PersonAlytics© User’s Guide

Stephen Tueller, Ty Ridenour, Derek Ramirez

2019-12-16

Table of Contents

# Introduction

The purpose of this user’s guide is to illustrate how to use PersonAlytics to analyze

1. Single case time series data, also known as single subject or N-of-1 studies. Interrupted time series design designs can be used to introduce one (or more) interventions which results in two (or more) phases (e.g., pre-intervention and post-intervention).
2. Small sample intensive longitudinal design data.
3. Idiographic clinical trial data (ICT), which a single-case or small sample longitudinal data set where all participants experience two or more treatment phases (e.g., baseline and follow-up).

These three types of data are analyzed using longitudinal mixed effects models, also known as hierarchical linear models, multilevel models, latent growth curve models, or mixed method trajectory analysis. There are already several R packages such as nlme and gamlss that can be used to implement these models.

The purpose of the PersonAlytics package is to automate the following aspects of analyzing these types of data:

1. Model selection for detecting the shape of the trajectory over time. Using a good fitting trajectory shape yields parameters that correspond to the shape and improve interpretability. Current options include polynomial growth (e.g., linear, quadratic, cubic, etc.) and piecewise growth (e.g., linear growth within each phase).
2. Model selection for detecting an appropriate time series model for the residuals. Getting a good fitting residual correlation model improves standard error estimation.
3. Automated, paralellized, high throughput analyses of mixed effects models in situations where the combination of predictors, outcomes, and/or individuals (i.e., in the case where individual level models are desired such as in personalized medicine) yields a number of analyses to unwieldy implement manually. This is especially true if the model selection process for the trajectory shape and the residual correlation model is conducted for each combination of predictors, outcomes, and/or individuals. Included in PersonAlytics are options for the user to specify Type I error rate or False Discovery Rate (FDR) corrections.

**Parallelization**. High throughput analyses are achieved through parallelization. Jobs are split among two or more processors on a computer and are run in parallel to each other. Results are recombined at the end of the process.

**High Throughput Example 1: Migraine Triggers**. 346 migraine patients were followed for 90 days. They recorded information on 71 potential migraine and non-migraine headache triggers such as alcohol, weather, and exercise. Individual models were required to determine the five person-specific triggers with the largest effect size. Even though 90 time points is large, it was deemed to small to estimate all 71 trigger effects simultaneously, so trigger effects were estimated one at a time. The analysis required 346 patients X 2 outcomes X 71 triggers = 49,132 PersonAlytic runs.

**High Throughput Example 2: THC Metabolomics**. 17 patients participated in a two-phase design. The baseline phase had 2 hours of observation. The intervention was 25mg of THC and the intervention phase had 6 hours of observation. A total of 20 time points generated blood metabolomic data and outcomes including sleepiness, reaction time, memory, and behavior. The analysis required 18,023 chemical compounds (the metabolites) X 8 outcomes = 144,184 PersonAlytic runs.

# The PersonAlytic framework

Ty to write content here

# Installing PersonAlytics

First the user must install R, available at https://cran.r-project.org/. It is also suggested that the user use a modern code editor such as Rstudio (https://www.rstudio.com/). It is assumed that the reader is familiar with basic R use. If needed, an internet search will provide tutorials to help the user become familiar with R.

Installing PersonAlytics is done by using the install\_github function of the devtools package. First install devtools using

install.packages('devtools')

Then use

devtools::install\_github("https://github.com/ICTatRTI/PersonAlytics")

The install\_github function may give the user the option update other R packages needed by PersonAlytics that the user already have. Unless the user are experienced with R package versions and updates, it is recommended that the user ignore this step and press enter without selecting an option to update packages. If the user does decide to update packages and run into an error, they may need to restart R, manually update the package in the error message, and then rerun the install\_github command.

Once install\_github starts running, it will install all of the other R packages required by PersonAlytics if the user does not already have them.

# Starting PersonAlytics

Now we can load PersonAlytics using

library(PersonAlytics)

# Basic PersonAlytics Use

We will illustrate the basic options of the PersonAlytic() function using the OvaryICT data from modified from the Ovary data in the nlme package to represent the structure of an ICT. The PersonAlytic() function is the primary user interface for the PersonAlytics package. the user can access the documentation for PersonAlytic() by typing

?PersonAlytic

## The OvaryICT Data

The first six rows of the OvaryICT data are shown in Table x where it can be seen that the original Ovary data set has been augmented with Phase variables which are an essential component of an ICT but which are not required by the PersonAlytics package. Note that the data are in ‘long format’. In the example below, Mare 1’s first six time points are represented in separate rows. The time points are days prior to ovulation (for negative values), at ovulation (for values of 0 or 1) or between ovulations (for positive values other than 0 and 1). There are also six randomly generated predictors named Target1 to Target6 which will be used to illustrate other PersonAlytics features.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mare | Time | follicles | Phase | Phase2 | Target1 | Target2 | Target3 | Target4 | Target5 | Target6 |
| 83 | 1 | -0.14 | 9 | 0 | 1 | 2 | 1 | 4 | -0.53 | 0.18 | 0.74 |
| 84 | 1 | -0.09 | 9 | 0 | 1 | 2 | 1 | 2 | -1.37 | 1.34 | 0.63 |
| 85 | 1 | -0.05 | 7 | 0 | 1 | 4 | 4 | 4 | -2.21 | 1.20 | 1.24 |
| 86 | 1 | 0.00 | 6 | 0 | 1 | 1 | 3 | 4 | 1.82 | 0.87 | 0.23 |
| 87 | 1 | 0.05 | 7 | 0 | 1 | 3 | 1 | 4 | -0.65 | -0.12 | -0.31 |
| 88 | 1 | 0.09 | 6 | 0 | 1 | 1 | 2 | 3 | -0.28 | 0.34 | 1.50 |

