# Spline Modeling

*2018-05-05*

A few spline models (also known as piecewise models). As in previous posts, 'affect' is the name given to values of $y$ throughout.

## 1) Growth and Even More Growth

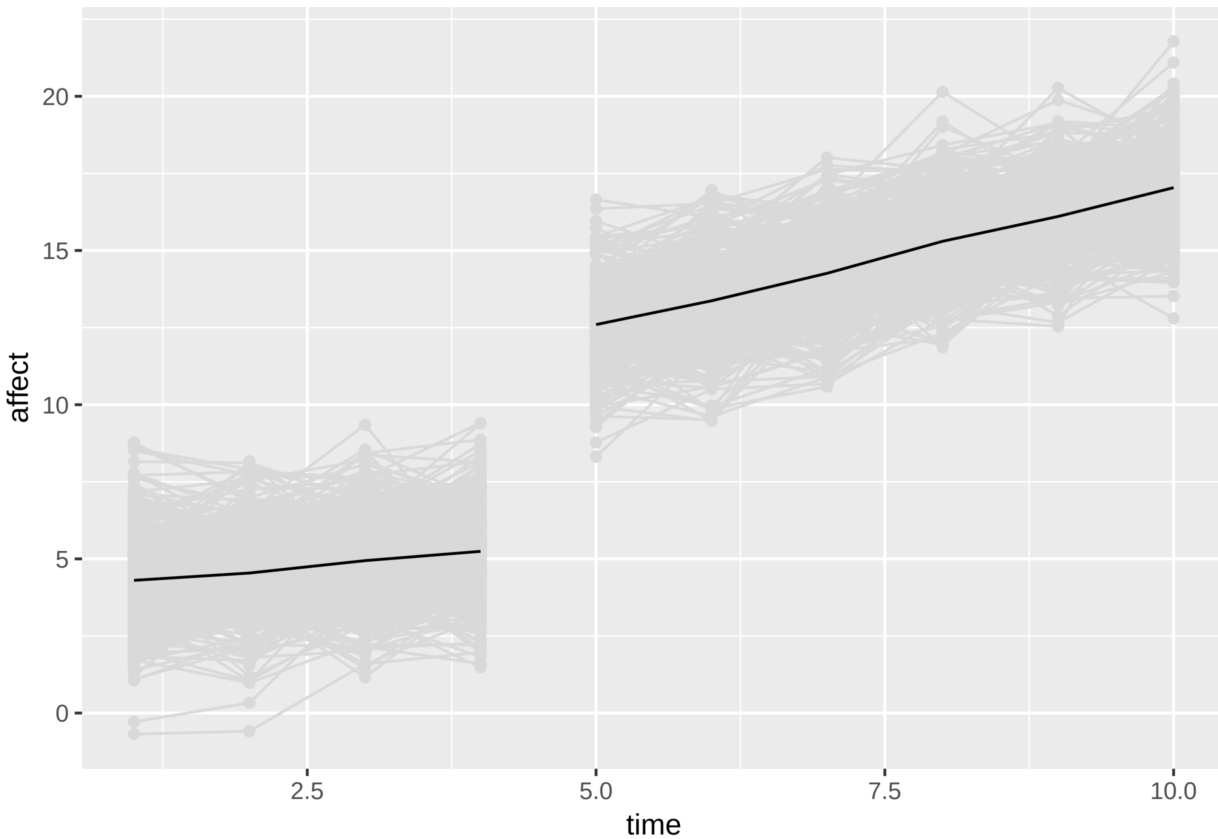
A model that captures a process that increases initially and then increases at an even greater rate once it reaches time point 5. The data generating process:

$$y_{it} = \begin{cases} 4 + 0.3t + error_t, & \text{if time} < 5 \\ 8 + 0.9t + error_t, & \text{otherwise} \end{cases} \tag{1}$$

The data generating code and plot

```
library(tidyverse)
library(lavaan)
library(ggplot2)
library(MASS)

N <- 400
time <- 10

intercept_1 <- 4
intercept_2 <- 8

growth1 <- 0.3
growth2 <- 0.9


df_matrix <- matrix(, ncol = 3, nrow = N*time)


count <- 0

for(i in 1:N){

  unob_het_y <- rnorm(1,0,1)


  for(j in 1:time){

    count <- count + 1

    if(j < 5){
    df_matrix[count, 1] <- i
    df_matrix[count, 2] <- j
    df_matrix[count, 3] <- intercept_1 + growth1*j + unob_het_y + rnorm(1,0,1)
```

```r
    }else{

      df_matrix[count, 1] <- i
      df_matrix[count, 2] <- j
      df_matrix[count, 3] <- intercept_2 + growth2*j + unob_het_y + rnorm(1,0,1)


    }
  }

}

df <- data.frame(df_matrix)

names(df) <- c('id', 'time', 'affect')

df1 <- df %>%
  filter(time < 5)

df2 <- df %>%
  filter(time >= 5)

df_sum1 <- df1 %>%
  group_by(time) %>%
  summarise(
    affect = mean(affect)
  )

df_sum2 <- df2 %>%
  group_by(time) %>%
  summarise(
    affect = mean(affect)
  )

ggplot() +
  geom_point(data = df1, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df1, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_point(data = df2, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df2, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df_sum1, aes(x = time, y = affect)) +
  geom_line(data = df_sum2, aes(x = time, y = affect))
```




Estimating the parameters using SEM:

```
library(lavaan)

df_wide <- reshape(df, idvar = 'id', timevar = 'time', direction = 'wide')


spline_string <- '

# latent intercept for first half

level1_affect =~ 1*affect.1 + 1*affect.2 + 1*affect.3 + 1*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect

# latent intercept for second half

level2_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 1*affect.5 + 1*affect.6 + 1*affect

# latent slope for first half basis coefficients

slope1_affect =~ 1*affect.1 + 2*affect.2 + 3*affect.3 + 4*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect

# latent slope for second half basis coefficients

slope2_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 5*affect.5 + 6*affect.6 + 7*affect

# means and variance of latent factors

level1_affect ~~ level1_affect
```

```
level2_affect ~~ level2_affect
slope1_affect ~~ slope1_affect
slope2_affect ~~ slope2_affect

# covariance between latent factors

level1_affect ~~ level2_affect
level1_affect ~~ slope1_affect
level1_affect ~~ slope2_affect

level2_affect ~~ slope1_affect
level2_affect ~~ slope2_affect

slope1_affect ~~ slope2_affect

# constrain means of indicators to zero across time

affect.1 ~ 0
affect.2 ~ 0
affect.3 ~ 0
affect.4 ~ 0
affect.5 ~ 0
affect.6 ~ 0
affect.7 ~ 0
affect.8 ~ 0
affect.9 ~ 0
affect.10 ~ 0

# constrain residual variance to equality across time

affect.1 ~~ res_var*affect.1
affect.2 ~~ res_var*affect.2
affect.3 ~~ res_var*affect.3
affect.4 ~~ res_var*affect.4
affect.5 ~~ res_var*affect.5
affect.6 ~~ res_var*affect.6
affect.7 ~~ res_var*affect.7
affect.8 ~~ res_var*affect.8
affect.9 ~~ res_var*affect.9
affect.10 ~~ res_var*affect.10

'

spline_model <- growth(spline_string, data = df_wide)
summary(spline_model, fit.measures = T)
```

```
## lavaan 0.6-2 ended normally after 75 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         24
##   Number of equality constraints                     9
##
##   Number of observations                           400
##
```

```
##   Estimator                                         ML
##   Model Fit Test Statistic                      48.906
##   Degrees of freedom                                50
##   P-value (Chi-square)                           0.517
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             1843.619
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.001
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -6184.692
##   Loglikelihood unrestricted model (H1)      -6160.239
##
##   Number of free parameters                         15
##   Akaike (AIC)                               12399.383
##   Bayesian (BIC)                             12459.255
##   Sample-size adjusted Bayesian (BIC)        12411.659
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000
##   90 Percent Confidence Interval          0.000  0.031
##   P-value RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.033
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   level1_affect =~
##     affect.1          1.000
##     affect.2          1.000
##     affect.3          1.000
##     affect.4          1.000
##     affect.5          0.000
##     affect.6          0.000
##     affect.7          0.000
##     affect.8          0.000
##     affect.9          0.000
```

```
##     affect.10         0.000
##   level2_affect =~
##     affect.1          0.000
##     affect.2          0.000
##     affect.3          0.000
##     affect.4          0.000
##     affect.5          1.000
##     affect.6          1.000
##     affect.7          1.000
##     affect.8          1.000
##     affect.9          1.000
##     affect.10         1.000
##   slope1_affect =~
##     affect.1          1.000
##     affect.2          2.000
##     affect.3          3.000
##     affect.4          4.000
##     affect.5          0.000
##     affect.6          0.000
##     affect.7          0.000
##     affect.8          0.000
##     affect.9          0.000
##     affect.10         0.000
##   slope2_affect =~
##     affect.1          0.000
##     affect.2          0.000
##     affect.3          0.000
##     affect.4          0.000
##     affect.5          5.000
##     affect.6          6.000
##     affect.7          7.000
##     affect.8          8.000
##     affect.9          9.000
##     affect.10        10.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   level1_affect ~~
##     level2_affect     1.098    0.183    6.002    0.000
##     slope1_affect    -0.022    0.048   -0.454    0.650
##     slope2_affect    -0.009    0.019   -0.455    0.649
##   level2_affect ~~
##     slope1_affect    -0.004    0.047   -0.094    0.925
##     slope2_affect     0.022    0.035    0.628    0.530
##   slope1_affect ~~
##     slope2_affect    -0.000    0.005   -0.013    0.990
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .affect.1          0.000
##    .affect.2          0.000
##    .affect.3          0.000
##    .affect.4          0.000
##    .affect.5          0.000
```

```
##    .affect.6          0.000
##    .affect.7          0.000
##    .affect.8          0.000
##    .affect.9          0.000
##    .affect.10         0.000
##     level1_affect     3.953    0.083   47.614    0.000
##     level2_affect     8.041    0.105   76.502    0.000
##     slope1_affect     0.322    0.022   14.357    0.000
##     slope2_affect     0.898    0.012   77.345    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     lvl1_ff           1.207    0.200    6.036    0.000
##     lvl2_ff           0.926    0.328    2.821    0.005
##     slp1_ff          -0.005    0.015   -0.341    0.733
##     slp2_ff          -0.005    0.004   -1.213    0.225
##    .affct.1 (rs_v)    1.033    0.030   34.641    0.000
##    .affct.2 (rs_v)    1.033    0.030   34.641    0.000
##    .affct.3 (rs_v)    1.033    0.030   34.641    0.000
##    .affct.4 (rs_v)    1.033    0.030   34.641    0.000
##    .affct.5 (rs_v)    1.033    0.030   34.641    0.000
##    .affct.6 (rs_v)    1.033    0.030   34.641    0.000
##    .affct.7 (rs_v)    1.033    0.030   34.641    0.000
##    .affct.8 (rs_v)    1.033    0.030   34.641    0.000
##    .affct.9 (rs_v)    1.033    0.030   34.641    0.000
##    .affc.10 (rs_v)    1.033    0.030   34.641    0.000
```

