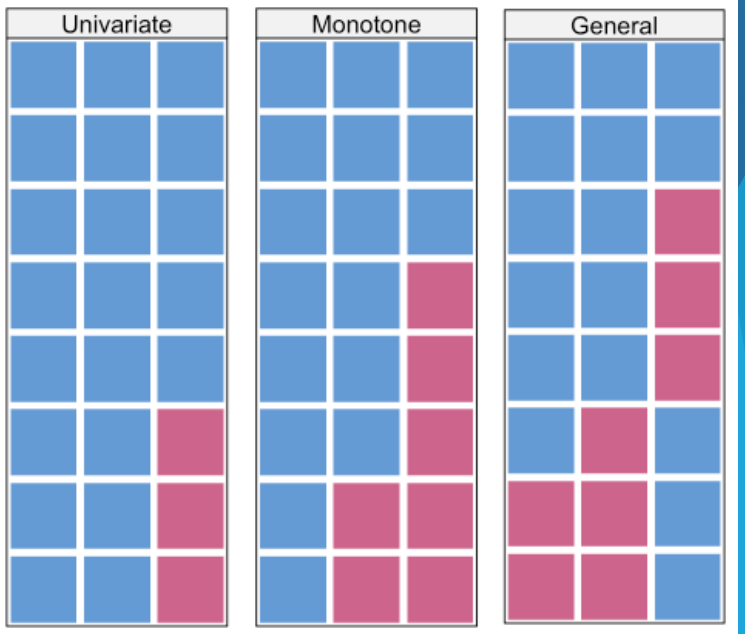
# Dealing with Missing Data – Multiple Imputation

Dr Lisanne A Gitsels
29 March 2019

# Is there missing data?

- What is missing?
  - R: >summary(*dataset*)
- How much is missing?
  - R: for number use >summary(*dataset*)
    for percentage use >prop.table(table(is.na(*dataset*$*covariate*)))*100
- Why is there missing data?
  - Is there a pattern? R: >library(mice) >md.pattern(*dataset*)
  - Who are the subjects with missing data?
  - What type of missing data?
  - More on next slides…

```
    SMOKING          regstat           weight      
Min.   :1.000   Min.   : 1.00   Min.   : 10.00  
1st Qu.:1.000   1st Qu.: 2.00   1st Qu.: 65.30  
Median :1.000   Median : 2.00   Median : 75.90  
Mean   :1.426   Mean   :22.03   Mean   : 77.17  
3rd Qu.:2.000   3rd Qu.: 5.00   3rd Qu.: 87.40  
Max.   :3.000   Max.   :99.00   Max.   :177.00  
NA's   :29314                   NA's   :108710  
```



# Types of missing data

- Missing completely at random (MCAR)
  - No systematic difference between observed and unobserved data
  - For example, cholesterol readings missing due to breakdown equipment
- Missing at random (MAR)
  - Systematic difference between observed and unobserved data and this can completely be explained by observed data
  - For example, cholesterol readings more likely to be missing in younger patients
- Missing not at random (MNAR)
  - Systematic difference between observed and unobserved data and this can at least partly be explained by unobserved data
  - For example, cholesterol readings are more likely to be missing in patients who do not adhere to drug therapy, which is not in the data

- No formal test but could assess associations between missingness and other covariates

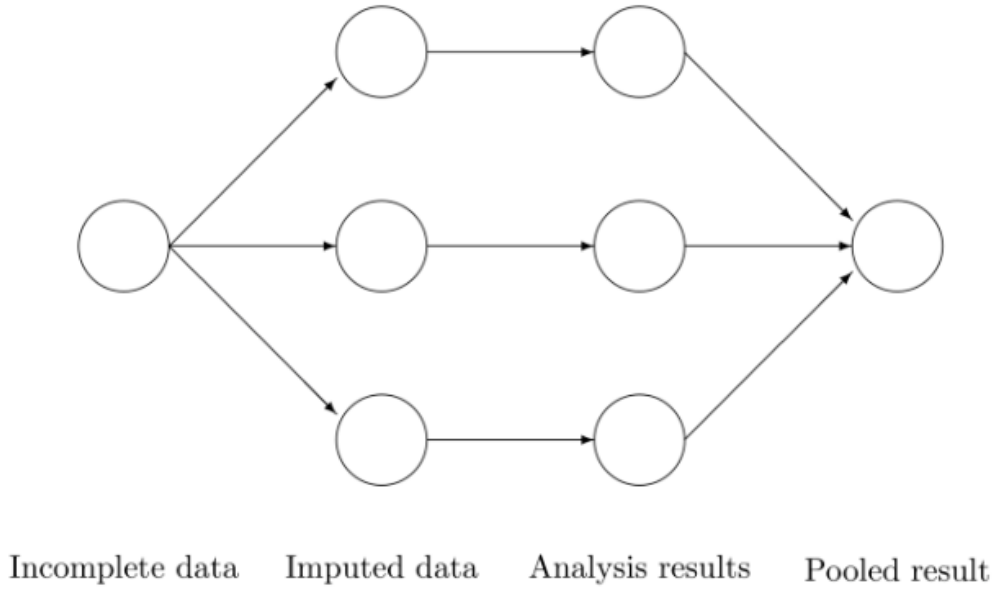
# Missingness in primary care data

- Systematic difference between observed and unobserved data
- Recording of medical, lifestyle, and socio-demographic information related to
  - Frequency doctor visits (e.g. ill-health and women)
  - Specific medical condition
- Introduction of the Quality and Outcomes Framework (QOF) improved recording in primary care
  - QOF is a pay scheme to improve the quality of the health care provided by general practitioners
- Standard approach to assume that patients who do not have a diagnosis or prescription, do not have the medical condition or receive the treatment.

# Methods to deal with missing data

- Complete case analysis
  - Assumes that complete cases represent full dataset and MCAR
  - Reduces sample size and statistical power of tests
- Exclude variables with incomplete records
  - Leads to biased estimates if covariate with missing data is a confounder
- Create missing data category
  - Leads to biased estimates because it distorts correlations with other covariates
- Single imputation
  - If substituting with mean, then correlation structure would be distorted
  - If substituting with regression estimate, then false precision

# Multiple imputation




- Goal: making valid statistical inferences by reflecting the uncertainty in missing data
  - NB: imputation is not prediction, i.e. imputing the true values
- Assumes missing data are MAR
  - Use when 5-50% of data is missing

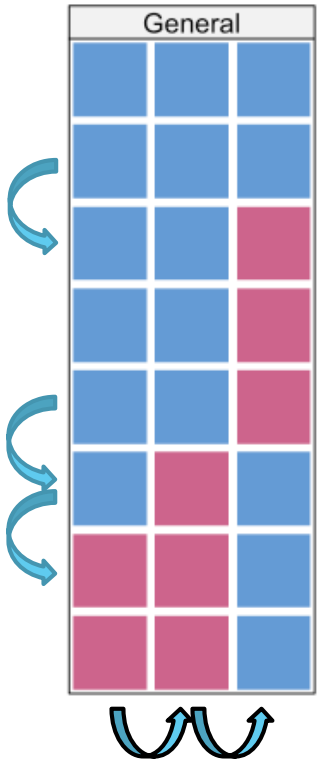
# Imputation model

- Should reflect analysis model
  - i.e. outcome, exposure, confounders, interactions and random effect
- Include other covariates that are associated with missingness
- Include up to 30 covariates
  - Although 15+ hardly increases the explained variance in the imputed covariate
- Type depends on measurement scale of covariate with missing data
  - Continuous data (linear regression), binary data (logistic regression) and categorical data (multinomial/ordinal logistic regression)
  - Impute on original measurement scale and transform after imputation

# Imputation process

## Joint modelling

- Imputes by missing data pattern (i.e. row-by-row basis)
- Advantage: better theoretical properties (data conform to modelling assumptions), less computation intensive (i.e. faster)
- Disadvantage: less robust under imputation model misspecification (could potentially be solved by more iterations)
- R package jomo

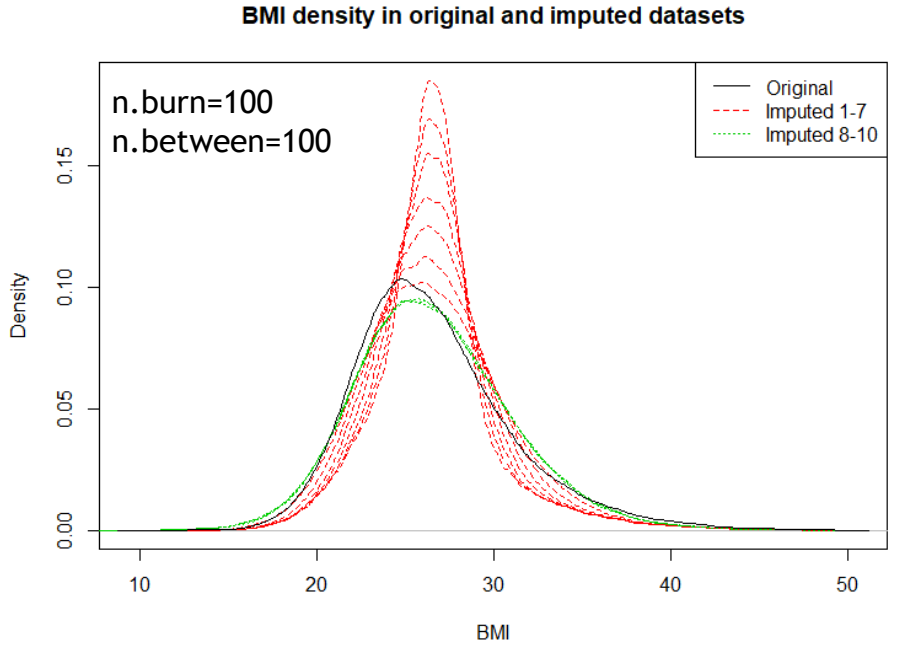


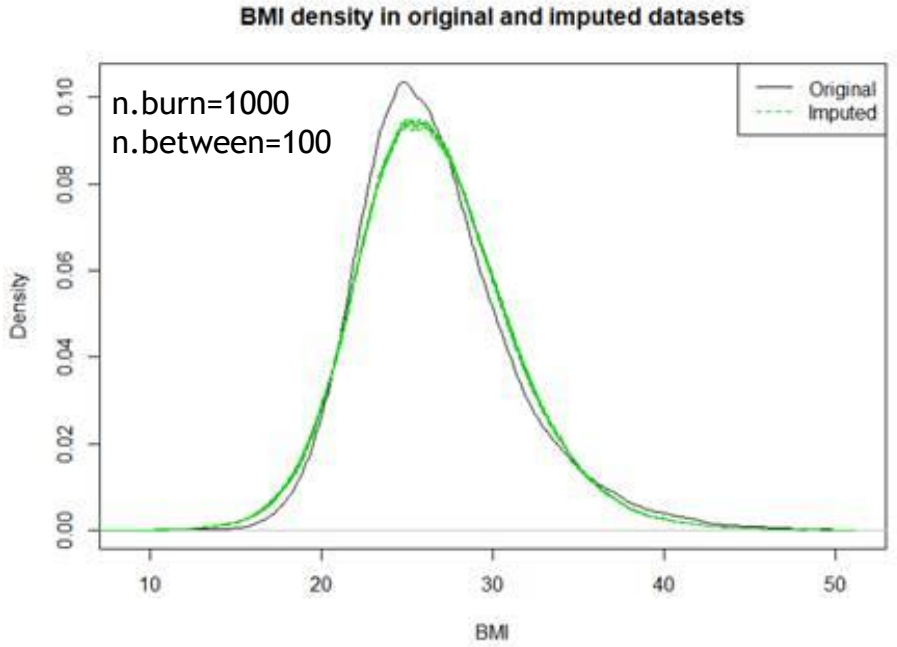

## Chained-equations

- Also known as fully conditional
- Imputes on a variable-by-variable basis
- Advantage: more flexible specifications, more robust under misspecification
- Disadvantage: more computation intensive (i.e. slower)
- R package mice

# Imputation iteration process

- Burn-in length: iteration process until convergence in the estimated regression coefficients
  - Check by plotting the coefficients of the imputation model against the iterations and plotting imputed values by bar and density plots. If no convergence, then increase burn-in length.
  - Ranging between 5 and 10,000s, default 100/1000
- Between iterations: number of iterations between imputed datasets
  - To ensure independent datasets
  - Ranging between 5 and 10,000s, defaults 100/1000
- Number of imputed datasets:
  - Roughly the percentage of missing records
  - Could use as little as 5, default is at least 10
- If missing data are MAR, then imputed values should have a similar distributions as the recorded observations



# Multiple imputation in R

- >library(jomo)
- *Imputation_dataset*: create dataset with all information for imputation model, where missing observations are set to NA and binary/categorical covariates are saved as factors.
  - Note that with survival data, the survival outcome is included in the imputation model as two covariates: continuous ‘log(survival_time)’ and binary ‘survival_status’
- *Cluster*: create vector of clustering covariate (here practice id)
- >imp=jomo(*imputation_dataset*,clus=*cluster*,nburn=100,nbetween=100,nimp=10)
  - Result will be dataset *imp* which is *Imputation_dataset* with *nimp* imputed datasets and extra column ‘Imputation’ where 0=original data and 1:*nimp*=imputed data. In other words, the original data and imputed data are stacked in one dataset in a long format.

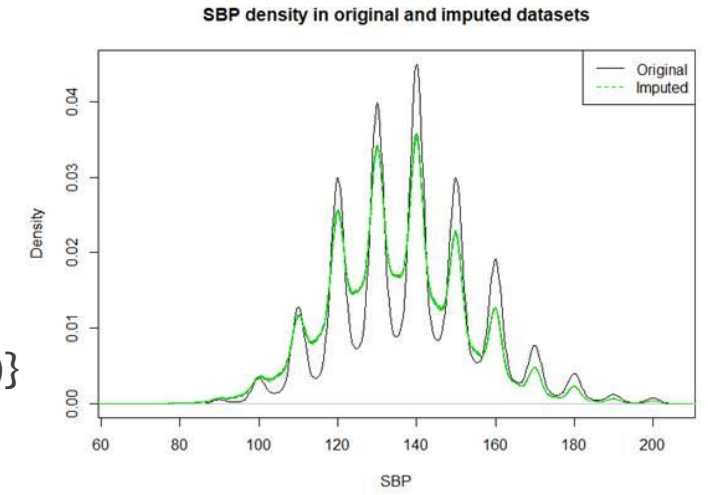
# Check imputations

- Continuous covariate:

>plot(density(*imp*$*imputed_covariate*[*imp*$Imputation==0],na.rm=TRUE),lty=1, col=1,xlab='*Name imputed covariate*',ylab='Density',main='*Name imputed covariate* in original and imputed datasets')

>for (i in 1:nimp) {lines(density(*imp*$*imputed_covariate*[*imp*$Imputation==i]),lty=2,col=3)}

>legend('topright',legend=c("Original","Imputed"),lty=c(1,2),col=c(1,3))

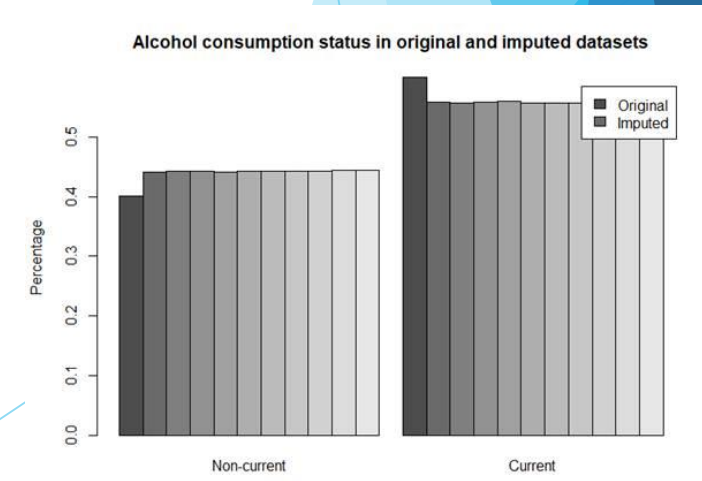



- Categorical covariate:

>Imputed_contrasts=matrix(nrow = 1+nimp, ncol = ncat)

>for (i in 0:nimp)
{Imputed_contrasts[i+1,c(1:ncat)]=table(*imp*$*imputed_covariate*[*imp*$Imputation==i])}

>barplot(prop.table(*imputed_covariate*,1),beside=TRUE,ylab='Percentage',main='*Name imputed covariate* in original and imputed datasets', names.arg = c("*Category names*"),legend=c("Original","Imputed"))



# Run survival model on imputed datasets

R packages needed: survival, mice, jomo, dplyr, and mitml

#save data in mitml object
>data_mitml<-jomo2mitml.list(*imp*)

#fit survival model on imputed datasets saved in the mitml object
>model_fit<- (data = data_mitml, exp = coxph(Surv(survival_time, survival_status) ~ exposure + others + frailty(pracid, sparse=TRUE))

#pool results (details on next slide)
>pooled_hr_95ci(model_fit)

#check proportional hazards assumption (i=imputed dataset 1 to nimp)
> cox.zph(model_fit[[i]])

# Pooling results of imputed datasets

```
#write own pool function for frailty
pooled_hr_95ci=function(fitted_model){
    ##covariate names
    name_coeff=names(fitted_model[[1]]$coefficients)
    ##number of estimates in model
    no_coeff=length(fitted_model[[1]]$coefficients)
    ##number of imputed datasets / fitted models
    no_imp=length(fitted_model)
    ##pooling of estimates: mean across imputed datasets
    matrix_individual_coef=matrix(,nrow=no_coeff,ncol=no_imp)
    for (i in 1:no_imp){
        matrix_individual_coef[,i]=fitted_model[[i]]$coefficients
        }
    pooled_coef=rowMeans(matrix_individual_coef)
    pooled_hr=exp(pooled_coef)

    #pooling of variance: calculate variance within and between imputed datasets
    var_within_matrix=matrix(,nrow=no_coeff,ncol=no_imp)
    for (i in 1:no_imp) {
        var_within_matrix[,i]=diag(fitted_model[[i]]$var)
        }
    var_within=rowMeans(var_within_matrix)
    var_between_matrix=matrix(,nrow=no_coeff,ncol=no_imp)
    for (i in 1:no_imp) {
        var_between_matrix[,i]=(fitted_model[[i]]$coef-pooled_coef)^2
        }
    var_between=rowSums(var_between_matrix)*(1/(no_imp-1))
    var_total=var_within+(1+(1/no_imp))*var_between
    se_coef=sqrt(var_total)
```

Rubin's Rules:

$$\hat{\beta} = \frac{1}{m}\sum_{j=1}^{m}\hat{\beta}_j,$$

$$\widehat{\mathrm{Var}}(\hat{\beta}) = \widehat{W} + \left(1+\frac{1}{m}\right)\widehat{B}, \text{where}$$

$$\widehat{W} = \frac{1}{m}\sum_{j=1}^{m}\widehat{W}_j,$$

$$\widehat{B} = \frac{1}{m-1}\sum_{j=1}^{m}(\hat{\beta}_j - \hat{\beta})^2.$$

Total variance from 3 sources:
1. W, sample variance
2. B, extra variance caused by missing data
3. B/m, extra simulation variance due to finite imputed datasets

# Pooling results of imputed datasets – cont.

- Test whether coefficient is significantly different from zero
  - Since the total variance is not known a priori, the estimate follows a t-distribution rather than the normal.

$$t = \frac{\hat{\beta}}{\widehat{\mathrm{se}}(\hat{\beta})}, \qquad df = (m-1)\left(1+\frac{1}{r}\right)^2, \text{where}$$

$$r = \frac{\widehat{B}(1+1/m)}{\widehat{W}}.$$

```
t_statistic=pooled_coef/se_coef
t_df=(no_imp-1)*(1+(no_imp*var_within)/((no_imp+1)*var_between))^2
t_p=2*(1-(pt(abs(t_statistic),t_df)))

hr_lcl=exp(pooled_coef-(qt(0.975,t_df)*se_coef))
hr_ucl=exp(pooled_coef+(qt(0.975,t_df)*se_coef))

results=as.data.frame(cbind('Covariate'=name_coeff,'HR'=round(pooled_hr,3),
'95%LCI'=round(hr_lcl,3),'95%UCI'=round(hr_ucl,3)))
print(results)
}
```

# More information

- Book on multiple imputation by Allison: https://us.sagepub.com/en-us/nam/missing-data/book9419
- Book on multiple imputation by Stef van Buuren: https://stefvanbuuren.name/fimd/
- R package mice (by Stef van Buuren): https://cran.r-project.org/web/packages/mice/mice.pdf
  Tutorial of mice https://www.jstatsoft.org/article/view/v045i03
- R package jomo: https://cran.r-project.org/web/packages/jomo/jomo.pdf
- Multilevel multiple imputation tutorial https://www.jstatsoft.org/article/view/v045i05
  (REALCOM-IMPUTE software but still insightful how multiple imputation works for hierarchical data)