## Required PersonAlytics Parameters

The four required parameters for PersonAlytic() are

1. data: the name of the user’s data set. This data needs to have been read into R prior to using PersonAlytic(). A web search can be used to learn how to read in multiple types of data into R, including (but not limited to) csv, xlsx, sas, stata, spss, and most database formats. The data must be structured in ‘long format’ where time points are repeated within individual as was illustrated above for the OvaryICT data. In the example below, we use the OvaryICT data set.
2. ids: the name of the identification variable for individuals. This must be a quoted variable name matching the user’s ID variable in the data set that the user provided to the parameter data.
3. dvs: the name of the dependent variable. This must be a quoted variable name matching the user’s dependent variable in the data set that the user provided to the parameter data. If the user has multiple dependent variables, PersonAlytics will iterate over them one at a time. Multiple dependent variables are specified as a character list, e.g., dvs=list('dv1', 'dv2', etc.). This is a high throughput option.
4. time: the name of the time variable. This must be a quoted variable name matching the user’s dependent variable in the data set that the user provided to the parameter data.

**list()**. In R, a list is a collection of variables or R objects that may or may not be of the same type. For example, a list might all be character strings, as is the case with dvs. In other situations, a list may mix data frames, character vectors, or other data types.

Here is an illustration using the four basic parameters. This example also includes setting autoDetect=NULL, which turns off automated model comparisons for selecting the trajectory shape and residual correlation structure until this is discussed in its own section.

eg\_required <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 time="Time",  
 autoDetect=NULL)

It is important to assign the results of a call to PersonAlytic to an R object. In the example above, the object is named eg\_required. If the user neglects to assign the results to an R object, minimal output will be printed to the screen and the remaining results will be lost. If high throughput is initialized by including more than one dependent variable (discussed above), more than one target independent variables (see parameter target\_ivs below), or if individual models are requested (see parameter individual\_mods below), the resulting object is a data.frame concatenating results across of the options submitted to the high throughput run. Reading the output is detailed in the next section.

# Reading the Output

The argument output is character string that will be used to name a file for saving output. If left NULL, the default is ‘PersonAlytic\_Output’. Do not give a file extension, these will be added automatically depending on whether a single analysis was run (yielding a txt file) versus when high throughput options are invoked (yielding a csv file).

## Single Analysis Output

Here we fit the same model using nlme and gamlss and illustrate options for viewing and manipulating the output. If the user is familiar with the lme package, this is an implementation of the linear (normal) mixed effects model. Using the package parameter, the gamlss package can alternatively be used in conjunction with the family parameter for outcomes with non-normal distributions. The family parameter is described in more detail later, the default distribution assumed when using package="gamlss" is the normal distribution.

First, we run the model using package="nlme". Since this is the default, we need to explicitly specify the package parameter. The output will be saved the the R object eg\_nlme for further use inside the R console. In addition, output="nlme\_example" will create a file in the working directory named ‘nlme\_example.txt’. If the user are unsure what the user’s current working directory is, type getwd() into the R console. A full path using forward slashes ‘/’ instead of backslashes ’\' can also be used. For example, output='C:/MyResults'. For convenience, we will also turn off the autoDetect parameter by setting it to NULL. The autoDetect parameter is discussed in detail below.

eg\_nlme <- PersonAlytic(output="nlme\_example",  
 data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 autoDetect=NULL)

Then we run the same model using package="gamlss". The output will be saved the the R object eg\_gamlss for further use inside the R console. In addition, output="gamlss\_example" will create a file in the working directory named ‘gamlss\_example.txt’.

eg\_gamlss <- PersonAlytic(output="gamlss\_example",  
 data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 package="gamlss",  
 autoDetect=NULL)

In these examples, the R object eg\_nlme is a class lme object, and the R object eg\_gamlss is a class gamlss object.

class(eg\_nlme)

## [1] "lme"

class(eg\_gamlss)

## [1] "gamlss" "gam" "glm" "lm"

Both lme and gamlss objects have a summary() method which prints detailed results to the R console. For the nlme example we get the following.

summary(eg\_nlme)

Linear mixed-effects model fit by REML  
 Data: tempData   
 AIC BIC logLik  
 1670.449 1700.185 -827.2244  
  
Random effects:  
 Formula: ~Time | Mare  
 Structure: General positive-definite, Log-Cholesky parametrization  
 StdDev Corr   
(Intercept) 2.751709 (Intr)  
Time 3.871676 -0.202  
Residual 3.317972   
  
Fixed effects: follicles ~ Time \* Phase   
 Value Std.Error DF t-value p-value  
(Intercept) 10.66163 0.9016267 294 11.824883 0.0000  
Time -0.86898 1.7667760 294 -0.491845 0.6232  
Phase 10.97209 1.3126329 294 8.358845 0.0000  
Time:Phase -8.64385 1.9742483 294 -4.378299 0.0000  
 Correlation:   
 (Intr) Time Phase   
Time -0.318   
Phase -0.105 0.135   
Time:Phase 0.175 -0.504 -0.817  
  
