Advances in ML: Theory Meets Practice

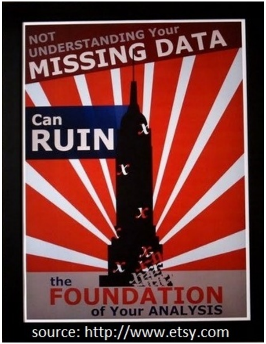
Julie Josse

Review on Missing Values Methods with Demos

Lausanne, 26 January

Dealing with missing values

- PCA with missing values/Matrix completion
- Categorical/mixed data



PCA (complete)

Find the subspace t ghat best represents the data

Figure 1: Camel or dromedary?

$\Rightarrow$ Best approximation with projection

$\Rightarrow$ Best representation of the variability

$\Rightarrow$ Do not distort the distances between individuals

PCA (complete)

Find the subspace t ghat best represents the data

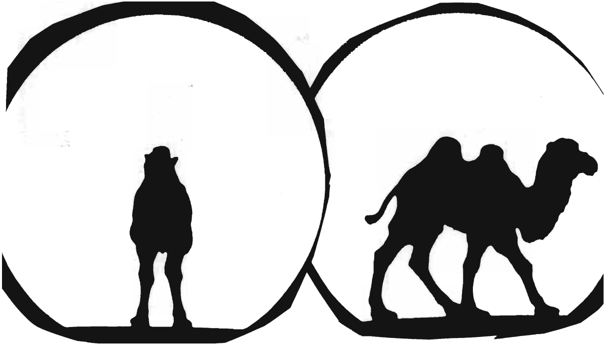


Figure 1: Camel or dromedary? source J.P. Fénelon

⇒ Best approximation with projection

⇒ Best representation of the variability

⇒ Do not distort the distances between individuals

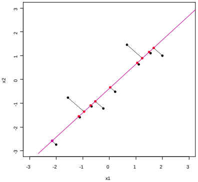
PCA reconstruction

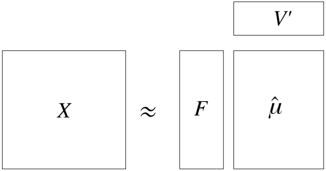
X

-2.00	-2.74
-1.56	-0.77
-1.11	-1.59
-0.67	-1.13
-0.22	-1.22
0.22	-0.52
0.67	1.46
1.11	0.63
1.56	1.10
2.00	1.00

$\hat{\mu}$

-2.16	-2.58
-0.96	-1.35
-1.15	-1.55
-0.70	-1.09
-0.53	-0.92
0.04	-0.34
1.24	0.89
1.05	0.69
1.50	1.15
1.67	1.33





$\Rightarrow$ Minimizes distance between observations and their projection

$\Rightarrow$ Approx $X_{n\times p}$ with a low rank matrix $S < p$ $\|A\|_2^2 = \mathrm{tr}(AA^\top)$:

$$\mathrm{argmin}_\mu \left\{ \|X - \mu\|_2^2 : \mathrm{rank}\,(\mu) \le S \right\}$$

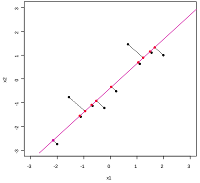
PCA reconstruction

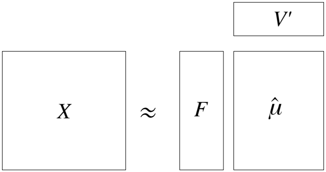
X

-2.00	-2.74
NA	-0.77
-1.11	-1.59
-0.67	-1.13
-0.22	NA
0.22	-0.52
0.67	1.46
NA	0.63
1.56	1.10
2.00	1.00

$\hat{\mu}$

-2.16	-2.58
-0.96	-1.35
-1.15	-1.55
-0.70	-1.09
-0.53	-0.92
0.04	-0.34
1.24	0.89
1.05	0.69
1.50	1.15
1.67	1.33





$\Rightarrow$ Minimizes distance between observations and their projection

$\Rightarrow$ Approx $X_{n \times p}$ with a low rank matrix $S < p$ $\|A\|_2^2 = \mathrm{tr}(AA^\top)$:

$$\mathrm{argmin}_{\mu} \left\{ \|X - \mu\|_2^2 : \mathrm{rank}\,(\mu) \leq S \right\}$$

SVD X: $\hat{\mu}^{\mathrm{PCA}} = U_{n \times S} \Lambda^{\frac{1}{2}}_{S \times S} V'_{p \times S} = F_{n \times S} V'_{p \times S}$

$F = U\Lambda^{\frac{1}{2}}$ PC - scores

V principal axes - loadings

Missing values in PCA

$\Rightarrow$ PCA: least squares

$$\text{argmin}_{\mu} \left\{ \| X_{n \times p} - \mu_{n \times p} \|_2^2 : \text{rank}(\mu) \leq S \right\}$$

$\Rightarrow$ PCA with missing values: weighted least squares

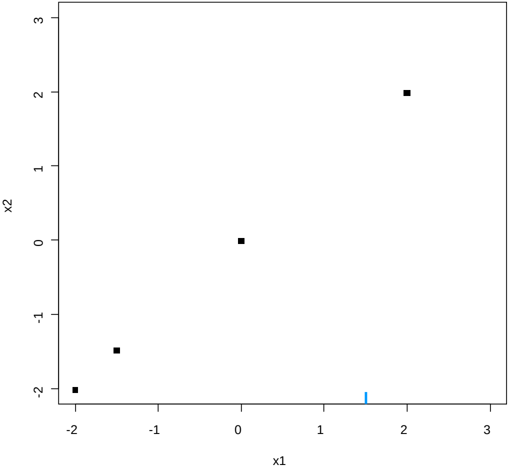
$$\text{argmin}_{\mu} \left\{ \| W_{n \times p} * (X - \mu) \|_2^2 : \text{rank}(\mu) \leq S \right\}$$

with $W_{ij} = 0$ if X_{ij} is missing, $W_{ij} = 1$ otherwise; $*$ elementwise multiplication

Many algorithms: weighted alternating least squares (Gabriel & Zamir, 1979); iterative PCA (Kiers, 1997)

Iterative PCA

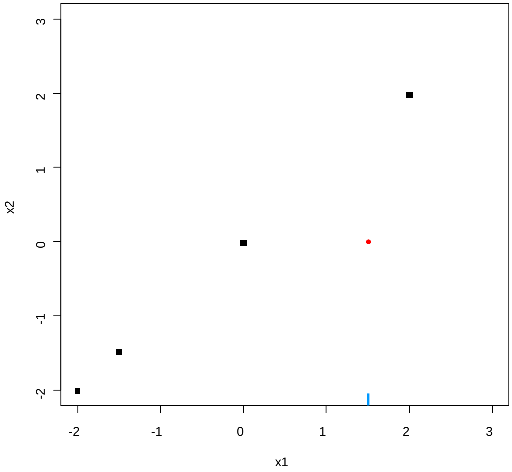
x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	NA
2.0	1.98



Iterative PCA

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	NA
2.0	1.98

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.00**
2.0	1.98



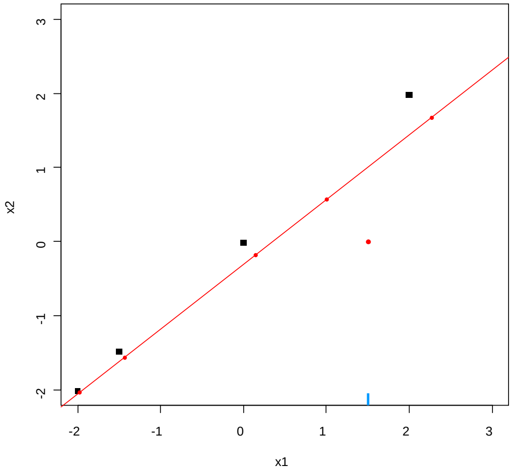
Initialization $\ell = 0$: X^0 (mean imputation)

Iterative PCA

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	NA
2.0	1.98

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.00**
2.0	1.98

$\widehat{x1}$	$\widehat{x2}$
-1.98	-2.04
-1.44	-1.56
0.15	-0.18
1.00	0.57
2.27	1.67



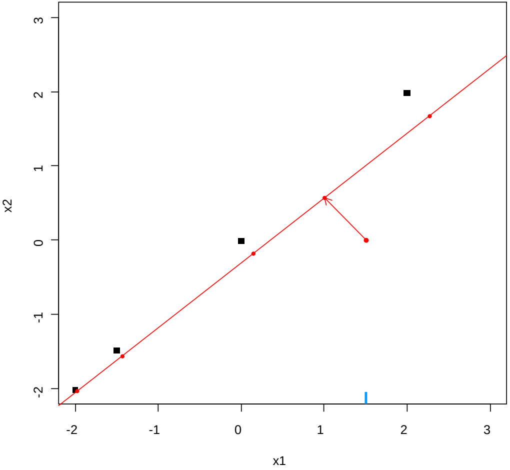
PCA on the completed data set $\rightarrow (U^{\ell}, \Lambda^{\ell}, V^{\ell})$;

Iterative PCA

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**NA**
2.0	1.98

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.00**
2.0	1.98

$\widehat{x1}$	$\widehat{x2}$
-1.98	-2.04
-1.44	-1.56
0.15	-0.18
1.00	**0.57**
2.27	1.67



Missing values imputed with the fitted matrix $\hat{\mu}^{\ell} = U^{\ell} {\Lambda^{1/2}}^{\ell} V^{\ell\prime}$

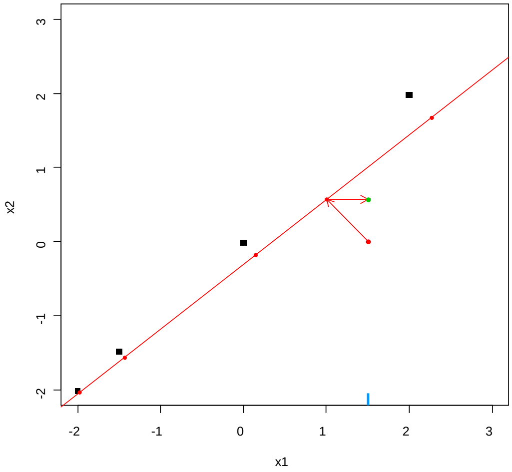
Iterative PCA

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	NA
2.0	1.98

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.00**
2.0	1.98

$\widehat{x1}$	$\widehat{x2}$
-1.98	-2.04
-1.44	-1.56
0.15	-0.18
1.00	**0.57**
2.27	1.67

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.57**
2.0	1.98

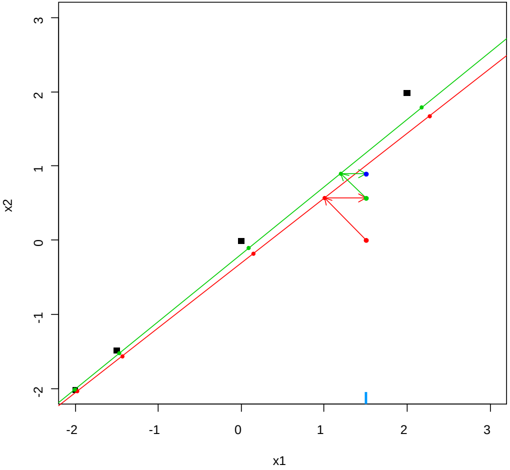


The new imputed dataset is $\hat{X}^{\ell} = W * X + (\mathbf{1} - W) * \hat{\mu}^{\ell}$

Iterative PCA

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	NA
2.0	1.98

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.57**
2.0	1.98



x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.57**
2.0	1.98

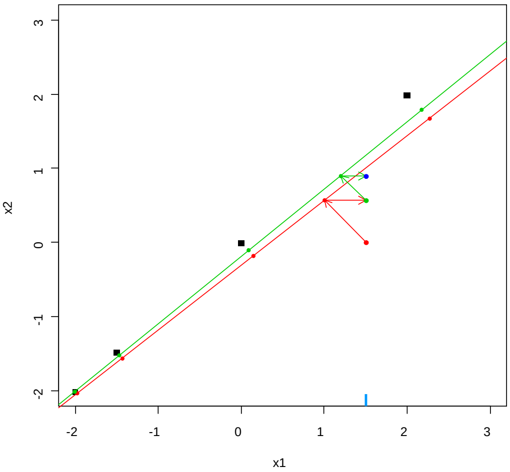
Iterative PCA

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	NA
2.0	1.98

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.57**
2.0	1.98

$\hat{x1}$	$\hat{x2}$
-2.00	-2.01
-1.47	-1.52
0.09	-0.11
1.20	**0.90**
2.18	1.78

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.90**
2.0	1.98



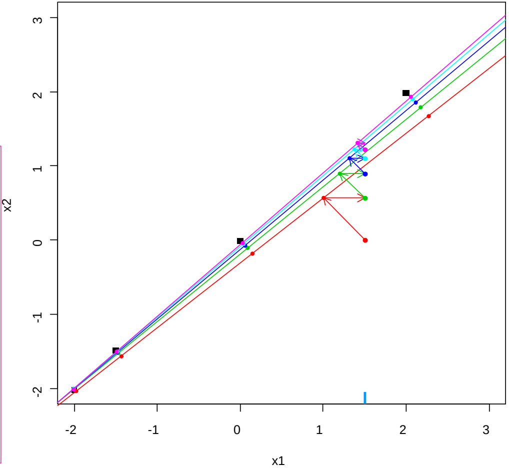
Iterative PCA

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**NA**
2.0	1.98

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.00**
2.0	1.98

$\widehat{x1}$	$\widehat{x2}$
-1.98	-2.04
-1.44	-1.56
0.15	-0.18
1.00	0.57
2.27	1.67

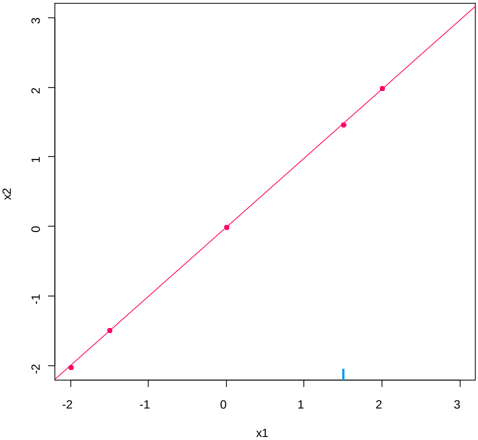
x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**0.57**
2.0	1.98



Steps are repeated until convergence

Iterative PCA

x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**NA**
2.0	1.98



x1	x2
-2.0	-2.01
-1.5	-1.48
0.0	-0.01
1.5	**1.46**
2.0	1.98

PCA on the completed data set $\rightarrow (U^\ell, \Lambda^\ell, V^\ell)$

Missing values imputed with the fitted matrix $\hat{\mu}^\ell = U^\ell \Lambda^{1/2^\ell} V^{\ell\prime}$

Iterative PCA

1. initialization $\ell = 0$: X^0 (mean imputation)
2. step ℓ:
   (a) PCA on the completed data $\rightarrow (U^\ell, \Lambda^\ell, V^\ell)$;
   S dimensions kept
   (b) missing values are imputed with $(\hat{\mu}^S)^\ell = U^\ell \Lambda^{1/2^\ell} V^{\ell\prime}$
   the new imputed data is $\hat{X}^\ell = W * X + (\mathbf{1} - W) * (\hat{\mu}^S)^\ell$
3. steps of estimation and imputation are repeated

Iterative PCA

1. initialization $\ell = 0$: X^0 (mean imputation)
2. step ℓ:
   (a) PCA on the completed data $\rightarrow (U^\ell, \Lambda^\ell, V^\ell)$;
   S dimensions kept
   (b) missing values are imputed with $(\hat{\mu}^S)^\ell = U^\ell {\Lambda^{1/2}}^\ell V^{\ell\prime}$
   the new imputed data is $\hat{X}^\ell = W * X + (\mathbf{1} - W) * (\hat{\mu}^S)^\ell$
3. steps of estimation and imputation are repeated

$\Rightarrow$ $\hat{\mu}$ from incomplete data: EM algo $X = \mu + \varepsilon$, $\varepsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \sigma^2\right)$
with μ of low rank , $x_{ij} = \sum_{s=1}^{S} \sqrt{\tilde{\lambda}_s} \tilde{u}_{is} \tilde{v}_{js} + \varepsilon_{ij}$

$\Rightarrow$ Completed data: good imputation (matrix completion, Netflix)

Iterative PCA

1. initialization $\ell = 0$: X^0 (mean imputation)
2. step ℓ:
   (a) PCA on the completed data $\rightarrow (U^\ell, \Lambda^\ell, V^\ell)$;
   S dimensions kept
   (b) missing values are imputed with $(\hat{\mu}^S)^\ell = U^\ell {\Lambda^{1/2}}^\ell V^{\ell\prime}$
   the new imputed data is $\hat{X}^\ell = W * X + (\mathbf{1} - W) * (\hat{\mu}^S)^\ell$
3. steps of estimation and imputation are repeated

$\Rightarrow$ $\hat{\mu}$ from incomplete data: EM algo $X = \mu + \varepsilon$, $\varepsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \sigma^2\right)$
with μ of low rank , $x_{ij} = \sum_{s=1}^{S} \sqrt{\tilde{\lambda}_s} \tilde{u}_{is} \tilde{v}_{js} + \varepsilon_{ij}$

$\Rightarrow$ Completed data: good imputation (matrix completion, Netflix)

Reduction of variability (imputation by $U\Lambda^{1/2}V'$)

Selecting S? Generalized cross-validation (J. & Husson, 2012)

Soft thresholding iterative SVD

$\Rightarrow$ Overfitting issues of iterative PCA: many parameters ($U_{n\times S}$, $V_{S\times p}$)/observed values (S large - many NA); noisy data

$\Rightarrow$ Regularized versions. Init - estimation - imputation steps:

imputation $\hat{\mu}_{ij}^{\text{PCA}} = \sum_{s=1}^{S} \sqrt{\lambda_s} u_{is} v_{js}$ is replaced by

a "shrunk" impute $\hat{\mu}_{ij}^{\text{Soft}} = \sum_{s=1}^{p} \left(\sqrt{\lambda_s} - \lambda\right)_+ u_{is} v_{js}$

$$X = \mu + \varepsilon \qquad \text{argmin}_\mu \left\{ \|W * (X - \mu)\|_2^2 + \lambda \|\mu\|_* \right\}$$

SoftImpute for large matrices. T. Hastie, R. Mazumber, 2015, Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares. *JMLR* Implemented in `softImpute`

Regularized iterative PCA

$\Rightarrow$ Init. - estimation - imputation steps. In `missMDA` (Youtube)

The imputation step:

$$\hat{\mu}_{ij}^{\text{PCA}} = \sum_{s=1}^{S} \sqrt{\lambda_s} u_{is} v_{js}$$

is replaced by a "shrunk" imputation step (Efron & Morris 1972):

$$\hat{\mu}_{ij}^{\text{rPCA}} = \sum_{s=1}^{S} \left(\frac{\lambda_s - \hat{\sigma}^2}{\lambda_s} \right) \sqrt{\lambda_s} u_{is} v_{js} = \sum_{s=1}^{S} \left(\sqrt{\lambda_s} - \frac{\hat{\sigma}^2}{\sqrt{\lambda_s}} \right) u_{is} v_{js}$$

σ^2 small $\rightarrow$ regularized PCA $\approx$ PCA
σ^2 large $\rightarrow$ mean imputation

$$\hat{\sigma}^2 = \frac{RSS}{\text{ddl}} = \frac{n \sum_{s=S+1}^{p} \lambda_s}{np - p - nS - pS + S^2 + S} \qquad (X_{n \times p}; U_{n \times S}; V_{p \times S})$$

Properties

$\Rightarrow$ Results of PCA obtained from an incomplete data set: graph of observations and correlation circle. Missing values are skipped $||W * (X - \mu)||^2$

$\Rightarrow$ Very good quality of imputation. Using similarities between individuals and relationship between variables. Popular in machine learning with recommandation systems (Netflix: 99% missing).

Model makes sense: Data = structure of rank S + noise

(Udell & Townsend Nice Latent Variable Models Have Log-Rank, 2017)

$\Rightarrow$ Different noise regime

- low noise: iterative PCA (tuning S: cross-validation, GCV)
- moderate: iterative regularized PCA (tuning σ, S)
- high noise (SNR low, S large): soft thresholding (tuning λ, σ) Implemented in R packages denoiseR (Josse, Wager, Sardy)

The imputed data set should be analysed with caution with other methods

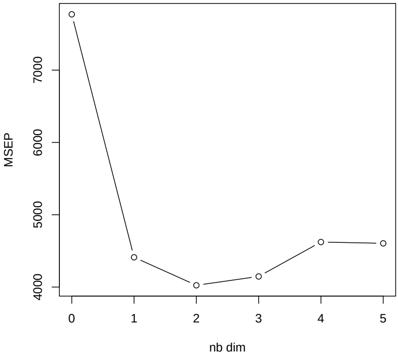
Incomplete ozone

	O3	T9	T12	T15	Ne9	Ne12	Ne15	Vx9	Vx12	Vx15	O3v
0601	87	15.6	18.5	18.4	4	4	8	NA	-1.7101	-0.6946	84
0602	82	NA	18.4	17.7	5	5	7	NA	NA	NA	87
0603	92	NA	17.6	19.5	2	5	4	2.9544	1.8794	0.5209	82
0604	114	16.2	NA	NA	1	1	0	NA	NA	NA	92
0605	94	17.4	20.5	NA	8	8	7	-0.5	NA	-4.3301	114
0606	80	17.7	NA	18.3	NA	NA	NA	-5.6382	-5	-6	94
0607	NA	16.8	15.6	14.9	7	8	8	-4.3301	-1.8794	-3.7588	80
0610	79	14.9	17.5	18.9	5	5	4	0	-1.0419	-1.3892	NA
0611	101	NA	19.6	21.4	2	4	4	-0.766	NA	-2.2981	79
0612	NA	18.3	21.9	22.9	5	6	8	1.2856	-2.2981	-3.9392	101
0613	101	17.3	19.3	20.2	NA	NA	NA	-1.5	-1.5	-0.8682	NA
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
0919	NA	14.8	16.3	15.9	7	7	7	-4.3301	-6.0622	-5.1962	42
0920	71	15.5	18	17.4	7	7	6	-3.9392	-3.0642	0	NA
0921	96	NA	NA	NA	3	3	3	NA	NA	NA	71
0922	98	NA	NA	NA	2	2	2	4	5	4.3301	96
0923	92	14.7	17.6	18.2	1	4	6	5.1962	5.1423	3.5	98
0924	NA	13.3	17.7	17.7	NA	NA	NA	-0.9397	-0.766	-0.5	92
0925	84	13.3	17.7	17.8	3	5	6	0	-1	-1.2856	NA
0927	NA	16.2	20.8	22.1	6	5	5	-0.6946	-2	-1.3681	71
0928	99	16.9	23	22.6	NA	4	7	1.5	0.8682	0.8682	NA
0929	NA	16.9	19.8	22.1	6	5	3	-4	-3.7588	-4	99
0930	70	15.7	18.6	20.7	NA	NA	NA	0	-1.0419	-4	NA

Imputation with PCA in practice

⇒ Step 1: Estimation of the number of dimensions
(Cross Validation, Bro, 2008; GCV, Josse & Husson, 2011)

```
> library(missMDA)
> nb <- estim_ncpPCA(don, method.cv = "Kfold")
> nb$ncp    #2
> plot(0:5, nb$criterion, xlab = "nb dim", ylab ="MSEP")
```



Imputation with PCA in practice

⇒ Step 2: Imputation of the missing values

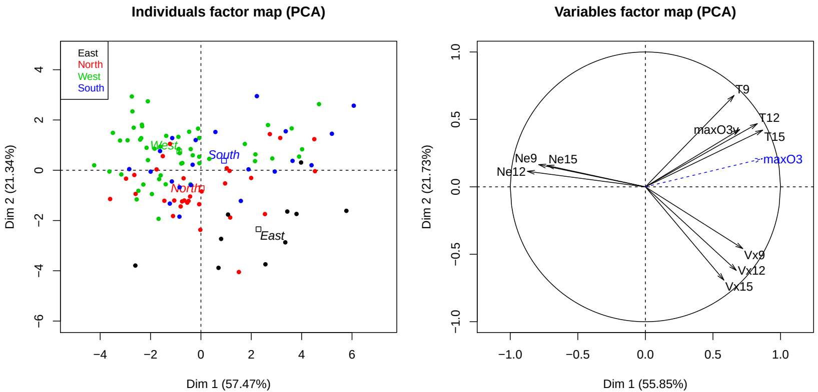
```
> res.comp <- imputePCA(don, ncp = 2)
> res.comp$completeObs[1:3, ]
     maxO3    T9   T12   T15 Ne9 Ne12 Ne15   Vx9  Vx12  Vx15 maxO3v
0601    87 15.60 18.50 20.47   4 4.00 8.00  0.69 -1.71 -0.69     84
0602    82 18.51 20.88 21.81   5 5.00 7.00 -4.33 -4.00 -3.00     87
0603    92 15.30 17.60 19.50   2 3.98 3.81  2.95  1.97  0.52     82
```

Complete ozone

```
         maxO3     T9    T12    T15   Ne9  Ne12  Ne15    Vx9   Vx12   Vx15  maxO3v
20010601  87.000 15.600 18.500 20.471 4.000 4.000 8.000  0.695 -1.710 -0.695  84.000
20010602  82.000 18.505 20.870 21.799 5.000 5.000 7.000 -4.330 -4.000 -3.000  87.000
20010603  92.000 15.300 17.600 19.500 2.000 3.984 3.812  2.954  1.951  0.521  82.000
20010604 114.000 16.200 19.700 24.693 1.000 1.000 0.000  2.044  0.347 -0.174  92.000
20010605  94.000 18.968 20.500 20.400 5.294 5.272 5.056 -0.500 -2.954 -4.330 114.000
20010606  80.000 17.700 19.800 18.300 6.000 7.020 7.000 -5.638 -5.000 -6.000  94.000
20010607  79.000 16.800 15.600 14.900 7.000 8.000 6.556 -4.330 -1.879 -3.759  80.000
20010610  79.000 14.900 17.500 18.900 5.000 5.000 5.016  0.000 -1.042 -1.389  99.000
20010611 101.000 16.100 19.600 21.400 2.000 4.691 4.000 -0.766 -1.026 -2.298  79.000
20010612 106.000 18.300 22.494 22.900 5.000 4.627 4.495  1.286 -2.298 -3.939 101.000
20010613 101.000 17.300 19.300 20.200 7.000 7.000 3.000 -1.500 -1.500 -0.868 106.000
.....

20010915  69.000 17.100 17.700 17.500 6.000 7.000 8.000 -5.196 -2.736 -1.042  71.000
20010916  71.000 15.400 18.091 16.600 4.000 5.000 5.000 -3.830  0.000  1.389  69.000
20010917  60.000 15.283 18.565 19.556 4.000 5.000 4.000  0.000  3.214  0.000  71.000
20010918  42.000 14.091 14.300 14.900 8.000 7.000 7.000 -2.500 -3.214 -2.500  60.000
20010919  65.000 14.800 16.425 15.900 7.000 7.982 7.000 -4.341 -6.062 -5.196  42.000
20010920  71.000 15.500 18.000 17.400 7.000 7.000 6.000 -3.939 -3.064  0.000  65.000
20010924  76.000 13.300 17.700 17.700 5.631 5.883 5.453 -0.940 -0.766 -0.500  65.139
20010925  75.573 13.300 18.434 17.800 3.000 5.000 5.001  0.000 -1.000 -1.286  76.000
20010927  77.000 16.200 20.800 20.499 5.368 5.495 5.177 -0.695 -2.000 -1.473  71.000
20010928  99.000 18.074 22.169 23.651 3.531 3.610 3.561  1.500  0.868  0.868  93.135
20010929  83.000 19.855 22.663 23.847 5.374 5.000 3.000 -4.000 -3.759 -4.000  99.000
20010930  70.000 15.700 18.600 20.700 7.000 6.405 7.000 -2.584 -1.042 -4.000  83.000
```

```
> library(missMDA)
> res.comp <- imputePCA(ozo[, 1:11])
> res.comp$comp
```

Cherry on the cake: PCA on incomplete data!



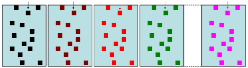

```
> imp <- cbind.data.frame(res.comp$completeObs, ozo[, 12])
> res.pca <- PCA(imp, quanti.sup = 1, quali.sup = 12)
> plot(res.pca, hab = 12, lab = "quali"); plot(res.pca, choix = "var")
> res.pca$ind$coord #scores (principal components)
```

Multiple imputation

⇒ Aim: provide estimation of the parameters and of their variability (taken into account the variability due to missing values)

Single imputation: a single value can't reflect the uncertainty of prediction ⇒ underestimate the standard errors

1. Generating M imputed data sets: variance of prediction



2. Performing the analysis on each imputed data set
3. Combining: variance = within + between imputation variance

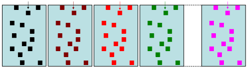
$$\hat{\beta} = \frac{1}{M}\sum_{m=1}^{M}\hat{\beta}_m \quad T = \frac{1}{M}\sum \widehat{Var}\left(\hat{\beta}_m\right) + \left(1+\frac{1}{M}\right)\frac{1}{M-1}\sum\left(\hat{\beta}_m - \hat{\beta}\right)^2$$

Multiple imputation

$\Rightarrow$ Aim: provide estimation of the parameters and of their variability (taken into account the variability due to missing values)

Single imputation: a single value can't reflect the uncertainty of prediction $\Rightarrow$ underestimate the standard errors

1. Generating M imputed data sets: variance of prediction



   1) Variance of estimation of the parameters + 2) Noise

2. Performing the analysis on each imputed data set
3. Combining: variance = within + between imputation variance

$$\hat{\beta} = \frac{1}{M}\sum_{m=1}^{M}\hat{\beta}_m \quad T = \frac{1}{M}\sum \widehat{Var}\left(\hat{\beta}_m\right) + \left(1+\frac{1}{M}\right)\frac{1}{M-1}\sum\left(\hat{\beta}_m - \hat{\beta}\right)^2$$

Joint modeling

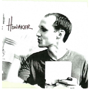
$\Rightarrow$ Hypothesis $x_{i.} \sim \mathcal{N}(\mu, \Sigma)$

Algorithm Expectation Maximization Bootstrap:

1. Bootstrap rows: X^1, ... , X^M
   EM algorithm: $(\hat{\mu}^1, \hat{\Sigma}^1)$, ... , $(\hat{\mu}^M, \hat{\Sigma}^M)$
2. Imputation: x_{ij}^m drawn from $\mathcal{N}\left(\hat{\mu}^m, \hat{\Sigma}^m\right)$

Easy to parallelized. Implemented in `Amelia` (website)

Amelia Earhart



James Honaker

Gary King

Matt Blackwell

Fully conditional modeling

⇒ Hypothesis: one model/variable

1. Initial imputation: mean imputation
2. For a variable j

   2.2 Imputation of the missing values in variable j with a model of X_j on the other X_{-j}: stochastic regression x_{ij} from $\mathcal{N}\left((x_{i,-j})'\hat{\beta}^{-j}, \hat{\sigma}^{-j}\right)$
3. Cycling through variables

⇒ Iteratively refine the imputation.

⇒ With continuous variables and a regression/variable: $\mathcal{N}(\mu, \Sigma)$

Implemented in `mice` (website) and Python

"There is no clear-cut method for determining whether the MICE algorithm has converged"

Stef van Buuren

Fully conditional modeling

⇒ Hypothesis: one model/variable

1. Initial imputation: mean imputation
2. For a variable j
   - 2.1 $(\hat{\beta}^{-j}, \hat{\sigma}^{-j})$ drawn from a Bootstrap: $(\hat{\beta}^{-j}, \hat{\sigma}^{-j})^1, ..., (\hat{\beta}^{-j}, \hat{\sigma}^{-j})^M$
   - 2.2 Imputation of the missing values in variable j with a model of X_j on the other X_{-j}: stochastic regression x_{ij} from $\mathcal{N}\left((x_{i,-j})'\hat{\beta}^{-j}, \hat{\sigma}^{-j}\right)$
3. Cycling through variables

Get M imputed data

⇒ Iteratively refine the imputation.

⇒ With continuous variables and a regression/variable: $\mathcal{N}(\mu, \Sigma)$

Implemented in `mice` (website) and Python

Stef van Buuren

"*There is no clear-cut method for determining whether the MICE algorithm has converged*"

Joint / Conditional modeling

⇒ Both seen imputed values are drawn from a Joint distribution (even if joint does not exist)
⇒ Conditional modeling takes the lead?

- Flexible: one model/variable. Easy to deal with interactions and variables of different nature (binary, ordinal, categorical...)
- Many statistical models are conditional models!
- Tailor to your data
- Appears to work quite well in practice

⇒ Drawbacks: one model/variable... tedious...

Joint / Conditional modeling

$\Rightarrow$ Both seen imputed values are drawn from a Joint distribution (even if joint does not exist)
$\Rightarrow$ Conditional modeling takes the lead?

- Flexible: one model/variable. Easy to deal with interactions and variables of different nature (binary, ordinal, categorical...)
- Many statistical models are conditional models!
- Tailor to your data
- Appears to work quite well in practice

$\Rightarrow$ Drawbacks: one model/variable... tedious...

$\Rightarrow$ What to do with high correlation or when $n < p$?

- JM shrinks the covariance $\Sigma + k\mathbb{I}$ (selection of k?)
- CM: ridge regression or predictors selection/variable $\Rightarrow$ a lot of tuning ... not so easy ...

Multiple imputation with Bootstrap PCA

$$x_{ij} = \mu_{ij} + \varepsilon_{ij} = \sum_{s=1}^{S} \sqrt{\tilde{\lambda}_s} \tilde{u}_{is} \tilde{v}_{js} + \varepsilon_{ij} \ , \ \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

1. Variability of the parameters, M plausible: $(\hat{\mu}_{ij})^1, \ldots, (\hat{\mu}_{ij})^M$

2. Noise: for $m = 1, \ldots, M$, missing values x_{ij}^m drawn $\mathcal{N}(\hat{\mu}_{ij}^m, \hat{\sigma}^2)$

Implemented in missMDA (website)

François Husson

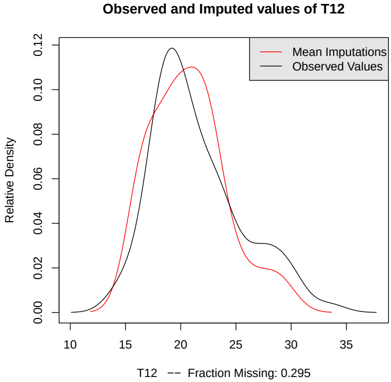
Multiple imputation in practice

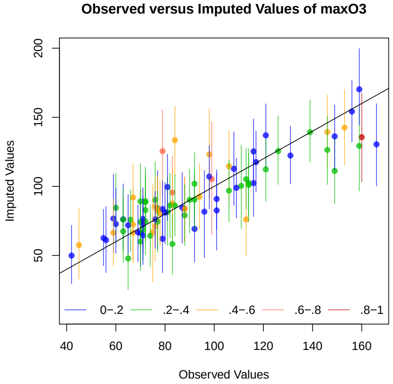
$\Rightarrow$ Step 1: Generate M imputed data sets

```
> library(Amelia)
> res.amelia <- amelia(don, m = 100)

> library(mice)
> res.mice <- mice(don, m = 100, defaultMethod = "norm.boot")

> library(missMDA)
>  res.MIPCA <- MIPCA(don, ncp = 2, nboot  = 100)
> res.MIPCA$res.MI
```

Multiple imputation in practice

⇒ Step 2: visualization



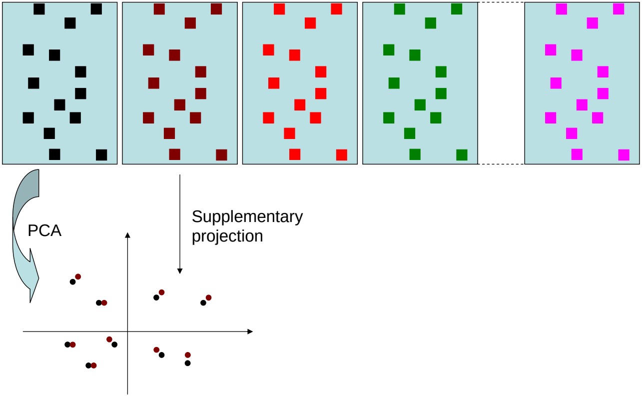



```
# library(Amelia)
> res.amelia <- amelia(don, m = 100)
> compare.density(res.amelia, var = "T12")
> overimpute(res.amelia, var = "maxO3")

# library(missMDA)
res.over<-Overimpute(res.MIPCA)
```

Multiple imputation in practice

⇒ Step 2: visualization

⇒ Individuals position (and variables) with other predictions



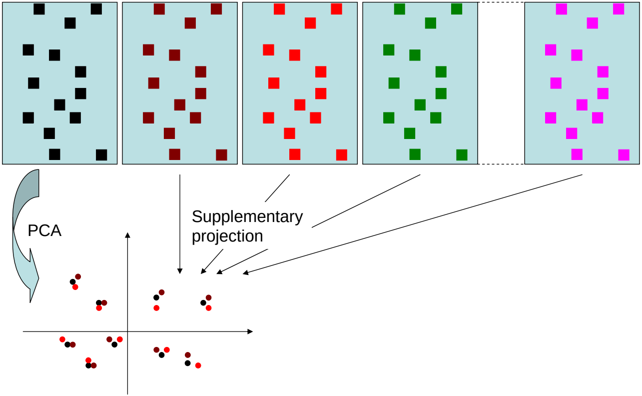
Regularized iterative PCA

⇒ reference configuration

Multiple imputation in practice

⇒ Step 2: visualization

⇒ Individuals position (and variables) with other predictions



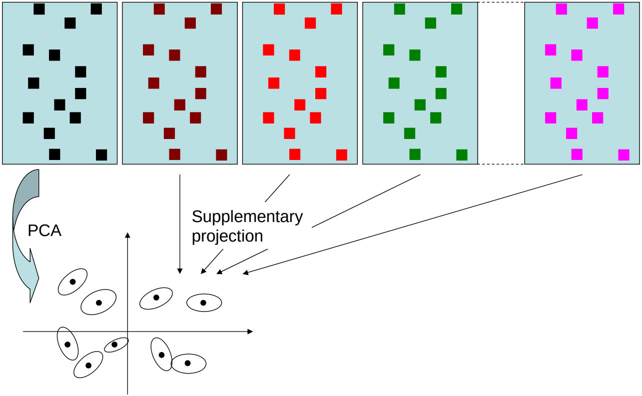
Regularized iterative PCA

⇒ reference configuration

Multiple imputation in practice

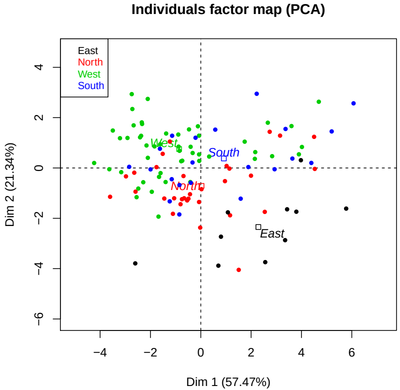
⇒ Step 2: visualization

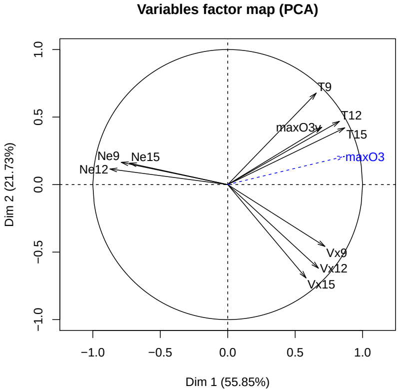
⇒ Individuals position (and variables) with other predictions



Regularized iterative PCA

⇒ reference configuration



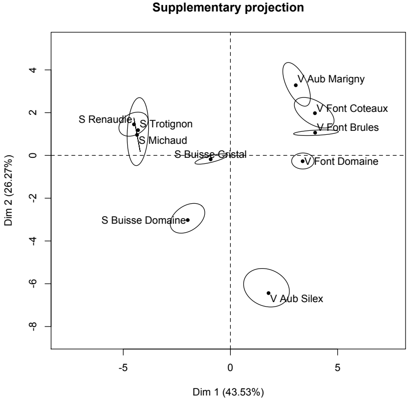


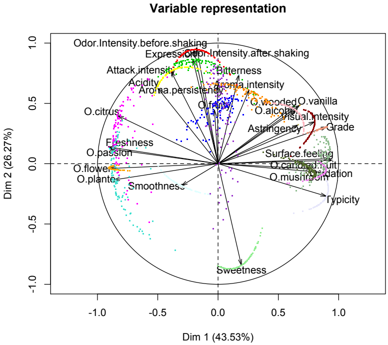

```
> imp <- cbind.data.frame(res.comp$completeObs, ozo[, 12])
> res.pca <- PCA(imp,quanti.sup = 1, quali.sup = 12)
> plot(res.pca, hab =12, lab = "quali"); plot(res.pca, choix = "var")
> res.pca$ind$coord #scores (principal components)
```

Multiple imputation in practice

⇒ Step 2: visualization

```
> res.MIPCA <- MIPCA(don, ncp = 2)
> plot(res.MIPCA, choice = "ind.supp"); plot(res.MIPCA, choice = "var")
```





⇒ Percentage of NA?

Multiple imputation in practice

$\Rightarrow$ Step 3. Regression on each table and pool the results

$$\hat{\beta} = \frac{1}{M}\sum_{m=1}^{M}\hat{\beta}_m$$
$$T = \frac{1}{M}\sum_m \widehat{Var}\left(\hat{\beta}_m\right) + \left(1+\frac{1}{M}\right)\frac{1}{M-1}\sum_m\left(\hat{\beta}_m - \hat{\beta}\right)^2$$

```
> library(mice)
> res.mice <- mice(don, m = 100)
> imp.micerf <- mice(don, m = 100, defaultMethod = "rf")
> lm.mice.out <- with(res.mice, lm(maxO3 ~ T9+T12+T15+Ne9+...+Vx15+maxO3v))
> pool.mice <- pool(lm.mice.out)
> summary(pool.mice)
```

```
                est    se     t    df Pr(>|t|)  lo 95 hi 95 nmis  fmi lambda
(Intercept) 19.31 16.30  1.18 50.48     0.24 -13.43 52.05   NA 0.46   0.44
T9          -0.88  2.25 -0.39 26.43     0.70  -5.50  3.75   37 0.71   0.69
T12          3.29  2.38  1.38 27.54     0.18  -1.59  8.18   33 0.70   0.68
....
Vx15         0.23  1.33  0.17 39.00     0.87  -2.47  2.93   21 0.57   0.55
maxO3v       0.36  0.10  3.65 46.03     0.00   0.16  0.56   12 0.50   0.48
```

Categorical data

Survey data

```
region                    sex       age          year          edu         drunk        alcohol       glasses
Ile de France      :8120  F:29776   18_25: 6920  2005:27907    E1:12684   0     :44237  <1/m :12889   0  : 2812
Rhone Alpes        :5421  M:23165   26_34: 9401  2010:25034    E2:23521   1-2   : 4952  0    : 6133   0-2:37867
Provence Alpes     :4116            35_44:10899                E3:6563    10-19:  839   1-2/m: 7583   10+:  590
Nord Pas de Calais :3819            45_54: 9505                E4:10100   20-29:  212   1-2/w: 9526   3-4: 9401
Pays de Loire      :3152            55_64: 9503                NA:73      3-5   : 1908  3-4/w: 6815   5-6: 1795
Bretagne           :3038            65_+ : 6713                           30+   :  404  5-6/w: 3402   7-9:  391
(Other)            :25275                                                 6-9   :  389  7/w  : 6593          NA: 85

binge                Pbsleep                Tabac
<2/m:10323           Never:20605            Frequent  : 9176
0   :34345           Often: 10172           Never     :39080
1/m : 6018           Rare :22134            Occasional: 4588
1/w : 1800           NA:  30                NA:  97
7/w :  374
NA  :   81
```

INPES `http://www.inpes.sante.fr`

Principal components method: Multiple Correpondence Analysis Single imputation based on MCA for categorical data

Multiple Correspondence Analysis (MCA)

$X_{n\times m}$ m categorical variables coded with indicator matrix A

$$X = \begin{array}{|c|c|c|} y & \dots & attack \\ y & \dots & attack \\ y & \dots & attack \\ n & \dots & suicide \\ & & \\ & & \\ n & \dots & accident \\ n & \dots & suicide \end{array} \quad A = \begin{array}{|cc|c|ccc|} 1 & 0 & \dots & 1 & 0 & 0 \\ 1 & 0 & \dots & 1 & 0 & 0 \\ 1 & 0 & \dots & 1 & 0 & 0 \\ 0 & 1 & \dots & 0 & 1 & 0 \\ & & & & & \\ & & & & & \\ 0 & 1 & \dots & 0 & 0 & 1 \\ 0 & 1 & \dots & 0 & 1 & 0 \end{array} \quad D_p = \begin{bmatrix} p_1 & & 0 \\ & \ddots & \\ 0 & & p_J \end{bmatrix}$$

For a category c, the frequency of the category: $p_c = n_c/n$.

A SVD on weighted matrix: $Z = \frac{1}{\sqrt{mn}}(A - 1p^T)D_p^{-1/2} = U\Lambda V'$

The PC ($F = U\Lambda^{1/2}$) satisfies: $\arg\max_{F_s \in \mathbb{R}^n} \quad \frac{1}{m}\sum_{j=1}^m \eta^2(F_s, X_j)$

$$\eta^2(F, X_j) = \frac{\sum_{c=1}^{C_j} n_c(F_{.c} - F_{..})^2}{\sum_{i=1}^n \sum_{c=1}^{C_j}(F_{ic})^2} = \frac{\text{RSS between}}{\text{RSS tot}}$$

Benzecri, 1973 :*"In data analysis the mathematical problems reduces to computing eigenvectors; all the science (the art) is in finding the right matrix to diagonalize"*

Regularized iterative MCA *(Chavent et al., 2012)*

Iterative MCA algorithm:

	V1	V2	V3	…	V14
ind 1	a	NA	g	…	u
ind 2	NA	f	g		u
ind 3	a	e	h		v
ind 4	a	e	h		v
ind 5	b	f	h		u
ind 6	c	f	h		u
ind 7	c	f	NA		v
…	…	…	…		…
ind 1232	c	f	h		v

	V1_a	V1_b	V1_c	V2_e	V2_f	V3_g	V3_h	…
ind 1	1	0	0	NA	NA	1	0	…
ind 2	NA	NA	NA	0	1	1	0	…
ind 3	1	0	0	1	0	0	1	…
ind 4	1	0	0	1	0	0	1	…
ind 5	0	1	0	0	1	0	1	…
ind 6	0	0	1	0	1	0	1	…
ind 7	0	0	1	0	1	NA	NA	…
…	…	…	…	…	…	…	…	…
ind 1232	0	0	1	0	1	0	1	…

```
library(missMDA); ?imputeMCA
```

Regularized iterative MCA *(Chavent et al., 2012)*

Iterative MCA algorithm:

1. initialization: imputation of the indicator matrix (proportion)

	V1	V2	V3	...	V14
ind 1	a	NA	g	...	u
ind 2	NA	f	g		u
ind 3	a	e	h		v
ind 4	a	e	h		v
ind 5	b	f	h		u
ind 6	c	f	h		u
ind 7	c	f	NA		v
...	...	...	...		...
ind 1232	c	f	h		v

	V1_a	V1_b	V1_c	V2_e	V2_f	V3_g	V3_h	...
ind 1	1	0	0	0.41	0.59	1	0	...
ind 2	0.20	0.30	0.50	0	1	1	0	...
ind 3	1	0	0	1	0	0	1	...
ind 4	1	0	0	1	0	0	1	...
ind 5	0	1	0	0	1	0	1	...
ind 6	0	0	1	0	1	0	1	...
ind 7	0	0	1	0	1	0.27	0.78	...
...	...	...	...	...	...	...	...	...
ind 1232	0	0	1	0	1	0	1	...

```
library(missMDA); ?imputeMCA
```

Regularized iterative MCA *(Chavent et al., 2012)*

Iterative MCA algorithm:

1. initialization: imputation of the indicator matrix (proportion)
2. iterate until convergence
   (a) estimation: MCA on the completed data $\rightarrow U, \Lambda, V$

	V1	V2	V3	...	V14
ind 1	a	NA	g	...	u
ind 2	NA	f	g		u
ind 3	a	e	h		v
ind 4	a	e	h		v
ind 5	b	f	h		u
ind 6	c	f	h		u
ind 7	c	f	NA		v
...	...	...	...		...
ind 1232	c	f	h		v

	V1_a	V1_b	V1_c	V2_e	V2_f	V3_g	V3_h	...
ind 1	1	0	0	0.41	0.59	1	0	...
ind 2	0.20	0.30	0.50	0	1	1	0	...
ind 3	1	0	0	1	0	0	1	...
ind 4	1	0	0	1	0	0	1	...
ind 5	0	1	0	0	1	0	1	...
ind 6	0	0	1	0	1	0	1	...
ind 7	0	0	1	0	1	0.27	0.78	...
...	...	...	...	...	...	...	...	...
ind 1232	0	0	1	0	1	0	1	...

```
library(missMDA); ?imputeMCA
```

Regularized iterative MCA *(Chavent et al., 2012)*

Iterative MCA algorithm:

1. initialization: imputation of the indicator matrix (proportion)
2. iterate until convergence
   (a) estimation: MCA on the completed data $\rightarrow U, \Lambda, V$
   (b) imputation with the fitted matrix $\hat{\mu} = U_S \Lambda_S^{1/2} V_S'$

	V1	V2	V3	...	V14
ind 1	a	NA	g	...	u
ind 2	NA	f	g		u
ind 3	a	e	h		v
ind 4	a	e	h		v
ind 5	b	f	h		u
ind 6	c	f	h		u
ind 7	c	f	NA		v
...	...	...	...		...
ind 1232	c	f	h		v

	V1_a	V1_b	V1_c	V2_e	V2_f	V3_g	V3_h	...
ind 1	1	0	0	0.65	0.35	1	0	...
ind 2	0.11	0.20	0.69	0	1	1	0	...
ind 3	1	0	0	1	0	0	1	...
ind 4	1	0	0	1	0	0	1	...
ind 5	0	1	0	0	1	0	1	...
ind 6	0	0	1	0	1	0	1	...
ind 7	0	0	1	0	1	0.30	0.40	...
...	...	...	...	...	...	...	...	...
ind 1232	0	0	1	0	1	0	1	...

```
library(missMDA); ?imputeMCA
```

Regularized iterative MCA *(Chavent et al., 2012)*

Iterative MCA algorithm:

1. initialization: imputation of the indicator matrix (proportion)
2. iterate until convergence
   (a) estimation: MCA on the completed data $\rightarrow U, \Lambda, V$
   (b) imputation with the fitted matrix $\hat{\mu} = U_S \Lambda_S^{1/2} V_S'$
   (c) column margins are updated

	V1	V2	V3	...	V14
ind 1	a	NA	g	...	u
ind 2	NA	f	g		u
ind 3	a	e	h		v
ind 4	a	e	h		v
ind 5	b	f	h		u
ind 6	c	f	h		u
ind 7	c	f	NA		v
...	...	...	...		...
ind 1232	c	f	h		v

	V1_a	V1_b	V1_c	V2_e	V2_f	V3_g	V3_h	...
ind 1	1	0	0	0.65	0.35	1	0	...
ind 2	0.11	0.20	0.69	0	1	1	0	...
ind 3	1	0	0	1	0	0	1	...
ind 4	1	0	0	1	0	0	1	...
ind 5	0	1	0	0	1	0	1	...
ind 6	0	0	1	0	1	0	1	...
ind 7	0	0	1	0	1	0.30	0.40	...
...	...	...	...	...	...	...	...	...
ind 1232	0	0	1	0	1	0	1	...

```
library(missMDA); ?imputeMCA
```

Regularized iterative MCA *(Chavent et al., 2012)*

Iterative MCA algorithm:

1. initialization: imputation of the indicator matrix (proportion)
2. iterate until convergence
   (a) estimation: MCA on the completed data $\rightarrow U, \Lambda, V$
   (b) imputation with the fitted matrix $\hat{\mu} = U_S \Lambda_S^{1/2} V_S'$
   (c) column margins are updated

	V1	V2	V3	...	V14
ind 1	a	**NA**	g	...	u
ind 2	**NA**	f	g		u
ind 3	a	e	h		v
ind 4	a	e	h		v
ind 5	b	f	h		u
ind 6	c	f	h		u
ind 7	c	f	**NA**		v
...	...	...	...		...
ind 1232	c	f	h		v

	V1_a	V1_b	V1_c	V2_e	V2_f	V3_g	V3_h	...
ind 1	1	0	0	**0.71**	**0.29**	1	0	...
ind 2	**0.12**	**0.29**	**0.59**	0	1	1	0	...
ind 3	1	0	0	1	0	0	1	...
ind 4	1	0	0	1	0	0	1	...
ind 5	0	1	0	0	1	0	1	...
ind 6	0	0	1	0	1	0	1	...
ind 7	0	0	1	0	1	**0.37**	**0.63**	...
...	...	...	...	...	...	...	...	...
ind 1232	0	0	1	0	1	0	1	...

$\Rightarrow$ the imputed values can be seen as degree of membership

```
library(missMDA); ?imputeMCA
```

Regularized iterative MCA *(Chavent et al., 2012)*

Iterative MCA algorithm:

1. initialization: imputation of the indicator matrix (proportion)
2. iterate until convergence
   - (a) estimation: MCA on the completed data $\rightarrow U, \Lambda, V$
   - (b) imputation with the fitted matrix $\hat{\mu} = U_S \Lambda_S^{1/2} V_S'$
   - (c) column margins are updated

	V1	V2	V3	...	V14
ind 1	a	**e**	g	...	u
ind 2	**c**	f	g		u
ind 3	a	e	h		v
ind 4	a	e	h		v
ind 5	b	f	h		u
ind 6	c	f	h		u
ind 7	c	f	**g**		v
...	...	...	...		...
ind 1232	c	f	h		v

	V1_a	V1_b	V1_c	V2_e	V2_f	V3_g	V3_h	...
ind 1	1	0	0	**0.71**	**0.29**	1	0	...
ind 2	**0.12**	**0.29**	**0.59**	0	1	1	0	...
ind 3	1	0	0	1	0	0	1	...
ind 4	1	0	0	1	0	0	1	...
ind 5	0	1	0	0	1	0	1	...
ind 6	0	0	1	0	1	0	1	...
ind 7	0	0	1	0	1	**0.37**	**0.63**	...
...	...	...	...	...	...	...	...	...
ind 1232	0	0	1	0	1	0	1	...

Two ways to obtain categories: majority or draw

```
library(missMDA); ?imputeMCA
```

Multiple imputation with MCA

1. Variability of the parameters: M sets $(U_{n\times S}, \Lambda_{S\times S}, V^{\top}_{m\times S})$ using a non-parametric bootstrap

$\hat{X}_1$

1	0	...	1	0	0
1	0	...	1	0	0
1	0	...			
			0.01	0.80	0.19
			0	0	1
0.25	0.75				
0	1		0	0	1

$\hat{X}_2$

1	0	...	1	0	0
1	0	...	1	0	0
1	0	...			
			0.60	0.2	0.20
			0	0	1
0.26	0.74				
0	1		0	0	1

...

$\hat{X}_M$

1	0	...	1	0
1	0	...	1	0
1	0	...		
			0.11	0.74
			0	0
0.20	0.80			
0	1		0	0

2. Categories drawn from multinomial disribution using the values in $\left(\hat{X}_m\right)_{1\leq m\leq M}$

y	...	Attack
y	...	Attack
y	...	
		Suicide
	...	Accident
n		
n	...	S

y	...	Attack
y	...	Attack
y	...	
		Attack
	...	Accident
n		
n	...	B

...

y	...	Attack
y	...	Attack
y	...	
		Suicide
	...	Accident
n		
n	...	Suicide

```
library(missMDA); MIMCA()
```

To conclude

Take home message:

- *"**The idea of imputation is both seductive and dangerous**. It is seductive because it can lull the user into the pleasurable state of believing that the data are complete after all, and it is dangerous because it lumps together situations where the problem is sufficiently minor that it can be legitimately handled in this way and situations where standard estimators applied to the real and imputed data have substantial biases."* (Dempster and Rubin, 1983)

- Single imputation aims to complete a dataset as best as possible (prediction)

- Multiple imputation aims to perform other statistical methods after and to estimate parameters and their variability taking into account the missing values uncertainty

- Single imputation can be appropriate for point estimates

To conclude

Take home message:

- Principal component methods powerful for single & multiple imputation of quanti & categorical data (rare categories): dimensionality reduction and capture similarities between obs and variables. (be careful some implementations do not handle well categorical data)

  ⇒ Correct inferences for analysis model based on relationships between pairs of variables

  ⇒ SVD can be distributed! Master - Slave, privacy preserving

  ⇒ Requires to choose the number of dimensions S

- Handling missing values in PCA, MCA, FAMD, Multiple Factor Analysis (MFA), Correspondence analysis for contingency tables
- Preprocessing before clustering
- Package R `missMDA` (youtube, website, blog)

Challenges

$\Rightarrow$ MI theory:

- Imputation model as complex as the analysis one (interaction)
- Good theory for regression parameters: others?
- MI theory with new asymptotic small n, large p ?
  $\Rightarrow$ Still an active area of research
  $\Rightarrow$ Imputation/Multiple imputation for prediction.
  $\Rightarrow$ Variable selection

$\Rightarrow$ Some practical issues:

- Imputation not in agreement (X and X^2): missing passive, Imputation out of range?, Problems of logical bounds (> 0)
- Multiple imputation is appealing .... but ... with large data?

Ressources implementation

Package `missMDA`:

http://factominer.free.fr/missMDA/index.html

Youtube: https://www.youtube.com/watch?v=OOM8_FH6_8o&list=PLnZgp6epRBbQzxFnQrcxg09kRt-PA66T_playlist

Article JSS: https://www.jstatsoft.org/article/view/v070i01

Ressources

R-miss-tastic https://rmisstastic.netlify.com/R-miss-tastic

J., I. Mayer, N. Tierney & N. Vialaneix

Project funded by the R consortium (Infrastructure Steering Committee)[1]

Aim: a reference platform on the theme of missing data management

- list existing packages
- available literature
- tutorials
- analysis workflows on data
- main actors

⇒ Federate the community

⇒ Contribute!

[1] https://www.r-consortium.org/projects/call-for-proposals

Ressources

Examples:

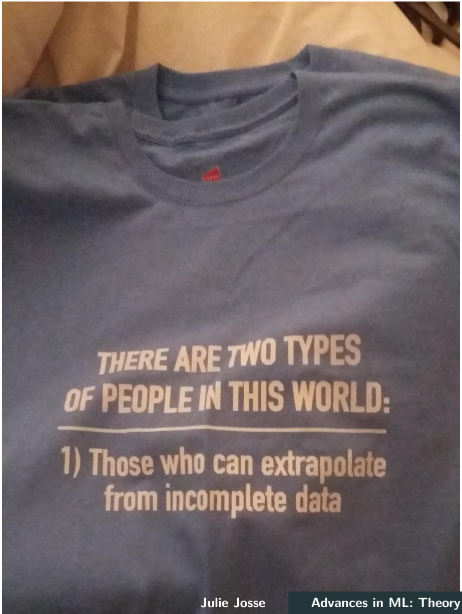
- Lecture [2] - General tutorial : Statistical Methods for Analysis with Missing Data (Mauricio Sadinle)
- Lecture - Multiple Imputation: `mice` by Nicole Erler [3]
- Longitudinal data, Time Series Imputation (Steffen Moritz - very active contributor of r-miss-tastic), Principal Component Methods[4]

[2] https://rmisstastic.netlify.com/lectures/

[3] https://rmisstastic.netlify.com/tutorials/erler_course_multipleimputation_2018/erler_practical_mice_2018

[4] https://rmisstastic.netlify.com/tutorials/Josse_slides_imputation_PCA_2018.pdf

Thank you


THERE ARE TWO TYPES
OF PEOPLE IN THIS WORLD:
1) Those who can extrapolate
from incomplete data