influence.ME

influence.ME allows you to compute measures of influential data for mixed effects models generated by lme4.

influence.ME provides a collection of tools for detecting influential cases in generalized mixed effects models. It analyses models that were estimated using lme4. The basic rationale behind identifying influential data is that when iteratively single units are omitted from the data, models based on these data should not produce substantially different estimates.

To standardize the assessment of how influential a (single group of) observation(s) is, several measures of influence are common practice, such as DFBETAS and Cook's Distance. In addition, we provide a measure of percentage change of the fixed point estimates and a simple procedure to detect changing levels of significance.

`influence()` is the workhorse function of the influence.ME package. Based on a priorly estimated mixed effects regression model (estimated using lme4), the `influence()` function iteratively modifies the mixed effects model to neutralize the effect a grouped set of data has on the parameters, and which returns returns the fixed parameters of these iteratively modified models. These are used to compute measures of influential data.

1 Computing DFBETAs with R

- This function computes the DFBETAS based on the information returned by the estex() function.
- The dfbeta refers to how much a parameter estimate changes if the observation or case in question is dropped from the data set.
- Cook's distance is presumably more important to you if you are doing predictive modeling, whereas dfbeta is more important in explanatory modeling.
- The DFBETAS statistics are the scaled measures of the change in each parameter estimate and are calculated by deleting the th observation:

Missing Formula

where is the th element of . In general, large values of DFBETAS indicate observations that are influential in estimating a given parameter.

- **Belsley, Kuh, and Welsch (1980)** recommend 2 as a general cutoff value to indicate influential observations and as a size-adjusted cutoff.

The `R` programming language has a variety of methods used to study each of the aspects for a linear model. While linear models and GLMS can be studied with a wide range of well-established diagnostic technqiues, the choice of methodology is much more restricted for the case of LMEs.

For an `lme` object, such as our fitted model `JS.roy1`, the predicted values for each subject can be determined using the `coef.lme` function.

```
> JS.roy1 %>% coef %>% head(5)
methodJ   methodS
74     84.31724  91.08404
36     91.54994  97.05548
3      81.16581  96.48653
62     92.09493  90.89073
31     88.41411 103.38802
```

The `CookD` fucntion , from the predictmeans R package, produces Cooks distance plots for an LME model (***predictmeans***)

```
library(predictMeans)
CookD(model, group=method, plot=TRUE, idn=5, newwd=FALSE)
```

2 DFbetas for Blood Data

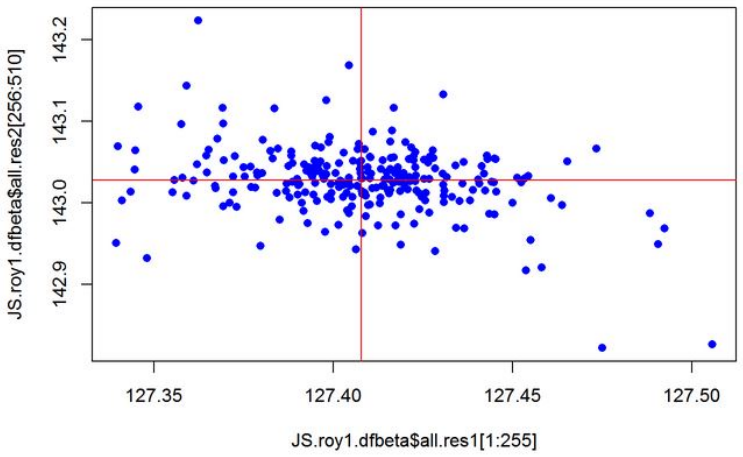

Figure 1:

```
plot(JS.ARoy20091.dfbeta$all.res1[1:255],JS.ARoy20091.dfbeta$all.res2[256:510],
pch=16,col="blue")
abline(v=JS.ARoy20091.dfbeta$all.res1[256],col="red")
abline(h=JS.ARoy20091.dfbeta$all.res2[1],col="red")
```

3 The `logLik` Function

`logLik.lme` returns the log-likelihood value of the linear mixed-effects model represented by object evaluated at the estimated coefficients. It is also possible to determine the restricted log-likelihood, if relevant, using this function. For the Blood Data Example, the loglikelihood of the JS.roy1 model can be computed as follows.

```
> logLik(JS.roy1)
'log Lik.' -2030.736 (df=8)
```

4 `Influence()` - Description

`influence()` is the workhorse function of the `influence.ME` package.

Based on a priorly estimated mixed effects regression model (estimated using lme4), the `influence()` function iteratively

modifies the mixed effects model to neutralize the effect a grouped set of data has on the parameters, and which

returns returns the fixed parameters of these iteratively modified models.

These are used to compute measures of influential data.

Usage

```
influence(model, group=NULL, select=NULL, obs=FALSE,
gf="single", count = FALSE, delete=TRUE, ...)
```

The `influence()` function was known as the `estex()` command in previous versions of the influence.ME pacakge

5 Leave-One-Out Diagnostics with `lmeU`

Galecki et al discuss the matter of LME influence diagnostics in their book, although not into great detail.

The command `lmeU` fits a model with a particular subject removed. The identifier of the subject to be removed is passed as the only argument

A plot ofthe per-observation diagnostics individual subject log-likelihood contributions can be rendered.

6 Partitioning Matrices

Without loss of generality, matrices can be partitioned as if the $i-$th omitted observation is the first row; i.e. $i = 1$.

7 Permutation Test, Power Tests and Missing Data

This section explores topics such as dependent variable simulation and power analysis, introduced by Galecki & Burzykowski (2013), and implementable with their ***nlmeU*** `R` package. Using the ***predictmeans*** `R` package, it is possible to perform permutation t-tests for coefficients of (fixed) effects and permutation F-tests.

The matter of missing data has not been commonly encountered in either Method Comparison Studies or Linear Mixed Effects Modelling. However Roy (2009) deals with the relevant assumptions regrading missing data. Galecki & Burzykowski (2013) approaches the subject of missing data in LME Modelling. The ***nlmeU*** package includes the `patMiss` function, which "*allows to compactly present pattern of missing data in a given vector/matrix/data frame or combination of thereof*".