Standardized Within-Group Residuals:  
 Min Q1 Med Q3 Max   
-2.49764683 -0.65159342 -0.01330142 0.61976543 2.61092061   
  
Number of Observations: 308  
Number of Groups: 11

The fixed effects results in the section titled ‘Fixed Effects’ is what gets saved to the file ‘nlme\_example.txt’. For the gamlss example we get the following:

summary(eg\_gamlss)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Family: c("NO", "Normal")   
  
Call: gamlss::gamlss(formula = self$formula, sigma.formula = sigma.formula,   
 family = currentFamily, data = tempData, control = ctrl)   
  
Fitting method: RS()   
  
------------------------------------------------------------------  
Mu link function: identity  
Mu Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 10.6898 0.3404 31.402 < 2e-16 \*\*\*  
Time -0.8637 1.2787 -0.675 0.5   
Phase 10.9725 1.2654 8.671 3.28e-16 \*\*\*  
Time:Phase -8.6439 1.9033 -4.541 8.25e-06 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
------------------------------------------------------------------  
Sigma link function: log  
Sigma Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 1.16333 0.04029 28.87 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
------------------------------------------------------------------  
NOTE: Additive smoothing terms exist in the formulas:   
 i) Std. Error for smoothers are for the linear effect only.   
ii) Std. Error for the linear terms may not be reliable.   
------------------------------------------------------------------  
No. of observations in the fit: 308   
Degrees of Freedom for the fit: 23.39381  
 Residual Deg. of Freedom: 284.6062   
 at cycle: 2   
   
Global Deviance: 1590.679   
 AIC: 1637.467   
 SBC: 1724.728   
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The fixed effects results in the section titled ‘Mu Coefficients’ and the variance coefficients in the section titled ‘Sigma Coefficients’ is what gets saved to the file ‘gamlss\_example.txt’.

It is beyond the scope of this document to detail the differences between the nlme and gamlss approaches, but we will highlight one important difference. Notice that the parameter estimates in the ‘Value’ column of the ‘Fixed Effects’ section of the eg\_nlme output and the ‘Estimate’ column of the ‘Mu Coefficients’ section of the eg\_gamlss output are very similar. However, the standard errors are of the gamlss model are lower. This is because the gamlss approach models not only the mean, but also the variance. If there is any heteroscedasticity of variance, the gamlss results will consequently have lower standard errors than the nlme results. In the gamlss output, the variance parameter(s) are in the ‘Sigma Coefficients’ section.

**Heteroscedasticity**. heteroscedasticity occurs in data when the variability of the dependent variable is unequal across the range of values of one (or more) predictors.

## High throughput output

Here we illustrate a high throughput example by first creating two additional outcomes and fitting a basic growth model to each outcome. Note that creating variables that are the square and the root of the outcome is not advised unless the user has substantive reasons to do so, this is simply for illustration purposes.

OvaryICT$follicles2 <- OvaryICT$follicles^2  
OvaryICT$folliclesr <- sqrt(OvaryICT$follicles2)  
eg\_htp <- PersonAlytic(output="htp\_example",  
 data=OvaryICT,  
 ids="Mare",  
 dvs=list("follicles", "follicles2", "folliclesr"),  
 phase="Phase",  
 time="Time",  
 autoDetect=NULL)

##   
##   
## Model fitting starting...

##   
 |   
 |========================== | 33%  
 |   
 |==================================================== | 67%  
 |   
 |==============================================================================| 100%

##   
##   
## Model fitting took:  
## Time difference of 4.660966 secs.

The output will be saved in the R object eg\_htp for further use inside the R console and output="htp\_example" will create a file in the working directory named ‘htp\_example.csv’ that is the same thing as the R object eg\_htp. This csv file can be opened in any spreadsheet program. The rows are all possible combinations of dvs, target\_ivs, and ids (if individual\_mods=TRUE). The column variables are listed below and come in five sets:

1. Variables identify the combination of user inputs that lead to a given row’s analysis. Most column names correspond to their respective parameter name in PersonAlytic. Other variables include information on the version of ‘PersonAlytics’, date, time, and the directory in which the models were run.
   * ‘Mare’: This is is the name of the id variable passed to the ids parameter. In the current example, the value is ‘All Cases’. If individual models are requested by setting individual\_mods=TRUE, this column will give the id for each case in the data set.
   * ‘ids’: This is the name of the ids variable which is the column name of column 1.
   * ‘dv’: The name of the dependent variable.
   * ‘time’: The name of the time variable.
   * ‘ivs’: The names of the indepedent variables (describe below).
   * ‘target\_iv’: The name of a target independent variable (described below).
   * ‘interactions’: The names of the interaction terms (described below).
   * ‘time\_power’: The shape of the trajectories over time (described below).
   * ‘alignPhase’: How time was realigned by phase, if any (described below).
   * ‘correlation’: The residual correlation structure (described below). If the value is NULL, this corresponds to no within-group correlations (see ?lme).
   * ‘family’: The distribution for the dependent variable (described below).
   * ‘standardize’: Which variables were standardized (described below).
   * ‘method’: The estimation method.
   * ‘package’: Which R package was used to fit the models.
   * ‘PersonAlytics’: The version of the PersonAlytics used for the analysis.
   * ‘Date\_Time’: The date and time the model was run.
   * ‘estimator’: What model estimator was used (described below).
   * ‘analyzed\_N’: The number of observations analyzed. This is the sum of time points (per case if multiple cases were analyzed).
   * ‘call’: The model formula created from the user inputs.
   * ‘wasLRTrun’: If ‘target\_ivs’ were provided, a likelihood ratio test (LRT) for models with and without the target independent variable will be attempted, and if succesfull, ‘wasLRTrun’ will be ‘TRUE’.
   * ‘targ\_ivs\_lrt\_pvalue’: If the LRT was run, the p-value is recorded here.
   * ‘fixed’: The fixed effects portion of the ‘call’.
   * ‘random’: The random effects portion of the ‘call’.
   * ‘formula’: This is similar to the ‘call’ variable which gives the intended formula. If the model will not converge using the intended formula, simplifications of the formula are attempted in the following order:
     + No correlation structure
     + No correlation structure and no random slopes
     + If the model is piecewise, drop the phase by time interaction
   * ‘correlation0’: The correlation portion of the ‘call’.
   * ‘directory’: The directory where model output is saved.
   * ‘date’: The date the output was saved (which may be different from the ‘Date\_Time’ model was run for long runs).
2. Variables describing whether the analysis converged and variables to help diagnose a failed run (e.g., zero variance in an outcome or independent variable).
   * ‘N\_participants’: The number of unique cases in the ids variable.
   * ‘N\_time\_points’: The number of unique time points.
   * ‘Nobs’: The number of individuals across all unique time points. If the time points are all unique, this number will be the same as ‘N\_time\_points’. If cases share time points, ‘Nobs’ will be smaller than ‘N\_time\_points’.
   * ‘dvVar’: The variance of the dependent variable. If this is zero, the model will not converge.
   * ‘timeVar’: A check whether the time variable is monotonically increasing. If it is not, the PersonAlytic function will stop with an error.
   * ‘ivVar’: The variance of the independent variables (if any are included in the model).
   * ‘target\_ivVar’: The variance of the target independent variable (if one is included in the model).
   * ‘converge’: Model convergence status.
3. Descriptive statistics in pairs with the first column describing the statistic with the prefix statName and the second column in each pair with the prefix statValue giving the statistic’s value.
4. Model results with a parameter estimates, standard error, t-value, degrees of freedom, and p-value.
5. If the finite population correction (FPC) is specified (see below for details), the model results repeated with FPCs for the standard errors (and consequently, the p-values). The example below doesn’t include the FPC, but if it did the parameter estimates, standard errors, t-values, degrees of freedom, and p-values would be repeated with the suffix fpc.

[1] "Mare" "ids" "dv"   
 [4] "time" "phase" "ivs"   
 [7] "target\_iv" "interactions" "time\_power"   
[10] "alignPhase" "correlation" "family"   
[13] "gamlss.family" "standardize" "method"   
[16] "package" "Personalytics" "Date\_Time"   
[19] "N\_participants" "N\_time\_points" "N\_time\_points\_complete"  
[22] "dvVariance" "timeVariance" "ivVariance"   
[25] "target\_ivVariance" "converge" "estimator"   
[28] "analyzed\_N" "call" "wasLRTrun"   
[31] "targ\_ivs\_lrt\_pvalue" "fixed" "random"   
[34] "formula" "correlation0" "directory"   
[37] "date" "statName1" "statValue1"   
[40] "statName2" "statValue2" "statName3"   
[43] "statValue3" "X.Intercept..Value" "X.Intercept..Std.Error"  
[46] "X.Intercept..DF" "X.Intercept..t.value" "X.Intercept..p.value"   
[49] "Time.Value" "Time.Std.Error" "Time.DF"   
[52] "Time.t.value" "Time.p.value" "Phase.Value"   
[55] "Phase.Std.Error" "Phase.DF" "Phase.t.value"   
[58] "Phase.p.value" "Time.Phase.Value" "Time.Phase.Std.Error"   
[61] "Time.Phase.DF" "Time.Phase.t.value" "Time.Phase.p.value"

# Autodetection of the Residual Autocorrelation Structure, Time Order, and Distribution

The autodetection options automate the tedious process of conducting ML model comparisons to determine any of three options described in this section. The value NULL (autoDetect=NULL), or an empty list (autoDetect=list()) turns all options off. Leaving any option out of the list will turn that option off (e.g., autoDetect=list(TO=list(polyMax=3)) will only implement autodetection of the time order). The default is autoDetect=list(AR = list(P = 3, Q = 3), TO = list(polyMax = 3), DIST = list()). As noted above, a list in R is a collection of objects (or variables) that need not be of the same type. In this example,

* autoDetect is a list that has length 3.
* The first object in autoDetect is another list, AR = list(P = 3, Q = 3).
  + AR has length 2
  + The first value is named P and has a value of 3.
  + The second value is named Q and also have a value of 3.
* The second object in autoDetect is also a list, TO = list(polyMax = 3). In this example TO has length 1 containing a value named polyMax with a value of 3.
* The final object in autoDetect is a list DIST = list(). DIST has length 0, i.e., it is an empty list.

This preceding discussion is meant to familiarize new R users with lists, and AR, TO, and DIST are explained in more detail in the remaind of this section.

