in% c("psi_11", "psi_12",
## "psi_22")] = c(rep(-2, : number of items to replace is not a multiple of
## replacement length
```

```
  res <- dynr.cook(model)
```

```
  coef(res)
```

```
##       phi11       phi21       phi12       phi22    lambda21    lambda31
##  0.42045986  0.36392167 -0.03780942  0.45171148  2.04807562  1.02716269
##    lambda52    lambda62        int1        int2        int3        int4
##  2.01686334  1.01327201  0.07979067  0.24471569  0.22645780  0.20659852
##        int5        int6      psi_11      psi_12      psi_22      var_e1
##  0.18984755  0.10645776  2.48464504  2.02878633  8.35279124  0.49062543
##      var_e2      var_e3      var_e4      var_e5      var_e6
##  0.53900831  0.36952354  0.47519325  0.54501583  0.39554848
```

```r
summary(res)
```

```
## Coefficients:
##          Estimate Std. Error t value ci.lower ci.upper Pr(>|t|)
## phi11     0.42046    0.09765   4.306  0.22908  0.61184   <2e-16 ***
## phi21     0.36392    0.17776   2.047  0.01551  0.71233   0.0216 *
## phi12    -0.03781    0.04797  -0.788 -0.13183  0.05621   0.2162
## phi22     0.45171    0.08691   5.197  0.28137  0.62205   <2e-16 ***
## lambda21  2.04808    0.08870  23.090  1.87423  2.22192   <2e-16 ***
## lambda31  1.02716    0.05048  20.348  0.92822  1.12610   <2e-16 ***
## lambda52  2.01686    0.04125  48.891  1.93601  2.09772   <2e-16 ***
## lambda62  1.01327    0.02441  41.515  0.96543  1.06111   <2e-16 ***
## int1      0.07979    0.22751   0.351 -0.36612  0.52570   0.3633
## int2      0.24472    0.45266   0.541 -0.64249  1.13192   0.2950
## int3      0.22646    0.23112   0.980 -0.22654  0.67945   0.1647
## int4      0.20660    0.51720   0.399 -0.80710  1.22030   0.3452
## int5      0.18985    1.03760   0.183 -1.84381  2.22351   0.4276
## int6      0.10646    0.52328   0.203 -0.91915  1.13207   0.4196
## psi_11    2.48465    0.36886   6.736  1.76169  3.20760   <2e-16 ***
## psi_12    2.02879    0.46467   4.366  1.11806  2.93952   <2e-16 ***
## psi_22    8.35279    1.10996   7.525  6.17732 10.52826   <2e-16 ***
## var_e1    0.49063    0.07694   6.377  0.33983  0.64142   <2e-16 ***
## var_e2    0.53901    0.19478   2.767  0.15725  0.92077   0.0034 **
## var_e3    0.36952    0.06506   5.680  0.24201  0.49703   <2e-16 ***
## var_e4    0.47519    0.07638   6.221  0.32548  0.62490   <2e-16 ***
## var_e5    0.54502    0.20567   2.650  0.14191  0.94812   0.0047 **
## var_e6    0.39555    0.07081   5.586  0.25677  0.53433   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -2 log-likelihood value at convergence = 2539.05
## AIC = 2585.05
## BIC = 2650.10
```