Satterthwaite's Method for Degrees of Freedom in Linear Mixed Models

Rune Haubo B Christensen

Jan 2018 – (last edit: 2020-04-15)

Contents

Introduction

Satterthwaite's method for degrees of freedom has many applications with the Welch-Satterthwaite two-sample t-test allowing for different variances as the most well-known example. Confidence intervals for linear functions of mean squares is another example as is t and F-tests for contrasts of the mean-value parameters or so-called fixed-effects in linear mixed models which is the topic of this note.

In applications of linear mixed models to balanced datasets it is often possible to derive known integer denominator degrees of freedom for t-statistics for individual parameters (or one-dimensional contrasts) and F-statistics for model terms (or multi-dimensional contrasts). These statistics are convenient since they follow exact t and F-distributions thus exempting the need for asymptotic Wald or likelihood ratio tests whose small-sample behavior is dubious and often anti-conservative – even for rather large samples. [We can think of the Wald test as an F-test with infinite denominator df so in experiments which are balanced except for one or a few missing observations the denominator df from the corresponding balanced experiment provides an upper bound for the denominator df with the unbalanced dataset. The likelihood ratio test can be viewed as a small adjustment to the Wald test but otherwise with the same caveats.] But balance is a fragile aspect which is easily lost due to a missing observation even in well-designed experiments. In such cases it is concerning and inconvenient having to resort to the asymptotic tests when intuitively we would expect an *approximate* F-test to be much more accurate. Essentially, Satterthwaite's method produces such an approximate t or F-test and if in fact an exact F-test exists this is produced. Simulations [1, 2] have shown these approximate F-tests to be an important improvement over the asymptotic tests in many situations.

In this note we describe the development of Satterthwaite's method [3] for estimation of denominator degrees of freedom in t and F-test for contrasts of mean-value parameters in linear mixed models. It is worth noting that the Welch-Satterthwaite two-sample t-test is in fact a special case of the more general case presented

1

here. We also exemplify how the actual computations can be done using the implementation of linear mixed models in the **lme4** package [4] for R.

In the remainder of this section we briefly introduce a class of multivariate normal models and the class of linear mixed models. We also briefly introduce the t-test for one-dimensional contrasts of the mean-value parameters.

In the next section we describe the development of Satterthwaite's method for estimation of denominator degrees of freedom in t-tests of one-dimensional contrasts. After describing how we address the computational challenges involved we present an example where t-tests of the mean-value parameters are evaluated in an unbalanced dataset.

The following section extends the application of Satterthwaite's method for denominator degrees of freedom from one-dimensional contrasts leading to t-tests on one numerator degree of freedom to multi-dimensional contrasts leading to F-tests on multiple numerator degrees of freedom. Also here is an example given on an unbalanced dataset.

The multivariate normal and linear mixed models

Satterthwaite's method for denominator degrees of freedom as presented here is not only applicable to linear mixed models, but also to the more general class of multivariate normal models of which linear mixed models are a special case. The multivariate normal model can be written

$$Y = X\beta + \varepsilon$$

with $\varepsilon \sim N(0, V_\tau)$, X being a $n \times p$ design matrix, V_τ being the variance-covariance matrix of the residuals ε (as well as of the observations Y), and τ is the (usually small) k-vector of unique variance-covariance parameters. The p-vector of mean-value parameters β can be profiled out of the likelihood function since the estimate $\hat{\beta}_\tau$ given a value for τ can be expressed by the generalized least squares (GLS) estimator as the solution to

$$X^\top V_\tau^{-1} X \hat{\beta} = X^\top V_\tau^{-1} y$$

where y is the observed value of Y. The asymptotic variance-covariance matrix of $\hat{\beta}$ is

$$\mathsf{V}_\tau(\hat{\beta}) = \left(X^\top V_\tau^{-1} X \right)^{-1}$$

which is a function of τ (only) and denoted $\mathsf{V}_{\hat{\tau}}(\hat{\beta})$ when evaluated at the (ML or REML) estimate $\hat{\tau}$. Note that the distribution of Y and therefore the likelihood function for the model, only depends on the, usually small, parameter vector, τ.

The linear mixed model may be written

$$Y = X\beta + Zb + \epsilon$$

with random-effects $b \sim N(0, G_\tau)$ and residuals $\epsilon \sim N(0, R_\tau)$. Noting that $\mathsf{E}(Y) = X\beta$ and $\mathsf{V}(Y) \equiv V_\tau = ZG_\tau Z^\top + R_\tau$ we essentially have the multivariate normal model with some structure imposed on V_τ with the same GLS estimator of β and its asymptotic variance-covariance matrix.

Tests of vector contrasts

In this context we are interested in assessing a hypothesis concerning a linear function of the mean-value or fixed-effect parameters:

$$H_0 : L^\top \beta = 0$$

where $L^\top$ is a known contrast vector, we will consider the t-statistic

$$t = \frac{L^\top \hat{\beta}}{\sqrt{L^\top \mathsf{V}_{\hat{\tau}}(\hat{\beta}) L}}$$

This statistic will in general not follow an exact t-distribution, but we may choose the degrees of freedom, ν, so that it follows approximately a t-distribution. The next section describes a method known as Satterthwaite's method for choosing or approximating ν.

Satterthwaite's method for t-statistics

The method of finding ν attributed to Satterthwaite (1946) [3] begins by matching the t-statistic above with a variable following an exact t-distribution, so we begin with the definition of the Student's t-distribution (see for example this wiki page):

The random variable

$$t_\nu = \frac{Z}{\sqrt{V/\nu}}$$

is t_ν-distributed with ν degrees of freedom if $Z \sim N(0,1)$, $V \sim \chi^2_\nu$ and Z and V are independent.

Considering the t-variable above, we divide both numerator and denominator with the (true, but unknown) variance of $L^\top \hat{\beta}$:

$$t = \frac{L^\top \hat{\beta}}{\sqrt{L^\top \mathsf{V}_{\hat{\tau}}(\hat{\beta})L}} = \frac{L^\top \hat{\beta}/\sqrt{L^\top \mathsf{V}_\tau(\hat{\beta})L}}{\sqrt{L^\top \mathsf{V}_{\hat{\tau}}(\hat{\beta})L}\Big/\sqrt{L^\top \mathsf{V}_\tau(\hat{\beta})L}} = \frac{L^\top \hat{\beta}/\sqrt{L^\top \mathsf{V}_\tau(\hat{\beta})L}}{\sqrt{L^\top \mathsf{V}_{\hat{\tau}}(\hat{\beta})L/L^\top \mathsf{V}_\tau(\hat{\beta})L}}$$

Now assuming that $L^\top \hat{\beta}$ is normally distributed with expectation $\mathsf{E}[L^\top \hat{\beta}] = 0$ and (true, unknown) variance $\mathsf{V}[L^\top \hat{\beta}] = L^\top \mathsf{V}_\tau(\hat{\beta})L$ leads to

$$L^\top \hat{\beta}/\sqrt{L^\top \mathsf{V}_\tau(\hat{\beta})L} \sim N(0,1)$$

under the null hypothesis. Matching the t-variable with the definition of the t-distribution leads us to assume that

$$\frac{\nu(L^\top \mathsf{V}_{\hat{\tau}}(\hat{\beta})L)}{L^\top \mathsf{V}_\tau(\hat{\beta})L} \sim \chi^2_\nu$$

is approximately χ^2-distributed with ν degrees of freedom. Note that this statistic has the familiar form $\nu S^2/\sigma^2$ which is χ^2_ν-distributed if $(\nu + 1)$ independent normally distributed random variables has variance σ^2 and sum-of-squares S^2.

Denoting $S^2 = L^\top \mathsf{V}_{\hat{\tau}}(\hat{\beta})L$ and $\sigma^2 = L^\top \mathsf{V}_\tau(\hat{\beta})L$ we have that (approximately)

$$S^2 \sim \frac{\sigma^2}{\nu}\chi^2_\nu$$

with expectation

$$\mathsf{E}(S^2) = \mathsf{E}\left(\frac{\sigma^2}{\nu}\chi^2_\nu\right) = \frac{\sigma^2}{\nu}\mathsf{E}(\chi^2_\nu) = \frac{\sigma^2}{\nu}\nu = \sigma^2$$

and variance

$$\mathsf{V}(S^2) = \left(\frac{\sigma^2}{\nu}\right)^2 \mathsf{V}(\chi^2_\nu) = \left(\frac{\sigma^2}{\nu}\right)^2 2\nu = \frac{2(\sigma^2)^2}{\nu}$$

The expectation doesn't bring anything new to the table, but isolating ν the expression for the variance leads to the following estimator for ν (depending on the data through τ):

$$\hat{\nu}(\tau) = \frac{2(\sigma^2)^2}{\mathsf{V}(S^2)} = \frac{2(L^\top \mathsf{V}_\tau(\hat{\beta})L)^2}{\mathsf{V}(L^\top \mathsf{V}_{\hat{\tau}}(\hat{\beta})L)}$$

We may obtain the estimate $\hat{\nu}(\hat{\tau})$ (a value, for $\hat{\nu}(\tau)$) by evaluating at $\tau = \hat{\tau}$. This is straight forward for the numerator, but we don't directly have the denominator so we need an extra step here.