## Residual Correlation Structure AR

AR is the autoregressive moving-average (ARMA) order of the residual correlation structure. Determining a good fitting correlation structure will provide more accurate standard errors. The default, AR=list(P=3, Q=3) will search all combinations of p=c(0,1,2,3) and q=c(0,1,2,3), as well as the default with no within-group correlations (correlation=NULL), for the best fitting ARMA correlation structure. This is done with a fit index instead of ML likelihood ratio tests (LRT) because not all ARMA models are nested (a requirement of the LRT). The fit index that will be used is set using the whichIC parameter with options BIC or AIC. It is beyond the scope of this document to discuss the LRT or choosing between BIC and AIC, and users should leave the default as BIC untill they have familiarized themselves with the differences between them. If or individual\_mods=TRUE, correlation model selection is implemented using the auto.arima function of the forecast package. See ?auto.arima.

Here is an example of using the AR option of the autoDetect parameter without the TO or DIST options:

eg\_autoDetect\_AR <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 method="ML",  
 package="gamlss",  
 autoDetect = list(AR=list(P=3, Q=2)))

##   
##   
## PersonAlytics: Automatic detection of the  
## residual correlation structure starting...

##   
 |   
 |========================== | 33%  
 |   
 |==================================================== | 67%  
 |   
 |==============================================================================| 100%

##   
## Automatic detection of the residual  
## correlation structure took: Time difference of 10.14193 secs.

##   
## The best correlation structure among those tested is nlme::corARMA(p=0,q=2)

## The Trajectory Shape TO

This parameter sets the shape of the trajectory using the polynomial order of the time/outcome relationship. Determining a good shape for the trajectory over time helps with model interpretation. Automatic detection is implemented using likelihood ratio test of the model with vs. where is the polynomial order of the time variable. The default is TO=list(polyMax=3), where polyMax is the largest values of to test. An alternative to using a polynomial order for time is to approximate the trajectory shape using a separate linear model within each of the phases. This can be set using the piecewise option of the alignPhase parameter as described below. As noted above, time\_power is used when the user wants to specify a value for , and doing so requires setting TO=list() or leaving it out of the autoDetect parameter.

Here is an example of using the TO option of the autoDetect parameter without the AR or DIST options:

eg\_autoDetect\_TO <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 method="ML",  
 package="gamlss",  
 autoDetect = list(TO=list(polyMax=4)))

##   
##   
## PersonAlytics: Automatic detection of the  
## time/outcome relationship starting...

##   
## Automatic detection of the time/outcome  
## relationship took: Time difference of 36.50381 secs.  
##   
## The best polynomial order for time from 1 to 4 was 3

## Dependent Variable Distributional Assumption DIST

Autodetection of the best fitting distribution of the dependent variable using the fitDist function of the gamlss package. The default is DIST=list(), which initializes this option. To turn off autodetection of the best fitting distribution, remove DIST from the autoDetect list. To see all of the available options, type

?gamlss.family

into the console. Here is an example of using the DIST option of the autoDetect parameter without the TO or AR options:

eg\_autoDetect\_DIST <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 package="gamlss",  
 autoDetect = list(DIST=list()))

##   
## The variable follicles has the following characteristics:   
## Integer : TRUE   
## Binary : FALSE   
## Proportion : FALSE   
## Positive : TRUE   
## Multinomial : FALSE (integer with >2 & <= 5 categories)   
## Count : TRUE (positive integer)   
## Continuous : FALSE (non-integer)   
##   
## Error in solve.default(oout$hessian) :   
## Lapack routine dgesv: system is exactly singular: U[4,4] = 0  
## Error in solve.default(oout$hessian) :   
## Lapack routine dgesv: system is exactly singular: U[3,3] = 0  
## Error in solve.default(oout$hessian) :   
## Lapack routine dgesv: system is exactly singular: U[4,4] = 0  
## Error in solve.default(oout$hessian) :   
## Lapack routine dgesv: system is exactly singular: U[4,4] = 0  
##   
## Family: c("WEI2", "Weibull type 2")   
## Fitting method: "nlminb"   
##   
## Call: gamlssML(formula = y, family = DIST[i], data = sys.parent())   
##   
## Mu Coefficients:  
## [1] -6.686  
## Sigma Coefficients:  
## [1] 0.9418  
##   
## Degrees of Freedom for the fit: 2 Residual Deg. of Freedom 306   
## Global Deviance: 1856.73   
## AIC: 1860.73   
## SBC: 1868.19

##   
## To explore this distribution install `gamlss.demo` and type  
##   
## dev.new()  
## gamlss.demo::demoDist()  
##   
## into the console, find your distribution, and use the  
## slider bars to select the parameters printed above  
## (mu, sigma, nu, and tau).

##   
## Descriptive statistics by phase:  
## mean median sd skewness kurtosis  
## 0 10.52 10 4.07 0.23 2.92  
## 1 13.65 13 5.47 0.12 2.00  
## OverAll 12.04 12 5.04 0.38 2.54

When running this code, the following is printed to the screen

The variable follicles has the following characteristics:   
Integer : TRUE   
Binary : FALSE   
Proportion : FALSE   
Positive : TRUE   
Multinomial : FALSE (integer with >2 & <= 5 categories)   
Count : TRUE (positive integer)   
Continuous : FALSE (non-integer)

The best fitting distribution is the “Weibull type 2” distribution. The resulting object eg\_autoDetect\_DIST is a gamlss object fit using the Weibull type 2 distribution.

