ACRM 2023 Longitudinal Data Analysis Workshop

Keith Lohse, PhD, PStat, and Allan Kozlowski, PhD, BsC (PT)

Practical Session 3: Non-Linear Models

Specifically Splines and Exponential Functions

In Practical Session 1, we went through the basics of building mixed-effect models for time-series data and, in Practical Session 2, we explored hypothesis testing to see which factors affected the trajectories we were estimating in part one. Although some of those models were pretty complex, all of those models were **linear** or **curvilinear** (that is, linear in their parameters). In Practical Session 3, we now want to build truly nonlinear models and we will focus on two specific types of models: **simple splines** and a three parameter **negative exponential** function.

As before, you will need to open the five packages we will be using for this session using the library function. We will also be adding the *nlme* package, which we will use to build our negative exponential model.

```
# Loading the essential libraries.
library("ggplot2"); library("lme4");
library("car"); library("dplyr"); library("lmerTest");library("nlme")
```

If you have not already installed these packages, you will need to use the install.packages() function first. This can take some time and will require an internet connection.

```
# If these packages are not installed already, run the following code:
install.packages("ggplot2"); install.packages("lme4");
install.packages("car"); install.packages("dplyr");
install.packages("lmerTest"); install.packages("nlme")
```

Finally, we will import the data and create a new variable called "*year*" to shrink the scale of our time variable (which is currently in months).

```
# We will read these data in from the web:
DATA <- read.csv("https://raw.githubusercontent.com/keithlohse/LMER_Clinical_Science/master/ACRM_2018/da
                 stringsAsFactors = TRUE, na.strings=c("NA","NaN"," ",""))

DATA$year <- DATA$month/12
```

2.1 The One-Knot Spline Model

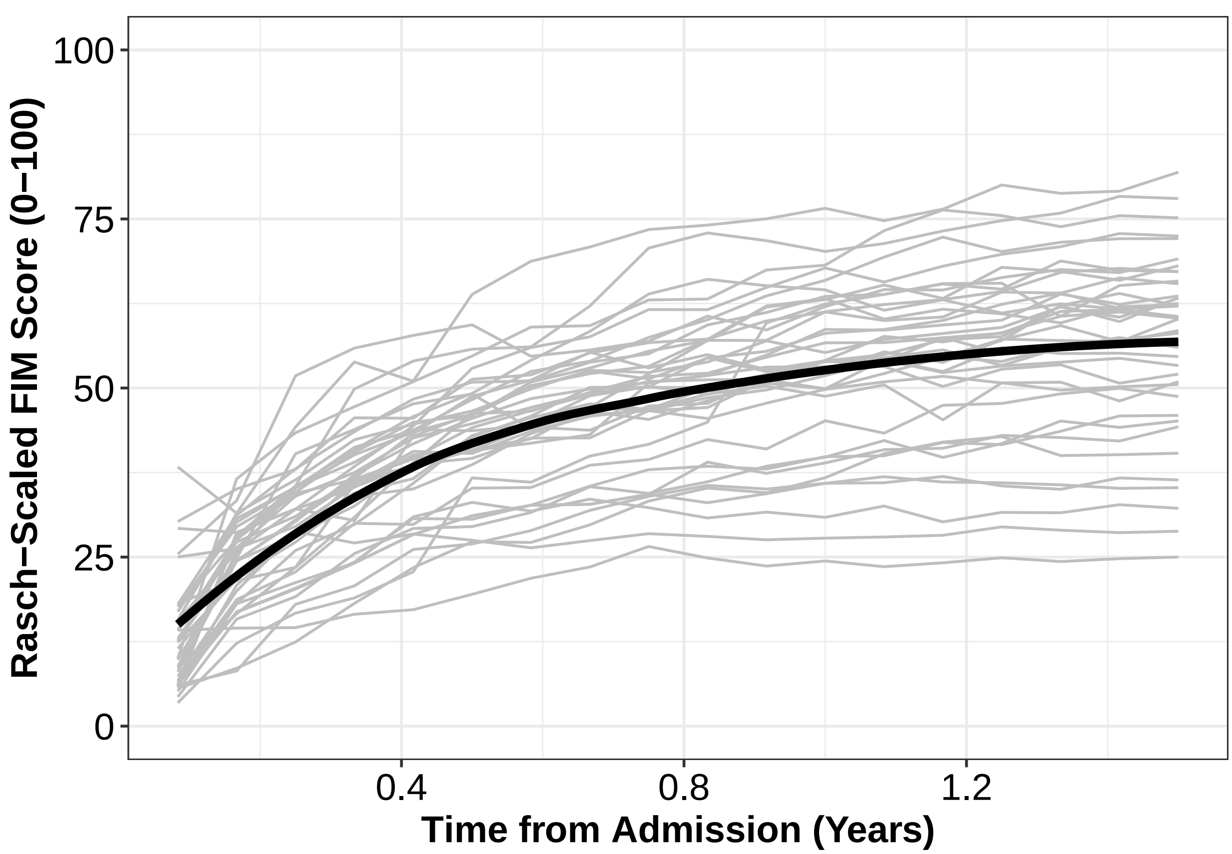
A spline is a specialized mathematical function that is defined piecewise by polynomials.The spline is defined "*piecewise*" because the function changes at specific points along the x-axis. For instance, before $X = 5$ the function might be $y = 1 + 0.5(x)$, however if $x \geq 5$ then the slope changes to $2(x)$. Thus, we have a linear function (first-order polynomial) that changes at discrete intervals along the x-axis (piecewise). Additionally, the point at which the function changes is called a *knot*. You can have different numbers of knots, the knots can be equally or unequally space, and you can have different order polynomials between the different knots. As a starting point, however, we will use a *univariate linear spline* with a *single knot*.

Although not commonly encountered, splines are tremendously useful in modeling longitudinal data. Because you can place the knot if different locations, you can generate complex curves with a relatively small number of parameters. In fact, splines tend to provide a better fit than a polynomial with a comparable number of degrees of freedom. As such, splines are super helpful. The trade-off however, is that splines are often a little harder to interpret relative to polynomials, as we shall see in the details below.

To understand the spline, let's first create a spaghetti plot of the data with a smooth regression line drawn over the top. This smoothed line will help us see how nonlinear the data might be and help to suggest where we might want to place the knot for our spline.

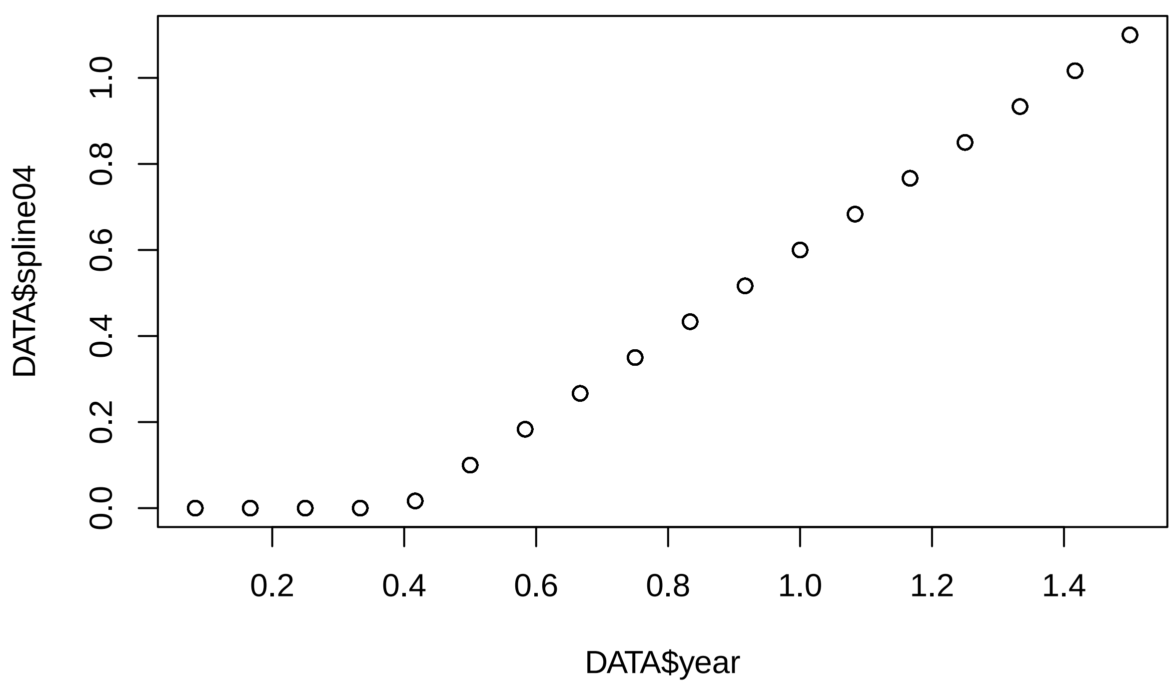
```
# Adding Fixed-Effect to Make Conditional Models ---
ggplot(DATA, aes(x = year, y = rasch_FIM)) +
  geom_line(aes(group=subID), col="grey") +
  stat_smooth(method="loess", col="black", lwd=1.5, se=FALSE)+
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100)) +
  theme_bw() +
  theme(axis.text=element_text(size=14, colour="black"),
        axis.title=element_text(size=14,face="bold")) +
  theme(strip.text.x = element_text(size = 14))+ theme(legend.position="none")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



It looks like there is a change in the rate of change at around 0.4 years. To make the spline function, we will create a new variable called *spline04*. This variable will change piecewise. If the current year is > 0.4 then the spline variable will be $= Year - 0.4$, whereas if the current year is ≤ 0.4 then the value of the spline variable will be zero (as shown in the plot).

```
# Let's create an example spline with a knot at 0.4 years.
DATA$spline04 <- ifelse(DATA$year>0.4, # Test
                        DATA$year-0.4, # Value if True
                        0) # Value if False

plot(y=DATA$spline04, x=DATA$year)
```



Next, we can set up a linear mixed-effects model similar to what we have done before but we will include a linear effect of *year* and our new *spline* variable as fixed-effects. You should also test whether the inclusion of a random-effect of the spline variable is necessary, but in the interest of time, lets move forward with this model. (In practice, if you only have a single knot spline, you probably won't want a random-effect of both time and the spline because those effects will be highly correlated.)

```
# Fitting a spline model
mod01 <- lmer(rasch_FIM~
                   #Fixed Effects
                   1+year+spline04+
                   # Random Effects
                   (1+year|subID), data=DATA, REML=FALSE)

anova(mod01)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## year      18781   18781     1 209.46  1980.4 < 2.2e-16 ***
## spline04  15540   15540     1 640.00  1638.7 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using the *summary()* function, we can see the values for the actual coefficients from the fitted model. Using those values, I will create a new data set that contains the spline models predictions for each year. We can then plot out spline model's predictions over the top of the raw data in the spaghetti plot below.

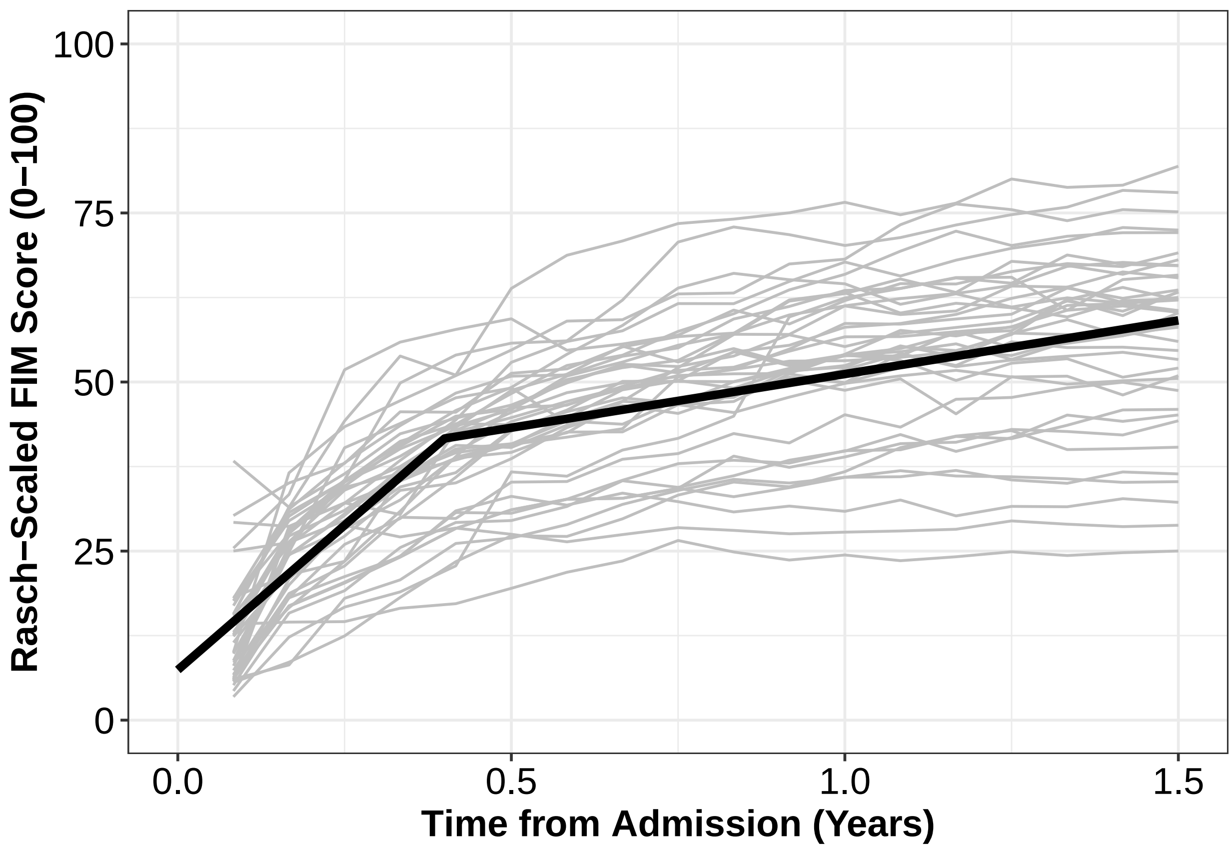
```
MEAN <- data.frame(year=seq(from=0.0, to=1.5, by=0.1))
MEAN$spline04 <- ifelse(MEAN$year>0.4, # Test
                        MEAN$year-0.4, # Value if True
                        0) # Value if False
head(MEAN, 10)
```

```
##    year spline04
## 1   0.0      0.0
## 2   0.1      0.0
## 3   0.2      0.0
## 4   0.3      0.0
## 5   0.4      0.0
## 6   0.5      0.1
## 7   0.6      0.2
## 8   0.7      0.3
## 9   0.8      0.4
## 10  0.9      0.5
```

```
MEAN$rasch_FIM <- 7.428+85.571*MEAN$year-69.697*MEAN$spline04
head(MEAN, 10)
```

```
##    year spline04 rasch_FIM
## 1   0.0      0.0    7.4280
## 2   0.1      0.0   15.9851
## 3   0.2      0.0   24.5422
## 4   0.3      0.0   33.0993
## 5   0.4      0.0   41.6564
## 6   0.5      0.1   43.2438
## 7   0.6      0.2   44.8312
## 8   0.7      0.3   46.4186
## 9   0.8      0.4   48.0060
## 10  0.9      0.5   49.5934
```

```
# Plotting a Basic Spline model ---
ggplot(DATA, aes(x = year, y = rasch_FIM)) +
  geom_line(aes(group=subID), col="grey") +
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100)) +
  theme_bw() +
  theme(axis.text=element_text(size=14, colour="black"),
        axis.title=element_text(size=14,face="bold")) +
  theme(strip.text.x = element_text(size = 14))+ theme(legend.position="none")+
  geom_line(data=MEAN, aes(x=year, y=rasch_FIM), col="black", lwd=1.5)
```



Hopefully that helps to illustrate what the spline function "does" at a conceptual level, but an important question we have left out is where to place the knot for the spline? In the example, we picked 0.4 years because visually that looked like where the rate of change might change. However, I want to emphasize that is **not** is a great way to select your knot location. Ideally, we can place the knot at some theoretically meaningful point (e.g., at the end of an intervention, at discharge from the hospital, or at the point of concussion).

In the absence of such a point, we can for search one empirically, testing many different knots and extracting the AIC for each spline. There are different packages that can determine optimal knot location for you, but one way we can do this with the tools we already have it to write a *for*-loop that will fit a spline with a knot at different locations, save the AIC value for each one, and then determine the best placement for the knot. Because it is easiest to increment the for-loop with integers, I will use the *months* variable for running the loop, then we will convert back to the *year* variable for fitting the actual model.

```
# Write a for-loop to select best fitting spline ----
SPLINES <- data.frame()

# Because the loop works with integers, we will use the months variable rather
# than the year variable to determine our "optimal" knot.

# Let's look at months 2 to 12
2:17
```

```
##  [1]  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
```

```r
length(2:17)
```

```
## [1] 16
```

```r
length(1:16)
```

```
## [1] 16
```

```r
for(i in 1:16){
  print(i)

  # Create the spline
  DATA$spline <- ifelse(DATA$month>i+1, # Test
                        DATA$month-i+1, # Value if True
                        0) # Value if False


  # Define the model
  mod01 <- lmer(rasch_FIM~
                  #Fixed Effects
                  1+month+spline+
                  # Random Effects
                  (1+month|subID), data=DATA, REML=FALSE)


  # Store the AIC, other parameters can be added
  SPLINES[i, 1] <- i+1
  SPLINES[i,2] <- AIC(mod01)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00372151 (tol = 0.002, component 1)

## [1] 7

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00293186 (tol = 0.002, component 1)

## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
```

```
## [1] 13
## [1] 14
## [1] 15
## [1] 16
```

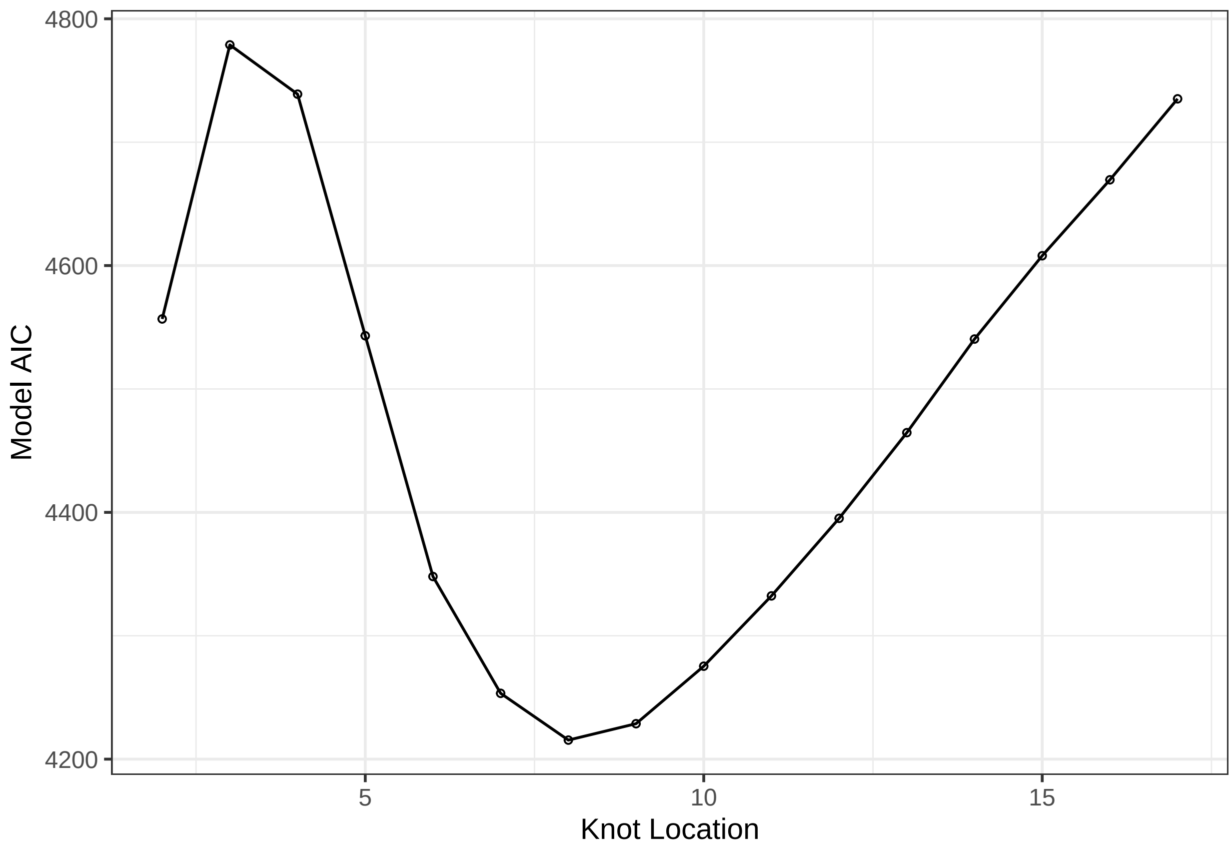
```r
SPLINES
```

```
##    V1       V2
## 1   2 4556.739
## 2   3 4778.857
## 3   4 4738.940
## 4   5 4543.121
## 5   6 4348.024
## 6   7 4253.340
## 7   8 4215.438
## 8   9 4228.731
## 9  10 4275.344
## 10 11 4332.311
## 11 12 4395.170
## 12 13 4464.598
## 13 14 4540.460
## 14 15 4607.977
## 15 16 4669.526
## 16 17 4735.149
```

```r
# Pick the row with lowest AIC
SPLINES[SPLINES$V2==min(SPLINES$V2),]
```

```
##   V1       V2
## 7  8 4215.438
```

```r
# Month 8 appears to be the best knot

# Plot the AIC as a function of knot location
ggplot(data=SPLINES, aes(x=V1, y=V2))+
  geom_line(col="black") +
  geom_point(shape=1, size=1)+
  theme_bw()+
  scale_x_continuous(name="Knot Location") +
  scale_y_continuous(name="Model AIC")
```



As shown in the figure, or more concretely the code to find the row with the *min()* AIC, the best fitting model had the spline placed at month 8. We will convert months back into years ($8/12 = 0.667$), and then fit an **unconditional** model with the best fitting spline.

```
# Modeling the Best Fitting Spline ----
8/12
```

```
## [1] 0.6666667
```

```
DATA$spline66 <- ifelse(DATA$year>(8/12), # Test
                        DATA$year-(8/12), # Value if True
                        0) # Value if False

mod01 <- lmer(rasch_FIM~
                #Fixed Effects
                1+year+spline66+
                # Random Effects
                (1+year|subID), data=DATA, REML=FALSE)

anova(mod01)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## year      15278   15278     1  76.03  1303.7 < 2.2e-16 ***
## spline66  14109   14109     1 640.00  1203.9 < 2.2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To visualize what this model is doing, we can take the fitted values from the model and plot the model predictions over the top of the raw data in a spaghetti plot. The difference between placing the knot at 0.4 years and 0.66 years may not seem like a lot, but if you scroll back and forth between the two figures, the difference is actually quite striking.

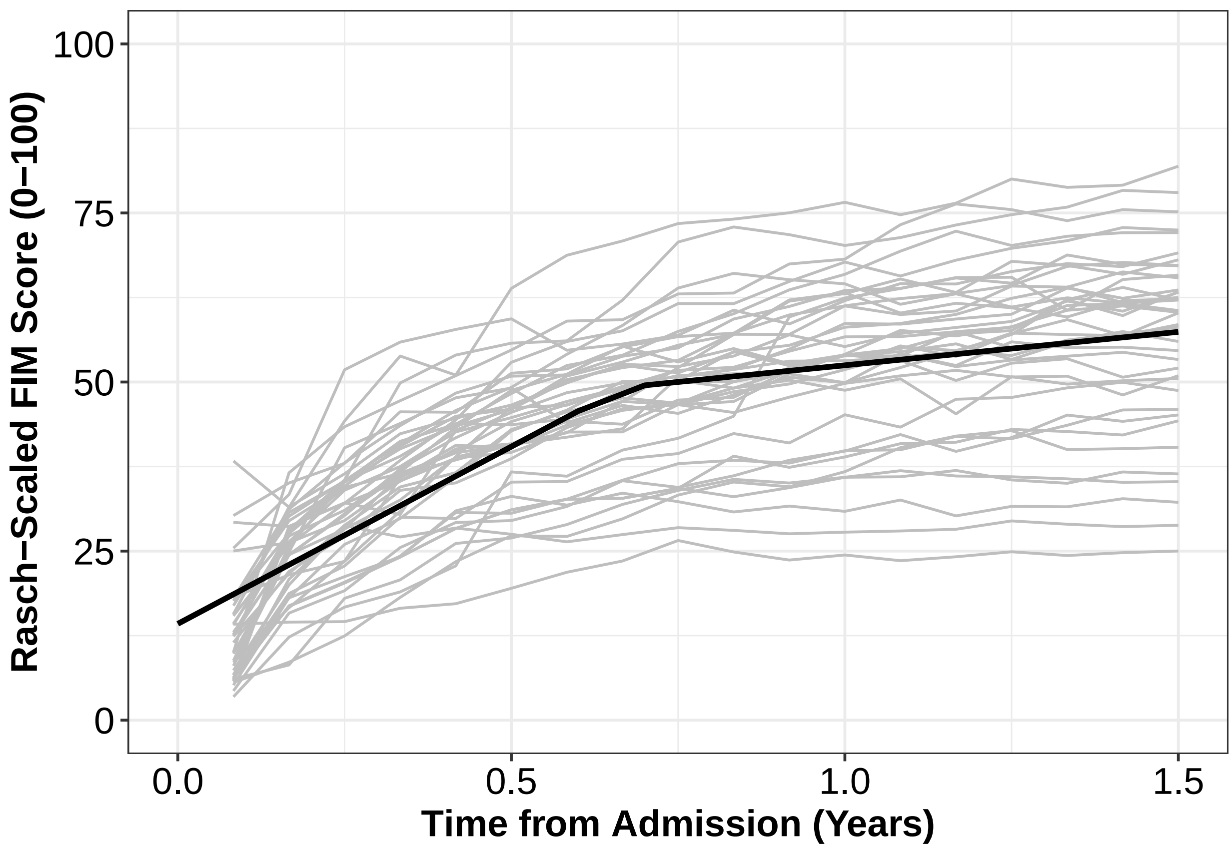
```
MEAN <- data.frame(year=seq(from=0.0, to=1.5, by=0.1))
MEAN$spline66 <- ifelse(MEAN$year>(8/12), # Test
                        MEAN$year-(8/12), # Value if True
                        0) # Value if False
head(MEAN, 10)
```

```
##    year   spline66
## 1   0.0 0.00000000
## 2   0.1 0.00000000
## 3   0.2 0.00000000
## 4   0.3 0.00000000
## 5   0.4 0.00000000
## 6   0.5 0.00000000
## 7   0.6 0.00000000
## 8   0.7 0.03333333
## 9   0.8 0.13333333
## 10  0.9 0.23333333
```

```
MEAN$rasch_FIM <- 14.241+52.413*MEAN$year-42.533*MEAN$spline66
head(MEAN, 10)
```

```
##    year   spline66 rasch_FIM
## 1   0.0 0.00000000  14.24100
## 2   0.1 0.00000000  19.48230
## 3   0.2 0.00000000  24.72360
## 4   0.3 0.00000000  29.96490
## 5   0.4 0.00000000  35.20620
## 6   0.5 0.00000000  40.44750
## 7   0.6 0.00000000  45.68880
## 8   0.7 0.03333333  49.51233
## 9   0.8 0.13333333  50.50033
## 10  0.9 0.23333333  51.48833
```

```
ggplot(DATA, aes(x = year, y = rasch_FIM)) +
  geom_line(aes(group=subID), col="grey") +
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100)) +
  theme_bw() +
  theme(axis.text=element_text(size=14, colour="black"),
        axis.title=element_text(size=14,face="bold")) +
  theme(strip.text.x = element_text(size = 14))+ theme(legend.position="none") +
  geom_line(data=MEAN, aes(x=year, y=rasch_FIM), col="black", lwd=1)
```



Finally, if we want to create a **conditional** spline model, we can add in a fixed-effect of *AIS_grade* and its interactions in precisely the same way that we have done before. We will add the interaction with both the linear effect of time and the spline in one step, then use the *anova()* function to compare the conditional to the unconditional model.

```
# Adding Fixed-Effects of Group ----
mod02 <- lmer(rasch_FIM~
                #Fixed Effects
                1+year*AIS_grade+spline66*AIS_grade+
                # Random Effects
                (1+year|subID), data=DATA, REML=FALSE)

anova(mod01, mod02)
```

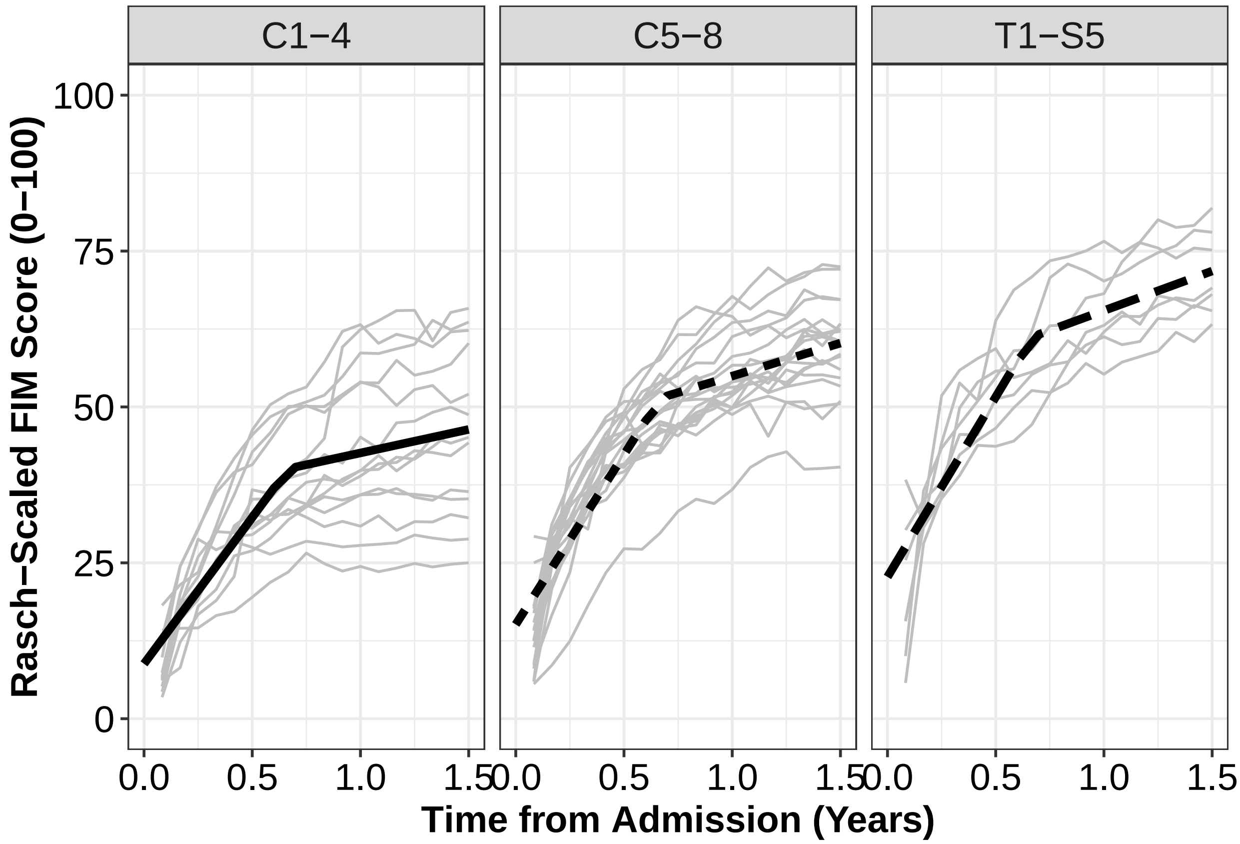
```
## Data: DATA
## Models:
## mod01: rasch_FIM ~ 1 + year + spline66 + (1 + year | subID)
## mod02: rasch_FIM ~ 1 + year * AIS_grade + spline66 * AIS_grade + (1 + year | subID)
##       npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## mod01    7 4119.6 4151.7 -2052.8   4105.6
## mod02   13 4090.3 4149.8 -2032.2   4064.3 41.28  6   2.55e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As you can see, the conditional model is a pretty large improvement beyond the unconditional model, $\Delta AIC > 20$. To understand the model, and especially the spline interactions, we'll extract the fixed-effects to get the model's prediction for each group, collect those predictions in a dataframe, and then plot the model's predictions over the top of the raw data in a spaghetti plot.

```
# Plotting the group level interaction
MEAN <- data.frame(year=rep(seq(from=0.0, to=1.5, by=0.1),3))

MEAN$spline66 <- ifelse(MEAN$year>(8/12), # Test
                        MEAN$year-(8/12), # Value if True
                        0) # Value if False
MEAN$AIS_grade <- factor(c(rep("C1-4", 16),
                           rep("C5-8",16),
                           rep("T1-S5", 16)))

fixef(mod02)
```

```
##          (Intercept)                 year          AIS_gradeC5-8
##             8.805181            46.956590               6.309606
##        AIS_gradeT1-S5              spline66     year:AIS_gradeC5-8
##            13.936953           -39.414498               7.519133
##   year:AIS_gradeT1-S5  AIS_gradeC5-8:spline66 AIS_gradeT1-S5:spline66
##            10.768901            -4.493187              -5.625241
```

```
MEAN$rasch_FIM <- as.numeric(with(MEAN, ifelse(AIS_grade == "C1-4", # Test
    yes= 8.805181+46.956590*year-39.414498*spline66,
    no = ifelse(AIS_grade=="C5-8",
                yes = (8.805181+6.309606)+(46.956590+7.519133)*year+
                  (-39.414498-4.493187)*spline66,
                no = ifelse(AIS_grade=="T1-S5",
                            yes=(8.805181+13.936953)+
                              (46.956590+10.768901)*year+
                              (-39.414498-5.625241)*spline66,
                            no = "ERROR!")
                        ))))

ggplot(DATA, aes(x = year, y = rasch_FIM)) +
  geom_line(aes(group=subID), col="grey") +
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100)) +
  facet_wrap(~AIS_grade) +
  theme_bw() +
  theme(axis.text=element_text(size=14, colour="black"),
        axis.title=element_text(size=14,face="bold")) +
  theme(strip.text.x = element_text(size = 14))+ theme(legend.position="none") +
  geom_line(data=MEAN, aes(x=year, y=rasch_FIM, lty=AIS_grade), col="black", lwd=1.5)
```



