

4.2 Models for correlated data

There is extensive support within SAS and R for correlated data regression models, including repeated measures, longitudinal, time series, clustered, and other related methods. Throughout this section we assume that repeated measurements are taken on a subject or cluster denoted by variable `id`.

4.2.1 Linear models with correlated outcomes

SAS

HELP example: see 4.6.10

```
proc mixed data=ds;  
  class id;  
  model y = x1 ... xk;  
  repeated / type=vartype subject=id;  
run;
```

or

```
proc mixed data=ds;
  class id;
  model y = x1 ... xk / outpm=dsname;
  repeated ordervar / type=covtypename subject=id;
run;
```

Note: The `solution` option to the `model` statement can be used to get fixed effects parameter estimates in addition to ANOVA tables. The `repeated ordervar` syntax is used when observations within a cluster are a) ordered (as in repeated measurements) b) the placement in the order affects the covariance structure (as in most structures other than independence and compound symmetry) and c) observations may be missing from the beginning or middle of the order. Predicted values for observations can be found using the `outpm` option to the `model` statement as demonstrated in the second block of code. To add to the `outpm` dataset the outcomes and transformed residuals (scaled by the inverse Cholesky root of the marginal covariance matrix), add the `vciry` option to the `model` statement.

The structure of the covariance matrix of the observations is controlled by the `type` option to the `repeated` statement. As of SAS 9.2, there are 36 available structures. Particularly useful options include `un` (unstructured), `cs` (compound symmetry), and `ar(1)` (first-order autoregressive). The full list is available through the on-line help: Contents; SAS Products; SAS Procedures; MIXED; Syntax; REPEATED.

R

```
glsres <- gls(y ~ x1 + ... + xk,
  correlation=corSymm(form = ~ ordervar | id),
  weights=varIdent(form = ~1 | ordervar), ds)
```

Note: The `gls()` function supports estimation of generalized least squares regression models with arbitrary specification of the variance covariance matrix. In addition to a formula interface for the mean model, the analyst specifies a within group correlation structure as well as a description of the within-group heteroscedasticity structure (using the `weights` option). The statement `ordervar | id` implies that associations are assumed within `id`. Other covariance matrix options are available, see `help(corClasses)`.

4.2.2 Linear mixed models with random intercepts

See also 4.2.3 (random slope models), 4.2.4 (random coefficient models), and 6.1.2 (empirical power calculations)

SAS

```
proc mixed data=ds;
  class id;
  model y = x1 ... xk;
  random int / subject=id;
run;
```

Note: The `solution` option to the `model` statement may be required to get fixed effects parameter estimates in addition to ANOVA tables. The `random` statement describes the design matrix for the random effects. Unlike the fixed effects design matrix, specified as usual with the `model` statement, the random effects design matrix includes a random intercept only if it is specified as above. The predicted random intercepts can be printed with the `solution` option to the `random` statement and saved into a dataset using the `ODS`

system, e.g an ods output `solutionr=reffs` statement. Predicted values for observations can be found using the `outp=datasetname` and `outpm=datasetname` options to the `model` statement; the `outp` dataset includes the predicted random effects in the predicted values while the `outpm` predictions include only the fixed effects.

R

```
library(nlme)
lmeint <- lme(fixed= y ~ x1 + ... + xk, random = ~ 1 | id,
             na.action=na.omit, data=ds)
```

Note: Best linear unbiased predictors (BLUP's) of the sum of the fixed effects plus corresponding random effects can be generated using the `coef()` function, random effect estimates using the `random.effects()` function, and the estimated variance covariance matrix of the random effects using `VarCorr()`. Normalized residuals (using a Cholesky decomposition, see pages 238-241 of Fitzmaurice et al [22]) can be generated using the `type="normalized"` option when calling `residuals()` using an NLME option (more information can be found using `help(residuals.lme)`). A plot of the random effects can be created using `plot(lmeint)`.

4.2.3 Linear mixed models with random slopes

HELP example: see 4.6.11

See also 4.2.2 (random intercept models) and 4.2.4 (random coefficient models)

SAS

```
proc mixed data=ds;
  class id;
  model y = time x1 ... xk;
  random int time / subject=id type=covtypename;
run;
```

Note: The `solution` option to the `model` statement can be used to get fixed effects parameter estimates in addition to ANOVA tables. Random effects may be correlated with each other (though not with the residual errors for each observation). The structure of the the covariance matrix of the random effects is controlled by the `type` option to the `random` statement. The option most likely to be useful is `type=un` (unstructured); by default, `proc mixed` uses the variance component (`type=vc`) structure, in which the random effects are uncorrelated with each other. The predicted random effects can be printed with the `solution` option to the `random` statement and saved into a dataset using the ODS system, e.g an ods output `solutionr=reffs` statement. Predicted values for observations can be found using the `outp=datasetname` and `outpm=datasetname` options to the `model` statement; the `outp` dataset includes the predicted random effects in the predicted values while the `outpm` predictions include only the fixed effects.

R

```
library(nlme)
lmeslope <- lme(fixed=y ~ time + x1 + ... + xk, random = ~ time | id,
              na.action=na.omit, data=ds)
```

Note: The default covariance for the random effects is unstructured (see `help(reStruct)` for other options). Best linear unbiased predictors (BLUP's) of the sum of the fixed effects plus corresponding random effects can be generated using the `coef()` function, random effect estimates using the `random.effects()` function, and the estimated variance covariance matrix of the random effects using `VarCorr()`. A plot of the random effects can be created using `plot(lmeint)`.

4.2.4 More complex random coefficient models

We can extend the random effects models introduced in 4.2.2 and 4.2.3 to 3 or more subject-specific random parameters (e.g., a quadratic growth curve or spline/“broken stick” model [22]). We use $time_1$ and $time_2$ to refer to 2 generic functions of time.

SAS

```
proc mixed data=ds;
  class id;
  model y = time1 time2 x1 ... xk;
  random int time1 time2 / subject=id type=covtypepname;
run;
```

Note: The `solution` option to the `model` statement can be used to get fixed effects parameter estimates in addition to ANOVA tables. Random effects may be correlated with each other, though not with the residual errors for each observation. The structure of the covariance matrix of the random effects is controlled by the `type` option to the `random` statement. The option most likely to be useful is `type=un` (unstructured); by default, `proc mixed` uses the variance component (`type=vc`) structure, in which the random effects are uncorrelated with each other. The predicted random effects can be printed with the `solution` option to the `random` statement and saved into a dataset using the ODS system, e.g. `ods output solutionr=reffs`. Predicted values for observations can be found using the `outp` and `outpm` options to the `model` statement; the `outp` dataset includes the predicted random effects in the predicted values while the `outpm` predictions include only the fixed effects.

R

```
library(nlme)
lmestick <- lme(fixed= y ~ time1 + time2 + x1 + ... + xk,
  random = ~ time1 time2 | id, data=ds, na.action=na.omit)
```

Note: The default covariance for the random effects is unstructured (see `help(reStruct)` for other options). Best linear unbiased predictors (BLUP’s) of the sum of the fixed effects plus corresponding random effects can be generated using the `coef()` function, random effect estimates using the `random.effects()` function, and the estimated variance covariance matrix of the random effects using `VarCorr()`. A plot of the random effects can be created using `plot(lmeint)`.