# High Throughput Options

## Multiple Dependent Variables

## Independent Variables and Target Independent Variables

Independent, or predictor, variables can be added using the ivs parameter. This might be a grouping variables (e.g., treatment/control) or demographic variable (e.g., age or sex). If there are more than one dependent (or outcome) variables in dvs, the variables in ivs will be included as predictors for each dependent variable.

eg\_ivs <- PersonAlytic(output="MyResults",  
 data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 ivs=list("Target1", "Target2", "Target3"),  
 autoDetect=NULL)

If the user wishes to iterate over multiple independent variables one at a time, use target\_ivs. This is a high throughput option.

eg\_target\_ivs <- PersonAlytic(output="MyResults",  
 data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 target\_ivs=list("Target4", "Target5", "Target6"),  
 autoDetect=NULL)

##   
##   
## Fitting models of the dependent variable `follicles` for 11 cases in `Mare`  
## and for 3 target indepented variables in `target\_ivs`.

##   
 |   
 |========================== | 33%  
 |   
 |==================================================== | 67%  
 |   
 |==============================================================================| 100%

##   
##   
## Model fitting for the dependent variable `follicles` took:  
## Time difference of 4.649978 secs.

If the user has some independent variables that should be in every model and some that should be added one at a time, both ivs and target\_ivs can be used. Here, Target1-3 will be included in every model, but Target4-6 will be added one at a time.

eg\_target\_ivs2 <- PersonAlytic(output="MyResults",  
 data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 ivs=list("Target1", "Target2", "Target3"),  
 target\_ivs=list("Target4", "Target5", "Target6"),  
 autoDetect=NULL)

##   
##   
## Fitting models of the dependent variable `follicles` for 11 cases in `Mare`  
## and for 3 target indepented variables in `target\_ivs`.

##   
 |   
 |========================== | 33%  
 |   
 |==================================================== | 67%  
 |   
 |==============================================================================| 100%

##   
##   
## Model fitting for the dependent variable `follicles` took:  
## Time difference of 4.207966 secs.

If the user wants variables to be dummy coded, the user should first specify this in the data set. For example, this code specifies that the Target1 variable should be treated as a categorical variable and dummy coded in any future analyses:

OvaryICT$Target1 <- factor(OvaryICT$Target1)

If the user is unfamiliar with factors in R, they should read the documentation by typing ?factor in the console or conducting an internet search for tutorials in factors in R.

The interactions parameter can be used to specify pairs of independent variables to be included in interaction terms. Main effects for all variables in the interaction terms will be estimated even if they are not listed in the ivs or target\_ivs parameter as shown in this example.

eg\_interactions <- PersonAlytic(output="MyResults",  
 data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 interactions=list(c("Target1", "Target2"), c("Target1", "Target3"),   
 c("Target2", "Target3")),  
 autoDetect=NULL)

## Individual Level Models

PersonAlytics can automate the task of fitting individual models to each case in the data set. This is useful for applications such as individualized medicine. This is enabled by simply setting individual\_mods=TRUE. The resulting output will be the data.frame saved to the assigned object in your R sessios. This will also be saved to a csv file named using the output parameter. In each of these two equivalent forms of output, each row will correspond to a separate participant.

## Combinations of dvs, target\_ivs and individuals\_mods

Any combination dvs, target\_ivs and individuals\_mods can be specified. The migraine triggers example used all three, and the THC metabolomics example used both target independent variables (the metabolites) and dependent variables (the 8 outcomes). PersonAlytic sets up the parallelization to minimize redundant operations depending on which combination of high throughput options the user requests.

# Optional PersonAlytic() Parameters

## The phase variable

phase: the name of the phase variable. A phase variable is required for an ICT, though PersonAlytic() does not require a phase variable for other analyses such as time series analyses and small sample intensive longitudinal data. The phase variable must be a quoted variable name matching the dependent variable in the data set that the user provided to the parameter data.

eg\_phase <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 autoDetect=NULL)

By default, the phase by time interaction will be included in the model. If the model type is alignPhase=piecewise, a piecewise growth model (see below), the time by phase interaction will be dropped if it model fails to converge with this interaction.

## The Shape of the Trajectory

By default, a linear growth model is fit to the data. If the user desires a quadratic growth model, use time\_power=2. Any integer may be supplied to time\_power to specify the corresponding trajectory shape (e.g., linear, quadratic, cubic, etc.). Alternatively, PersonAlytics can automatically detect the order for time using the TO option of the autoDetect parameter as described below. This example specifies a quadratic growth trajectory.

eg\_time\_power <- PersonAlytic(output="MyResults",  
 data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 time\_power=2,  
 autoDetect=NULL)

## Residual Correlation Structure

The residual autocorrelation structure can be specified by the user using the correlation option. Any option listed in the ?corStruct documentation of the nlme package can be specified (see also the correlation parameter in ?lme). The correlation option defaults to NULL, corresponding to no within-group correlations. This example specifies an ARMA(3,4) residual correlation structure. Note that the parameters p and q must be specified explicitly and the entire correlation structure must be in quotes.

eg\_correlation <- PersonAlytic(output="MyResults",  
 data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 correlation="corARMA(p=3,q=4)",  
 autoDetect=NULL)

