# Supervised learning with missing values

Julie Josse Ecole Polytechnique, INRIA

8 September 2020

Workshop on Missing Data Challenges in Computation, Statistics and Applications, Princeton



# Introduction

# Collaborators on supervised learning with missing values

- M. Le Morvan, Postdoc at INRIA, Paris.
- E. Scornet, Associate Professor at Ecole Polytechnique, IP Paris. Topic: random forests.
- G. Varoquaux, Senior researcher at INRIA, Paris.

Topic: machine learning. Creator of Scikitlearn in python.



- $\Rightarrow$  Random Forests with missing values
- 1. Consistency of supervised learning with missing values. (2019). Revis JMLR.
- $\Rightarrow$  Linear regression with missing values MultiLayer perceptron
- 2. Linear predictor on linearly-generated data with missing values: non consistency and solutions. AISTAT2020.

3. Neumann networks: differential programming for supervised learning with missing values. Submitted Neurips2020.

# Traumabase project: decision support for trauma patients.

- 20000 trauma patients
- 250 continuous and categorical variables: heterogeneous
- 11 hospitals: multilevel data
- 4000 new patients/ year

Center	Accident	Age	Sex	Lactactes	BP	Shock	Platelet	
Beaujon	fall	54	m	NM	180	yes	292000	
Pitie	gun	26	m	NA	131	no	323000	
Beaujon	moto	63	m	3.9	NR	yes	318000	
Pitie	moto	30	W	Imp	107	no	211000	
HEGP	knife	16	m	2.5	118	no	184000	

# Traumabase project: decision support for trauma patients.

- 20000 trauma patients
- 250 continuous and categorical variables: heterogeneous
- 11 hospitals: multilevel data
- 4000 new patients/ year

Center	Accident	Age	Sex	Lactactes	BP	Shock	Platelet	
Beaujon	fall	54	m	NM	180	yes	292000	
Pitie	gun	26	m	NA	131	no	323000	
Beaujon	moto	63	m	3.9	NR	yes	318000	
Pitie	moto	30	W	Imp	107	no	211000	
HEGP	knife	16	m	2.5	118	no	184000	
:								۰.

#### $\Rightarrow$ Estimate causal effect: Administration of the treatment

"tranexamic acid" (within 3 hours after the accident) on the **outcome** mortality for traumatic brain injury patients.  $^1$ 

 $<sup>^{1}\</sup>mbox{Doubly}$  robust treatment effect estimation with incomplete confounders. Mayer, Wager, J. Annals Of Applied Statistics 2020.

# Traumabase project: decision support for trauma patients.

- 20000 trauma patients
- 250 continuous and categorical variables: heterogeneous
- 11 hospitals: multilevel data
- 4000 new patients/ year

Center	Accident	Age	Sex	Lactactes	ΒP	Shock	Platelet	
Beaujon	fall	54	m	NM	180	yes	292000	
Pitie	gun	26	m	NA	131	no	323000	
Beaujon	moto	63	m	3.9	NR	yes	318000	
Pitie	moto	30	W	Imp	107	no	211000	
HEGP	knife	16	m	2.5	118	no	184000	
:								· .

 $\Rightarrow$  **Predict** platelet levels given pre-hospital features

Ex linear regression/ random forests with covariates with missing values

### **Missing values**



**Different pattern**: sporadic & systematic (missing variable in one hospital) **Different types**: MCAR, MAR, MNAR

# Random Forests with missing values

# Missing values in a predictive framework (not inferential)

- <u>Aim</u>: target an outcome Y (not estimate parameters and their variance)
- Specificities: train & test sets with missing values
- <u>Methods 1</u>: (in practice) imputation prior to prediction
  - Separate: impute train and test separately (with a different model)
  - Grouped/ semi-supervised: impute train and test simultaneously but the predictive model is learned only on the training imputed data set.
  - Imputation train and test sets with the same model Issue: methods (missForest) are "black-boxes" *i.e.* take as an input the incomplete data and output the completed data

Easy for univariate imputation: mean of each colum of the train.

<sup>&</sup>lt;sup>1</sup>Rmistatic platform to organize ressources - Task view: more than 150 packages

#### Mean imputation is bad for estimation



PCA with mean imputation

library(FactoMineR)
PCA(ecolo)
Warning message: Missing
are imputed by the mean
of the variable:
You should use imputePCA
from missMDA

#### EM-PCA

library(missMDA)
imp <- imputePCA(ecolo)
PCA(imp\$comp)</pre>

J. (2016). miss-MDA: Handling Missing Values in Multivariate Data Analysis, JSS.

Ecological data: <sup>2</sup> n = 69000 species - 6 traits. Estimated correlation between Pmass & Rmass  $\approx 0$  (mean imputation) or  $\approx 1$  (EM PCA)

<sup>2</sup>Wright, I. et al. (2004). The worldwide leaf economics spectrum. *Nature*.

