Longitudinal Data Analysis: Estimation in the General Linear Model

Arnab Maity

NCSU Department of Statistics ~ 5240 SAS Hall ~ 919-515-1937 ~ amaity[at]ncsu.edu

Contents

Introduction

We know that the three main steps in modeling longitudinal data are as follows:

- modeling the mean,
- modeling the covariance,
- and selecting the distribution of the data $\boldsymbol{Y}$.

We have alredy established that we can write a *general linear model* for the i-subject:

$$\boldsymbol{Y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{e}_i,$$

where $\boldsymbol{e}_i$ is m_i-dimensional vector of random deviations, $\boldsymbol{\beta}$ is the fixed effects parameter and corresponds to the model matrix $\boldsymbol{X}_i$. The *mean regression parameter* $\boldsymbol{\beta}$ is the main object of inference.[1] The term $\boldsymbol{e}_i$ is the deviation from the systematic component, which has a multivariate random distribution with mean 0 and covariance matrix $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_i(\omega)$. Here ω is a vector of unknown parameter in the covariance model; these parameter in ω are often called *variance components*.

In this chapter we assume that the responses are *normally distributed*, that is,

$$\boldsymbol{Y}_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta}, \boldsymbol{\Sigma}_i(\omega)).$$

This approach separates the modeling of the mean (systematic component) and the correlation of the errors (random component). Modeling the correlation in longitudinal data is important to be able to obtain correct inferences on regression coefficients $\boldsymbol{\beta}$. The correlation model does not change the interpretation of the $\boldsymbol{\beta}$ parameters.

[1] Inference refers to creating confidence intervals and hypothesis testing.

Estimation of the regression parameters ($\boldsymbol{\beta}$)

Maximum Likelihood Estimatator

When full distributional assumptions have been made about the vector of responses[2] , a standard approach is to employ *Maximum Likelihood Estimation (MLE)*.

[2] Recall that we have assumed $\boldsymbol{Y}$ follows a multivariate normal distribution in our case.

Main idea of MLE

The main idea in the MLE is to estimate the parameters by the values that make the observed data most likely to have occurred, under the specified model.

For simplicity assume first that the covariance parameters ω are *known*. In other words, $\boldsymbol{\Sigma}_i(\omega)$ is known. As usual, we use hat to denote parameter estimators (e.g., $\widehat{\boldsymbol{\beta}}$ will denote an estimator $\boldsymbol{\beta}$).

Math behind MLE: not needed for exam

To obtain the MLE of $\boldsymbol{\beta}$ we need to maximize the following log-likelihood function:

$$\begin{aligned}\ell(\boldsymbol{\beta}) &= \frac{1}{2}(\sum_{i=1}^{n} m_i)\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log|\boldsymbol{\Sigma}_i| \\ &\quad -\frac{1}{2}\left\{\sum_{i=1}^{n}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta})\right\};\end{aligned}$$

Thus,

$$\widehat{\boldsymbol{\beta}}_{MLE} = \text{argmax}_{\beta}\ell(\boldsymbol{\beta}).$$

Since $\boldsymbol{\beta}$ does not appear in the first two terms, it follows that maximization of the log-likelihood function $\ell(\boldsymbol{\beta})$ is equivalent to minimization of the last term only. We can obtain the estimator of $\boldsymbol{\beta}$ as

$$\widehat{\boldsymbol{\beta}}_{MLE} = \text{argmin}_{\beta}\sum_{i=1}^{n}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}).$$

We can show that the *maximum likelihood estimator* is:

$$\widehat{\beta}_{MLE} = \left\{\sum_{i=1}^{n}(\boldsymbol{X}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{X}_i)\right\}^{-1}\sum_{i=1}^{n}(\boldsymbol{X}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{Y}_i).$$

The solution presented above is also refered to as the *generalized least squares (GLS)* estimator of $\boldsymbol{\beta}$, also denoted as $\widehat{\beta}_{GLS}$. From now on, we will simply refer the estimator as $\widehat{\beta}$.

Properties of $\widehat{\beta}$

The GLS estimator has the following properties.

1) $\widehat{\boldsymbol{\beta}}$ is an *unbiased estimator* of $\boldsymbol{\beta}$: $E(\widehat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$. This is true *even if we wrongly specify the covariance structure*.

2) *If the error distribution is multivariate normal*, then the sampling distribution of $\widehat{\boldsymbol{\beta}}$ is *multivariate normal*, that is,

$$\widehat{\boldsymbol{\beta}} \sim N\left(\boldsymbol{\beta}, \left\{\sum_{i=1}^{n}(\boldsymbol{X}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{X}_i)\right\}^{-1}\right).$$

3) *If the error distribution is multivariate normal*[3] *and the covariance is correctly specified*, then $\widehat{\boldsymbol{\beta}}$ is the uniformly minimum variance unbiased estimator (UMVUE) for $\boldsymbol{\beta}$. Specifically, it has "smaller" variance than any other estimator (linear or nonlinear) of $\boldsymbol{\beta}$.

[3] Irrespective of the error distribution, normal or otherwise, the GLS estimator $\widehat{\boldsymbol{\beta}}$ is the best linear unbiased estimator (BLUE) for $\boldsymbol{\beta}$. Specifically, it has "smaller" variance than any other *linear* estimator of $\boldsymbol{\beta}$.

Estimation of the variance components (ω)

In practice, the covariance parameter ω is not known. Typically Maximum Likelihood (ML) or Restricted Maximum Likelihood (REML) estimation is used to obtain an estimate for ω. *The MLE of the variance component ω is typically biased*. The theory of restricted maximum likelihood (REML) was precisely developed to address this limitation. [4] *REML produces estimates of the variance/covariance parameters that are less biased*. REML estimation is thus the default method used to estimate the variance component parameters for many algorithms.

[4] Bias arises because the ML estimate $\widehat{\omega}$ does not take into account that $\boldsymbol{\beta}$ is also estimated. The REML likelihood is the function for the marginal distribution of the residuals.

Math details for REML: not needed for exam

REML approach uses a ML function calculated to a transformed data in a way that ensures that the mean parameters have no effect on the transformed data. The REML log-likelihood function can be computed as:

$$\begin{aligned}\ell_{REML}(\boldsymbol{\beta},\omega) &= \frac{\sum_{i=1}^{n} m_i}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log|\boldsymbol{\Sigma}_i(\omega)| \\ &\quad -\frac{1}{2}\left\{\sum_{i=1}^{n}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T\boldsymbol{\Sigma}_i(\omega)^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta})\right\} \\ &\quad -\frac{1}{2}\sum_{i=1}^{n}\log|\boldsymbol{X}_i^T\boldsymbol{\Sigma}_i^{-1}(\omega)\boldsymbol{X}_i|;\end{aligned}$$

The ML/REML estimator $\widehat{\omega}$ does not have a close form simple expression – numerical algorithms are used to obtain $\widehat{\omega}$. When such an estimate is obtained then $\widehat{\boldsymbol{\Sigma}}_i = \boldsymbol{\Sigma}_i(\widehat{\omega})$ is substituted in the expression of $\widehat{\boldsymbol{\beta}}$. *When the sample size n is large*, the resulting estimator $\widehat{\boldsymbol{\beta}}$ will *approximately* have all the same properties as if ω were known.

Example: The Vlagtwedde-Vlaardingen Study

This epidemiologic study was conducted in two different areas in the Netherlands - the rural area of Vlagtwedde (N-E) and the urban, industrial area of Vlaardingen (S-W).[5] The dataset consists of a subsample of $n = 133$ individuals who were 36 years or older at their entry into the study and whose smoking status did not change over the 19 years of follow-up. A measure of *forced expiratory volume (FEV1)* was obtained every three years for the first 15 years of the study, and also at year 19. Information on smoking status (0 = former, 1 = current) was also collected at each follow-up.

[5] Source: van der Lende, R., Kok, T.J., Peset, R., Quanjer, P.H., Schouten, J.P. and Orie, N.G.M. (1981). Decreases in VC and FEV1 with time: Indicators for effects of smoking and air pollution. Bulletin of European Physiopathology and Respiration, 17, 775-792.