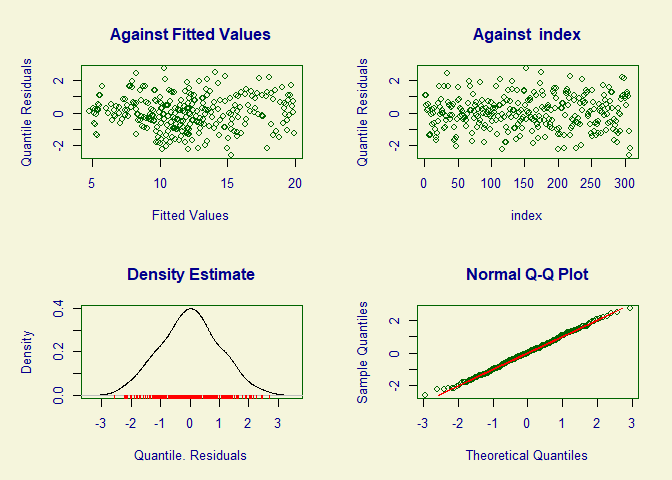
Users unfamiliar with residual correlation structures (or other features of the nlme package) should consult the book “Mixed-Effects Models in S and S-PLUS” by Pinheiro and Bates (2000). Alternatively, PersonAlytics can automatically detect the values for p and q for an ARMA(p,q) model using the AR option of the autoDetect parameter as described below.

## Distributional Assumptions

By default, a linear mixed effects model is fit assuming a normal distribution for the outcome (or, more precisely, for the residuals). If the package parameter described below is set to 'gamlss', the family parameter can be used to specify any of the distributions available in the ?gamlss.family package. Once a model is fit, the plot option can be used to examine the residuals. This only works if a single model is fit (i.e., dvs has only one variable, target\_ivs is not used, and individual\_mods is FALSE). Here is an example of a normal model fit using gamlss.

eg\_normal <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 family=NO(),  
 package="gamlss",  
 autoDetect=NULL)

plot(eg\_normal)

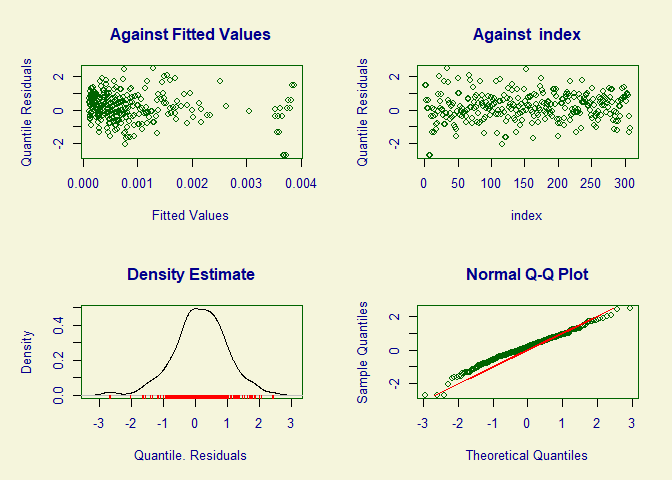


## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
## Summary of the Quantile Residuals  
## mean = 4.314702e-17   
## variance = 1.003257   
## coef. of skewness = 0.02660023   
## coef. of kurtosis = 2.684892   
## Filliben correlation coefficient = 0.998804   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The above example is repeated, this time using a Weibull Type 2 distribution.

eg\_weibull2 <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 family=WEI2(),  
 package="gamlss",  
 autoDetect=NULL)

plot(eg\_weibull2)



## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
## Summary of the Quantile Residuals  
## mean = 0.1554844   
## variance = 0.680279   
## coef. of skewness = -0.2875389   
## coef. of kurtosis = 3.794029   
## Filliben correlation coefficient = 0.9942695   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Model comparisons can be implemented to automatically select the best fitting distribution using the approach implemented in the fitDist function of the gamlss package. See the section on the autoDetect parameter.

## Subgroup Analysis

If the user wants the model to be fit to a subset of the data, use the subgroup parameter. A logical (true/false) or binary (0/1) vector can be used to specify which cases to use. For example, we can restrict the analysis to the first 5 mares using the following:

eg\_subgroup <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 subgroup=OvaryICT$Mare<6,  
 autoDetect=NULL)

where the code OvaryICT$Mare<6 creates a logical vector which is true if the mare id is less than 6.

## Variable Standardization

Variable standardization (mean zero and unit variance) is facilitated using the standardize parameter which is a list with three options that can be set to TRUE or FALSE (the default for all three is FALSE):

* dv: if set to TRUE, the dependent variable is standardized. If there are multiple dependent variables in dvs, each dependent variable will be standardized.
* ivs: if set to TRUE, the the independent variables in ivs and target\_ivs are standardized.
* byids: if set to TRUE, standardization is done within each individual. This option should be set to TRUE if individual\_mods=TRUE (this parameter is discussed below).

Here is an example where the target independent variables are standardized. This is useful for determining which target independent variable has the largest effect size since standardization put their respective effect sizes on the same scale. All continuous variables should be standardized.

eg\_standardize <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 target\_ivs=list("Target1", "Target2", "Target3"),  
 standardize=list(dv=FALSE, ivs=TRUE, byids=FALSE),  
 autoDetect=NULL)

##   
## PersonAlytics is standardizing the variables in `targe\_ivs`.

##   
##   
## Fitting models of the dependent variable `follicles` for 11 cases in `Mare`  
## and for 3 target indepented variables in `target\_ivs`.

