Plotting Growth Curves

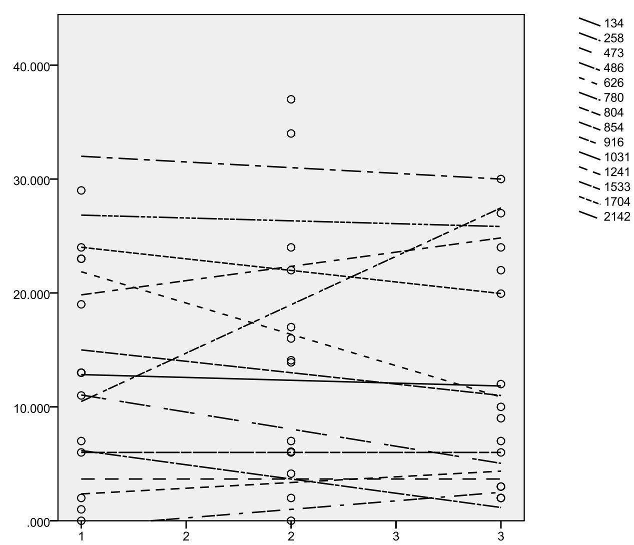
Below is an example of how to plot example growth curves in SPSS using the `GGRAPH` command. Selection of a random sample of cases needs to be done when the data are in a wide (repeated measures) format. If they data are already in wide format, then the first step illustrated below with the `VARSTOCASES` command is not needed. If your data set is in long (person x period) format initially, then you need to convert them first using the `CASESTOVARS` command, then select the 20 cases, and then reconvert data with the `VARSTOCASES` format.[1]

Basic Linear Growth Curves

```
COMPUTE W1=0.
COMPUTE W2=1.
COMPUTE W3=2.

SAMPLE 20 FROM 294.

VARSTOCASES
  /MAKE time "Wave Number" FROM w1 w2 w3
  /MAKE dep "Depression" FROM cesdtot1 cesdtot2 cesdtot3
  /KEEP=RID.

LIST VARS=rid time dep.

GGRAPH
  /GRAPHDATASET NAME="GraphDataset" VARIABLES= time dep rid
  /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
SOURCE: s=userSource( id( "GraphDataset" ) )
DATA: time=col( source(s), name( "time" ) )
DATA: dep=col( source(s), name( "dep" ) )
DATA: rid = col(source(s), name("rid"), unit.category())
ELEMENT: point( position(time * dep))
ELEMENT: line(position(smooth.linear(time * dep)), shape(rid))
END GPL.
```

Output

```
RID     TIME   DEP

 139      .00 12.00
 139     1.00 15.00
 139     2.00 8.000
1241      .00 1.000
1241     1.00 6.072
1241     2.00 3.000
1273      .00 13.00
1273     1.00 12.00
1273     2.00 8.000
1380      .00 15.00
1380     1.00 12.00
1380     2.00 25.00
1958      .00 41.00
1958     1.00 33.00
1958     2.00 38.00

Number of cases read:  15    Number of cases
listed:  15
```



Individual Observed Values By Case

The following syntax generates observed data points using a different version of the `GGRAPH` command, which is useful for examining each case more carefully.[2] I selected only 6 cases at random for this example, and each is given a different color and line style.

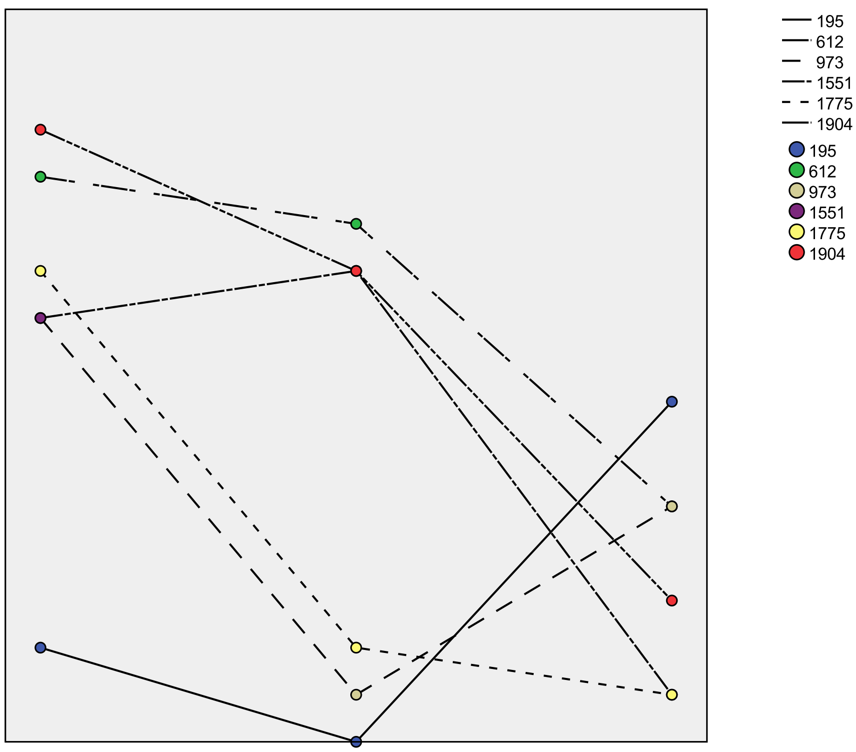
---

[1] See the handout "Plotting Within-Group Regression Lines: SPSS, R, and HLM" for an illustration of the `CASESTOVARS` procedure.
[2] For individual observed points, the older method used the IGRAPH command based on examples by Singer & Willett (2003) to produce a series of separate dot plots for each case:

```
IGRAPH
  /X1=var(time) TYPE=SCALE
```

SPSS Syntax

```
GGRAPH
  /GRAPHDATASET NAME="GraphDataset" VARIABLES= time dep rid
  /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
SOURCE: s=userSource( id( "GraphDataset" ) )
DATA: time=col( source(s), name( "time" ) )
DATA: dep=col( source(s), name( "dep" ) )
DATA: rid = col(source(s), name("rid"), unit.category())
GUIDE: axis(dim(1), ticks(null()))
GUIDE: axis(dim(2), ticks(null()))
ELEMENT: line( position(time*dep), shape(rid))
ELEMENT: point( position(time * dep),color(rid))
END GPL.
```

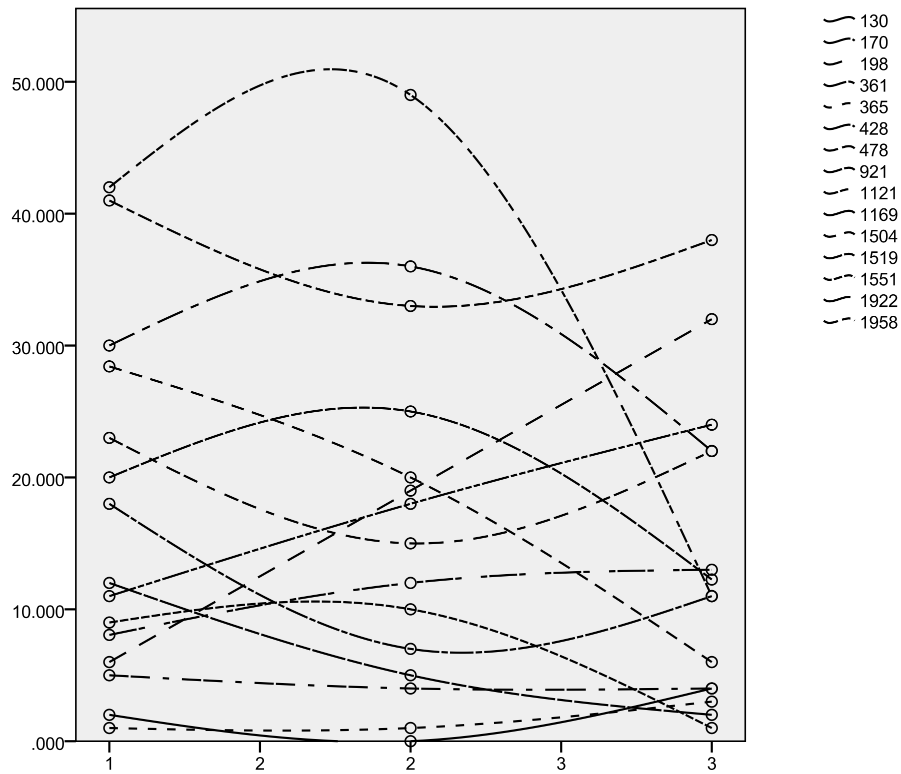


Nonlinear Curves

The following syntax explores curvilinear trajectories using the `GGRAPH` command. I selected only 6 cases at random for this example. Notice that the only differences in the `GGRAPH` command used for the linear plot is the last `ELEMENT` command which uses *`smooth.spline`* instead of *`smooth. linear`*.

```
GGRAPH
  /GRAPHDATASET NAME="GraphDataset" VARIABLES= time dep rid
  /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
SOURCE: s=userSource( id( "GraphDataset" ) )
DATA: time=col( source(s), name( "time" ) )
DATA: dep=col( source(s), name( "dep" ) )
DATA: rid = col(source(s), name("rid"), unit.category())
ELEMENT: point( position(time * dep))
ELEMENT: line(position(smooth.spline(time * dep)), shape(rid))
END GPL.
```

```
  /Y=var(support) TYPE=SCALE
  /STYLE=VAR(w1id)
  /FITLINE METHOD=REGRESSION LINEAR LINE=TOTAL MEFFECT SPIKE=OFF
  /SCALERANGE=VAR(support) MIN=0 MAX=4.
```



Growth Curve Plots in R

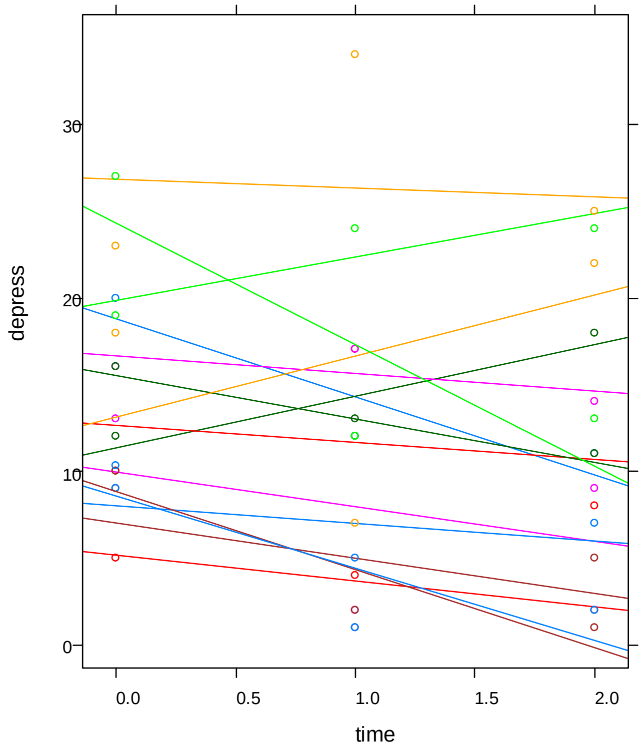
This plotting method gives OLS plots for a sample of cases

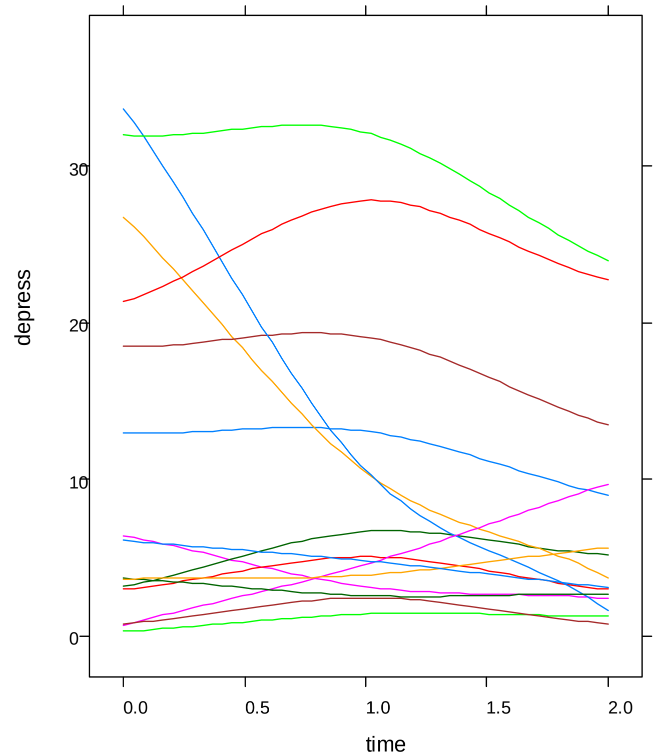
Linear

```
rm(mydata2)
#sample 15 cases
ids <- sample(unique(mydata$rid), 15)
mydata2 <- mydata[mydata$rid %in% ids, ]

#View(mydata2)

## change rid to factor variable for
plotting
mydata2 <- within(mydata2, {
  rid <- factor(rid)
})

library(lattice)
xyplot(depress~time, mydata2, group = rid,
type=c('p','r'), points = TRUE)
```

Quadratic

```
rm(mydata2)
#sample 15 cases
ids <- sample(unique(mydata$rid), 15)
mydata2 <- mydata[mydata$rid %in% ids, ]

#View(mydata2)

## change rid to factor variable for plotting
mydata2 <- within(mydata2, {
  rid <- factor(rid)
})

#prints a loess curve, smoothed to span of 2,
which may need be smaller or larger depending
on the time values

xyplot(depress~time, mydata2, group = rid,
type='smooth',span=2)
```





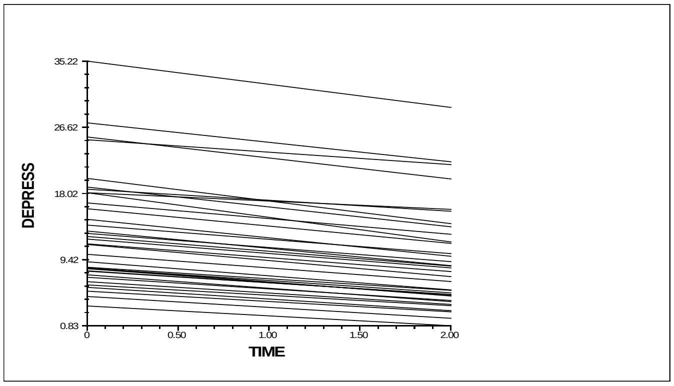
Alternatively, model-based graphs (EB estimates instead of OLS) can be produced with the slightly more complicated code demonstrated in the handout "Plotting Within-Group Regression Lines: SPSS, R, HLM"

Growth Plots in HLM

HLM software has a number of plots preprogrammed.

Linear Growth Lines

Follow these menus: *File → Graph Equations → Level-1 equation graphing*. Then choose TIME as the X-focus and select the number cases to use (I chose a random selection of 15% in this example).



The slopes look more homogeneous because they are model-based lines (i.e., based on empirical Bayes estimates), so there is shrinkage compared to the OLS estimates generated in SPSS due to the low reliability of our slope estimates.

One can also choose to plot the average slope, by following *File → Graph Equations → Model graphs*, and then choosing TIME as the x-focus.

Nonlinear Growth Lines

I generated the following graph of individual quadratic curves after testing the curvilinear model with CTIME and CTIMESQ in the model. Follow *File → Graph Equations → Level-1 equation graphing*. Then choose CTIME as the X-focus and select the number cases to use. Under *Categories/transforms/interactions*, click on a number button, and choose *power or x/z.* Then choose the quadratic variable, CTIMESQ, in the drop down menu on the left. The linear variable, CTIME, will appear in the center box automatically. Enter "2" in the last box on the right, because CTIMESQ is the square of CTIME (i.e., raised to the power of 2).