The original dataset is available at `https://content.sph.harvard.edu/fitzmaur/ala2e/`.

```r
# read data
smoking <- read.table("data/smoking.txt")
names(smoking) <-  c("id", "smoker", "time", "FEV1")
head(smoking)
```

```
##   id smoker time FEV1
## 1  1      0    0 3.40
## 2  1      0    3 3.40
## 3  1      0    6 3.45
## 4  1      0    9 3.20
## 5  1      0   15 2.95
## 6  1      0   19 2.40
```

```r
## Observations per subject
table(table(smoking$id))
```

```
##
##  1  5  6  7
##  1 54 46 32
```

```r
## Observations in each smoking group (0=former, 1=current)
## grouped by time points
table(smoking$smoker, smoking$time, dnn = list("smoker", "time"))
```

```
##       time
## smoker  0  3  6  9 12 15 19
##      0 23 27 28 30 29 24 28
##      1 85 95 89 85 81 73 74
```

Our questions of interest as as follows.

- How does FEV1 mean trend change over time?
- Is the mean trend different for current smokers than for former ones?

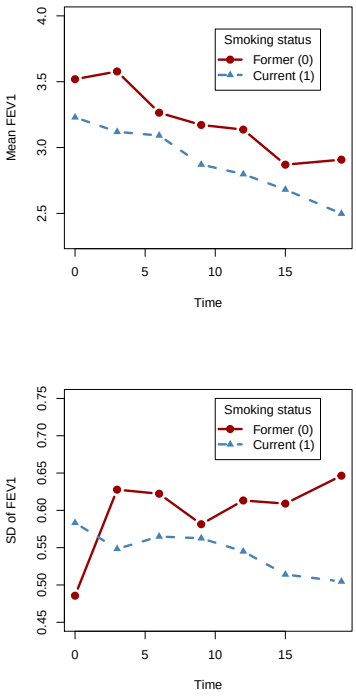

Figure 1: Mean and sd profiles for the Vlagtwedde-Vlaardingen Study data.

Figure 1 shows the mean profiles (top panel) and the sd profiles (bottom panel) for the two groups. It is evident that a linear function of `time` is sufficient for both the groups. Also, it seems the standard deviations across time do not vary by much.

Next, we write down a parametric model for both mean and covariance. Following the discussion in the previous chapter, we write the mean model as

$$Y_{ij} = \eta_1 + \eta_2 t_{ij} + \eta_3 S_{ij} + \eta_4 t_{ij} S_i + e_{ij},$$

where t_{ij} are observed time point for the j-measurement of the i-th person, and S_i is a dummy variable for whether the i-th person is a smoker or not (1 = current, 0 = former).

For the covariance model, we will investigate the following options:

- *Compound symmetric*, assuming both groups have the *same covariance matrix*, and a *common variance* for each time point: assume that any two measurements within a specific subject has the same correlation ρ, and the same variance σ^2 for each time point.[6] Here we have

$$\Sigma(\omega) = \sigma^2 \begin{bmatrix} 1 & \rho & \dots & \rho \\ & 1 & \dots & \rho \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

  Thus, we assume that value of σ^2 and ρ are the same for the two groups, and that σ^2 does not change over time.

- *Compound symmetric*, assuming both groups have the *same covariance* but *unequal variance* across time: same structure as above, but value of σ^2 changes over time. Specifically, we assume $var(e_{ij}) = \sigma_j^2$ and $corr(e_{ij}, e_{i\ell}) = \rho, j \neq \ell$. We can write the covariance matrix[7]

$$\Sigma(\omega) = \boldsymbol{D} \begin{bmatrix} 1 & \rho & \dots & \rho \\ & 1 & \dots & \rho \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix} \boldsymbol{D},$$

  where $\boldsymbol{D} = diag(\sigma_1, \sigma_2, \dots, \sigma_7)$ – here σ_j denotes the standard deviation of the errors at time point t_j.

- *Unstructured covariance, assuming both groups have the same covariance* structure. Specifically, $\boldsymbol{\Sigma}$ is unstructuted and does not depend on smoking status. For example, correlation between observations taken at time points 1 and 3 in the "current" group is the same as in "former" group.

To fit this model in R, we can use the `gls()` function in the `nlme` package.[8]

gls(model, data, correlation, weights, subset, method, na.action, control, verbose)

- *model*: a formula that specifies the mean model. This is typically of the form `y ~ x1 + x2` where `y` is the response variable, and `x1`, `x2` etc are predictors
- *data*: dataset (a data frame) that we have
- *correlation*: the corrrelation structure we are going to use[9]

[6] Specifically, we assume $var(e_{ij}) = \sigma^2$ and $corr(e_{ij}, e_{i\ell}) = \rho, j \neq \ell$.

[7] We specify the covariance matrix in this way since we will separately specify correlation and variance parameters when fitting the model in R.

[8] See "?gls" for more details.

[9] Specifies the correlation matrix, e.g., compound symmetry.

- *weights*: any specification of variances we might have[10]
- *subset*: if we intend to use only a subset of the dataset (e.g. `smoker==0`)
- *method:* "REML" or "ML" to estimate $\boldsymbol{\beta}$ and ω.

[10] Specifies the variances, e.g., equal variance, unequal variances across time etc.

We now set the formula of the mean trend in R format.[11]

[11] We can do this step directly in `gls`; however, I recommend to define the formula separately especially if it involves a lot ot covariate terms and interactions.

```
meanform <- FEV1 ~ time + smoker + time:smoker
```

Recall smoker takes values 0 and 1. Thus we are fitting a linear function in `time` and allowing the intercept and slope to be different in the `smoker=0` and `smoker=1` groups. This will model the baseline (`smoker=0` group) intercept and slope (β_1 and β_2) directly; then model the *change* in intercept and slope in the other group (β_3 and β_4). Thus we have

$$\text{Former smokers}(S_i = 0) : E(Y_{ij}) = \eta_1 + \eta_2 t_{ij},$$

$$\text{Current smokers}(S_i = 1) : E(Y_{ij}) = (\eta_1 + \eta_3) + (\eta_2 + \eta_4) t_{ij}.$$

Let us examine the design matrix for a particular individual (that is, $\boldsymbol{X}_i$ for some i). We can do so by using the `model.matrix()` function. This is a good way to check whether the formula we intend to use is indeed the one we want to fit.[12]

[12] Note that for $i = 1$, smoker = 0 in the dataset, and the observation corresponding to time "12" is missing.

```
model.matrix(meanform, data = smoking[smoking$id==1,])
```

```
##   (Intercept) time smoker time:smoker
## 1           1    0      0           0
## 2           1    3      0           0
## 3           1    6      0           0
## 4           1    9      0           0
## 5           1   15      0           0
## 6           1   19      0           0
## attr(,"assign")
## [1] 0 1 2 3
```

Fitting the compound symmetry covariance model with equal variance over time

```
library(nlme)
gls.fit.exch <- gls(model = meanform, data = smoking,
    correlation = corCompSymm(form = ~1 | id))
```

Let us examine each part of the code above.

- *The "model" argument* specifies the formula specified in `meanform` we defined before, that is, `FEV1 ~ time + smoker + smoker:time`.[13] Note that even though we have not included an intercept directly in the formula, R will automatically include the intercept term.

[13] This fits the mean model model: $E(Y_{ij}) = \eta_1 + \eta_2 t_{ij} + \eta_3 S_i + \eta_4 t_{ij} S_i$, where Y_{ij} is the FEV1 measure of the i-th subject at time t_{ij}, S_i is the dummy variable for smoking status.

- *The "data" argument* specified the dataset we are using. So R knows where to find the variables refered in the formula (e.g., `FEV1`, `time`, `smoker` etc.)
- *The argument* `correlation = corCompSymm( , form= ~ 1 | id )` specifies the *correlation structure*.[14] Specifically, the `corCompSymm()` function specifies the compound symmetric corrrelation structure. The argument in this function, "`form= ~ 1 | id`", defines that compound symmetric correlation structure should be defined for *each id* (that is, observations with the same `id` value belong to the same subject).
- Note that we have left *the* `weights` *argument* unspecified. This enforces R to use same variance for all time points.[15]

[14] This argument specifies the correlation matrix

$$\begin{bmatrix} 1 & \rho & \cdots & \rho \\ & 1 & \cdots & \rho \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

[15] This specifies $var(e_{ij}) = \sigma^2$, that is, variance for each time point is same.

```
summary(gls.fit.exch)
```

```
## Generalized least squares fit by REML
##   Model: meanform
##   Data: smoking
##        AIC      BIC    logLik
##   323.6031 351.4581 -155.8016
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | id
##  Parameter estimate(s):
##       Rho
## 0.8595179
##
## Coefficients:
##                 Value  Std.Error   t-value p-value
## (Intercept)  3.507677 0.09804324  35.77684  0.0000
## time        -0.033852 0.00262840 -12.87912  0.0000
## smoker      -0.272676 0.11239910  -2.42596  0.0155
## time:smoker -0.004570 0.00301829  -1.51419  0.1304
##
##  Correlation:
##             (Intr) time   smoker
## time        -0.250
## smoker      -0.872  0.218
## time:smoker  0.218 -0.871 -0.246
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.97625810 -0.65102769  0.01890054  0.68738226  3.14895565
##
```

```
## Residual standard error: 0.5711548
## Degrees of freedom: 771 total; 767 residual
```

Ley us now examin each segment of the output above. The first block "Generalized least squares fit by REML" tells us that the REML procedure was used to estimate the parameters.

```
## Generalized least squares fit by REML
##   Model: meanform
##   Data: smoking
##        AIC      BIC    logLik
##   323.6031 351.4581 -155.8016
```

This indicates that the estimated variance components are unbiased as well. This block also gives us the model specification and data used, and also some goodness-of-fit measurements (AIC, BIC).[16] The `logLik` value gives the log-likelihood value that can be used to formally test goodness-of-fit between two models – we will discuss this topic later.

[16] We will not use these measures yet – they are used to select models given various choices of mean and covariance models.

The second block gives us the estimate of the variance parameters.

```
## Correlation Structure: Compound symmetry
##  Formula: ~1 | id
##  Parameter estimate(s):
##       Rho
## 0.8595179
```

It varifies that we have fit the "Compound symmetry" correlation structure and provids an estimate of the correlation as $\widehat{\rho} \approx 0.86$.

The third block gives the estimated regression coefficients, $\widehat{\boldsymbol{\beta}}$.[17]

[17] Can you interpret the p-values? We will talk about hypothesis testing later.

```
## Coefficients:
##                 Value  Std.Error   t-value p-value
## (Intercept)  3.507677 0.09804324  35.77684  0.0000
## time        -0.033852 0.00262840 -12.87912  0.0000
## smoker      -0.272676 0.11239910  -2.42596  0.0155
## time:smoker -0.004570 0.00301829  -1.51419  0.1304
```

Assuming that we have specified the correct covariance, we can rely on the p-values to test for significance for each $\boldsymbol{\beta}$ coefficients.

The fourth block gives the correlation among the *estimated coefficients* in $\widehat{\boldsymbol{\beta}}$.

```
##  Correlation:
##             (Intr) time   smoker
## time        -0.250
## smoker      -0.872  0.218
## time:smoker  0.218 -0.871 -0.246
```

We can also obtain the *variance-covariance matrix of the estimated regression coefficient*, that is, $cov(\widehat{\boldsymbol{\beta}})$, as below.[18] Entries of this matrix will be used to test for specific contrasts.

[18] Recall that $\widehat{\boldsymbol{\beta}} = (\widehat{\eta}_1, \widehat{\eta}_2, \widehat{\eta}_3, \widehat{\eta}_4)^T$. Can you tell what is the estimated standard error of $\widehat{\eta}_1$, that is, $SE(\widehat{\eta}_1)$?

```
gls.fit.exch$varBeta
```

```
##                (Intercept)          time        smoker   time:smoker
## (Intercept)  9.612476e-03 -6.454309e-05 -9.612476e-03  6.454309e-05
## time        -6.454309e-05  6.908510e-06  6.454309e-05 -6.908510e-06
## smoker      -9.612476e-03  6.454309e-05  1.263356e-02 -8.344302e-05
## time:smoker  6.454309e-05 -6.908510e-06 -8.344302e-05  9.110083e-06
```

We can compute the corresponding correlation matrix, $corr(\widehat{\boldsymbol{\beta}})$, as follows.

```
# Correlation
round(cov2cor(gls.fit.exch$varBeta), 3)
```

```
##             (Intercept)   time smoker time:smoker
## (Intercept)       1.000 -0.250 -0.872       0.218
## time             -0.250  1.000  0.218      -0.871
## smoker           -0.872  0.218  1.000      -0.246
## time:smoker       0.218 -0.871 -0.246       1.000
```

Notice that the correlation matrix above is indeed the same one as from the output of `gls()` call.

The fifth block `Standardized residuals` provides a summary of residuals (scaled by their SD).

```
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.97625810 -0.65102769  0.01890054  0.68738226  3.14895565
```

If they are indeed normal, most values should be -3 to 3. In our case, this seems to be the case indeed.

The last two lines give the estimated SD of errors "Residual standard error: 0.5711548", that is, $\widehat{\sigma} \approx 0.57$, and the degrees of freedoms of the model componets "Degrees of freedom: 771 total; 767 residual". The total degrees of freedom (DF) is essentially the number of rows in the data matrix. The residual degrees of fredom is (total DF - number of parameters in the mean model) = 771 - 4 = 767.[19] These values will be used later to formally test goodness-of-fit.

[19] Recall, we have 4 parameters $\eta_1, \ldots, \eta_4$ in our mean model.

We can also obtain $\widehat{\sigma}$ also by calling the `sigma()` function:

```
sigma(gls.fit.exch)
```

```
## [1] 0.5711548
```

The covariance/correlation matrix of a particular subject can be extracted as below.[20]

[20] Recall that the first individual ($i = 1$) only has six observations (rather than seven). Thus the variance covariance matrix is 6×6 rather than 7×7.

```
# Covariance matrix
Sigma <- getVarCov(gls.fit.exch, individual = 1)
round(Sigma, 2)
```

```
## Marginal variance covariance matrix
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0.33 0.28 0.28 0.28 0.28 0.28
## [2,] 0.28 0.33 0.28 0.28 0.28 0.28
## [3,] 0.28 0.28 0.33 0.28 0.28 0.28
## [4,] 0.28 0.28 0.28 0.33 0.28 0.28
## [5,] 0.28 0.28 0.28 0.28 0.33 0.28
## [6,] 0.28 0.28 0.28 0.28 0.28 0.33
##   Standard Deviations: 0.57446 0.57446 0.57446 0.57446 0.57446 0.57446
```

The "standard deviation" line repeats the same number "0.57446" six times. This is because we have fit a model which assumes *equal variance/sd for each time point*, that is, $\widehat{sd}(e_{ij}) = \widehat{\sigma} \approx 0.57$.

```
# Correlation matrix
round(cov2cor(Sigma), 2)
```

```
## Marginal variance covariance matrix
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 1.00 0.86 0.86 0.86 0.86 0.86
## [2,] 0.86 1.00 0.86 0.86 0.86 0.86
## [3,] 0.86 0.86 1.00 0.86 0.86 0.86
## [4,] 0.86 0.86 0.86 1.00 0.86 0.86
## [5,] 0.86 0.86 0.86 0.86 1.00 0.86
## [6,] 0.86 0.86 0.86 0.86 0.86 1.00
##   Standard Deviations: 1 1 1 1 1 1
```

Notice that the correlation structure is indeed "compound symmetry" as we intended, and the common correlation is indeed "0.86" as we saw in "gls()" output under "Correlation Structure:" block.

Let us now estimate the mean trajectories of the two groups (former and current smokers). Recall, our baseline was "Former" ($S = 0$) group.

```
# Estimated coefficients
cf <- gls.fit.exch$coefficients
round(cf, 3)
```

```
## (Intercept)        time      smoker time:smoker
##       3.508      -0.034      -0.273      -0.005
```

Recall that we have the two mean trends as follows.

$$\text{Former smokers}(S_i = 0) : E(Y_{ij}) = \beta_1 + \beta_2 t_{ij},$$
$$\text{Current smokers}(S_i = 1) : E(Y_{ij}) = (\beta_1 + \beta_3) + (\beta_2 + \beta_4) t_{ij}.$$

Thus the estimated mean trends are as follows:

$$\begin{aligned} \text{Former smokers}(S_i = 0) : E(Y_{ij}) &= 3.508 + (-0.034) t_{ij}, \\ \text{Current smokers}(S_i = 1) : E(Y_{ij}) &= (3.508 - 0.273) + (-0.034 - 0.005) t_{ij} \\ &= 3.235 + (-0.039) t_{ij}. \end{aligned}$$

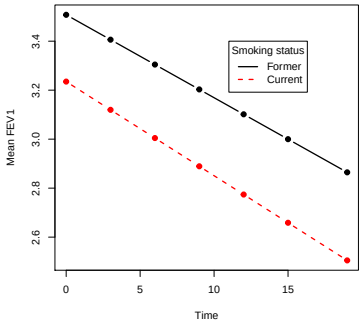

Figure 2: Estimated mean profiles for the former and current smokers.

The two mean profiles are shown in Figure 2.

We can create a few diagnostic plots to assess whether the model fits the data well and whether the normality assumption is reasonable for the errors – see Figure 3.

```
par(mfrow = c(2, 1))
# Get the residuals (standardized)
residual <- residuals(gls.fit.exch, type = "pearson")
# The fitted values
fitted <- fitted(gls.fit.exch)
# Plot residuals vs fitted
plot(fitted, residual, pch = 19, cex = 0.6)
abline(h = 0)
# Normal QQ plot
qqnorm(residual, pch = 19, cex = 0.6)
abline(0, 1, lwd = 2)
```

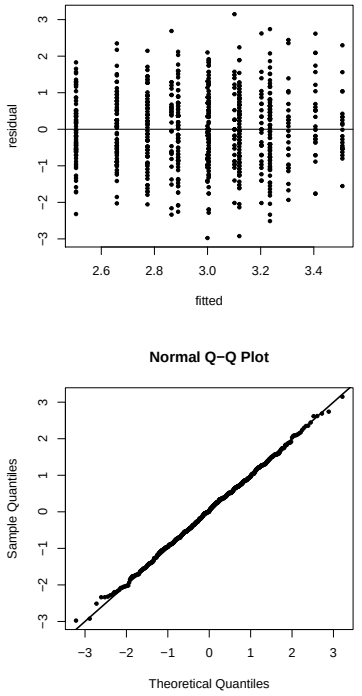

Figure 3: Some diagnostic plots for the model fit.

The command `residuals()` is used to extract residuals from the model fit. If we omit the `type='pearson'` argument, we will only obtain the raw residuals; these are difficult to visualize especially if the variance varies over time. The "pearson residuals" are the standardized residuals (raw residuals divided by the corresponding standard errors). See `?residuals.gls` for more details.

The top panel in Figure 3 shows the usual residual plot. We see that there is no visible trend/pattern indicating a good model fit. The bottom panel shows a normal Q-Q plot of the residuals; the straight line pattern indicates that the normality assumption is reasonable.

Compound symmetry covariance model with unequal variance over time

Recall that we have seven unique time points, $t_1 = 0, t_2 = 3, \ldots, t_6 = 15$ and $t_7 = 19$. We want to model $var(e_{ij}) = \sigma^2_j, j = 1, \ldots, 7$. Thus variance of the observations for each time point is different. To specify unequal variances over time, we need to use the `weights` argument and the `varIdent()` function.[21]

[21] See "?varClasses" for a list of available variance function in the nlme library.

```
gls.cs.uv <- gls(model = meanform, data = smoking,
    correlation = corCompSymm(form = ~1 | id),
    weights = varIdent(form = ~1 | time))
```

The line *weights = varIdent(form = ~ 1 | time)* specifies that each time point has possibly different variances. Specifically, *form = ~ 1 | time* enforces that the variance depends on `time`.

We see the resulting output below. We can interpret the results as explained before.

```
summary(gls.cs.uv)
```

```
## Generalized least squares fit by REML
##   Model: meanform
##   Data: smoking
##        AIC      BIC    logLik
##   331.1515 386.8613 -153.5757
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | id
##  Parameter estimate(s):
##      Rho
## 0.860502
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | time
##  Parameter estimates:
##         0         3         6         9        15        19        12
## 1.0000000 0.9660445 0.9428716 0.9524425 0.9261668 0.9170897 0.9168496
##
## Coefficients:
##                 Value  Std.Error   t-value p-value
## (Intercept)  3.504139 0.10119171  34.62871  0.0000
## time        -0.033629 0.00265466 -12.66775  0.0000
## smoker      -0.256179 0.11606395  -2.20722  0.0276
## time:smoker -0.004861 0.00304898  -1.59444  0.1113
##
##  Correlation:
```

```
##             (Intr) time   smoker
## time        -0.384
## smoker      -0.872  0.335
## time:smoker  0.335 -0.871 -0.381
##
## Standardized residuals:
##           Min            Q1           Med            Q3           Max
## -3.0409686866 -0.6627688481  0.0005168067  0.6730519635  3.2520442372
##
## Residual standard error: 0.6034954
## Degrees of freedom: 771 total; 767 residual
```

We will focus on the "Correlation Structure" section of the output shown above. We can obtain just this part by using the command

```
summary(gls.cs.uv$modelStruct)
```

```
## Correlation Structure: Compound symmetry
##  Formula: ~1 | id
##  Parameter estimate(s):
##      Rho
## 0.860502
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | time
##  Parameter estimates:
##         0         3         6         9        15        19        12
## 1.0000000 0.9660445 0.9428716 0.9524425 0.9261668 0.9170897 0.9168496
```

Under the section *`Variance function`*, we see information about the variances. Specifically, the *`Parameter estimates`* part gives the so-called inflation factors for the *standard deviations*:

$$1, \sigma_2/\sigma_1, \ldots, \sigma_m/\sigma_1.$$

Recall that the estimate of σ_1 (sd at the first time point $t_1 = 0$) can be obtained as follows:[22]

[22] Also from the gls() output in the line "Residual standard error: 0.6034954".

```
sigma(gls.cs.uv)
```

```
## [1] 0.6034954
```

Therefore, the sd at the 2nd time point ($t_2 = 3$), σ_2 is estimated as $\widehat{\sigma}_2 = \sigma_1 \times$(2nd inflation factor) = $0.6034954 \times 0.9660445 \approx 0.583$. The sd at the 3rd time point ($t_3 = 6$), σ_3 is estimated as $\widehat{\sigma}_3 = \sigma_1 \times$(3rd inflation factor) = $0.6034954 \times 0.9428716 \approx 0.569$, and so on.

The covariance matrix for one subject, here the first subject (id=1), is as follows:

```
# Covariance matrix
Sigma <- getVarCov(gls.cs.uv, individual = 1)
Sigma
```

```
## Marginal variance covariance matrix
##         [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
## [1,] 0.36421 0.30276 0.29550 0.29850 0.29026 0.28742
## [2,] 0.30276 0.33989 0.28546 0.28836 0.28041 0.27766
## [3,] 0.29550 0.28546 0.32378 0.28144 0.27368 0.27100
## [4,] 0.29850 0.28836 0.28144 0.33039 0.27646 0.27375
## [5,] 0.29026 0.28041 0.27368 0.27646 0.31241 0.26620
## [6,] 0.28742 0.27766 0.27100 0.27375 0.26620 0.30632
##   Standard Deviations: 0.6035 0.583 0.56902 0.57479 0.55894 0.55346
```

Notice that the standard deviations (of the first three time points) match with what we calculated manually in the previous paragraph. Also recall that the first individual has observation at tme points $0, 3, 6, 9, 15$ and 19, but *not* at time 12. Thus the standard deviations displayed in the output above does *not* display the sd corresponding to $t = 12$.[23]

[23] Can you compute the sd corresponding to $t = 12$ manually?

It seems that our original decision of fitting an equal variance model is justified since the SD values accros time points are very similar (equivalently, the inflation factors in the sd estimates in the gls() output are all close to 1).

Unstructured covariance model with unequal variances over time

The most general covariance model is the unstructured model (no specific pattern among the correlation/covariances) and each time point has a different variance parameter. To fit this model, we need to use the fucntion `corSymm()` as our correlation structure.[24]

[24] See `?corSymm` for more details.

One particular issue about using the corSymm() structure is that *the covariate ("time" in our case) for this correlation structure must be connsecutive integers*. Thus we first *re-label* the time points at $1, 2, \ldots, 7$.

```
smoking$timefact <- as.numeric( factor(smoking$time, labels = 1:7) )
head(smoking)
```

```
##   id smoker time FEV1 timefact
## 1  1      0    0 3.40        1
## 2  1      0    3 3.40        2
## 3  1      0    6 3.45        3
## 4  1      0    9 3.20        4
## 5  1      0   15 2.95        6
## 6  1      0   19 2.40        7
```

Notice that "timefact" for the first individual is missing the value 5 corresponding to the time point $t = 12$.

```
# mean fomula
meanform <- FEV1 ~ time + smoker + time:smoker
# fit unstructured
gls.un <- gls( model = meanform, data = smoking,
                    correlation = corSymm(form= ~ timefact | id ),
                    weights = varIdent(form = ~ 1 | timefact) )
```

The line *correlation = corSymm(, form= ~ timefact | id)* specifies that the subjects are specified using `id` (all observations with the same `id` value belongs to one subject) and that separate correlation parameters should be use for each pair of time point.[25]

The line *weights = varIdent(form = ~ 1 | timefact)* specifies unequal variances, that is, we set a different variance parameter at each time point. Specifically, `form = ~ 1|timefact` enforces that the variance depends on `time`.[26]

We need to notice the following:

1. The mean formula ("meanform") does not change – we are still using "time" (the original time points) as a continuous covariate
2. The newly created variable "timefact" (re-labeled time points) is only used to specify the correlation structure and weights.

The estimated covariance information is shown below.

[25] Specifically, we model $corr(e_{ij}, e_{ik}) = \rho_{jk}$, and *not* $corr(e_{ij}, e_{ik}) = \rho$.

[26] Specifically, we model $var(e_{ij}) = \sigma^2_j$, and *not* $var(e_{ij}) = \sigma^2$.

```
summary(gls.un$modelStruct)
```

```
## Correlation Structure: General
##  Formula: ~timefact | id
##  Parameter estimate(s):
##  Correlation:
##   1     2     3     4     5     6
## 2 0.863
## 3 0.846 0.888
## 4 0.838 0.833 0.833
## 5 0.855 0.862 0.890 0.886
## 6 0.839 0.874 0.868 0.876 0.932
## 7 0.831 0.824 0.834 0.840 0.858 0.893
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | timefact
##  Parameter estimates:
##         1         2         3         4         6         7         5
## 1.0000000 0.9779813 0.9509405 0.9550631 0.9679811 0.9198126 0.9387328
```

Be careful to note that the sd inflation factors are given in the order $1, 2, 3, 4$, then $6, 7$ and then 5.[27]

It is evident that all the pair-wise correlations – shown in the *Correlations* section of the output – are about 0.85. Also, all the sd inflation constants are close to 1. This observation further validates that our original dicision to fit a equal variance compound symmetric model is indeed reasonable.

The covariance matrices for one subject are shown below.

[27] Recall, that in the original time scale, these points correspond to $0, 3, 6, 9, 15, 19$, and 12.

```
# Covariance matrix
Sigma <- getVarCov(gls.un, individual = 1)
Sigma
```

```
## Marginal variance covariance matrix
##         [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
## [1,] 0.35514 0.29967 0.28564 0.28418 0.28842 0.27141
## [2,] 0.29967 0.33968 0.29323 0.27629 0.29385 0.26318
## [3,] 0.28564 0.29323 0.32115 0.26878 0.28368 0.25897
## [4,] 0.28418 0.27629 0.26878 0.32394 0.28769 0.26197
## [5,] 0.28842 0.29385 0.28368 0.28769 0.33277 0.28230
## [6,] 0.27141 0.26318 0.25897 0.26197 0.28230 0.30047
##   Standard Deviations: 0.59594 0.58282 0.5667 0.56916 0.57686 0.54815
```

As before, since the individual corrresponding "id=1" has a missing observation at time point 12 (or re-labled time 5), the standard deviation values presented above correspond to $t = 0, 3, 6, 9, 15$, and 19, respectively.[28]

[28] How can you compute sd corresponding to $t = 12$?

```
# Correlation matrix
cov2cor(Sigma)
```

```
## Marginal variance covariance matrix
##         [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
## [1,] 1.00000 0.86281 0.84580 0.83785 0.83899 0.83085
## [2,] 0.86281 1.00000 0.88781 0.83291 0.87404 0.82380
## [3,] 0.84580 0.88781 1.00000 0.83332 0.86776 0.83366
## [4,] 0.83785 0.83291 0.83332 1.00000 0.87623 0.83968
## [5,] 0.83899 0.87404 0.86776 0.87623 1.00000 0.89277
## [6,] 0.83085 0.82380 0.83366 0.83968 0.89277 1.00000
##   Standard Deviations: 1 1 1 1 1 1
```

Notice that The "Standard Deviations" line shows all entries to be 1 in the output immediately above. This also indicates that the matrix above is a *correlation matrix*.[29]

[29] Can you obtain this correlation matrix directly/manually from looking at the gls() output?

Robust estimator of covariance matrix of $\widehat{\boldsymbol{\beta}}$

We need to remember that one needs to *specify a variance-covariance structure* to estimate $\boldsymbol{\beta}$ using GLS. In practice, our posited covariance model might not correctly specified. In fact, it is quite challenging to identify a "correct" covariance model for the data at hand. Thus, the standard errors of the estimated regression parameters, called the *model based standard errors*[30], might not be reliable. Specifically, the model based covariance matrix of $\widehat{\boldsymbol{\beta}}$ is

[30] These are default standard errors in gls() output.

$$\widehat{\boldsymbol{V}}_{\widehat{\beta},\text{model}} = \{\sum_{i=1}^{n}(\boldsymbol{X}_i^T\widehat{\boldsymbol{\Sigma}}_i^{-1}\boldsymbol{X}_i)\}^{-1},$$

which assumes that the posited covariance model, represented by $\boldsymbol{\Sigma}_i$, is correct. *The model based covariance matrix can be from the gls() output using the "varBeta" field.* For example, for the compund symmetry model fit with equal variance over time[31], $\widehat{\boldsymbol{V}}_{\widehat{\beta},\text{model}}$ is as follows:

[31] Results saved in the output "gls.fit.exch" in the previous sections.

```
V.model <- gls.fit.exch$varBeta
V.model
```

```
##                (Intercept)          time        smoker   time:smoker
## (Intercept)  9.612476e-03 -6.454309e-05 -9.612476e-03  6.454309e-05
## time        -6.454309e-05  6.908510e-06  6.454309e-05 -6.908510e-06
## smoker      -9.612476e-03  6.454309e-05  1.263356e-02 -8.344302e-05
## time:smoker  6.454309e-05 -6.908510e-06 -8.344302e-05  9.110083e-06
```

Therefore the model based standard errors of $\widehat{\boldsymbol{\beta}}$ is as follows.[32]

[32] The diagonal entries of "V.model" are the variances of the elements of $\widehat{\boldsymbol{\beta}}$. Standard errors are obtained by taking square root.

```
se.model <- sqrt(diag(V.model))
round(se.model, 4)
```

```
## (Intercept)        time      smoker time:smoker
##      0.0980      0.0026      0.1124      0.0030
```

To obtain proper inference, we need to obtain standard errors of $\widehat{\boldsymbol{\beta}}$ taking into account that our covariance model might not be correctly specified. We can indeed do so by computing the *robust empirical covariance matrix*[33] of $\widehat{\boldsymbol{\beta}}$.

[33] Also known as robust *sandwich* covariance matrix.

Robust empirical covariance matrix: not needed for exam

The robust empirical covariance matrix of $\widehat{\boldsymbol{\beta}}$ is:

$$\widehat{\boldsymbol{V}}_{\widehat{\beta},\text{robust}} = \widehat{\boldsymbol{V}}_{\widehat{\beta},\text{model}}\boldsymbol{B}\widehat{\boldsymbol{V}}_{\widehat{\beta},\text{model}},$$

where

$$\boldsymbol{B} = \sum_{i=1}^{n}\boldsymbol{X}_i^T\widehat{\boldsymbol{\Sigma}}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\widehat{\boldsymbol{\beta}})(\boldsymbol{Y}_i - \boldsymbol{X}_i\widehat{\boldsymbol{\beta}})^T\widehat{\boldsymbol{\Sigma}}_i^{-1}\boldsymbol{X}_i.$$

We can use the `vcovCR()` function in the `clubSandwich` library to compute the robust covariance matrix as follows.[34]

[34] See "?vcovCR()" for details. There are various "smalll sample adjustments" as well. The option "CRo" does not make any small sample corrections.

```
library(clubSandwich)
# Robust covariance matrix of betahat
V.robust <- vcovCR(gls.fit.exch, type = "CR0")
# Robust se
se.robust <- sqrt(diag(V.robust))
round(se.robust, 4)
```

```
## (Intercept)        time      smoker time:smoker
##      0.1080      0.0033      0.1213      0.0037
```

Note the following points:

1) The *model based covariance matrix* of $\widehat{\boldsymbol{\beta}}$ is computed based on the assumption that out posited error covariance model is *correct*.

2) In contrast to pint in 1), the *robust covariance matrix* accounts for the possibility that we have incorrectly specified the error covariance matrix. Thus it is robust to possible misspecification of the error covariance model.

3) In practice, it is preferable to make inference on $\boldsymbol{\beta}$ based on the *robust covariance matrix*. This is done to protect against the possibility of specifying an incorrect error covariance model.

Inference of the regression parameters

In this section we discuss how to make inferences about $\boldsymbol{\beta}$. Specifically we consider the construction of *confidence intervals* and *tests of hypotheses*. To this end we use the estimator $\widehat{\boldsymbol{\beta}}$ and its estimated covariance matrix $\widehat{\boldsymbol{V}}$ – here $\widehat{\boldsymbol{V}}$ is either model based $\widehat{\boldsymbol{V}}_{\beta,\text{model}}$ or robust $\widehat{\boldsymbol{V}}_{\beta,\text{robust}}$.[35]

[35] We often prefer the robust covariance of $\widehat{\boldsymbol{\beta}}$.

Let us continue with the Vlagtwedde-Vlaardingen example. Recall that we used the following model:

$$Y_{ij} = \eta_1 + \eta_2 t_{ij} + \eta_3 S_{ij} + \eta_4 t_{ij} S_i + e_{ij},$$

where t_{ij} are observed time point for the j-measurement of the i-th person, and S_i is a dummy variable for whether the i-th person is a smoker or not (1 = current, 0 = former).[36]

[36] Group means are as follows:

$$\text{Former: } E(Y_{ij}) = \eta_1 + \eta_2 t_{ij}$$

$$\text{Current: } E(Y_{ij}) = (\eta_1 + \eta_3) + (\eta_2 + \eta_4) t_{ij}$$

The quantities of interest are usually expressed as linear combination of elements $\boldsymbol{\beta}$. Specifically, we are interested in quantities of the form $\boldsymbol{L\beta}$, where $\boldsymbol{L}$ is a *known* matrix. Here are a few examples – recall that in our example, $\boldsymbol{\beta} = (\eta_1, \ldots, \eta_4)^T$:

1) Suppose we are interested *rate of change* in Y over time for "former" smokers. Thus we are interested in the following parameters: $\eta_2 = [0,1,0,0]\boldsymbol{\beta}$. Here $\boldsymbol{L} = [0,1,0,0]$.

2) Similarly, the *rate of change* in Y over time for "current" smokers is $\eta_2 + \eta_4 = [0,1,0,1]\boldsymbol{\beta}$.

3) Suppose we want to estimate $E(Y)$ for $t = 12$ in the current smokers ($S = 1$) group. Here we are interested in

$$(\eta_1 + \eta_3) + 12(\eta_2 + \eta_4) = [1,12,1,12]\boldsymbol{\beta}.$$

4) Suppose we want to test whether the mean trajectories for the two groups are same or not. We can write the hypotheses as $H_0 : \eta_3 = \eta_4 = 0$, or equivalently,

$$H_0 : \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

We will discuss about constructing confidence intervals of elements of $\boldsymbol{L\beta}$ and conducting hypothesis tests about $\boldsymbol{L\beta}$.

Confidence intervals

Recall that, we can write the result $\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \widehat{\boldsymbol{V}})$, *approximately*. Thus, for a known matrix $\boldsymbol{L}$, we can write *approximately*

$$\boldsymbol{L}\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{L\beta}, \boldsymbol{L}\widehat{\boldsymbol{V}}\boldsymbol{L}^T).$$

Using the result shown above, we can construct *approximate* (large sample) confidence intervals for a single component of β, say β_k, by setting

$$\boldsymbol{L} = [0,\ldots,0,1,0,\ldots,0],$$

where the entry "1" appears at the k-th position. Also $\boldsymbol{L}\widehat{\boldsymbol{V}}\boldsymbol{L}$ becomes the k-th diagonal entry of $\widehat{\boldsymbol{V}}$, that is $\widehat{var}(\widehat{\beta}_k)$. Thus a $100(1-\alpha)\%$ confidence interval of β_k is

$$\left[\widehat{\beta}_k \pm t_{df}(\alpha/2)\widehat{SE}(\widehat{\beta}_k)\right] \equiv \left[\widehat{\beta}_k \pm t_{df}(\alpha/2)\sqrt{\widehat{var}(\widehat{\beta}_k)}\right],$$

where the degrees of freedom *df* is the *residual degrees of freedom*. Specifically, df $= \sum m_i - p$, where p is the number of parameters in the mean formula.[37] This confidence interval can still be used if the data are not normally distributed, but the number of units n is large.

In general, if $\boldsymbol{L}$ is a row vector (a matrix with only one row),[38], we can construct $100(1-\alpha)\%$ confidence interval of $\boldsymbol{L\beta}$ as

$$\left[\boldsymbol{L}\widehat{\boldsymbol{\beta}} \pm t_{df}(\alpha/2)\widehat{SE}(\boldsymbol{L}\widehat{\boldsymbol{\beta}})\right] \equiv \left[\boldsymbol{L}\widehat{\boldsymbol{\beta}} \pm t_{df}(\alpha/2)\sqrt{\boldsymbol{L}\widehat{\boldsymbol{V}}\boldsymbol{L}^T}\right],$$

[37] Note that $\sum m_i$ is the total number of observations – the number of rows in data "long format" data set.

[38] Example items 1, 2 and 3 above

If $\boldsymbol{L}$ is a matrix with multiple rows[39], then we can create intervals *for each row* of $\boldsymbol{L}$ separately.

For the Vlagtwedde-Vlaardingen example, suppose we want to construct 95% confidence intervals for the mean parameters $\boldsymbol{\beta}$ *using the robust covariance matrix* of $\widehat{\boldsymbol{\beta}}$. We will also use the model fit using compound symmetry with equal variance over time for errors. Note the computation of the degrees of freedom, df, below.[40]

[39] As in Example item 4 above.

[40] We can also see the *residual degrees of freedom* from the last line of the gls() output by calling *summary()*.

```
# Compound symmetry with equal variance fit
meanform <- FEV1 ~ time + smoker + time:smoker
gls.fit.exch <- gls(model = meanform, data = smoking,
                    correlation = corCompSymm(form = ~1 | id))
betahat <- gls.fit.exch$coefficients
# Robust covariance matrix and se of betahat
V.robust <- vcovCR(gls.fit.exch, type = "CR0")
se.robust <- sqrt(diag(V.robust))
# CI limits
df <- nrow(smoking) - length(betahat)
t.alpha <- qt(0.05/2, df = df, lower.tail = F)
lower <- betahat - t.alpha*se.robust
upper <- betahat + t.alpha*se.robust
# Display the estimates
tab <- rbind(betahat, se.robust, lower, upper)
tab
```

```
##           (Intercept)        time      smoker  time:smoker
## betahat      3.507677 -0.03385154 -0.27267604 -0.004570281
## se.robust    0.108024  0.00330702  0.12129015  0.003719316
## lower        3.295620 -0.04034343 -0.51077609 -0.011871527
## upper        3.719735 -0.02735966 -0.03457598  0.002730966
```

The rows "lower" and "upper" give the lower and upper limits of the 95% CI of elements of $\boldsymbol{\beta}$.

As another demonstration, suppose we want we want to estimate $E(Y)$ for $t = 12$ in the current smokers ($S = 1$) group.[41] We have already established that our quantity of interest is $(\eta_1 + \eta_3) + 12(\eta_2 + \eta_4) = [1, 12, 1, 12]\boldsymbol{\beta}$. We can create a 95% confidence interval as follows *using the robust covariance matrix* of $\widehat{\boldsymbol{\beta}}$.[42]

[41] Example item 4 as discusses before.

[42] Recall we have comtuted the robust covariance of $\widehat{\boldsymbol{\beta}}$ in "V.robust" in the previous code block.

```
# L matrix
L <- matrix(c(1, 12, 1, 12), nrow = 1)
# Estimate and SE
estimate <- L %*% betahat
SE <- sqrt( L %*% V.robust %*% t(L) )
# confidence limits
df <- nrow(smoking) - length(betahat)
```

```
t.alpha <- qt(0.05/2, df = df, lower.tail = F)
lower <- estimate - t.alpha*SE
upper <- estimate + t.alpha*SE
# results
tab <- data.frame(estimate, SE, lower, upper)
round(tab, 4)
```

```
##   estimate     SE  lower  upper
## 1   2.7739 0.0513 2.6732 2.8747
```

Hypothesis tests

Suppose we want to test $H_0 : \beta_k = 0$ versus the alternative $H_1 : \beta_k \neq 0$. We can still use the result that $\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \widehat{\boldsymbol{V}})$, *approximately*.[43] We can then use the approximate *t-test* statistic:

$$t = \frac{\widehat{\beta}_k - 0}{\widehat{SE}(\widehat{\beta}_k)} = \frac{\widehat{\beta}_k - 0}{\sqrt{\widehat{var}(\widehat{\beta}_k)}}.$$

We reject H_0 if $|t| > t_{\text{df}}(\alpha/2)$, where he degrees of freedom *df* is the *residual degrees of freedom*, as defined previously. The corresponding *p-value* is

$$\text{p} - \text{value} = 2[Pr(T_{\text{df}} > |t|)].$$

For our example, the test results of the individual components are shown below.[44]

[43] Consequantly, we can write, *approximately*,

$$\widehat{\beta}_k \sim N(\beta_k, var(\widehat{\beta}_k)).$$

[44] Recall that in the previous code blocks, "betahat" stores the estimated $\widehat{\boldsymbol{\beta}}$, "se.robust" stored the robust standard errors.

```
# df
df <- nrow(smoking) - length(betahat)
# t stats
t.robust <- betahat/se.robust
# p-values
p.value <- round( 2*pt(q = abs(t.robust), df = df, lower.tail = FALSE), 4 )
# results
tab <- data.frame(betahat, se.robust, t.robust, p.value)
tab
```

```
##                  betahat   se.robust   t.robust p.value
## (Intercept)  3.507677331 0.108024006  32.471276  0.0000
## time        -0.033851544 0.003307020 -10.236268  0.0000
## smoker      -0.272676037 0.121290153  -2.248130  0.0249
## time:smoker -0.004570281 0.003719316  -1.228796  0.2195
```

Notice that we have used the robust standard errors to construct the t-test. We can also use the model based standard errors[45] to obtain the corresponding test results.

[45] The model based standard errors is used by default in gls() to create the t-test table in the gls() output.

More generally, it may be of interest to construct tests that concern certain linear combinations of the components of $\boldsymbol{\beta}$, that is, $\boldsymbol{L\beta}$. Specifically, we want to test

$$H_0 : \boldsymbol{L\beta} = \boldsymbol{h} \text{ vs. } H_a : \boldsymbol{L\beta} \neq \boldsymbol{h},$$

where h is a known vector. More often than not, h is taken to be 0. In this case, we can use the *Wald test* or the *F-test*. The tests are described below.

- **Wald test:** The Wald test statistic is[46]

$$W = (\boldsymbol{L\widehat{\beta}} - \boldsymbol{h})^T (\boldsymbol{L\widehat{V}L}^T)^{-1} (\boldsymbol{L\widehat{\beta}} - \boldsymbol{h}).$$

  Under H_0, we can show that

$$W \sim \chi^2_c,$$

  where c is the *number of independent rows in* $\boldsymbol{L}$ *matrix*[47], that is, the number of parameters are being tested simultaneously. The p-value can be computed as p-value $= Pr(\chi^2_c > W)$.

- **F-test:** Wald test can sometimes have higher type I error since this test does not take into account variability associated with estimating the error variance components ω in the specification of $\boldsymbol{\Sigma}$. The *F-test* provides a more reliable alternative to the Wald test. The test statistic is

$$F = \frac{(\boldsymbol{L\widehat{\beta}} - \boldsymbol{h})^T (\boldsymbol{L\widehat{V}L}^T)^{-1} (\boldsymbol{L\widehat{\beta}} - \boldsymbol{h})}{c},$$

  where c is the *number of independent rows in* $\boldsymbol{L}$ *matrix* as defined above. Under H_0, we have that $F \sim F_{c,\text{df}}$, where df is the denominator degrees of freedom that is *estimated from the data*. There are several methods to estimate df[48] We will simply use df = *residual degrees of freedom*[49] as before. The corresponding p-value is p-value $= Pr(F_{c,\text{df}} > F)$.

[46] Recall that we have already established that $\boldsymbol{L\widehat{\beta}} \sim N[\boldsymbol{L\beta}, \boldsymbol{L\widehat{V}L}^T]$, approximately.

[47] This is the *rank* of $\boldsymbol{L}$ matrix.

[48] For example, the Satterthwaite method or the Kenward-Roger method – we will not discuss these methods.

[49] This is default in gls().

In our example, let us consider the problem of testing whether the mean trajectories for the two groups ("Former" and "current") are same or not. We can write the hypotheses as $H_0 : \eta_3 = \eta_4 = 0$, or equivalently,

$$H_0 : \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The Wald and F-tests are as follows.

```
# L matrix
L <- rbind(c(0, 0, 1, 0), c(0, 0, 0, 1))
```

```
cc <- nrow(L)
df <- nrow(smoking) - length(betahat)
# estimate and covariance matrix of L\beta
est <- L %*% betahat
varmat <- L %*% V.robust %*% t(L)
# Wald test
Wald <- c( t(est) %*% solve(varmat) %*% (est) )
p.value <- pchisq(q = Wald, df = cc, lower.tail=FALSE)
data.frame(Wald, p.value)
```

```
##       Wald     p.value
## 1 9.613951 0.008172541
```

```
# F-test
Fstat <- c( t(est) %*% solve(varmat) %*% (est) ) / cc
p.value <- pf(q = Fstat, df1 = cc, df2 =  df, lower.tail=FALSE)
data.frame(Fstat, p.value)
```

```
##      Fstat     p.value
## 1 4.806975 0.008420397
```

In general, the Wald and F-test methods may produce different results. However, when sample size n is large, the corresponding p-values become similar since the residual degrees of freedom become sufficiently large.

Information criteria

The information criteria provide an *informal approach* of comparing competing models, especially *multiple covariance models*.[50] These criteria are especially usefull when the competing models are *not nested*, that is, one model is not a special case of the other. We can use the information criteria to do the following:

[50] Formal testing of variance components will be discussed in next chapter.

- compare multiple covariance models, e.g., AR(1) vs. CS,
- compare two competing models for *mean trends* that *involve different combinations of covariates* and are not nested,
- a combination of the above two situations.

The information criteria are constructed based on the idea of balancing between the goodness of fit and the number of parameters involved in the model. We know that a larger model[51] will produce a better fit, that is, a larger likelihood. Thus to compare two models, we need to look at both the likelihood value *and* the number of parameters in the model.[52] Thus the information criteria are essentially

[51] Model involving more paramaters

[52] Comparing just the maximized loglikelihoods directly will always favor larger models.

penalized version of the maximized (log-)likelihood. We often use the following criteria:

- *Akaike's Information Criterion (AIC):*

$$AIC = -2\widehat{\ell}_{ML} + 2c$$

  where $\widehat{\ell}_{ML}$ is the maximized or fitted (ML) log-likelihood using the assumed model and c is the number of parameters included in this model, including both mean parameters and variance components.

- *Schwarz's Bayesian Information Criterion (BIC):*

$$BIC = -2\widehat{\ell}_{ML} + (\log N)c$$

  where c is the number of parameters included in the model of interest, and N is the total number of observations in the data $N = \sum_{i=1}^{n} m_i$. Because BIC penalizes drastically the number of components in the model, it *tends to select the most parsimonious (simplistic) model*.

Among all the models of interest, the one with the smallest AIC or BIC is preferred.

Comparing covariance models with same mean trend

We can construct the AIC and BIC based on "REML" loglikelihoods as well to compare competing covariance models **only when we use the same mean formulation in both models**.

In our example, we do have the *same mean model*

$$E(Y_{ij}) = \eta_1 + t_{ij}\eta_2 + S_i\eta_3 + S_i t_{ij}\eta_4,$$

but investigate three different covariance models.[53]

[53] The results were saved in outputs "gls.fit.exch", "gls.cs.uv" and "gls.un". The AIC and BIC for these models are shown below.

```
AIC(gls.fit.exch, gls.cs.uv, gls.un)
```

```
##              df      AIC
## gls.fit.exch  6 323.6031
## gls.cs.uv    12 331.1515
## gls.un       32 330.0506
```

```
BIC(gls.fit.exch, gls.cs.uv, gls.un)
```

```
##              df      BIC
## gls.fit.exch  6 351.4581
## gls.cs.uv    12 386.8613
## gls.un       32 478.6102
```

We can see that the compound symmetry model with equal variance ("gls.fit.exch") gives the smallest AIC and BIC. **Make note that AIC/BIC should not be used for formal hypothesis testing.**

Small sample inference

The inferential procedures (CI and p-value) we discussed so far are developed using asymptotic (large sample) theory. These results may not be accurate when sample size (n) is small. Specifically, the large sample results can be used when $n - p$ is large. Otherwise, we might need to use resampling techniques.

One major impact of having a small sample size is the calculation of degrees of freedom of the tests. A approximation used by `gls` can be unreliable when n is small. To this end, we can use **Kenward-Roger approximation**.[54] Specifically, Kenward and Roger (1997) developed an approximation that modify the usual F (or T) test statistic such that

- $cov(\widehat{\beta})$ is estimated using a small sample approximation
- The F-statistic is scaled approproately to account for small sample size
- The *corrected* error degrees of freedom is calculated accordingly.

The last two steps are done so that the expected value of the test statistic match approximately to that of an F distribution.

The package *`lavaSearch2`* implements an adaptation of Kenward-Roger approximation for *ML based testing* (for regression coeffi-cients). *We need to refit our model using ML*, and use a new function *`summary2()`*.

We demonstrate the Kenward-Roger approximation below.[55]

[54] Kenward, M., & Roger, J. (1997). Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood. Biometrics, 53(3), 983-997. doi:10.2307/2533558

[55] Note that we have used *method = "ML"* to get the ML estimates.

```
library(lavaSearch2)
gls.fit.ML <- gls(model = FEV1 ~ time + smoker + time:smoker,
                  data = smoking, correlation = corCompSymm(form= ~ 1 | id ),
                  method = "ML")
summary2(gls.fit.ML)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: FEV1 ~ time + smoker + time:smoker
##   Data: smoking
##        AIC      BIC    logLik
##   295.4603 323.3465 -141.7302
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | id
##  Parameter estimate(s):
##       Rho
## 0.8580072
```

```
##
## Coefficients:
##                 Value  Std.Error   t-value p-value       df
## (Intercept)  3.507698 0.09806414  35.76943  0.0000 150.2038
## time        -0.033853 0.00264242 -12.81153  0.0000 651.2183
## smoker      -0.272687 0.11242292  -2.42554  0.0165 149.6335
## time:smoker -0.004568 0.00303420  -1.50553  0.1327 651.0930
##
##  Correlation:
##             (Intr) time   smoker
## time        -0.252
## smoker      -0.872  0.220
## time:smoker  0.219 -0.871 -0.247
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.99688542 -0.65556536  0.01900769  0.69212175  3.17079532
##
## Residual standard error: 0.5672213
## Degrees of freedom: 771 total; 767 residual
```

Notice the extra *df* column in the t-test table in the "Coefficients" block. Since we have a fairly large sample size, the corrected tests do not differ too much fom our original results.[56]

We can also perform an F-test using the function `compare2()`. Suppose we want to test the "time:smoker" interaction.

[56] Notice that the tests are using the model based covariance of $\widehat{\beta}$. We can also replace these by robust covariance and construct corresponding tests.

```
compare2(gls.fit.ML, par = c("time:smoker = 0"))
```

```
##
##  - Wald test -
##
##  Null Hypothesis:
##  [time:smoker] = 0
##
## data:
## F-statistic = 2.2666, df1 = 1, df2 = 651.09, p-value = 0.1327
## sample estimates:
##                       Estimate   Std.Err      df        2.5%       97.5%
## [time:smoker] = 0 -0.004568065 0.0030342 651.093 -0.01052606 0.001389933
```

Notice that the argument `par = c("time:smoker = 0")` directly specifies which parameter we want to test. We can also test multiple parameters (and contrasts) together, as shown below.

```
compare2(gls.fit.ML, par = c("smoker=0", "time:smoker = 0"))
```

```
##
##  - Wald test -
##
##  Null Hypothesis:
##  [smoker] = 0
##  [time:smoker] = 0
##
## data:
## F-statistic = 5.2973, df1 = 2, df2 = 247.12, p-value = 0.005589
## sample estimates:
##                       Estimate   Std.Err       df        2.5%        97.5%
## [smoker] = 0      -0.272686554 0.1124229 149.6335 -0.49482802 -0.050545087
## [time:smoker] = 0 -0.004568065 0.0030342 651.0930 -0.01052606  0.001389933
```

For more such examples, see `?compare2`.

Final Remarks: main features and limitations

When confronted with a real data application, an important step is the selection of the appropriate covariance model. Such a covariance structure incorporates both sources of variation (among-units and between-units). Useful ideas in the selection of the covariance model are as follows.

1. Informal graphical/numerical summaries and other techniques may be used on a preliminary fit using OLS estimates of the regression parameters. [57]
2. We may use AIC and BIC, but some subjectivity is also involved, since different criteria might produce different models. Also, information criteria should not be used to conduct formal hypothesis testing – they should only be used as rules of thumb.
3. If no covariance model is truly appropriate, we can still rely on the estimates of $\widehat{\boldsymbol{\beta}}$ and use the robust empirical covariance of $\widehat{\boldsymbol{\beta}}$ to make an inference.[58]

[57] See *Introduction* chapter.

[58] The models used in the next chapter offer an alternative approach.

The following are some essential features of the regression approach:

1. The mean can be modeled smoothly over time; the rate of change is the slope of this function. Modeling of the mean in this fashion allows estimation of the mean at any time point, not just the observed times.
2. We can incorporate multiple groups/populations by appropriately manipulating the design matrix. Recall the explicit parameterization and the difference parameterizations.

3. This approach does not require a balanced time points design: the vectors of observations may have different lengths. One important aspect we should be aware of is the *missingness mechanism*. If the missingness is entirely unrelated to the issues under study[59] (e.g., a sample of a specific subject at a particular time is mistakenly destroyed/misplaced), then the analysis is ok. However, if the missingness is related to issues under study (e.g., two treatments are compared, and in one treatment a subject does not show up because they are too ill), then the missingness might contain information about the treatment. Then this analysis would not be valid.

[59] Often called *Missing Completely at Random (MCAR)*

Some limitations of this methodology are as follows.

1. The main focus is modeling of the mean trajectories over time; the reconstruction of the individual trajectories is not considered. Characterizing the subject trajectories may be of interest – the current framework does not allow such a study.
2. The modeling of the covariance matrix aggregates the two sources of variation: within-subject variation and between subjects variation. This approach does not allow analysts to understand the two sources separately. We will discuss this issue more in the next chapter.

Various references:

1. Modeling Longitudinal Data by Robert E. Weiss. New York: Springer.
2. Linear Mixed Models for Longitudinal Data by Geert Verbeke and Geert Molenberghs. New York: Springer.
3. Applied Longitudinal Analysis by Fitzmaurice by G.M., Laird, N.M., and Ware, J.H. New York: Wiley.