##   
 |   
 |========================== | 33%  
 |   
 |==================================================== | 67%  
 |   
 |==============================================================================| 100%

##   
##   
## Model fitting for the dependent variable `follicles` took:  
## Time difference of 4.607968 secs.

## Estimation Methods

A challenge of single subject and small sample mixed effect modeling is a lack of statistical power. One result of research in this area recommends that model comparisons be made using maximum likelihood (ML) estimation, while final model results should be estimated using restricted maximum likelihood (REML). The autodetection options detailed below use ML model comparisons by default, and final results reported in the output are estimated using REML by default. This cannot be changed, but for a single model without autodetection, the method parameter can be used select ML instead of REML:

eg\_method <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 method="ML",  
 autoDetect=NULL)

## Dealing with Invalid Variable Names

If the names of the target predictors in target\_ivs had to be edited to make valid variable names (see ?make.names), this parameter allows the user to put the illegal characters back in for the row variable names in high throughput output. For example, if the original variable name was “17.00\_832.2375m/z”, a letter would need to prefix the variable name and the “/” would need to be replaced with another character, e.g., “X17.00\_832.2375m.z”. To get the row names of the output back to original variable name, use charSub=list(c("X", ""), c("m.z", "m/z")).

Note that inputs to charSub must be in double quotes and are case sensitive. All duplicates will be substituted. For example, if the variable name was “X1X23.x” and charSub=list(c(“X”, “”)), the resulting row label for this variable would be “123.x”.

## Type I Error or False Discovery Rate Adjustment

For high throughput analyses, we recommend selecting either a Type I error rate adjustment (such as Bonferonni) or a False Discovery Rate (FDR) adjustments (such as the method of Benjamini, Hochberg, and Yekutieli). This is implemented using the p.method parameter which takes on any value available in the ?p.adjust function. In conjunction with the p.method is the Type I error rate alpha, which has a default of .05. Here is an example using the method of Benjamini, Hochberg, and Yekutieli (p.method=BY) and a Type I error rate of .1, adjusting for the FDR across the three predictors in target\_ivs. If there are multiple dependent variables in dvs, adjustments are made across target independent variables within each dependent variable. If individual\_mods=TRUE, adjustments are made across target independent variables within each dependent variable.

eg\_p.method <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 target\_ivs=list("Target1", "Target2", "Target3"),  
 p.method='BY',  
 alpha=.1,  
 autoDetect=NULL  
 )

##   
##   
## Fitting models of the dependent variable `follicles` for 11 cases in `Mare`  
## and for 3 target indepented variables in `target\_ivs`.

##   
 |   
 |========================== | 33%  
 |   
 |==================================================== | 67%  
 |   
 |==============================================================================| 100%

##   
##   
## Model fitting for the dependent variable `follicles` took:  
## Time difference of 4.186984 secs.

## Phase Alignment

The alignPhase parameter provides options for aligning the time variable with the phase variable.

1. alignPhase='none' is the default and the time variable is left as-is.
2. alignPhase='align' aligns the time variable at the transition from the first and second phase within each participant. This alignment makes it so the effect at time=0 is the start of the second phase. For example, if the second phase starts at time 6 for participant A, and at time 8 for participant B, and there are 15 time points for each starting at 1:
   * Participant A’s resulting time variable will be -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
   * Participant B’s resulting time variable will be -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7.
3. alignPhase='piecewise' create a piecewise straight-line growth model within each phase. This option can be use to simplify complex trajectories when the within-phase trajectory is approximately linear within each phase. Although this may make results easier to interpret than a curvelinear model (e.g., when O=3), the piecewise model may have many more random effects and therefore be less parsimonious than a model with polynomial time. If alignPhase='piecewise', the TO option of the autodDetect parameter and the time\_power parameter are ignored.

## Finite Population Correction

A finite population correction (FPC) can be made if the size of the population from which the user’s sample originated is known. As an example, population sizes may be known for rare diseases. The parameter fpc has a default of 0, which turns off the FPC. If the user’s population size is known, set fpc to the population size. In this example, the population size is set to 6,000.

A finite population correction (FPC) can be used when the sample more than 5% of the population without replacement. In this situation the central limit theorem doesn’t hold and the standard errors of the user’s parameter estimates will be to large. Before illustrating how to implement a finite population correction, we first discuss the conditions under which a finite population correction may make a difference in a power analysis. This will only occur in situation where a large number of the B replications have p-values just larger than alpha and the FPC results in these p-values being less than alpha. In our experience, this situation is rare but possible. The p-value distributions are usually smoothly positively skewed or near uniform. Neither of this distributions puts enough p-values near alpha for the FPC to have a large effect on power. Users can apply the FPC to real data analyses using the PersonAlytics packag but they should not expect the FPC to yield large improvements in power.

The argument fpc can be set to the user’s finite population size and an FPC will be included in the analyses and output.

eg\_fpc <- PersonAlytic(data=OvaryICT,  
 ids="Mare",  
 dvs="follicles",  
 phase="Phase",  
 time="Time",  
 ivs=list("Target1", "Target2", "Target3"),  
 fpc=6000,  
 autoDetect=NULL  
 )

Preliminary simulation studies have found that on average, the FPC generally reduces standard errors, but only by a small amount.

## Processors for Paralellization

The cores option allows the user to specify how many processors (or cores) on their computer can be devoted to a high throughput PersonAlytic run. By default, PersonAlytic detects the number of cores and uses one fewer than are on the machine. This allows the user to still do other work. If the user has a machine dedicated to analyses that won’t be needed until analyses are completed, setting the the number of cores to the maximum available will reduce computation time. Do not set this value to a number greater than the number of processors on the user’s machine or it may cause R or the user’s computer to crash. To determine the number of cores the user has, type the following into the R console:

parallel::detectCores()

## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE