

Session 5.2: Missing data imputation

Imperial College London

Learning Objectives

After this session you should be able to:

- Appreciate the importance of thinking about why data are missing, and stating your modelling assumptions
- Understand the disadvantages of complete case analysis
- Learn about how Bayesian methods can be used as 'statistically principled' for handling missing data
- Be able to run the above models in R-INLA

The topics covered in this lecture are covered in Chapter 12 of Gómez-Rubio (2020a).

Why we care about missing data

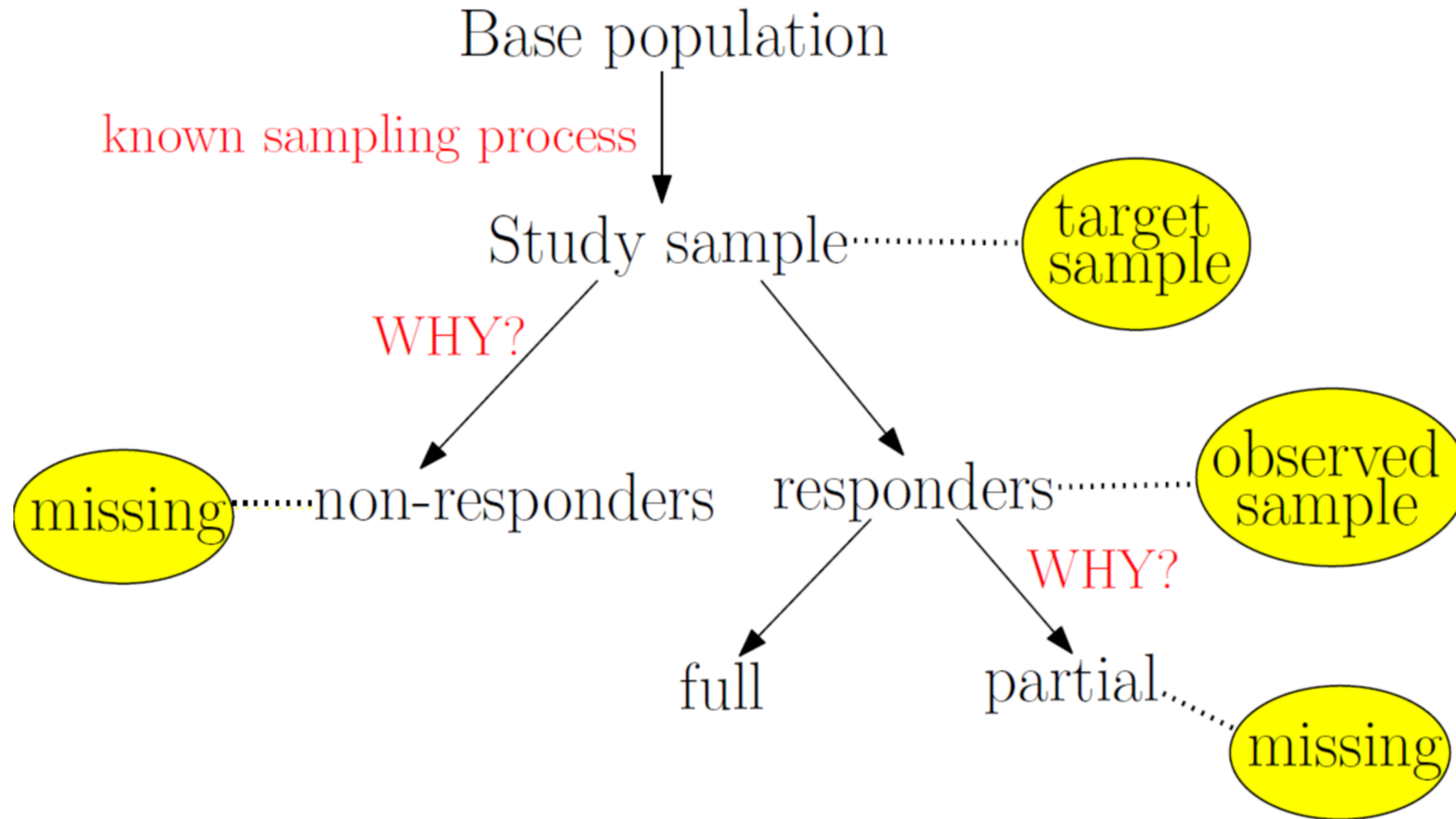
- Missing data are common!
- Usually inadequately handled in both observational and experimental research
- For example, reviewed 71 recently published BMJ, JAMA, Lancet and NEJM papers
 - 89% had partly missing outcome data
 - In 37 trials with repeated outcome measures, 46% performed complete case analysis
 - Only 21% reported sensitivity analysis
- reviewed articles using Multiple Imputation in BMJ, JAMA, Lancet and NEJM from 2002 to 2007
 - 59 articles found, with use doubling over 6 year period
 - However, the reporting was almost always inadequate

Outline

1. How do missing data arise?
2. Example: children height and weight
3. Bayesian imputation
4. Extending the model

How do missing data arise?

How do missing data arise?



Different types of missing data: MCAR

- There are three types of missing data, depending on why the missingness arise. Let's define m_i as the variable indicating if the i – th observation is missing

$$m_i \sim \text{Bernoulli}(p_i)$$

1. **Missing completely at random** (MCAR) occurs when the missing data are independent from the observed or unobserved data:

$$\text{logit}(p_i) = \theta_0$$

This means that the missing values can be ignored and the analysis can be conducted as usual.

Different types of missing data: MAR and MNAR

2. **Missing at random** (MAR) occurs when the missing data depends ONLY on the observed data:

$$\text{logit}(p_i) = \theta_0 + \mathbf{x}_i\theta_1$$

In this case, this can be introduced into the model so that missing observations are imputed as part of the model fitting.

Different types of missing data: MAR and MNAR

2. **Missing at random** (MAR) occurs when the missing data depends ONLY on the observed data:

$$\text{logit}(p_i) = \theta_0 + \mathbf{x}_i\theta_1$$

In this case, this can be introduced into the model so that missing observations are imputed as part of the model fitting.

3. **Missing non at random** (MNAR) occurs when the missing data depends on both the observed and missing data:

$$\text{logit}(p_i) = \theta_0 + \mathbf{x}_i\theta_1 + \lambda\mathbf{y}_i$$

This scenario is difficult to tackle since there is no information about the missingness mechanism and the missing data.

Missing response or missing covariates

- Additionally, it is crucial to distinguish between missing values **in the response** and **in the covariates**.
- When the missingness is in the response (and it is MCAR or MAR), these can naturally be predicted as the distribution of the response values is determined by the statistical model to be fit (posterior predictive distribution in the Bayesian approach)
- Missingness in the covariates requires additional steps:
 - a model needs to be specified on the covariate (imputation model) if the missingness mechanism is MCAR or MAR
 - an additional model of missingness needs to be specified if the mechanism is MNAR
 - Here we will consider only missing values in the response
 - R-INLA requires a certain degree of complexity to deal with missing values in covariates, if you are interested in learning more look at

Example: height and weight of children

Example: height and weight of children

- Information of 10,030 children measured within the Fifth Dutch Growth Study 2009
- Data available from the `fdgs` dataset in the `library(mice)`

Variable	Description	Missing
<code>id</code>	Child ID	0
<code>reg</code>	Region (5 levels)	0
<code>age</code>	Age (year)	0
<code>sex</code>	Sex	0
<code>hgt</code>	Height (cm)	23
<code>wgt</code>	Weight (kg)	20
<code>hgt.z</code>	Re-scaled height (as a Z-score)	23
<code>wgt.z</code>	Re-scaled weight (as a Z-score)	20

Summarising the data

- The data can be summarised using

```
> summary(fdgs)
```

id	reg	age	sex	hgt	wgt	hgt.z
Min. :100001	North: 732	Min. : 0.008214	boy :4829	Min. : 46.0	Min. : 2.585	Min. : -4.
1st Qu.:106353	East :2528	1st Qu.: 1.618754	girl:5201	1st Qu.: 83.8	1st Qu.: 11.600	1st Qu.: -0.
Median :203855	South:2931	Median : 8.084873		Median :131.5	Median : 27.500	Median : -0.
Mean :180091	West :2578	Mean : 8.157936		Mean :123.9	Mean : 32.385	Mean : -0.
3rd Qu.:210591	City :1261	3rd Qu.:13.547570		3rd Qu.:162.3	3rd Qu.: 51.100	3rd Qu.: 0.
Max. :401955		Max. :21.993155		Max. :208.0	Max. :135.300	Max. : 3.
				NA's :23	NA's :20	NA's :23

- Note that several variables in the dataset have missing observations. In particular, height (hgt) and weight (wgt), which are common variables used as response or predictors in models

Subsetting the data

- In order to provide a smaller dataset to speed up computations, only the children with missing values (in height and weight) and another 1000 ones taken at random will be used in the analysis

```
> # Subset 1, observations with NA's
> subset1 <- which(is.na(fdgs$wgt) | is.na(fdgs$hgt))
>
> #Subset 2, random sample of 1000 individuals
> set.seed(1)
> subset2 <- sample((1:nrow(fdgs))[-subset1], 1000)
> # Subset 1 + subset 2
> fdgs.sub <- fdgs[c(subset1, subset2), ]
> summary(fdgs.sub)
```

id	reg	age	sex	hgt	wgt	hgt.z
Min. :100098	North: 78	Min. : 0.07118	boy :493	Min. : 46.00	Min. : 2.585	Min. :-4.26
1st Qu.:106293	East :275	1st Qu.: 1.74264	girl:550	1st Qu.: 86.28	1st Qu.: 11.960	1st Qu.: -0.66
Median :204306	South:298	Median : 8.59411		Median :136.05	Median : 29.000	Median :-0.04
Mean :183214	West :250	Mean : 8.56536		Mean :127.04	Mean : 33.614	Mean :-0.02
3rd Qu.:211388	City :142	3rd Qu.:14.16427		3rd Qu.:165.10	3rd Qu.: 53.100	3rd Qu.: 0.64
Max. :401949		Max. :21.88364		Max. :199.00	Max. :117.300	Max. : 3.20
				NA's :23	NA's :20	NA's :23

Bayesian imputation

Model specification

- We first predict weight as a function of age and sex. We specify:

$$wgt_i \sim \text{Normal}(\alpha + \beta_1 sex_i + \beta_2 age_i, \sigma^2)$$

- where `wgt` is missing we will have NA in the corresponding vector, e.g.

```
> fdgs.sub$wgt[1:10]
```

```
[1] 23.200      NA      NA      NA  8.445      NA  6.960 10.420      NA 14.100
```

- Remember the posterior predictive distribution (presented in lecture 2.2):

$$p(\mathbf{y}^*|\mathbf{y}) = \int p(y^*|\theta)p(\theta|\mathbf{y})d\theta$$

– here \mathbf{y}^* identifies the missing values, while \mathbf{y} is the set of observed values for `wgt`

Running the model in R-INLA

```
> library("INLA")
> wgt.inla <- inla(wgt ~ age + sex, data = fdgs.sub,
+ control.predictor = list(compute = TRUE), control.compute=list(return.marginals.predictor=TRUE))
> wgt.inla$summary.fixed
```

	mean	sd	0.025quant	0.5quant	0.975quant	mode	kld
(Intercept)	6.259187	0.43576270	5.404524	6.259188	7.113849	6.259188	8.95248e-12
age	3.330942	0.03367352	3.264898	3.330942	3.396986	3.330942	8.61623e-12
sexgirl	-2.378945	0.44587122	-3.253432	-2.378945	-1.504455	-2.378945	8.93984e-12

```
> wgt.inla$summary.hyperpar
```

	mean	sd	0.025quant	0.5quant	0.975quant	mode
Precision for the Gaussian observations	0.01972943	0.0008729198	0.01805322	0.01971746	0.02147696	0.01969183

- Note that we need to include `control.predictor=list(compute = TRUE)` so INLA can estimate the predictive distribution for the missing observations, while `control.compute=list(return.marginals.predictor=TRUE)` tells inla that we want to access the entire posterior distribution of the prediction (rather than only the summary)

Getting the posterior prediction

We now subset the children indexes with missing values so we can report their predictive distributions:

```
> wgt.na <- which(is.na(fdgs.sub$wgt))  
> rownames(fdgs.sub)[wgt.na]
```

```
[1] "275" "1278" "1419" "2135" "2684" "2940" "3069" "3189" "3543" "4262" "5687" "6485" "7101" "7108" "7506"
```

```
> # Obtain the predictive distribution  
> wgt.inla$summary.fitted.values[wgt.na, c("mean", "sd")][1:5,]
```

	mean	sd
fitted.Predictor.0002	23.067927	0.3200569
fitted.Predictor.0003	27.617341	0.3328551
fitted.Predictor.0004	54.019923	0.3765046
fitted.Predictor.0006	13.144502	0.3926908
fitted.Predictor.0009	7.838159	0.3936613

Remember that you can also access the marginal posterior distributions (rather than the summary) using `wgt.inla$marginals.fitted.values`

Imputing hgt using the same approach

- Similarly, a model can be fit to explain height based on age and sex and to compute the predictive distribution of the missing observations:

```
> hgt.inla <- inla(hgt ~ age + sex, data = fdgs.sub,  
+ control.predictor = list(compute = TRUE),  
+ control.compute = list(return.marginals.predictor=TRUE))  
> hgt.inla$summary.fixed
```

	mean	sd	0.025quant	0.5quant	0.975quant	mode	kld
(Intercept)	77.148117	0.72109712	75.733811	77.148118	78.562415	77.148118	1.005731e-11
age	6.022575	0.05549259	5.913736	6.022575	6.131413	6.022575	1.096012e-11
sexgirl	-4.716069	0.73116459	-6.150109	-4.716071	-3.282014	-4.716071	1.096018e-11

```
> hgt.inla$summary.hyperpar
```

	mean	sd	0.025quant	0.5quant	0.975quant	m
Precision for the Gaussian observations	0.007403866	0.0003304386	0.006769282	0.00740418	0.008066786	0.007423

Imputing hgt using the same approach

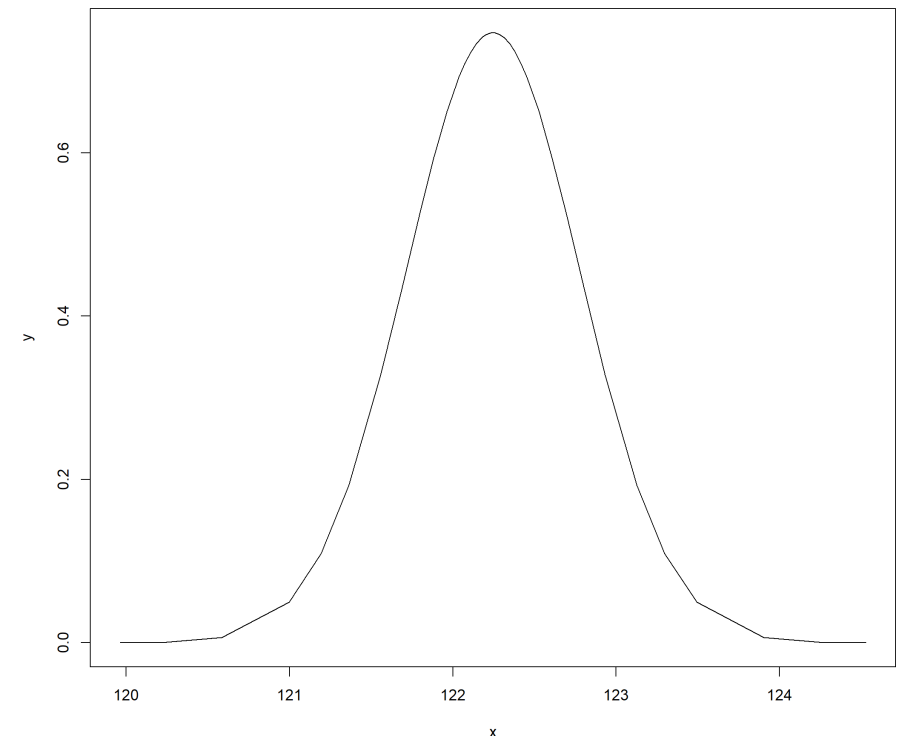
- We can obtain the predictions using

```
> hgt.na <- which(is.na(fdgs.sub$hgt))  
> hgt.inla$summary.fitted.values[hgt.na, c("mean",
```

	mean	sd
fitted.Predictor.0001	122.24529	0.5361333
fitted.Predictor.0005	83.24901	0.6840796
fitted.Predictor.0007	74.64156	0.6794198
fitted.Predictor.0008	82.80381	0.6866698
fitted.Predictor.0010	88.26140	0.6020496
fitted.Predictor.0012	96.55557	0.6143018
fitted.Predictor.0013	86.31595	0.6666480
fitted.Predictor.0016	81.84746	0.6922833
fitted.Predictor.0017	77.75821	0.7170109
fitted.Predictor.0018	81.59988	0.6379852

- We can plot the entire posterior distributions for each child with:

```
> # First child with missing value  
> plot(hgt.inla$marginals.fitted.values[[hgt.na[1]
```



Extending the model

Joint model of height and weight

- The two previous models consider height and weight separately, but it is clear that there is a high correlation between height and weight, which is caused by the age and sex of the child.

[1] 0.9999865

- We can build a joint model for height and weight to exploit a correlated effect between the coefficients of age in both models.

$$\begin{aligned} hgt_i &= \alpha_h + \beta_{h1}sex_i + \beta_{h2}age_i + \epsilon_{1i} \\ wgt_i &= \alpha_w + \beta_{w1}sex_i + \beta_{w2}age_i + \epsilon_{2i} \end{aligned}$$

with

- α_h, α_w model intercepts
- β_{h1}, β_{w1} the effect of sex
- β_{h2}, β_{w2} the effect of age
- ϵ_1, ϵ_2 the error terms (note that this specification is equivalent to the one above for the separate models)

Joint model of height and weight: prior

- The vectors (β_{1h}, β_{1w}) and (β_{2h}, β_{2w}) are modeled using a multivariate Gaussian distribution with mean 0 and covariance matrix with $1/\tau_{hj}$ and $1/\tau_{wj}$ as variances and $\rho_j / \sqrt{(\tau_{jh}\tau_{jw})}$ as the covariance where ρ_j is the correlation parameter.
- First, the bivariate response variable needs to be put in a two-column matrix given that the model will be made of two data distributions

```
> n <- nrow(fdgs.sub)
> y <- matrix(NA, nrow = 2 * n, ncol = 2)
> y[1:n, 1] <- fdgs.sub$hgt
> y[n + 1:n, 2] <- fdgs.sub$wgt
```

- Similarly, as we have two intercepts, we need to define these explicitly as covariates with all values equal to one:

```
> I <- matrix(NA, nrow = 2 * n, ncol = 2)
> I[1:n, 1] <- 1
> I[n + 1:n, 2] <- 1
```

Correlated effects

- Now we need to define the correlated effects. We will use the random effect specification similar to what we saw with the hierarchical models (`iid`), but modified to have the two coefficients as correlated `f(...,model=iid2d)`.
- In order to do so we need to modify the variables `age` and `sex` as they will be passed to the model as weights of the latent random effects `iid2d` specification. We need them to be twice as long to match the dimension of the response:

```
> age.joint <- rep(fdgs.sub$age, 2)
> sex.joint <- rep(fdgs.sub$sex, 2)
```

- Finally we need two index vectors to indicate which coefficient to use from the `iid2d` model is required. These indexes will be 1 for the first half of observations (to indicate that the coefficient is β_h and 2 for the second half (to indicate that the coefficient is β_w).

```
> idx.age = rep(1:2, each = n)
> idx.sex = rep(1:2, each = n)
```


Model fitting

The model is fit and summarized as seen below

```
> # Model formula
> joint.f <- y ~ -1 + I + f(idx.sex, sex, model="iid2d", n=2) + f(idx.age, age, model = "iid2d", n = 2)
> # Model fit
> fdgs.joint <- inla(joint.f,
+   data = list(y = y, I = I, sex = sex.joint, age = age.joint, idx.age = idx.age, idx.sex= idx.sex),
+   family = rep("gaussian", 2),
+   control.predictor = list(compute = TRUE))
> # Summary fixed (intercept)
> fdgs.joint$summary.fixed
```

	mean	sd	0.025quant	0.5quant	0.975quant	mode	kld
I1	81.124354	1.1467496	78.89268	81.118125	83.391094	81.117990	7.177745e-09
I2	8.512301	0.6799152	7.18675	8.509552	9.853253	8.509475	4.004876e-09

Hyperparameters and variable effects

- The hyperparameters can be obtained through

```
> fdgs.joint$summary.hyperpar
```

	mean	sd	0.025quant	0.5quant	0.975quant	
Precision for the Gaussian observations	0.007381468	0.0003158389	0.006760246	0.00738111	0.008003156	0.007
Precision for the Gaussian observations[2]	0.019730455	0.0008577595	0.017983707	0.01974962	0.021349126	0.019
Precision for idx.sex (component 1)	0.229098980	0.0421660917	0.147725213	0.22842312	0.310628523	0.234
Precision for idx.sex (component 2)	0.752987237	0.1325558166	0.530631196	0.73997939	1.051043080	0.711
Rho1:2 for idx.sex	0.913712999	0.0150280999	0.885675060	0.91371111	0.943606611	0.910
Precision for idx.age (component 1)	0.344366034	0.0564052256	0.252994487	0.33791276	0.474058828	0.321
Precision for idx.age (component 2)	1.132872604	0.1700804329	0.828579940	1.12244242	1.496724034	1.105
Rho1:2 for idx.age	0.843269536	0.0278518255	0.779593953	0.84634295	0.888782528	0.854

- While the coefficients for age and sex are part of the random effects of the model:

```
> #Sex  
> fdgs.joint$summary.random$idx.sex
```

```
> #Age  
> fdgs.joint$summary.random$idx.age
```

ID	mean	sd	0.025quant	0.5quant	0.975quant	ID	mean	sd	0.025quant	0.5quant	0.975quant	
1	1	-4.235610	0.6346222	-5.496296	-4.229926	-3.006711	640	2196480	109520628	-085.913552	6.021964	6.130
2	2	-2.285555	0.3740273	-3.026223	-2.282892	-1.559643	323	2821720	102352324	-083.263077	3.328916	3.394

Estimates of missing values

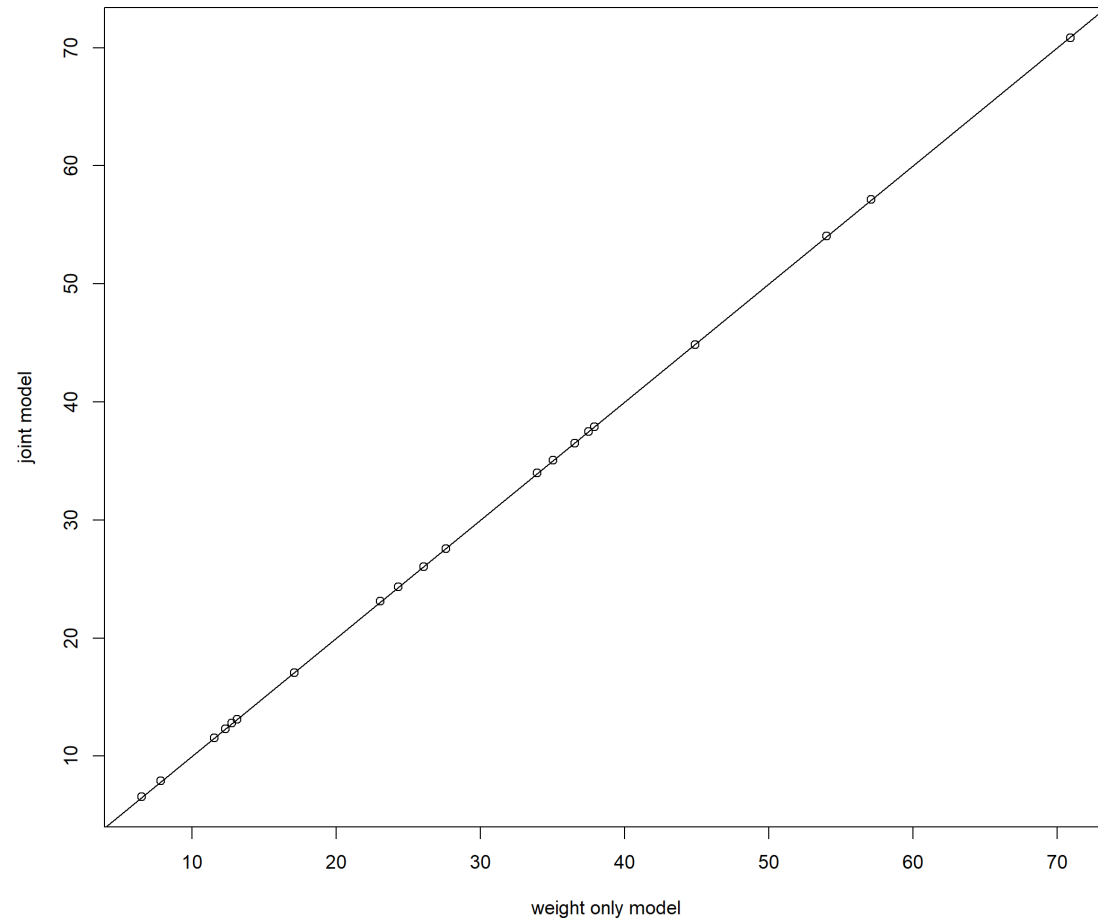
- Finally to access the predicted values for the children with missing data we need to remember that we now have the response y stacked (first height, then weight)
- As height is the first variable we can use `hgt.na` to get to the indexes of the children with missing data
- For weight we need to use `wgt.na` and add `n` to each index, as this is the total number of observations, so the data for weight will start on the index 1044

```
> joint.wgt.na = wgt.na+n
```

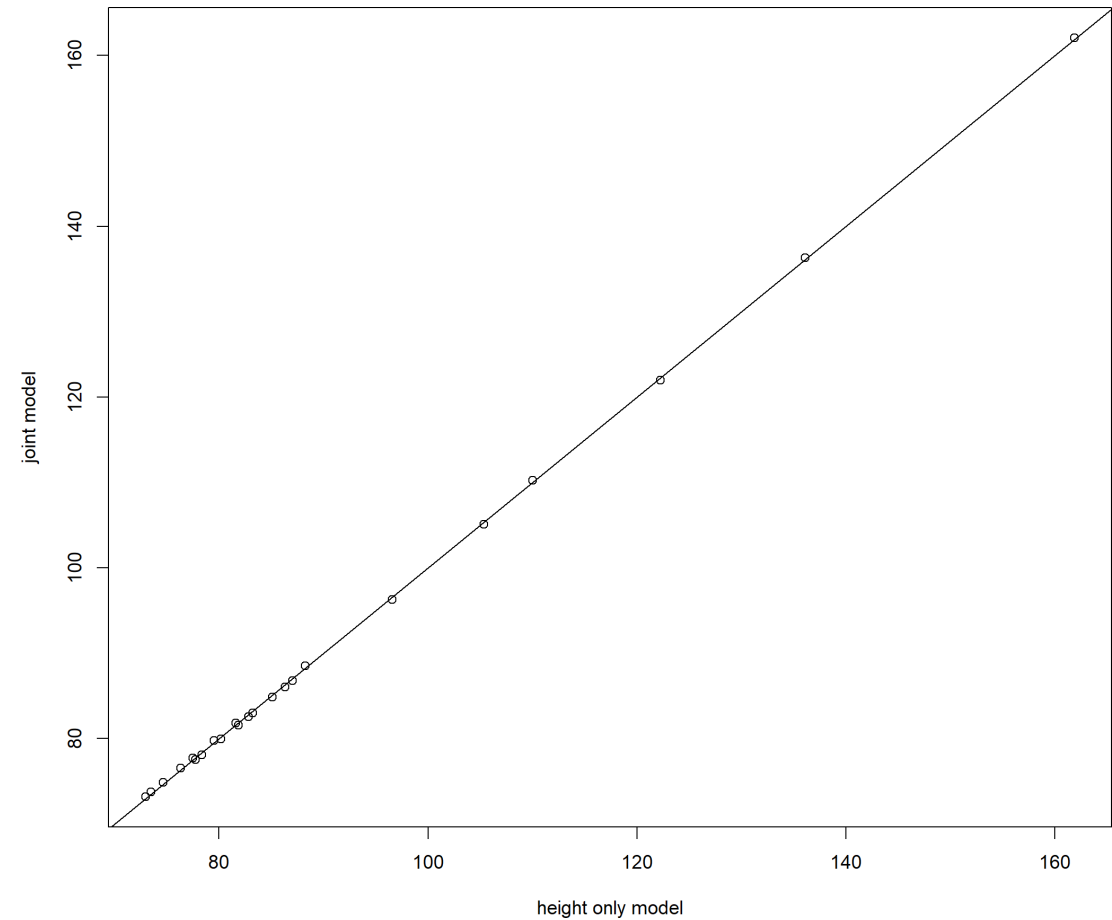
```
> #Height  
> fdgs.joint$summary.fitted.values[hgt.na, c("mean", "sd")][1:10,]  
> #Weight  
> fdgs.joint$summary.fitted.values[joint.wgt.na, c("mean", "sd")][1:10,]
```

Comparing the results of the joint and separate models

Weight



Height



References

Gómez-Rubio, V. (2020a). *Bayesian inference with INLA*. CRC Press.