Noting that $L^\top \mathsf{V}_\tau(\hat\beta)L \equiv f(\tau)$ is a function of τ and using the delta method we may approximate the variance of $f(\tau)$ at $\tau = \hat\tau$ by

$$\mathsf{V}(f(\hat\tau)) \approx f'(\hat\tau)^\top \mathsf{V}(\hat\tau)f'(\hat\tau)$$

where $f'(\hat\tau) = f'(\tau)|_{\tau=\hat\tau}$ is the vector of derivatives of $f(\tau)$ wrt. τ evaluated at $\hat\tau$ and $\mathsf{V}(\hat\tau)$ is the asymptotic variance-covariance matrix of the τ-vector evaluated at $\hat\tau$ (which can be obtained (numerically) as the inverse Hessian of the negative log-likelihood function).

Finally, we may evaluate the estimator of ν at $\hat\tau$ to obtain

$$\hat\nu(\hat\tau) = \frac{2(L^\top \mathsf{V}_{\hat\tau}(\hat\beta)L)^2}{f'(\hat\tau)^\top \mathsf{V}(\hat\tau)f'(\hat\tau)} = \frac{2f(\hat\tau)^2}{f'(\hat\tau)^\top \mathsf{V}(\hat\tau)f'(\hat\tau)}$$

Instead of directly evaluating the gradient of $f(\hat\tau) = L^\top \mathsf{V}_{\hat\tau}(\hat\beta)L$ we write the gradient as

$$f'(\hat\tau) = L^\top \{\nabla_\tau \mathsf{V}_\tau(\hat\beta)\}L \big|_{\tau=\hat\tau}$$

Here the Jacobian $\mathcal{J} \equiv \nabla_\tau \mathsf{V}_\tau(\hat\beta)$ is the 3-dimensional array with k faces of size $p \times p$. Left and right multiplying each face by $L^\top$ and L respectively (where L is $p \times 1$) leads to the k-vector $f'(\hat\tau)$.

This has the advantage that the gradient is not tied to a particular L and $\mathcal{J}$ may be re-used for other contrast matrices, and, as we shall see later, can also be used for multi-degree of freedom F-tests.

Computational approach

The computational challenge is essentially to evaluate the denominator in the expression for $\hat\nu(\hat\tau)$, which amounts to computing the gradient $f'(\hat\tau)$ and variance-covariance matrix of the variance parameter vector $\mathsf{V}(\hat\tau)$. We will now address these challenges in turn starting with the latter.

The variance-covariance matrix of the variance parameter vector $\mathsf{V}(\hat\tau)$ can be computed as the inverse of Hessian of the negative log-likelihood function with respect to the variance-parameters. Conveniently the **lme4** package provides a simple way to extract a deviance function from a linear mixed model fit. The deviance is a scaled version of the log-likelihood function for the model so numerically computing the Hessian of the deviance function provides an estimate of the covariance matrix of the variance parameters. One minor challenge is that the residual variance, σ is profiled out of the deviance function as implemented in **lme4**, so the deviance function cannot be used directly as it comes. A variant of the deviance function has to be written that operates on the variance parameters, τ. Using the `hessian` function from the **numDeriv** package [5] conveniently and accurately evaluates the Hessian numerically.

The variance-covariance matrix of β can also be expressed as a function of the variance parameters using the same deviance function. Using the `jacobian` function from **numDeriv** allows us to numerically evaluate $\mathcal{J}$ at $\hat\tau$. Having computed $\mathsf{V}(\hat\tau)$ and $\mathcal{J}$ its only a matter of putting the different quantities together to compute the estimate of the denominator degrees of freedom.

Example: Computing denominator df for a t-test using `lme4::lmer`

In this example we will consider the `ham` dataset from the **lmerTest** package [1] in which 81 consumers evaluated 4 products twice leading to 648 observations. The dataset is balanced, but we randomly select and use 580 observations (corresponding to approximately 90%) making the dataset unbalanced. An initial fit of the data using `lmer` from the **lme4** package reads:

```
library(lme4)
## Loading required package: Matrix
data(ham, package="lmerTest")
set.seed(12345)
model <- lmer(Informed.liking ~ Product + (1|Consumer),
              data = ham[sample(x=1:nrow(ham), size=580, replace=FALSE), ])
```

```
summary(model, corr=FALSE)
## Linear mixed model fit by REML ['lmerMod']
## Formula: Informed.liking ~ Product + (1 | Consumer)
##    Data: ham[sample(x = 1:nrow(ham), size = 580, replace = FALSE), ]
##
## REML criterion at convergence: 2560.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4794 -0.6766  0.1447  0.7477  2.4066
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Consumer (Intercept) 0.7408   0.8607
##  Residual             4.3152   2.0773
## Number of obs: 580, groups:  Consumer, 81
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  5.84346    0.19790  29.528
## Product2    -0.66197    0.24431  -2.709
## Product3     0.20102    0.24476   0.821
## Product4     0.08105    0.24656   0.329
```

Suppose now that we are interested in the contrast

```
L <- c(0, 1, 0, 0)
```

which simply picks out the second coefficient; the estimate of the difference between product 2 and product 1.

The estimate of this contrast $L^\top \beta$ is then computed with:

```
(estimate <- drop(t(L) %*% fixef(model)))
## [1] -0.6619655
```

Computing the variance-covariance matrix of the variance parameters

In this model the variance-covariance parameters are collected in the 2-vector $\tau = [\sigma_c, \sigma]$; the square root of the random-effect variance for consumers and the residual standard deviation. `lmer` however, profiles out the residual standard deviation of the (restricted) profile likelihood and operates with vector θ of *relative* variance parameters. In this model $\theta = \sigma_c/\sigma$.

The parameters that characterize the model can be summarized as

```
(parlist <- list(beta=fixef(model),
                 theta=getME(model, "theta"),
                 sigma=sigma(model)))
## $beta
## (Intercept)    Product2    Product3    Product4
##   5.84345560 -0.66196549  0.20102196  0.08105489
##
## $theta
## Consumer.(Intercept)
##            0.4143353
##
## $sigma
## [1] 2.077314
```

5

and the consumer random effect standard deviation can be retrieved with

```
with(parlist, theta*sigma)
## Consumer.(Intercept)
##          0.8607045
```

A function that directly evaluates the model deviance (REML or ML criterion) can be obtained directly from the model fit:

```
devfun <- update(model, devFunOnly=TRUE)
```

Being a function of θ, we can calculate the deviance at $\hat{\theta}$ with

```
devfun(parlist$theta)
## [1] 2563.524
```

To compute the variance-covariance matrix of τ by numerically evaluating the Hessian of the log-likelihood function, we need to re-implement the deviance function as a function of the unique variance-covariance parameters (τ) – not only the relative variance parameters (θ). Utilizing the deviance function as a function of θ and some particulars of the implementation of linear mixed models in `lmer` as described in [4] we can implement this as:

```
devfun_varpar <- function(varpar, devfun, reml) {
  # Computes deviance as a function of 'varpar=c(theta, sigma)'
  # devfun: deviance function as a function of theta only.
  # reml: TRUE if REML; FALSE if ML
  nvarpar <- length(varpar)
  sigma2 <- varpar[nvarpar]^2
  theta <- varpar[-nvarpar]
  df_envir <- environment(devfun)
  devfun(theta) # Evaluate deviance function at varpar
  n <- nrow(df_envir$pp$V)
  # Compute deviance for ML:
  dev <- df_envir$pp$ldL2() + (df_envir$resp$wrss() + df_envir$pp$sqrL(1))/sigma2 +
    n * log(2 * pi * sigma2)
  if(!reml) return(dev)
  # Adjust of REML is used:
  RX <- df_envir$pp$RX() # X'V^{-1}X ~ crossprod(RX^{-1}) = cov(beta)^{-1} / sigma^2
  dev + 2*c(determinant(RX)$modulus) - ncol(RX) * log(2 * pi * sigma2)
}
```

This function returns the same deviance but as a function of a different parameter vector:

```
is_reml <- getME(model, "is_REML")
varpar_opt <- unname(c(parlist$theta, parlist$sigma))
devfun_varpar(varpar_opt, devfun, reml=is_reml)
## [1] 2563.532
```

We can now obtain the hessian for τ of the deviance function with the numerical approximation provided in the **numDeriv** package:

```
library(numDeriv)
h <- hessian(func=devfun_varpar, x=varpar_opt, devfun=devfun, reml=is_reml)
```

Before inverting `h`, we check that it is positive definite by confirming that all its eigenvalues are positive:

```
eig_h <- eigen(h, symmetric=TRUE)
eig_h$values
## [1] 795.7477 353.1973
```

```
stopifnot(all(eig_h$values > 0))
```

Inverting and scaling `h` provides the asymptotic variance-covariance matrix of the variance parameters at their estimate $V(\hat{\tau})$:

```
h_inv <- with(eig_h, vectors %*% diag(1/values) %*% t(vectors))
(cov_varpar <- 2 * h_inv)
##              [,1]         [,2]
## [1,]  0.003828133 -0.001553014
## [2,] -0.001553014  0.004347785
```

Computing the gradient of $f(\hat{\tau})$

To compute $f'(\hat{\tau}) = L^\top \{\nabla_\tau V_\tau(\hat{\beta})\} L \big|_{\tau = \hat{\tau}}$ we first implement $V_\tau(\hat{\beta})$ as a function of τ:

```
get_covbeta <- function(varpar, devfun) {
  # Compute cov(beta) as a function of varpar
  #
  # varpar: c(theta, sigma)
  # devfun: deviance function, ie. update(model, devFunOnly=TRUE) - a function of theta
  # return: cov(beta) given specified varpar
  #
  nvarpar <- length(varpar)
  sigma <- varpar[nvarpar] # residual std.dev.
  theta <- varpar[-nvarpar] # ranef var-par
  devfun(theta) # evaluate REML or ML deviance 'criterion'
  df_envir <- environment(devfun) # extract model environment
  sigma^2 * tcrossprod(df_envir$pp$RXi()) # vcov(beta)
}
```

Then evaluate the gradient (Jacobian), $\mathcal{J} = \nabla_\theta V_\theta(\hat{\beta})$ numerically using the `jacobian` function from the **numDeriv** package and organize it as a list (of length k) of $p \times p$ matrices:

```
Jac <- jacobian(func=get_covbeta, x=varpar_opt, devfun=devfun)
Jac_list <- lapply(1:ncol(Jac), function(i)
  array(Jac[, i], dim=rep(length(parlist$beta), 2))) # k-list of jacobian matrices
```

Left and right multiplying each matrix by $L^\top$ and L respectively gives the gradient vector (of length k):

```
(grad_var_Lbeta <- vapply(Jac_list, function(x)
  sum(L * x %*% L), numeric(1L))) # = {L' Jac L}_i
## [1] 0.0008652581 0.0580662870
```

Putting it all together

We now have all elements for the denominator of $\hat{\nu}(\hat{\tau})$. The only element in the numerator, the estimated covariance matrix of the contrast $L^\top \beta$ is then simply $V(L^\top \hat{\beta}) = L^\top V_{\hat{\tau}}(\hat{\beta}) L$ is simple to evaluate since `vcov(model)` extracts $V_{\hat{\tau}}(\hat{\beta})$:

```
cov_beta <- as.matrix(vcov(model))
# Compute vcov(Lbeta)
(var_Lbeta <- drop(t(L) %*% cov_beta %*% L))
## [1] 0.05968911
# Alternative: (var_con <- get_covbeta(varpar_opt, devfun))
```

Collecting all elements gives the following coefficient table:

```
se.estimate <- sqrt(var_Lbeta)
satt_denom <- sum(grad_var_Lbeta * (cov_varpar %*% grad_var_Lbeta)) # g'Ag
ddf <- drop(2 * var_Lbeta^2 / satt_denom) # denominator DF
tstat <- estimate/se.estimate
pvalue <- 2 * pt(abs(tstat), df = ddf, lower.tail = FALSE)
data.frame(estimate, se=se.estimate, tstat, ddf, pvalue)
##    estimate        se     tstat      ddf      pvalue
## 1 -0.6619655 0.2443135 -2.709492 491.2089 0.006974127
```

Satterthwaite's method for multi-df F-tests

Often it is of interest to assess an hypothesis on more than one linear function of the parameters such as

$$H_0 : \ L\beta = 0$$

were the $q \times p$ matrix L has rank q with $1 < q \leq p$ and p is the length of the β-vector.

The conventional F-statistic for this hypothesis reads

$$F = \frac{1}{q}(L\hat{\beta})^\top (L\mathsf{V}_{\hat{\tau}}(\hat{\beta})L^\top)^{-1}(L\hat{\beta})$$

which is seen to reduce to the square of the t-statistic considered above for $q = 1$. Despite its name, this statistic only follows an exact F-distribution with q numerator degrees of freedom and known denominator degrees of freedom in special cases (such as orthogonal or balanced designs.)

In the general case we are interested in estimating an appropriate denominator degrees of freedom for F assuming q numerator degrees of freedom.

Following Fai and Cornelius (1996) [6] we eigen-decompose the variance-covariance matrix of $L\hat{\beta}$:

$$\mathsf{V}_{\hat{\tau}}(L\hat{\beta}) = L\mathsf{V}_{\hat{\tau}}(\hat{\beta})L^\top = PDP^\top$$

such that we may write

$$\begin{aligned}
qF \equiv Q &= (L\hat{\beta})^\top PD^{-1}P^\top (L\hat{\beta}) \\
&= \hat{\beta}^\top L^\top PD^{-1}P^\top L\hat{\beta} \\
&= (P^\top L\hat{\beta})^\top D^{-1}(P^\top L\hat{\beta}) \\
&= \sum_{m=1}^{q} \frac{(P^\top L\hat{\beta})_m^2}{d_m} = \sum_{m=1}^{q} t_m^2
\end{aligned}$$

where $(P^\top L\hat{\beta})_m$ denotes the m'th element of the q-vector $P^\top L\hat{\beta}$ ($P^\top$ is an orthonormal $q \times q$ rotation matrix), and d_m is the m'th diagonal element of D; the diagonal matrix of eigenvalues.

Thus Q is being rewritten as a sum of q independent variables that have the form of the square of t-statistics each on one "numerator-degree of freedom".

In equivalence with (and multidimensional extension of) the 1D case above let

$$f(\hat{\tau}) \equiv P^\top L\mathsf{V}_{\hat{\tau}}(\hat{\beta})L^\top P = \tilde{L}\mathsf{V}_{\hat{\tau}}(\hat{\beta})\tilde{L}^\top = D, \quad \text{with} \quad \tilde{L} = P^\top L$$

be the diagonal matrix of eigenvalues (of $\mathsf{V}_{\hat{\tau}}(L\hat{\beta})$) and let $f(\hat{\tau})_m = d_m$ denote the m'th eigenvalue.

Note also that the mth diagonal element of $f(\hat{\tau})$ can be written

$$f(\hat{\tau})_m = (\tilde{L}\mathsf{V}_{\hat{\tau}}(\hat{\beta})\tilde{L})_m = \tilde{L}_m \mathsf{V}_{\hat{\tau}}(\hat{\beta})\tilde{L}_m$$

The m'th t-statistic in the summation above can now be written as

$$t_m = \frac{\tilde{L}_m \hat{\beta}}{\sqrt{\tilde{L}_m \mathsf{V}_{\hat{\tau}}(\hat{\beta}) \tilde{L}_m}}$$

which clarifies the similarity with the t-statistic considered in the 1-degree of freedom case above.

We therefore proceed as in the 1D case to estimate the degrees of freedom ν_m for the q t_m-statistics with

$$\hat{\nu}_m(\hat{\tau}) = \frac{2 f(\hat{\tau})_m^2}{f'(\hat{\tau})_m^\top \mathsf{V}(\hat{\tau}) f'(\hat{\tau})_m} = \frac{2 d_m^2}{f'(\hat{\tau})_m^\top \mathsf{V}(\hat{\tau}) f'(\hat{\tau})_m}$$

in which the gradient vector of length k reads

$$f'(\hat{\tau})_m = \tilde{L}_m^\top \{ \nabla_\tau \mathsf{V}_\tau(\hat{\beta})|_{\tau=\hat{\tau}} \} \tilde{L}_m$$

as discussed in the 1D case above.

Having determined estimates of ν_m for $m = 1, \ldots, q$, the objective is now to determine the denominator degrees of freedom for the F-statistic above. Since an F-variable with q numerator and ν denominator degrees of freedom has expectation

$$\mathsf{E}(F_{q,\nu}) = \frac{\nu}{\nu - 2} \quad \text{for} \quad \nu > 2$$

and we may for our F-statistics evaluate this quantity via

$$\mathsf{E}(F_{q,\nu}) = \mathsf{E}(q^{-1} Q) = q^{-1} \mathsf{E}(Q)$$

we may consider as estimator for ν the solution to

$$q^{-1} \mathsf{E}(Q) = \frac{\nu}{\nu - 2} \quad \text{for} \quad \nu > 2$$

which leads to

$$\hat{\nu} = \frac{2 \mathsf{E}(Q)}{\mathsf{E}(Q) - q}$$

Note that we need to require that $\mathsf{E}(Q) > q$ since only then is $\hat{\nu}$ positive.

The only remaining step is to determine $\mathsf{E}(Q)$ where again we utilize the expectation of an F-variable with at least two denominator degrees of freedom:

$$\mathsf{E}(Q) = \sum_{m=1}^{q} \mathsf{E}(t_{\nu_m}^2) = \sum_{m=1}^{q} \mathsf{E}(F_{1,\nu_m}) = \sum_{m=1}^{q} \frac{\nu_m}{\nu_m - 2} \quad \text{for} \quad \nu_m > 2$$

Example: Computing denominator df for an F-test using `lme4::lmer`

Considering the example above, we start by defining the contrast matrix, L and form the contrast estimate $L\hat{\beta}$:

```
L <- rbind(c(0, 1, 0, 0),
           c(0, 0, 1, 0))
(Lbeta <- drop(L %*% parlist$beta))
## [1] -0.6619655  0.2010220
```

and compute the variance of the contrasts $\mathsf{V}_{\hat{\tau}}(L\hat{\beta}) = L \mathsf{V}_{\hat{\tau}}(\hat{\beta}) L^\top$:

```
cov_Lbeta <- L %*% cov_beta %*% t(L) # Var(contrast) = Var(Lbeta)
```

We then compute the eigen-decomposition $\mathsf{V}(L\hat{\beta}) = P^\top D P$ and extract the rank, eigenvectors and eigenvalues:

```
# Get eigen decomposition of vcov(Lbeta):
eig_VLbeta <- eigen(cov_Lbeta)
positive <- eig_VLbeta$values > 1e-8
q <- sum(positive) # rank(VLbeta)
(P <- eig_VLbeta$vectors)
##              [,1]         [,2]
## [1,] 0.7058316 -0.7083797
## [2,] 0.7083797  0.7058316
(d <- eig_VLbeta$values)
## [1] 0.08984076 0.02975399
```

The new contrast vectors are then $\tilde{L} = P^\top L$

```
PtL <- crossprod(P, L) # q x p matrix
```

from which we can compute the t^2 values and F-statistic:

```
t2 <- drop(PtL %*% parlist$beta)^2 / d
Fvalue <- sum(t2) / q
```

For the new contrast vectors $\tilde{L}_m = (P^\top L)_m$ for $m = 1, \ldots, q$ we compute the gradient $f'(\hat{\tau})_m$:

```
grad_PLcov <- lapply(1:q, function(m) {
  vapply(Jac_list, function(J) sum(PtL[m, ] * J %*% PtL[m, ]), numeric(1L))
})
```

The m'th denominator degree of freedom estimate is then $\hat{\nu}_m = 2d_m^2(f'(\hat{\tau})_m^\top A f'(\hat{\tau})_m^\top)^{-1}$:

```
nu_m <- vapply(1:2, function(m) {
  denom <- sum(grad_PLcov[[m]] * (cov_varpar %*% grad_PLcov[[m]])) # g'Ag
  2*(d[m])^2 / denom # 2d_m^2 / g'Ag
}, numeric(1L))
nu_m
## [1] 501.4952 494.6459
```

From $\mathsf{E}(Q)$ we then evaluate the estimated denominator degrees of freedom for the F-statistic:

```
EQ <- sum(nu_m / (nu_m - 2))
(ddf <- 2 * EQ / (EQ - q)) # nu
## [1] 498.0469
```

In summary we have the following approximate F-test:

```
pvalue <- pf(q=Fvalue, df1=q, df2=ddf, lower.tail=FALSE)
data.frame('F value'=Fvalue, ndf=q, ddf=ddf, 'p-value'=pvalue,
           check.names = FALSE)
##    F value ndf      ddf    p-value
## 1 6.856825   2 498.0469 0.00115446
```

Remarks

Remark 1: If L is rank deficient

If the contrast matrix L is row-rank deficient with $< q$ rows where q is the rank of L, then all that it is needed is to define $\tilde{L}$ as the first q rows of $P^\top L$. Notice that we can find q as the number of non-zero (save a tolerance) eigenvalues in D.

Remark 2: Sum of Squares and Mean Squares

The sums of squares (SSQ) associated with a contrast L can be written

$$\text{SSQ}(L\hat{\beta}) = Fq\hat{\sigma}^2$$

with mean square (MS)

$$\text{MS}(L\hat{\beta}) = \text{SSQ}(L\hat{\beta})/q = F\hat{\sigma}^2$$

These SSQ and MS do not correspond to those that can be derived for variance-component models but refer directly to the multivariate normal model.

Remark 3: $q = 1$

For $q = 1$ the F-value can be computed as the square of the t-statistics with its estimate of the denominator degrees of freedom without evaluation of $\mathsf{E}(Q)$.

Remark 4: $q \geq 2$

For $q \geq 2$ we need $\mathsf{E}(Q) > q$, but note that

$$\mathsf{E}(Q) = \sum_{m=1}^{q} \frac{\nu_m}{\nu_m - 2} > q$$

if $\nu_m > 2$ for all m, since

$$\frac{\nu_m}{\nu_m - 2} > 1 \quad \text{for} \quad \nu_m > 2$$

Thus the requirement reduces to $\nu_m > 2$ for $m = 1, \ldots, q$.

Remark 5: ν **tends to 2**

If, for some m, say m' $\nu_{m'} \to 2_+$ ($\nu_{m'}$ approaches 2 from above), then

$$\hat{\nu} = \frac{2\mathsf{E}(Q)}{\mathsf{E}(Q) - q} \to 2$$

since

$$\frac{\nu_{m'}}{\nu_{m'} - 2} \to \infty$$

and therefore

$$\mathsf{E}(Q) \to \infty$$

Remark 6: lower bound on ν

A property of the F-distribution is that if

$$X \sim F_{q,\nu} \quad \text{then} \quad X^{-1} \sim F_{\nu,q}$$

Thus we may consider 2 as a lower bound on ν (recall that $q \geq 2$) which may be used if any $\nu_m < 2$.

Remark 7: all q estimates ν_m are the same

If all q estimates ν_m are the same and $\nu_1 = \ldots = \nu_q$ we have that

$$\mathsf{E}(Q) = q\frac{\nu_m}{\nu_m - 2}$$

and therefore

$$\nu = \frac{2\mathsf{E}(Q)}{\mathsf{E}(Q) - q} = \frac{q\frac{2\nu_m}{\nu_m - 2}}{q\frac{\nu_m - \nu_m + 2}{\nu_m - 2}} = \frac{2\nu_m}{\nu_m - 2}\frac{\nu_m - 2}{2} = \nu_m$$

thus the estimate of the degrees of freedom ν for the F-statistic is just $\nu = \nu_1 = \ldots = \nu_q$.

References

[1] Kuznetsova, A., Brockhoff, P., & Christensen, R. (2017). "lmerTest Package: Tests in Linear Mixed Effects Models." Journal of Statistical Software, 82(13), 1–26. doi:10.18637/jss.v082.i13

[2] Schaalje GB, McBride JB, Fellingham GW (2002). "Adequacy of Approximations to Distributions of Test Statistics in Complex Mixed Linear Models." Journal of Agricultural, Biological, and Environmental Statistics, 7(4), 512–524. [doi:10.1198/108571102726](https://doi.org/10.1198/108571102726).

[3] Satterthwaite FE (1946). "An Approximate Distribution of Estimates of Variance Components." Biometrics Bulletin, 2(6), 110–114. doi:10.2307/3002019.

[4] Bates DM, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." Journal of Statistical Software, 67(1), 1–48. doi:10.18637/jss.v067.i01.

[5] Dahl DB (2016). numDeriv: Accurate Numerical Derivatives. R package version 2016.8–1, URL https://CRAN.R-project.org/package=numDeriv.

[6] Fai AH, Cornelius PL (1996). "Approximate F-Tests of Multiple Degree of Freedom Hypotheses in Generalised Least Squares Analyses of Unbalanced Split-Plot Experiments."" Journal of Statistical Computation and Simulation, 54(4), 363–378. doi:10.1080/00949659608811740.