The structure of the basis coefficients is the important piece that allows us to capture the change in slope:

```
'
# latent slope for first half basis coefficients

slope1_affect =~ 1*affect.1 + 2*affect.2 + 3*affect.3 + 4*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect

# latent slope for second half basis coefficients

slope2_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 5*affect.5 + 6*affect.6 + 7*affect

'
```

## 2) Growth and Negative Growth

A model that captures a process that goes up and then goes down. The data generating process:

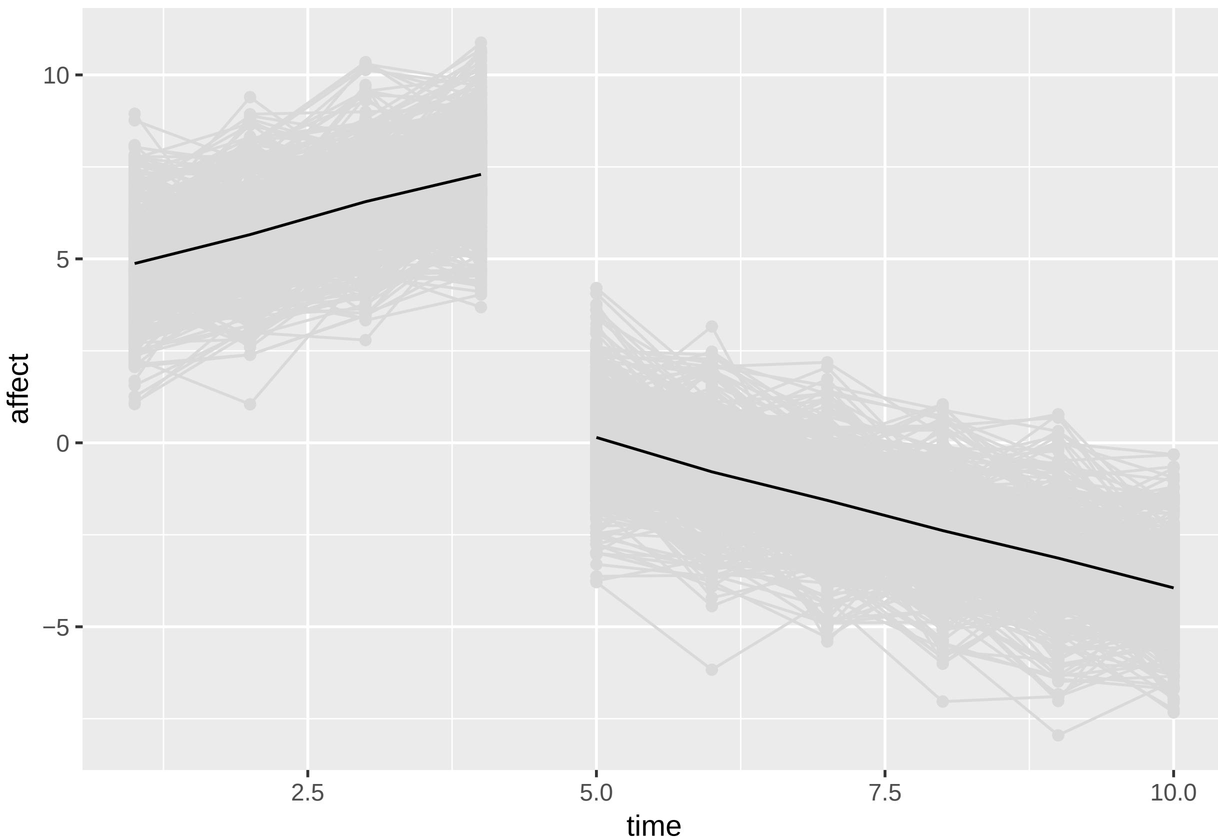
$$y_{it} = \begin{cases} 4 + 0.5t + error_t, & \text{if time} < 5 \\ 4 - 0.5t + error_t, & \text{otherwise} \end{cases} \tag{2}$$

The data generating code and plot

```
library(tidyverse)
library(lavaan)
library(ggplot2)
library(MASS)
```

```
N <- 400
time <- 10

intercept_1 <- 4
intercept_2 <- 4

growth1 <- 0.8
growth2 <- -0.8

df_matrix_b <- matrix(, ncol = 3, nrow = N*time)


count <- 0

for(i in 1:N){

  unob_het_y <- rnorm(1,0,1)


  for(j in 1:time){

    count <- count + 1

    if(j < 5){
      df_matrix_b[count, 1] <- i
      df_matrix_b[count, 2] <- j
      df_matrix_b[count, 3] <- intercept_1 + growth1*j + unob_het_y + rnorm(1,0,1)

    }else{

      df_matrix_b[count, 1] <- i
      df_matrix_b[count, 2] <- j
      df_matrix_b[count, 3] <- intercept_2 + growth2*j + unob_het_y + rnorm(1,0,1)


    }
  }

}

df_b <- data.frame(df_matrix_b)

names(df_b) <- c('id', 'time', 'affect')

df1_b <- df_b %>%
  filter(time < 5)

df2_b <- df_b %>%
  filter(time >= 5)

df_sum1_b <- df1_b %>%
  group_by(time) %>%
  summarise(
```

```r
    affect = mean(affect)
  )

df_sum2_b <- df2_b %>%
  group_by(time) %>%
  summarise(
    affect = mean(affect)
  )

ggplot() +
  geom_point(data = df1_b, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df1_b, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_point(data = df2_b, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df2_b, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df_sum1_b, aes(x = time, y = affect)) +
  geom_line(data = df_sum2_b, aes(x = time, y = affect))
```




Estimating the parameters using SEM:

```r
library(lavaan)


df_wide_b <- reshape(df_b, idvar = 'id', timevar = 'time', direction = 'wide')


spline_string_b <- '
```

```
# latent intercept for first half

level1_affect =~ 1*affect.1 + 1*affect.2 + 1*affect.3 + 1*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect

# latent intercept for second half

level2_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 1*affect.5 + 1*affect.6 + 1*affect

# latent slope for first half basis coefficients

slope1_affect =~ 1*affect.1 + 2*affect.2 + 3*affect.3 + 4*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect

# latent slope for second half basis coefficients

slope2_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 5*affect.5 + 6*affect.6 + 7*affect

# means and variance of latent factors

level1_affect ~~ level1_affect
level2_affect ~~ level2_affect
slope1_affect ~~ slope1_affect
slope2_affect ~~ slope2_affect

# covariance between latent factors

level1_affect ~~ level2_affect
level1_affect ~~ slope1_affect
level1_affect ~~ slope2_affect

level2_affect ~~ slope1_affect
level2_affect ~~ slope2_affect

slope1_affect ~~ slope2_affect

# constrain means of indicators to zero across time

affect.1 ~ 0
affect.2 ~ 0
affect.3 ~ 0
affect.4 ~ 0
affect.5 ~ 0
affect.6 ~ 0
affect.7 ~ 0
affect.8 ~ 0
affect.9 ~ 0
affect.10 ~ 0

# constrain residual variance to equality across time

affect.1 ~~ res_var*affect.1
affect.2 ~~ res_var*affect.2
affect.3 ~~ res_var*affect.3
affect.4 ~~ res_var*affect.4
```

```r
affect.5 ~~ res_var*affect.5
affect.6 ~~ res_var*affect.6
affect.7 ~~ res_var*affect.7
affect.8 ~~ res_var*affect.8
affect.9 ~~ res_var*affect.9
affect.10 ~~ res_var*affect.10

'

spline_model_b <- growth(spline_string_b, data = df_wide_b)
summary(spline_model_b, fit.measures = T)
```

```
## lavaan 0.6-2 ended normally after 92 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         24
##   Number of equality constraints                     9
##
##   Number of observations                           400
##
##   Estimator                                         ML
##   Model Fit Test Statistic                      40.196
##   Degrees of freedom                                50
##   P-value (Chi-square)                           0.838
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             1852.766
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.005
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -6142.519
##   Loglikelihood unrestricted model (H1)      -6122.421
##
##   Number of free parameters                         15
##   Akaike (AIC)                               12315.039
##   Bayesian (BIC)                             12374.911
##   Sample-size adjusted Bayesian (BIC)        12327.315
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000
##   90 Percent Confidence Interval          0.000  0.020
##   P-value RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
```

```
##   SRMR                                           0.030
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   level1_affect =~
##     affect.1          1.000
##     affect.2          1.000
##     affect.3          1.000
##     affect.4          1.000
##     affect.5          0.000
##     affect.6          0.000
##     affect.7          0.000
##     affect.8          0.000
##     affect.9          0.000
##     affect.10         0.000
##   level2_affect =~
##     affect.1          0.000
##     affect.2          0.000
##     affect.3          0.000
##     affect.4          0.000
##     affect.5          1.000
##     affect.6          1.000
##     affect.7          1.000
##     affect.8          1.000
##     affect.9          1.000
##     affect.10         1.000
##   slope1_affect =~
##     affect.1          1.000
##     affect.2          2.000
##     affect.3          3.000
##     affect.4          4.000
##     affect.5          0.000
##     affect.6          0.000
##     affect.7          0.000
##     affect.8          0.000
##     affect.9          0.000
##     affect.10         0.000
##   slope2_affect =~
##     affect.1          0.000
##     affect.2          0.000
##     affect.3          0.000
##     affect.4          0.000
##     affect.5          5.000
##     affect.6          6.000
##     affect.7          7.000
##     affect.8          8.000
##     affect.9          9.000
##     affect.10        10.000
```

```
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   level1_affect ~~
##     level2_affect     1.118    0.172    6.483    0.000
##     slope1_affect     0.038    0.043    0.878    0.380
##     slope2_affect    -0.029    0.018   -1.625    0.104
##   level2_affect ~~
##     slope1_affect    -0.007    0.047   -0.152    0.879
##     slope2_affect    -0.000    0.035   -0.001    0.999
##   slope1_affect ~~
##     slope2_affect     0.005    0.005    1.042    0.297
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .affect.1          0.000
##    .affect.2          0.000
##    .affect.3          0.000
##    .affect.4          0.000
##    .affect.5          0.000
##    .affect.6          0.000
##    .affect.7          0.000
##    .affect.8          0.000
##    .affect.9          0.000
##    .affect.10         0.000
##     level1_affect     4.055    0.077   52.935    0.000
##     level2_affect     4.119    0.106   38.688    0.000
##     slope1_affect     0.816    0.022   37.356    0.000
##     slope2_affect    -0.809    0.012  -69.597    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     lvl1_ff           0.856    0.172    4.991    0.000
##     lvl2_ff           1.172    0.335    3.498    0.000
##     slp1_ff          -0.008    0.015   -0.540    0.589
##     slp2_ff          -0.003    0.004   -0.683    0.494
##    .affct.1 (rs_v)    0.995    0.029   34.641    0.000
##    .affct.2 (rs_v)    0.995    0.029   34.641    0.000
##    .affct.3 (rs_v)    0.995    0.029   34.641    0.000
##    .affct.4 (rs_v)    0.995    0.029   34.641    0.000
##    .affct.5 (rs_v)    0.995    0.029   34.641    0.000
##    .affct.6 (rs_v)    0.995    0.029   34.641    0.000
##    .affct.7 (rs_v)    0.995    0.029   34.641    0.000
##    .affct.8 (rs_v)    0.995    0.029   34.641    0.000
##    .affct.9 (rs_v)    0.995    0.029   34.641    0.000
##    .affc.10 (rs_v)    0.995    0.029   34.641    0.000
```

Notice that the string syntax is the exact same because the process changes at the same point in time, it does not matter if the process changes to 'more positive' or 'more negative.'

## 3) Negative Growth, Growth, and Negative Growth

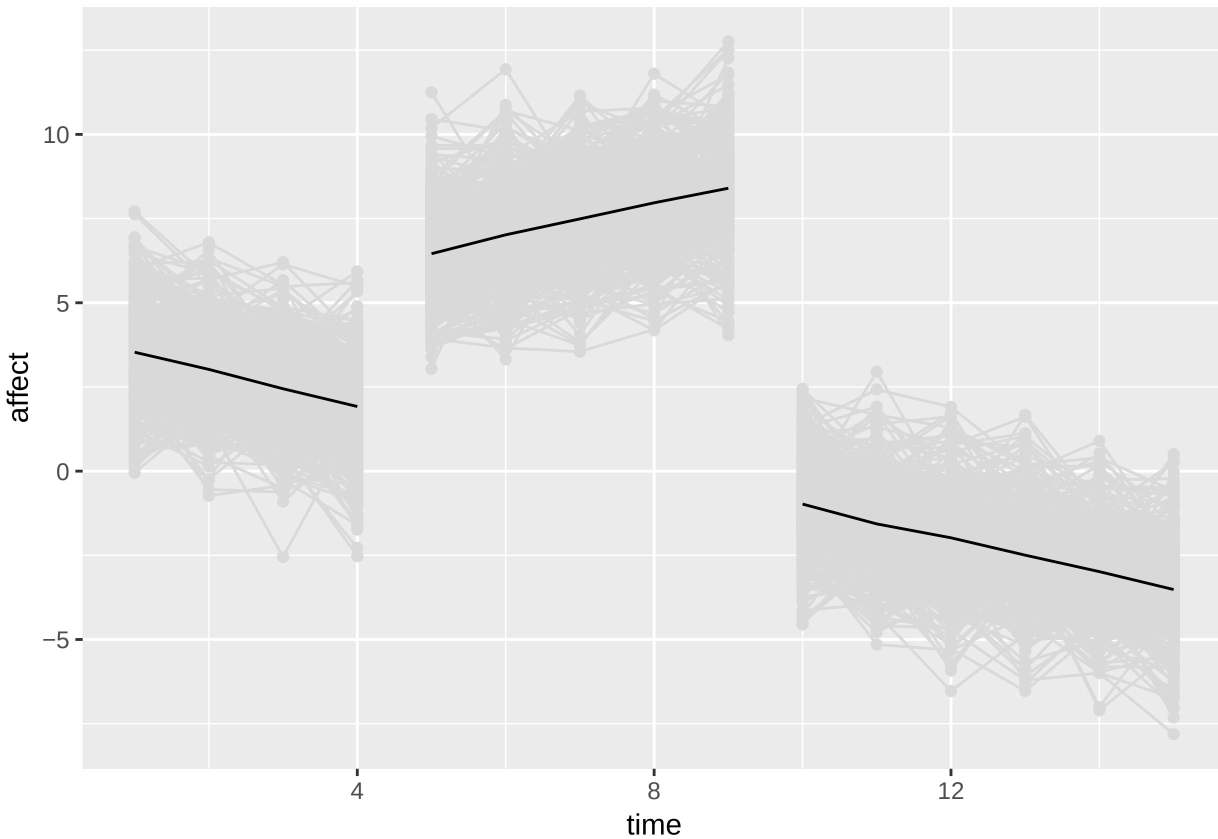
Now a process that goes down, goes up, and then goes back down. The data generating process:

$$y_{it} = \begin{cases} 4 - 0.5t + error_t, & \text{if time} < 5 \\ 4 + 0.5t + error_t, & \text{if } 5 < \text{time} < 10 \\ 4 - 0.5t + error_t, & \text{otherwise} \end{cases} \tag{3}$$

The data generating code and plot

```
library(tidyverse)
library(lavaan)
library(ggplot2)
library(MASS)

N <- 400
time <- 15

intercept_1 <- 4
intercept_2 <- 4
intercept_3 <- 4

growth1 <- -0.5
growth2 <- 0.5
growth3 <- -0.5


df_matrix_c <- matrix(, ncol = 3, nrow = N*time)


count <- 0

for(i in 1:N){

  unob_het_y <- rnorm(1,0,1)


  for(j in 1:time){

    count <- count + 1

    if(j < 5){
      df_matrix_c[count, 1] <- i
      df_matrix_c[count, 2] <- j
      df_matrix_c[count, 3] <- intercept_1 + growth1*j + unob_het_y + rnorm(1,0,1)

    }else if(j >= 5 && j < 10){

      df_matrix_c[count, 1] <- i
      df_matrix_c[count, 2] <- j
      df_matrix_c[count, 3] <- intercept_2 + growth2*j + unob_het_y + rnorm(1,0,1)


    }else{

      df_matrix_c[count, 1] <- i
      df_matrix_c[count, 2] <- j
```

```r
      df_matrix_c[count, 3] <- intercept_3 + growth3*j + unob_het_y + rnorm(1,0,1)

    }
  }

}

df_c <- data.frame(df_matrix_c)

names(df_c) <- c('id', 'time', 'affect')

df1_c <- df_c %>%
  filter(time < 5)

df2_c <- df_c %>%
  filter(time >= 5 & time < 10)

df3_c <- df_c %>%
  filter(time >= 10)

df_sum1_c <- df1_c %>%
  group_by(time) %>%
  summarise(
    affect = mean(affect)
  )

df_sum2_c <- df2_c %>%
  group_by(time) %>%
  summarise(
    affect = mean(affect)
  )

df_sum3_c <- df3_c %>%
  group_by(time) %>%
  summarise(
    affect = mean(affect)
  )

ggplot() +
  geom_point(data = df1_c, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df1_c, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_point(data = df2_c, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df2_c, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df_sum1_c, aes(x = time, y = affect)) +
  geom_line(data = df_sum2_c, aes(x = time, y = affect)) +
  geom_point(data = df3_c, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df3_c, aes(x = time, y = affect, group = id), color = 'gray85') +
  geom_line(data = df_sum3_c, aes(x = time, y = affect))
```




Now estimate the parameters using SEM:

```
library(lavaan)

df_wide_c <- reshape(df_c, idvar = 'id', timevar = 'time', direction = 'wide')


spline_string_c <- '

# latent intercept for first third

level1_affect =~ 1*affect.1 + 1*affect.2 + 1*affect.3 + 1*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect

# latent intercept for second third

level2_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 1*affect.5 + 1*affect.6 + 1*affect

# latent intercept for final third

level3_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect


# latent slope for first third basis coefficients

slope1_affect =~ 1*affect.1 + 2*affect.2 + 3*affect.3 + 4*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect

# latent slope for second third basis coefficients
```

```
slope2_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 5*affect.5 + 6*affect.6 + 7*affect

# latent slope for final third basis coefficients

slope3_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 + 0*affect.5 + 0*affect.6 + 0*affect



# means and variance of latent factors

level1_affect ~~ level1_affect
level2_affect ~~ level2_affect
level3_affect ~~ level3_affect
slope1_affect ~~ slope1_affect
slope2_affect ~~ slope2_affect
slope3_affect ~~ slope3_affect

# covariance between latent factors

level1_affect ~~ level2_affect
level1_affect ~~ level3_affect
level1_affect ~~ slope1_affect
level1_affect ~~ slope2_affect
level1_affect ~~ slope3_affect

level2_affect ~~ level3_affect
level2_affect ~~ slope1_affect
level2_affect ~~ slope2_affect
level2_affect ~~ slope3_affect

level3_affect ~~ slope1_affect
level3_affect ~~ slope2_affect
level3_affect ~~ slope3_affect

slope1_affect ~~ slope2_affect
slope1_affect ~~ slope3_affect

slope2_affect ~~ slope3_affect

# constrain means of indicators to zero across time

affect.1 ~ 0
affect.2 ~ 0
affect.3 ~ 0
affect.4 ~ 0
affect.5 ~ 0
affect.6 ~ 0
affect.7 ~ 0
affect.8 ~ 0
affect.9 ~ 0
affect.10 ~ 0

# constrain residual variance to equality across time
```

```r
affect.1 ~~ res_var*affect.1
affect.2 ~~ res_var*affect.2
affect.3 ~~ res_var*affect.3
affect.4 ~~ res_var*affect.4
affect.5 ~~ res_var*affect.5
affect.6 ~~ res_var*affect.6
affect.7 ~~ res_var*affect.7
affect.8 ~~ res_var*affect.8
affect.9 ~~ res_var*affect.9
affect.10 ~~ res_var*affect.10

'

spline_model_c <- growth(spline_string_c, data = df_wide_c)
summary(spline_model_c, fit.measures = T)
```

```
## lavaan 0.6-2 ended normally after 152 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         42
##   Number of equality constraints                     9
##
##   Number of observations                           400
##
##   Estimator                                         ML
##   Model Fit Test Statistic                     106.253
##   Degrees of freedom                               102
##   P-value (Chi-square)                           0.367
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             3175.539
##   Degrees of freedom                               105
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.999
##   Tucker-Lewis Index (TLI)                       0.999
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -9059.364
##   Loglikelihood unrestricted model (H1)      -9006.237
##
##   Number of free parameters                         33
##   Akaike (AIC)                               18184.727
##   Bayesian (BIC)                             18316.446
##   Sample-size adjusted Bayesian (BIC)        18211.735
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.010
```

```
##   90 Percent Confidence Interval          0.000  0.028
##   P-value RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.038
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   level1_affect =~
##     affect.1          1.000
##     affect.2          1.000
##     affect.3          1.000
##     affect.4          1.000
##     affect.5          0.000
##     affect.6          0.000
##     affect.7          0.000
##     affect.8          0.000
##     affect.9          0.000
##     affect.10         0.000
##     affect.11         0.000
##     affect.12         0.000
##     affect.13         0.000
##     affect.14         0.000
##     affect.15         0.000
##   level2_affect =~
##     affect.1          0.000
##     affect.2          0.000
##     affect.3          0.000
##     affect.4          0.000
##     affect.5          1.000
##     affect.6          1.000
##     affect.7          1.000
##     affect.8          1.000
##     affect.9          1.000
##     affect.10         0.000
##     affect.11         0.000
##     affect.12         0.000
##     affect.13         0.000
##     affect.14         0.000
##     affect.15         0.000
##   level3_affect =~
##     affect.1          0.000
##     affect.2          0.000
##     affect.3          0.000
##     affect.4          0.000
##     affect.5          0.000
##     affect.6          0.000
```

```
##     affect.7          0.000
##     affect.8          0.000
##     affect.9          0.000
##     affect.10         1.000
##     affect.11         1.000
##     affect.12         1.000
##     affect.13         1.000
##     affect.14         1.000
##     affect.15         1.000
##   slope1_affect =~
##     affect.1          1.000
##     affect.2          2.000
##     affect.3          3.000
##     affect.4          4.000
##     affect.5          0.000
##     affect.6          0.000
##     affect.7          0.000
##     affect.8          0.000
##     affect.9          0.000
##     affect.10         0.000
##     affect.11         0.000
##     affect.12         0.000
##     affect.13         0.000
##     affect.14         0.000
##     affect.15         0.000
##   slope2_affect =~
##     affect.1          0.000
##     affect.2          0.000
##     affect.3          0.000
##     affect.4          0.000
##     affect.5          5.000
##     affect.6          6.000
##     affect.7          7.000
##     affect.8          8.000
##     affect.9          9.000
##     affect.10         0.000
##     affect.11         0.000
##     affect.12         0.000
##     affect.13         0.000
##     affect.14         0.000
##     affect.15         0.000
##   slope3_affect =~
##     affect.1          0.000
##     affect.2          0.000
##     affect.3          0.000
##     affect.4          0.000
##     affect.5          0.000
##     affect.6          0.000
##     affect.7          0.000
##     affect.8          0.000
##     affect.9          0.000
##     affect.10        10.000
##     affect.11        11.000
##     affect.12        12.000
```

```
##     affect.13        13.000
##     affect.14        14.000
##     affect.15        15.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   level1_affect ~~
##     level2_affect     0.937    0.198    4.724    0.000
##     level3_affect     0.967    0.252    3.830    0.000
##     slope1_affect     0.020    0.045    0.451    0.652
##     slope2_affect     0.015    0.024    0.599    0.549
##     slope3_affect     0.002    0.018    0.100    0.920
##   level2_affect ~~
##     level3_affect     1.517    0.387    3.924    0.000
##     slope1_affect     0.027    0.053    0.510    0.610
##     slope2_affect     0.059    0.054    1.090    0.276
##     slope3_affect    -0.036    0.028   -1.264    0.206
##   level3_affect ~~
##     slope1_affect     0.024    0.069    0.346    0.729
##     slope2_affect    -0.065    0.048   -1.346    0.178
##     slope3_affect     0.011    0.056    0.194    0.846
##   slope1_affect ~~
##     slope2_affect    -0.005    0.007   -0.785    0.432
##     slope3_affect    -0.002    0.005   -0.457    0.648
##   slope2_affect ~~
##     slope3_affect     0.005    0.004    1.286    0.198
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .affect.1          0.000
##    .affect.2          0.000
##    .affect.3          0.000
##    .affect.4          0.000
##    .affect.5          0.000
##    .affect.6          0.000
##    .affect.7          0.000
##    .affect.8          0.000
##    .affect.9          0.000
##    .affect.10         0.000
##    .affect.11         0.000
##    .affect.12         0.000
##    .affect.13         0.000
##    .affect.14         0.000
##    .affect.15         0.000
##     level1_affect     4.082    0.079   51.441    0.000
##     level2_affect     4.080    0.121   33.596    0.000
##     level3_affect     3.977    0.156   25.473    0.000
##     slope1_affect    -0.541    0.022  -24.643    0.000
##     slope2_affect     0.484    0.015   31.436    0.000
##     slope3_affect    -0.498    0.012  -42.857    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     lvl1_ff           0.967    0.184    5.258    0.000
```

```
##     lvl2_ff           0.624    0.445    1.400    0.162
##     lvl3_ff           0.878    0.751    1.170    0.242
##     slp1_ff          -0.014    0.015   -0.942    0.346
##     slp2_ff          -0.009    0.007   -1.171    0.242
##     slp3_ff          -0.001    0.004   -0.222    0.824
##    .affct.1 (rs_v)    1.034    0.031   33.682    0.000
##    .affct.2 (rs_v)    1.034    0.031   33.682    0.000
##    .affct.3 (rs_v)    1.034    0.031   33.682    0.000
##    .affct.4 (rs_v)    1.034    0.031   33.682    0.000
##    .affct.5 (rs_v)    1.034    0.031   33.682    0.000
##    .affct.6 (rs_v)    1.034    0.031   33.682    0.000
##    .affct.7 (rs_v)    1.034    0.031   33.682    0.000
##    .affct.8 (rs_v)    1.034    0.031   33.682    0.000
##    .affct.9 (rs_v)    1.034    0.031   33.682    0.000
##    .affc.10 (rs_v)    1.034    0.031   33.682    0.000
##    .affc.11           0.987    0.077   12.844    0.000
##    .affc.12           0.984    0.075   13.127    0.000
##    .affc.13           1.068    0.081   13.191    0.000
##    .affc.14           0.874    0.070   12.515    0.000
##    .affc.15           0.919    0.081   11.325    0.000
```

Again, the basis coefficients are the important piece here:

```
'



# latent slope for first third basis coefficients

slope1_affect =~ 1*affect.1 + 2*affect.2 + 3*affect.3 + 4*affect.4 +
                 0*affect.5 + 0*affect.6 + 0*affect.7 + 0*affect.8 +
                 0*affect.9 + 0*affect.10 + 0*affect.11 + 0*affect.12 +
                 0*affect.13 + 0*affect.14 + 0*affect.15

# latent slope for second third basis coefficients

slope2_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 +
                 5*affect.5 + 6*affect.6 + 7*affect.7 + 8*affect.8 +
                 9*affect.9 + 0*affect.10 + 0*affect.11 + 0*affect.12 +
                 0*affect.13 + 0*affect.14 + 0*affect.15

# latent slope for final third basis coefficients

slope3_affect =~ 0*affect.1 + 0*affect.2 + 0*affect.3 + 0*affect.4 +
                 0*affect.5 + 0*affect.6 + 0*affect.7 + 0*affect.8 +
                 0*affect.9 + 10*affect.10 + 11*affect.11 + 12*affect.12 +
                 13*affect.13 + 14*affect.14 + 15*affect.15




'
```

```
## [1] "\n\n\n# latent slope for first third basis coefficients\n\nslope1_affect =~ 1*affect.1 + 2*affe
```

Bo$^2$m =)