MRC Clinical Trials Unit

Smarter Studies
Global Impact
Better Health

UCL

jomo: an R package for Multilevel Multiple Imputation

Matteo Quartagno

MRC Clinical Trials Unit at UCL

13th July 2020 (NASH)

R packages for joint modelling MI

- There are several R packages for Joint Modelling Multiple Imputation;
- `norm` and `Amelia` are possibly the most popular, but not available for multilevel data;
- `pan` extends to multilevel (aka *pan*el) data, but only continuous data;
- How is `jomo` different from these packages?

jomo: why is it different?

- jomo is a package for multilevel joint modelling imputation, using multivariate normal model;
- **Advantage #1**: handles binary/categorical data through latent normal variables;
- **Advantage #2**: allows for cluster-specific covariance matrices:
  - Helps to take into account heterogeneity between level 2 units;
  - Better with missing data in random slope variable;
- **Advantage #3**: allows for imputation of level 2 variables;
- **Advantage #4**: Allows for congeniality / compatibility / consistency of analysis and imputation model: What does this mean?

Compatibility

- **Settings:**
  - Collect data on two variables: Y and X;
  - *I* individuals nested in *J* clusters, e.g. *I* children in *J* schools;
  - Analysis model: a random intercept model with quadratic effect for X. R code:

```
fit <- lmer ( Y ~ X +I(X^2) + ( 1 | schoolID ) , data )
```

- **Example 1:** Missing data in Y only. What is the best imputation model?
  - Same as analysis model;
- **Example 2:** Missing data in both X and Y.
  - Most simple choice: bivariate normal model.
  - But this does not reflect quadratic association in analysis model between Y and X;
  - The two models are not compatible. Imputation may introduce bias, particularly for quadratic effect parameter.

`jomo` and compatibility

- Problems with compatibility generally arise whenever there are missing data in variables that are included in analysis model with:
  - Non-linear effects (e.g. quadratic, cubic, log);
  - Interactions;
  - Random slopes;
- We'll come back to this, but let's start from a simple example first;

Data: Junior School Project

	school	id	sex	fluent	ravens	english	behaviour	cons
14	1	307	1	2	17	NA	upper	1
15	1	308	0	2	22	NA	lowerquarter	1
16	1	309	1	NA	NA	15	lowerquarter	1
17	1	310	0	2	25	42	lowerquarter	1
18	1	312	0	NA	NA	10	lowerquarter	1
19	1	313	1	2	NA	47	upper	1
20	1	314	0	2	23	NA	upper	1
21	1	315	0	2	16	NA	upper	1
22	1	316	1	2	NA	31	upper	1
23	1	317	0	NA	NA	11	upper	1
24	1	318	0	2	20	NA	lowerquarter	1
25	1	324	1	NA	14	24	lowerquarter	1
26	2	328	1	NA	27	49	lowerquarter	1
27	2	330	0	2	NA	NA	upper	1
28	2	331	1	2	31	15	upper	1
29	2	332	1	2	28	55	upper	1
30	2	333	0	2	25	58	upper	1
31	2	334	1	0	19	13	lowerquarter	1
32	2	335	0	2	29	55	upper	1
33	2	336	1	2	NA	34	upper	1
34	2	337	1	2	28	41	upper	1

`jomo`: single level example

- Data from Junior School Project;
- Model of interest: linear regression of English score at 3 years over 1 year test score and sex. R code:

```
lm(english ~ ravens + sex, data=JSPmiss)
```

- Missing data in English and ravens. We can assume simple multivariate normal model with sex as covariate. `jomo` can fit and impute from this model.

jomo: single level example

```
Y <- JSPmiss[, c("english", "ravens")]

X <- JSPmiss[, c("cons", "sex")]

set.seed(1988)

imp <- jomo(Y = Y, X = X, nburn = 1000,
nbetween = 1000, nimp = 5)
```

jomo: single level example

```
> Y <- JSPmiss[, c("english", "ravens")]
> X <- JSPmiss[, c("cons", "sex")]
> set.seed(1988)
> imp <- jomo(Y = Y, X = X, nburn = 1000, nbetween = 1000, nimp = 5)
No clustering, using functions for single level imputation.
Found  2 continuous outcomes and no categorical. Using function jomo1con.
..................................................................................
First imputation registered.
..................................................................................
Imputation number  2 registered
..................................................................................
Imputation number  3 registered
..................................................................................
Imputation number  4 registered
..................................................................................
Imputation number  5 registered
The posterior mean of the fixed effects estimates is:
            cons        sex
english 38.18853  6.3720200
ravens  25.32846 -0.5515291

The posterior covariance matrix is:
         english   ravens
english 458.4993 64.70380
ravens   64.7038 36.37251
>
```

jomo: single level example

```
> Y <- JSPmiss[, c("english", "ravens")]
> X <- JSPmiss[, c("cons", "sex")]
> set.seed(1988)
> imp <- jomo(Y = Y, X = X, nburn = 1000, nbetween = 1000, nimp = 5)
No clustering, using functions for single level imputation.
Found  2 continuous outcomes and no categorical. Using function jomo1con.
..........................................................................
First imputation registered.
..........................................................
Imputation number  2 registered
..........................................................
Imputation number  3 registered
..........................................................
Imputation number  4 registered
..........................................................
Imputation number  5 registered
The posterior mean of the fixed effects estimates is:
             cons        sex
english 38.18853  6.3720200
ravens  25.32846 -0.5515291

The posterior covariance matrix is:
         english   ravens
english 458.4993 64.70380
ravens   64.7038 36.37251
> |
```

	english	ravens	cons	sex	id	Imputation
1113	*NA*	32.000000	1	1	1113	0
1114	*NA*	21.000000	1	1	1114	0
1115	18.000000	22.000000	1	1	1115	0
1116	75.000000	14.000000	1	0	1116	0
1117	82.000000	34.000000	1	0	1117	0
1118	28.000000	27.000000	1	0	1118	0
1119	*NA*	27.000000	1	1	1119	0
1120	39.000000	11.268014	1	1	1	1
1121	53.218409	15.000000	1	0	2	1
1122	65.000000	19.000000	1	1	3	1
1123	25.649187	22.000000	1	0	4	1
1124	30.000000	21.231007	1	1	5	1
1125	12.000000	8.582315	1	0	6	1
1126	20.000000	30.000000	1	0	7	1

jomo: single level example

- Once we have imputed data and stored them in `imp`, how do we fit model on each imputed data? How do we combine estimates?
- Can use several packages, here we use `mitools`:

```
> library(mitools)
>
> imp.list<-imputationList(split(imp, imp$Imputation)[-1])
> fit.imp<-with(imp.list, lm(english~ravens+sex))
> coefs<-MIextract(fit.imp, fun=coef)
> vars<-MIextract(fit.imp, fun=function(x){diag(vcov(x))})
> results<-MIcombine(coefs, vars)
> summary(results)
Multiple imputation results:
      MIcombine.default(coefs, vars)
              results         se     (lower    upper) missInfo
(Intercept) -7.155383 2.56682756 -12.190695 -2.120071      6 %
ravens       1.763872 0.09756369   1.572169  1.955574      9 %
sex          8.391990 1.24733980   5.915699 10.868280     22 %
>
```

Including categorical variables

- Easy to include categorical / binary variables. Just need to be treated as factor:

```
> JSPmiss<-within(JSPmiss,fluent<-factor(fluent))
> Y <- JSPmiss[, c("english", "ravens", "fluent")]  # Variables to be imputed
> X <- JSPmiss[, c("cons", "sex")]                  # Other fully observed variables
> set.seed(1988)                                    # Set seed for replicability
> imp <- jomo(Y = Y, X = X)
No clustering, using functions for single level imputation.
Found  2 continuous outcomes and  1 categorical. Using function jomo1mix.
.....................................................................................
First imputation registered.
.....................................................................................
Imputation number  2 registered
.....................................................................................
Imputation number  3 registered
.....................................................................................
Imputation number  4 registered
.....................................................................................
Imputation number  5 registered
The posterior mean of the fixed effects estimates is:
              cons         sex
english  38.337151  6.30391964
ravens   25.320582 -0.55017880
fluent.1 -1.213688 -0.09648138
fluent.2 -1.310895  0.04521813

The posterior covariance matrix is:
           english    ravens   fluent.1  fluent.2
english  516.41486 75.040040 -13.024148 -10.06556
ravens    75.04004 38.691419  -2.517502  -1.85167
fluent.1 -13.02415 -2.517502   1.000000   0.50000
fluent.2 -10.06556 -1.851670   0.500000   1.00000
> View(imp)
```

Random intercept model

- Motivation for developing `jomo` was multilevel data.
- So assume our substantive analysis model is:

```
lmer(english ~ ravens + sex + factor(fluent)
+ (1|school))
```

- If we have random intercept in *analysis* model, need to reflect that in the *imputation* model as well
- In `jomo`, only thing that changes is that we need to tell software about clustering.

jomo: multilevel data example

```
> clus<-JSPmiss$school
> Y <- JSPmiss[, c("english", "ravens", "fluent")]  # Variables to be imputed
> X <- JSPmiss[, c("cons", "sex")]                  # Other fully observed variables
> set.seed(1988)                                    # Set seed for replicability
> imp <- jomo(Y = Y, X = X, clus=clus, nburn = 2000)
Clustered data, using functions for two-level imputation.
Found  2 continuous outcomes and  1 categorical. Using function jomo1ranmix.
..........................................................................
First imputation registered.
..........................................................................
Imputation number  2 registered
..........................................................................
Imputation number  3 registered
..........................................................................
Imputation number  4 registered
..........................................................................
Imputation number  5 registered
The posterior mean of the fixed effects estimates is:
              cons         sex
english  38.228396  5.82846310
ravens   25.200419 -0.48311630
fluent.1 -1.220306 -0.07506056
fluent.2 -1.318883  0.06369684
```

jomo: multilevel data example

```
> clus <- JSPmiss$school
> Y <- JSPmiss[, c("english", "ravens", "fluent")]
> JSPmiss$cons <- 1
> X <- JSPmiss[, c("cons", "sex")]
> set.seed(1569)
> imp <- jomo(Y = Y, X = X, clus = clus, nburn = 2000, nbetween = 1000, nimp = 5)
The posterior mean of the random effects estimates is:
    english.z1    ravens.z1  fluent.1.z1  fluent.2.z1          x.
1   -9.9473341 -2.525482543  0.017103423  0.071558798 ..........
2   -0.5216873  0.177027863  0.033156548 -0.033906883 ..........
3    0.9868271  1.394930115 -0.094843642 -0.082051598
4   -1.2459841  0.059001753 -0.058214607  0.016746615 ..........
5    7.2371453  1.757267410 -0.105010471  0.008110850
                                                     ..........

46   7.2819165  1.883121563 -0.121513108  0.020228350 ..........
47   3.2192374  1.446750628  0.241459155  0.308610455
48  -2.3268527 -0.236753298 -0.089105900 -0.019258071 ..........
49   0.8526051  0.068906842 -0.078313340 -0.002422980
50   2.1296118  1.051757415 -0.061999949 -0.037407180

The posterior mean of the level 1 covariance matrices is:
           english    ravens   fluent.1  fluent.2
english  442.626203 64.822129 -10.571044 -8.769852
ravens    64.822129 35.585715  -2.275623 -1.923068
fluent.1 -10.571044 -2.275623   1.000000  0.500000
fluent.2  -8.769852 -1.923068   0.500000  1.000000

The posterior mean of the level 2 covariance matrix is:
            english.z1   ravens.z1 fluent.1.z1 fluent.2.z1
english*z1  65.0011603 12.59045739 -0.77743796 -0.34490660
ravens*z1   12.5904574  3.91499979 -0.17609956 -0.07043958
fluent.1*z1 -0.7774380 -0.17609956  0.08486756  0.01702867
fluent.2*z1 -0.3449066 -0.07043958  0.01702867  0.08145533
> View(imp)
```

jomo: multilevel data example

- Fitting the random intercept model on all imputed datasets and combining the estimates is no different than in the single level example.

```
> library(lme4)
>
> imp.list<-imputationList(split(imp, imp$Imputation)[-1])
> fit.imp<-with(imp.list, lmer(english~ravens+sex+factor(fluent)+(1|clus)))
> coefs<-MIextract(fit.imp, fun=fixef)
> vars<-MIextract(fit.imp, fun=function(x){diag(vcov(x))})
> results<-MIcombine(coefs, vars)
> summary(results)
Multiple imputation results:
      MIcombine.default(coefs, vars)
                   results         se     (lower    upper) missInfo
(Intercept)     -13.123020 3.72500806 -20.625916 -5.620125     33 %
ravens            1.576412 0.09321524   1.393585  1.759239      5 %
sex               6.391213 1.35221772   3.600182  9.182244     45 %
factor(fluent)1   2.861211 3.92975546  -5.067142 10.789565     34 %
factor(fluent)2  13.103559 3.10588468   6.669952 19.537167     47 %
```

Random slope model

- If the analysis model has a random slope:

```
lmer (english ~ ravens + sex + factor(fluent) +
(1+ravens|school))
```

- Or more generally if there is likely heterogeneity between clusters, option “random” must be used:

- `Imp <- jomo(Y = Y, X = X, clus = clus, meth = "random")`

jomo: two-level imputation

- Some variables may be at level 2, e.g. school-related variables. Let's considered Exam scores dataset:

	school	student	normexam	cons	standlrt	sex	schgend	avslrt	schav	vrband
1	1	1	*NA*	1	*NA*	girl	mixedsch	0.1661745	mid	vb1
2	1	2	0.13406679	1	0.20580196	girl	mixedsch	0.1661745	mid	vb2
3	1	3	-1.72388244	1	*NA*	boy	mixedsch	0.1661745	mid	vb3
4	1	4	*NA*	1	0.20580196	girl	mixedsch	0.1661745	mid	vb2
5	1	5	0.54434091	1	0.37110487	girl	mixedsch	0.1661745	mid	vb2
6	1	6	1.73489916	1	2.18943691	boy	mixedsch	0.1661745	mid	vb1
7	1	7	*NA*	1	-1.11662138	boy	mixedsch	0.1661745	mid	vb3
8	1	8	-0.12908468	1	-1.03396988	boy	mixedsch	0.1661745	mid	vb2
9	1	9	*NA*	1	-0.53806120	girl	mixedsch	0.1661745	mid	vb2
10	1	10	-1.21948552	1	*NA*	boy	mixedsch	0.1661745	mid	vb3
11	1	11	2.40869188	1	2.43739128	boy	mixedsch	0.1661745	mid	vb1
12	1	12	0.61072856	1	2.10678554	boy	mixedsch	0.1661745	mid	vb1
13	1	13	*NA*	1	0.04049904	boy	mixedsch	0.1661745	mid	vb2
14	1	14	-0.12908468	1	1.19761944	boy	mixedsch	0.1661745	mid	vb1
15	1	15	2.20312095	1	2.52004290	boy	mixedsch	0.1661745	mid	vb1
16	1	16	1.24053323	1	1.11496806	girl	mixedsch	0.1661745	mid	vb1
17	1	17	*NA*	1	1.03231657	girl	mixedsch	0.1661745	mid	vb1
18	1	18	1.31014240	1	0.78436220	boy	mixedsch	0.1661745	mid	vb1

jomo: two-level imputation

- Some variables may be at level 2, e.g. school-related variables;
- Need to make sure we impute properly, i.e. same value for all individuals in same school
- Easily dealt with by jomo.

```
> clus<-ExamScores$school
> Y <- ExamScores[, c("normexam", "standlrt")]  # Level 1 variables to be imputed
> Y2 <- ExamScores[, c("avslrt"), drop=FALSE]   # Level 2 variables to be imputed
> set.seed(1988)                                # Set seed for replicability
> imp <- jomo(Y = Y, Y2 = Y2, clus=clus)
2-level data, using functions for two-level imputation.
Found  2 level 1 continuous and  0 level 1 categorical outcomes,  1  level 2 continuo
us and  0  level 2 categorical outcomes. Using function jomo2com, assuming common cov
ariance matrix across clusters
.....................................................................................
First imputation registered.
.....................................................................................
Imputation number  2 registered
.....................................................................................
Imputation number  3 registered
```

Checking convergence

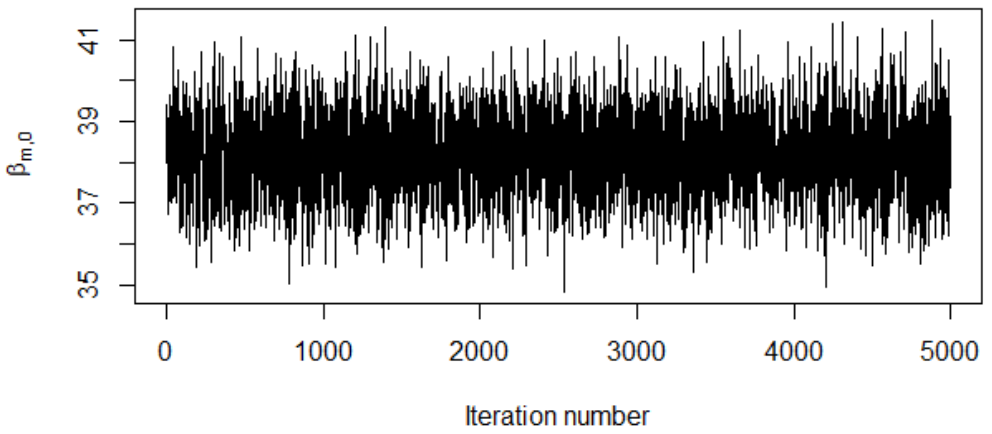
- We have seen how to generate imputed data with `jomo`. But can we trust these imputations?
- `jomo` imputes by running MCMC. Crucial to check chains have **converged** before registering imputations.
- We can do this with `jomo.MCMCchain` function, before running the actual imputation.

Checking convergence

```
> Y <- JSPmiss[, c("english", "ravens")]  # Variables to be imputed
> X <- JSPmiss[, c("cons", "sex")]        # Other fully observed variables
> set.seed(1988)                          # Set seed for replicability
> imp <- jomo.MCMCchain(Y = Y, X = X, nburn = 5000)
No clustering, using functions for single level imputation.
Found  2 continuous outcomes and no categorical. Using function jomo1con.
..................................................................................
The posterior mean of the fixed effects estimates is:
            cons       sex
english 38.18803  6.379945
ravens  25.32729 -0.552918

The posterior covariance matrix is:
         english   ravens
english 458.7564 64.81650
ravens   64.8165 36.40801
>
> plot(imp$collectbeta[1,1,], type="l",
+      xlab = "Iteration number",
+      ylab = expression(beta["m,0"]))
>
```

Checking convergence

```
> Y <- JSPmiss[, c("english", "ravens")]  # Variables to be imputed
> X <- JSPmiss[, c("cons", "sex")]        # Other fully observed variables
> set.seed(1988)                          # Set seed for replicability
> imp <- jomo.MCMCchain(Y = Y, X = X, nburn = 5000)
No clustering, using functions for single level imputation.
Found  2 continuous outcomes and no categorical. Using function jomo1con.
.....................................................................................
The posterior mean of the fixed effects estimates is:
            cons       sex
english 38.18803  6.379945
ravens  25.32729 -0.552918

The posterior covariance matrix is:
         english   ravens
english 458.7564 64.81650
ravens   64.8165 36.40801
>
> plot(imp$collectbeta[1,1,], type="l",
+      xlab = "Iteration number",
+      ylab = expression(beta["m,0"]))
>
```



Using `jomo` in practice

- `jomo` uses MCMC, i.e. a Bayesian method. So we need to choose priors;
- `jomo` assumes flat prior for all parameters but covariance matrices;
- For covariace matrices, inverse-Wishart. Default hyperparameter is identity, if not suitable either use expert-informed priors or derived from data;

```
JSPmiss <- within(JSPmiss, fluent <- factor(fluent))
Y <- JSPmiss[, c("english", "ravens", "fluent")]
X <- JSPmiss[, c("cons", "sex")]
imp1 <- jomo.MCMCchain(Y = Y, X = X)
l1cov.guess <- apply(imp1$collectomega, c(1, 2), mean)
l1cov.prior <- l1cov.guess*4
imp <- jomo(Y = Y, X = X, l1cov.prior = l1cov.prior)
```

Using `jomo` in practice

- Recommended workflow:
  1. Before running the imputation model (which may take some time), perform a 'dry run', i.e. run `jomo.MCMCchain` with nburn = 2 and check the output;
  2. Re-run the same function for a larger number of iterations (e.g. 5000) and analyse trace plots to choose sensible number of burn-in and between-imputation iterations;
  3. Run the **jomo** function for the chosen number of iterations
  4. Fit the substantive model on the imputed data sets and apply Rubin's rules, e.g. using `mitools`.

`mitml`: alternative interface to `jomo`

- Package `mitml` offers an alternative interface, based on the classic formula/data framework

```
> library(mitml)
*** This is beta software. Please report any bugs!
*** See the NEWS file for recent changes.
>
> fml <- english + ravens +fluent  ~ sex +  (1|school)
>
> imp <- jomoImpute(data = JSPmiss, formula = fml, n.burn = 1000, n.iter = 1000, m = 5,
+                   seed = 1569)
Running burn-in phase ...
Creating imputed data set ( 1 / 5 ) ...
Creating imputed data set ( 2 / 5 ) ...
Creating imputed data set ( 3 / 5 ) ...
Creating imputed data set ( 4 / 5 ) ...
Creating imputed data set ( 5 / 5 ) ...
Done!
>
```

mitml: alternative interface to jomo

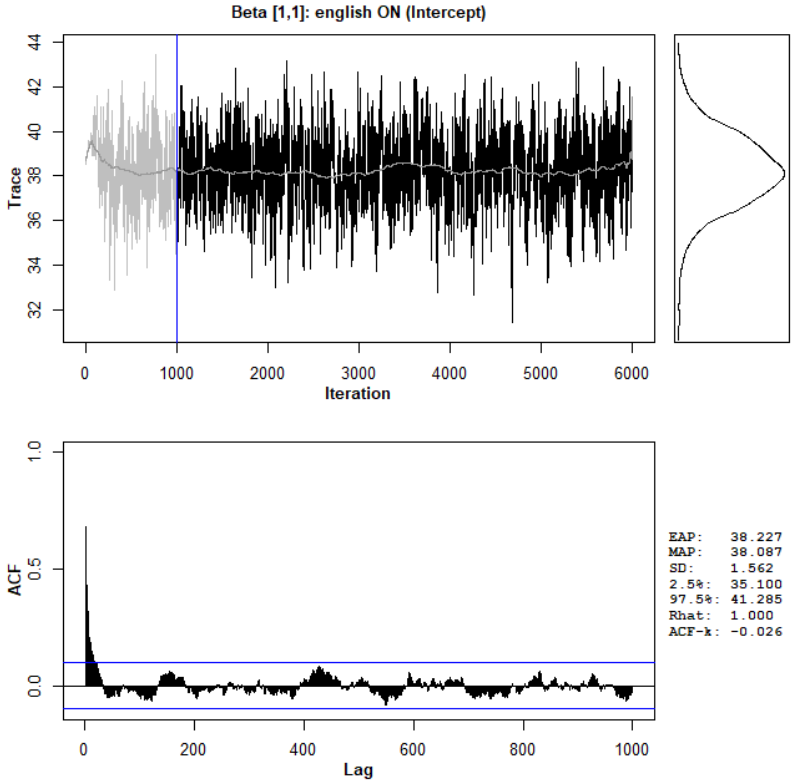
- Package mitml offers an alternative interface, based on the classic formula/data framework

```
> summary(imp)

Call:

jomoImpute(data = JSPmiss, formula = fml, n.burn = 1000, n.iter = 1000,
    m = 5, seed = 1988)

Cluster variable:         school
Target variables:         english ravens fluent
Fixed effect predictors:  (Intercept) sex
Random effect predictors: (Intercept)

Performed 1000 burn-in iterations, and generated 5 imputed data sets,
each 1000 iterations apart.

Potential scale reduction (Rhat, imputation phase):

         Min   25%  Mean Median   75%   Max
Beta:  1.000 1.001 1.004  1.003 1.005 1.010
Psi:   1.000 1.002 1.004  1.004 1.006 1.011
Sigma: 1.000 1.002 1.025  1.020 1.045 1.069

Largest potential scale reduction:
Beta: [1,3], Psi: [2,1], Sigma: [3,1]

Missing data per variable:
    school english ravens fluent id sex behaviour cons
MD% 0      21.1    22.0   21.0   0  0   0         0
```

`mitml`: alternative interface to `jomo`

- Package `mitml` offers an alternative interface, based on the classic formula/data framework

```
> plot(imp, trace = "all")
Waiting to confirm page change...
```



`jomo` and compatibility

- We said there were at least 4 advantages in using `jomo`.
- We have seen how to use it to impute (1) binary/categorical data, (2) 2-level data and (3) allowing for heterogeneity;
- Fourth advantage was it can solve compatibility problems;
  - Functions used thus far solve compatibility with random intercept and just approximately for random slopes
  - However, still no compatibility with interactions, non-linearities, survival models.
  - For this, new “Substantive Model Compatible” functions. SMC-jomo for friends…

SMC-jomo: a simple example

- For example, if our substantive model is random intercept model with quadratic effect:

```
> cldata<-within(cldata, sex<-factor(sex))
>
> # we define the data frame with all the variables
>
> data<-cldata[,c("measure","age", "sex", "city")]
> mylevel<-c(1,1,1,1)
>
> # And the formula of the substantive lm model
>
> formula<-as.formula(measure~sex+age+I(age^2)+(1|city))
>
> #And finally we run the imputation function:
>
> imp<-jomo.lmer(formula,data, level=mylevel, nburn=1000, nbetween=1000)
```

SMC-jomo: a simple example

- For example, if our substantive model is random intercept model with quadratic effect:

```
> cldata<-within(cldata, sex<-factor(sex))
>
> # we define the data frame with all the variables
>
> data<-cldata[,c("measu
> mylevel<-c(1,1,1,1)
>
> # And the formula of t
>
> formula<-as.formula(me
>
> #And finally we run th
>
> imp<-jomo.lmer(formula
```

```
The posterior mean of the substantive model fixed effects estimates is:
            (Intercept)     sex1         age     I(age^2)
measure   0.7298494 0.117094 -0.03459484 -0.006904436
The posterior mean of the substantive model residual variance is:
[1] 0.9862608
The posterior mean of the substantive model random effects covariance matrix is:
                (Intercept)
(Intercept)   0.2744676
The posterior mean of the substantive model random effects estimates is:
  (Intercept)
0  0.30451853
1  0.47946427
2 -0.44882364
3  0.95349486
4 -0.49812336
5  0.01817672
6  0.02429864
7 -0.16391304
8 -0.36161239
9 -0.19871937
```

SMC-jomo: a simple example

- For example, if our substantive model is random intercept model with quadratic effect:

```
> cldata<-within(cldata, sex<-factor(sex))
>
> # we define the data frame with all the variables
>
> data<-cldata[,c("measure","age", "sex", "city")]
> mylevel<-c(1,1,1,1)
>
> # And the formula of the substantive lm model
>
> formula<-as.formula(measure~sex+age+I(age^2)+
>
> #And finally we run the imputation function:
>
> imp<-jomo.lmer(formula,data, level=mylevel, nI
```

	measure	age	sex	clus	id	Imputation
995	-0.665733971	0	0	9	995	0
996	NA	0	1	9	996	0
997	1.716508187	-2	0	9	997	0
998	-0.056368925	0	1	9	998	0
999	0.632276731	-2	1	9	999	0
1000	0.807351223	0	1	9	1000	0
1001	-0.184765759	-1	1	0	1	1
1002	1.714421521	1	1	0	2	1
1003	1.520574561	-1	1	0	3	1
1004	1.916804809	3	1	0	4	1

SMC-jomo

- Currently SMC-jomo available for imputation compatible with `lm` and `lmer`, `glm` and `glmer` (binomial only), `coxph`, `polr` and `clmm`;
- Can handle random intercept and slopes with unstructured covariance matrix;
- Non-linearities and interactions;
- Future developments: splines, other variance structures, other models (e.g. frailty), …

Conclusions

- MI has become gold standard in handling missing data;
- However, needs to be used carefully. Compatibility of imputation and analysis model is key;
- `jomo` provides a software to impute compatibly when analysis model is multilevel, and when we want to impute data in a mix of continuous and categorical variables at level 1 and 2;
- Additionally the SMC-jomo functions can handle interactions, nonlinearities, and are the only ones 100% compatible with random slopes.

Limitations/extensions

- Important to always check the sampler has converged before starting registering imputations.
- `jomo` can be slow with large data sets, particularly with categorical variables with several categories;
  - Plan to speed up package in the future.
- We will keep maintaining the package and including new functions, for imputation compatible with other analysis models.

Acknowledgements

- Joint work with James Carpenter (MRC-CTU and LSHTM)
- Presentation based on R Journal paper with Simon Grund (Leibniz Institute for Science and Mathematics Education, Kiel, Germany)
- Quartagno M, Grund S, Carpenter JR, `jomo`: a flexible package for two-level joint modelling multiple imputation, R Journal, December 2019