To get the model output, we can use the *anova()* function (to see omnibus F- or Chi-squared-tests) and the *summary()* function (to see the fitted values for the individual coefficients).

```
anova(mod02)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                      Sum Sq Mean Sq NumDF  DenDF   F value    Pr(>F)
## year                14419.5 14419.5     1  81.55 1237.3509 < 2.2e-16 ***
## AIS_grade             385.9   192.9     2  51.79   16.5569 2.759e-06 ***
## spline66            12036.1 12036.1     1 640.00 1032.8257 < 2.2e-16 ***
## year:AIS_grade        105.6    52.8     2  81.55    4.5321    0.0136 *
## AIS_grade:spline66     42.1    21.1     2 640.00    1.8073    0.1649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod02)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: rasch_FIM ~ 1 + year * AIS_grade + spline66 * AIS_grade + (1 +
##     year | subID)
##    Data: DATA
##
##      AIC      BIC   logLik deviance df.resid
##   4090.3   4149.8  -2032.2   4064.3      707
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.5627 -0.4945  0.0193  0.5361  3.0797
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  subID    (Intercept) 21.96    4.686
##           year        49.94    7.067    -0.17
##  Residual             11.65    3.414
## Number of obs: 720, groups:  subID, 40
##
## Fixed effects:
##                         Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                8.805      1.419  51.790   6.203 9.31e-08 ***
## year                      46.957      2.341  81.549  20.062  < 2e-16 ***
## AIS_gradeC5-8              6.310      1.871  51.790   3.373  0.00141 **
## AIS_gradeT1-S5            13.937      2.459  51.790   5.669 6.46e-07 ***
## spline66                 -39.414      2.066 640.000 -19.076  < 2e-16 ***
## year:AIS_gradeC5-8         7.519      3.085  81.549   2.438  0.01696 *
## year:AIS_gradeT1-S5       10.769      4.054  81.549   2.656  0.00950 **
## AIS_gradeC5-8:spline66    -4.493      2.723 640.000  -1.650  0.09942 .
## AIS_gradeT1-S5:spline66   -5.625      3.579 640.000  -1.572  0.11648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) year   AIS_gC5-8 AIS_gT1-S5 spln66 y:AIS_C y:AIS_T AIS_C5-8:
## year        -0.373
## AIS_grdC5-8 -0.759  0.283
## AIS_grT1-S5 -0.577  0.215  0.438
## spline66     0.349 -0.551 -0.265    -0.201
## yr:AIS_C5-8  0.283 -0.759 -0.373    -0.163      0.418
## y:AIS_T1-S5  0.215 -0.577 -0.163    -0.373      0.318  0.438
## AIS_C5-8:66 -0.265  0.418  0.349     0.153     -0.759 -0.551  -0.241
## AIS_T1-S5:6 -0.201  0.318  0.153     0.349     -0.577 -0.241  -0.551   0.438
```

2.2 Negative Exponential Model

First, negative Exponential models are hard! Splines are hard too, but if you are anything like me, negative exponential models are intimidating and complicated. Let's take a second to think about the parts of three-parameter negative exponential model and what they mean in general: $y_i = \alpha + (\beta)e^{(\delta * Time_i)}$

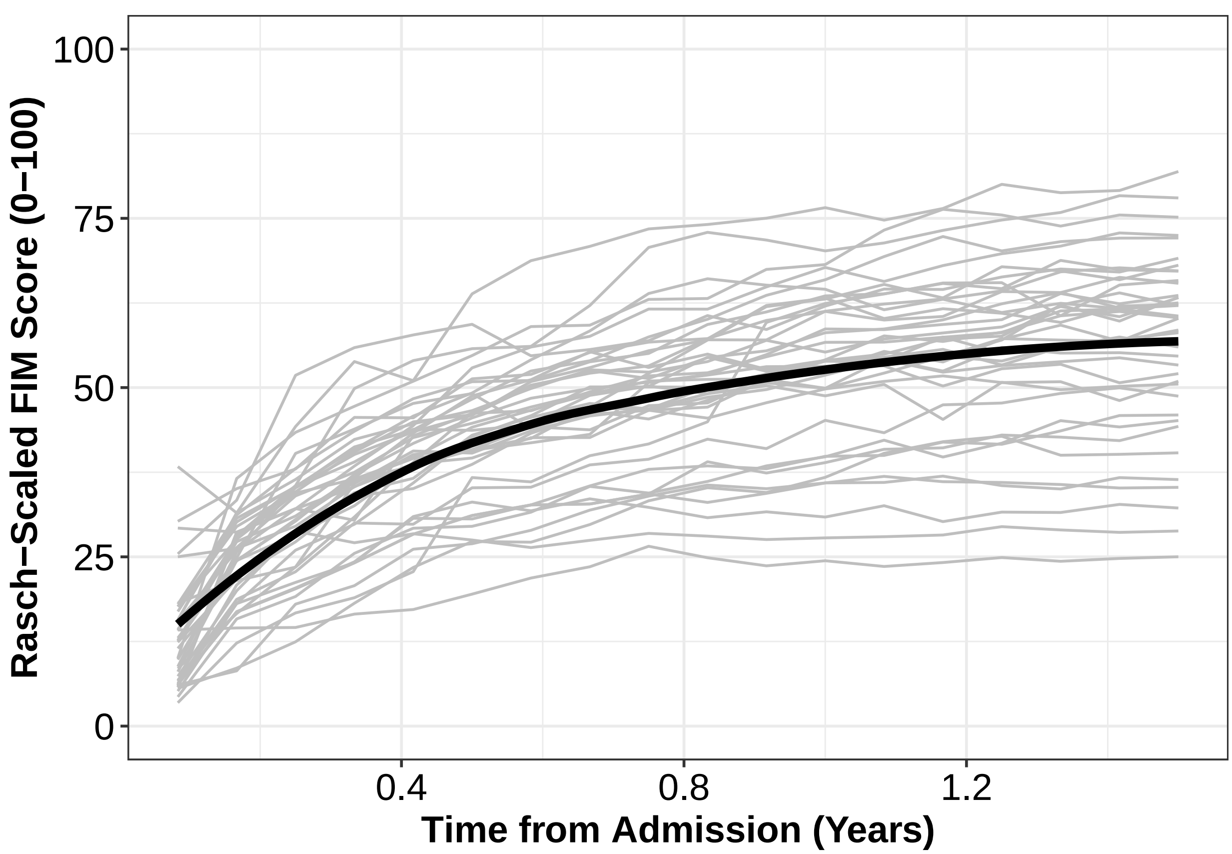
The α parameter defines an asymptote (i.e., the value that a curve will approach, but never reach on the y-axis). The β parameter defines the total change between a psuedo-intercept and the asymptote (i.e., β should capture the total change that person or group demonstrates over time). Finally, the δ parameter reflects the rate of change, the closer to 0 this parameter is the flatter the curve will be (less changing), the farther from 0 the faster the function will approach its asymptote. For instance, think about a situation where $Time = 1$ so that $\delta * Time$ simplifies to δ. You can then solve for $(\beta)e^{\delta}$… that constant will reflect how far a person/group is from their asymptote at $Time = 1$.

It is important to dwell on the meaning of these parameters because when we are estimating nonlinear models, we have an added wrinkle: we need to provide the model with starting values of these parameters. The reason for this is that the mathematics to estimate the best coefficients are very complicated. In order to make the computations feasible, you need to provide the model with reasonable estimates so that it can search the parameter space and converge on a reasonable solution. (This is a huge simplification, but I think it is as simple as possible while still being accurate.) Fortunately, finding reasonable values for α and β is not too difficult because we only need to search for the approximate maximum and minimum values in the data respectively. Estimating δ can be a bit more challenging, but the good news is that values of delta generally tend to be smaller and the algorithms searching for the best value are very efficient, so that even if your initial estimate is off, the model should still converge and find a reasonable δ value for you.

Let's start by plotting the data (ignoring group) to try and then try to fit an unconditional negative exponential model.

```
ggplot(DATA, aes(x = year, y = rasch_FIM)) +
  geom_line(aes(group=subID), col="grey") +
  stat_smooth(method="loess", col="black", lwd=1.5, se=FALSE)+
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100)) +
  theme_bw() +
  theme(axis.text=element_text(size=14, colour="black"),
        axis.title=element_text(size=14,face="bold")) +
  theme(strip.text.x = element_text(size = 14))+ theme(legend.position="none")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Looking at the data, it looks like most of the individual participants max-out around 80 points, and the group level average almost certainly does, so we can choose 80 as a reasonable starting point for our asymptote. Similarly, the minimum value for most participants is around 10, with the group level minimum around 12 or 15, so let's choose 10 as a minimum value, which means that our choice for the β parameter is actually -70 because the psuedo-intercept is $\alpha - \beta = 80 - 70 = 10$. For the starting value of the rate parameter, I will try -1 with the idea being that at Year=1, the group tends to be at about $80 - 70e^{(-1*1)} = 54.25$ points.

After I fit the model using the *nlme()* function, we can use the summary function to see how accurate our predictions were. Hopefully the model converges!

```
set.seed(100)
neg_exp_rand_mod <- nlme(rasch_FIM ~ b_1i +
                              (b_2)*(exp(b_3i * year)),
                         data = DATA,
                         fixed = b_1i + b_2 + b_3i ~ 1,
                         random = list(b_1i ~ 1, b_3i ~1),
                         groups = ~ subID,
                         start = c(80, -70, -1),
                         na.action = na.omit)
summary(neg_exp_rand_mod)
```

```
## Nonlinear mixed-effects model fit by maximum likelihood
##   Model: rasch_FIM ~ b_1i + (b_2) * (exp(b_3i * year))
##   Data: DATA
##        AIC      BIC    logLik
##   4117.913 4149.968 -2051.957
##
## Random effects:
##  Formula: list(b_1i ~ 1, b_3i ~ 1)
##  Level: subID
##  Structure: General positive-definite, Log-Cholesky parametrization
##          StdDev    Corr
## b_1i     10.339912 b_1i
## b_3i      1.325709 0.799
## Residual  3.347909
##
## Fixed effects:  b_1i + b_2 + b_3i ~ 1
##          Value Std.Error  DF   t-value p-value
## b_1i  69.29272 1.7084294 678  40.55931       0
## b_2  -62.37421 0.7609616 678 -81.96762       0
## b_3i  -1.97474 0.2153092 678  -9.17163       0
##  Correlation:
##      b_1i   b_2
## b_2  -0.191
## b_3i  0.768  0.069
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -5.59131619 -0.54091010  0.02240316  0.62022934  2.82591089
##
## Number of Observations: 720
## Number of Groups: 40
```

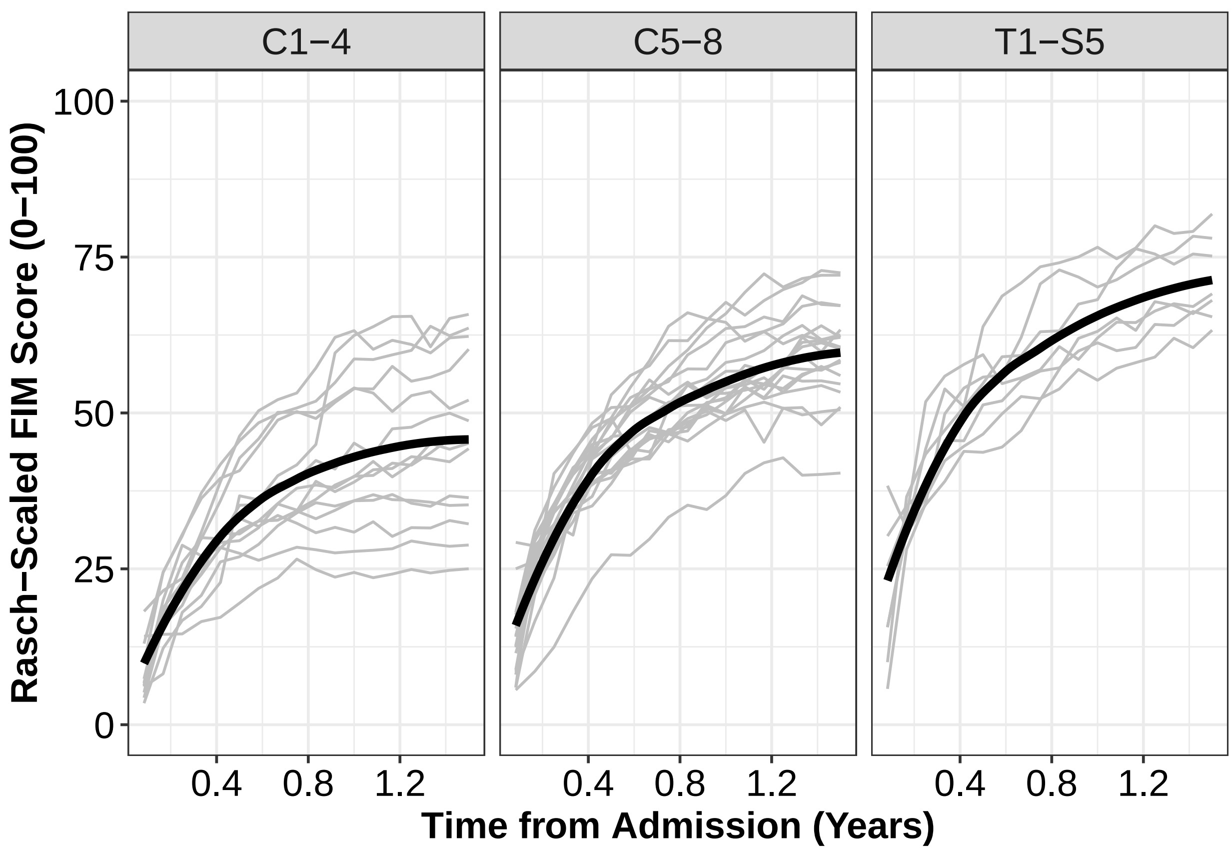
From the *summary()* output it looks like our initial guesses weren't too bad. The α parameter (called b_1i in the output) is 69.19 (we guessed 80), the β parameter (called b_2 in the output) is -62.37 (we guessed -70), and the δ parameter (called b_3i in the output) is -1.97 (we guessed -1). Note that I only put random-effects on the asymptote and the rate parameter. You could put random-effects on the change parameter as well, but in my experience that tends to be correlated with the random-effect of the asymptote. In research, you would determine the best fitting model objectively by trying different random-effects, but I have glossed over that for simplicity to focus on the fixed-effects. As you can see in my code though, I add an "i" sub-script to parameters with random-effects (indicating they vary by subject) and don't include sub-scripts on parameters that exist only as fixed-effects.

Hopefully that gives you a good sense of what the negative exponential function is doing overall. However, our primary interest is going to be how (or if!) these coefficients differ between groups. So, now we are going to be fitting a conditional negative exponential model with a factor of *AIS_grade* that interacts with both the asymptote parameter, the the change parameter, and the rate parameter.

Setting the starting values is going to be tricky in this situation, but again it helps to start with a plot of the data:

```
ggplot(DATA, aes(x = year, y = rasch_FIM)) +
  geom_line(aes(group=subID), col="grey") +
  stat_smooth(method="loess", col="black", lwd=1.5, se=FALSE)+
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100)) +
  facet_wrap(~AIS_grade)+
  theme_bw() +
  theme(axis.text=element_text(size=14, colour="black"),
        axis.title=element_text(size=14,face="bold")) +
  theme(strip.text.x = element_text(size = 14))+ theme(legend.position="none")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Based on these data, I am going to set the starting values to "*start = c(50, 15, 25, -35, -10, -15, -2, 0, 0)*" as shown below. Recall that R will use treatment coding by default and *C1-4* will be the reference group because it is alphabetically first. Thus, 50, -35, and -2 are the values for the asymptote, the change parameter, and the rate parameter in the *C1-4* group respectively. The othervalues then represent the difference relative to the *C1-4* group. That is, the asymptote for the *C5-8* group is about 15 points higher than the *C1-4* group and the asymptote for the *paraplegic (T1-S5)* group is about 25 points higher than the *C1-4* group. This can be a somewhat arduous process, but to choose starting values, you need to estimate the parameters for the reference group and then estimate how *different* those parameters are in the other groups.

```
set.seed(100)
neg_exp_group_mod2 <- nlme(rasch_FIM ~
                              (b_1i) +(b_2)*(exp(b_3i * year)),
                           data = DATA,
                           fixed = list(b_1i ~ 1 + AIS_grade,
                                        b_2 ~ 1 + AIS_grade,
                                        b_3i ~ 1+ AIS_grade),
                           random = list(b_1i ~ 1, b_3i ~1),
                           groups = ~ subID,
                           start = c(50, 15, 25, -35, -10, -15, -2, 0, 0),
                           na.action = na.omit)
```

With a model that successfully converges, we can now use the *anova()* function to obtain omnibus test results and the *summary()* function to obtain fitted values for the individual coefficients.

```
anova(neg_exp_group_mod2)
```

```
##                    numDF denDF  F-value p-value
## b_1i.(Intercept)     1   672 4018.956  <.0001
## b_1i.AIS_grade       2   672   26.755  <.0001
## b_2.(Intercept)      1   672 7221.980  <.0001
## b_2.AIS_grade        2   672   10.737  <.0001
## b_3i.(Intercept)     1   672  132.171  <.0001
## b_3i.AIS_grade       2   672    3.968  0.0194
```

```
summary(neg_exp_group_mod2)
```

```
## Nonlinear mixed-effects model fit by maximum likelihood
##   Model: rasch_FIM ~ (b_1i) + (b_2) * (exp(b_3i * year))
##   Data: DATA
##        AIC      BIC    logLik
##   4061.248 4120.779 -2017.624
##
## Random effects:
##  Formula: list(b_1i ~ 1, b_3i ~ 1)
##  Level: subID
##  Structure: General positive-definite, Log-Cholesky parametrization
##                    StdDev    Corr
## b_1i.(Intercept) 10.208357 b_1.(I
## b_3i.(Intercept)  1.760296 0.864
## Residual           3.283911
##
## Fixed effects:  list(b_1i ~ 1 + AIS_grade, b_2 ~ 1 + AIS_grade, b_3i ~ 1 + AIS_grade)
```

```
##                         Value Std.Error  DF   t-value p-value
## b_1i.(Intercept)     45.33124  2.785318 672  16.27507  0.0000
## b_1i.AIS_gradeC5-8   16.76469  3.687080 672   4.54687  0.0000
## b_1i.AIS_gradeT1-S5  26.24979  4.849582 672   5.41279  0.0000
## b_2.(Intercept)     -57.18711  1.304339 672 -43.84373  0.0000
## b_2.AIS_gradeC5-8    -1.53229  1.609707 672  -0.95191  0.3415
## b_2.AIS_gradeT1-S5   -8.28523  2.099583 672  -3.94613  0.0001
## b_3i.(Intercept)     -4.46463  0.500975 672  -8.91188  0.0000
## b_3i.AIS_gradeC5-8    1.81441  0.651277 672   2.78593  0.0055
## b_3i.AIS_gradeT1-S5   1.38897  0.849066 672   1.63588  0.1023
##  Correlation:
##                     b_1.(I b_1.AIS_C b_1.AIS_T b_2.(I b_2.AIS_C b_2.AIS_T
## b_1i.AIS_gradeC5-8  -0.755
## b_1i.AIS_gradeT1-S5 -0.574  0.434
## b_2.(Intercept)     -0.010  0.008     0.006
## b_2.AIS_gradeC5-8    0.008 -0.029    -0.005    -0.810
## b_2.AIS_gradeT1-S5   0.006 -0.005    -0.038    -0.621  0.503
## b_3i.(Intercept)     0.831 -0.628    -0.477     0.175 -0.142    -0.109
## b_3i.AIS_gradeC5-8  -0.639  0.839     0.367    -0.135  0.141     0.084
## b_3i.AIS_gradeT1-S5 -0.490  0.370     0.842    -0.103  0.084     0.126
##                     b_3.(I b_3.AIS_C
## b_1i.AIS_gradeC5-8
## b_1i.AIS_gradeT1-S5
## b_2.(Intercept)
## b_2.AIS_gradeC5-8
## b_2.AIS_gradeT1-S5
## b_3i.(Intercept)
## b_3i.AIS_gradeC5-8  -0.769
## b_3i.AIS_gradeT1-S5 -0.590  0.454
##
## Standardized Within-Group Residuals:
##          Min           Q1          Med           Q3          Max
## -5.142932509 -0.538440102 -0.001816655  0.506369689  4.884473976
##
## Number of Observations: 720
## Number of Groups: 40
```