# Constant (mean) imputation is consistent for prediction

$$ilde{X} = X \odot (1-M) + ext{NA} \odot M$$
. New feature space is  $\widetilde{\mathbb{R}}^d = (\mathbb{R} \cup \{ ext{NA}\})^d$ 

$$Y = \begin{pmatrix} 4.6\\ 7.9\\ 8.3\\ 4.6 \end{pmatrix} \quad \tilde{X} = \begin{pmatrix} 9.1 & \text{NA} & 1\\ 2.1 & \text{NA} & 3\\ \text{NA} & 9.6 & 2\\ \text{NA} & 5.5 & 6 \end{pmatrix} \quad X = \begin{pmatrix} 9.1 & 8.5 & 1\\ 2.1 & 3.5 & 3\\ 6.7 & 9.6 & 2\\ 4.2 & 5.5 & 6 \end{pmatrix} \quad M = \begin{pmatrix} 0 & 1 & 0\\ 0 & 1 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0 \end{pmatrix}$$

#### Find a prediction function that minimizes the risk.

Bayes rule: 
$$f^* \in \underset{f: \widetilde{\mathbb{R}}^d \to \mathbb{R}}{\arg \min} \mathbb{E}\left[\left(Y - f(\tilde{X})\right)^2\right]$$

$$f^{*}(\tilde{X}) = \mathbb{E}\left[Y \mid \tilde{X}\right] = \mathbb{E}\left[Y \mid X_{obs(M),M}\right]$$
$$= \sum_{m \in \{0,1\}^{d}} \mathbb{E}\left[Y \mid X_{obs(m)}, M = m\right] \mathbb{1}_{M = m}$$

 $\Rightarrow$  One model per pattern (2<sup>*d*</sup>) (Rubin, 1984, generalized propensity score)

# Constant (mean) imputation is consistent for prediction

$$ilde{X} = X \odot (1 - M) + \mathtt{NA} \odot M$$
. New feature space is  $\mathbb{\widehat{R}}^d = (\mathbb{R} \cup \{\mathtt{NA}\})^d$ 

$$Y = \begin{pmatrix} 4.6\\ 7.9\\ 8.3\\ 4.6 \end{pmatrix} \quad \tilde{X} = \begin{pmatrix} 9.1 & \text{NA} & 1\\ 2.1 & \text{NA} & 3\\ \text{NA} & 9.6 & 2\\ \text{NA} & 5.5 & 6 \end{pmatrix} \quad X = \begin{pmatrix} 9.1 & 8.5 & 1\\ 2.1 & 3.5 & 3\\ 6.7 & 9.6 & 2\\ 4.2 & 5.5 & 6 \end{pmatrix} \quad M = \begin{pmatrix} 0 & 1 & 0\\ 0 & 1 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0 \end{pmatrix}$$

#### Find a prediction function that minimizes the risk.

Bayes rule: 
$$f^* \in \underset{f: \widetilde{\mathbb{R}}^d \to \mathbb{R}}{\arg \min} \mathbb{E}\left[\left(Y - f(\tilde{X})\right)^2\right]$$

$$f^{*}(\tilde{X}) = \mathbb{E}\left[Y \mid \tilde{X}\right] = \mathbb{E}\left[Y \mid X_{obs(M),M}\right]$$
$$= \sum_{m \in \{0,1\}^{d}} \mathbb{E}\left[Y \mid X_{obs(m)}, M = m\right] \mathbb{1}_{M = m}$$

 $\Rightarrow$  One model per pattern (2<sup>*d*</sup>) (Rubin, 1984, generalized propensity score)

#### Framework - assumptions

- $Y = f(X) + \varepsilon$
- $X = (X_1, \dots, X_d)$  has a continuous density g > 0 on  $[0, 1]^d$
- $\|f\|_{\infty} < \infty$
- Missing data MAR on  $X_1$  with  $M_1 \perp X_1 | X_2, \ldots, X_d$ .
- $(x_2, \ldots, x_d) \mapsto \mathbb{P}[M_1 = 1 | X_2 = x_2, \ldots, X_d = x_d]$  is continuous
- $\varepsilon$  is a centered noise independent of  $(X, M_1)$

(remains valid when missing values occur for several variables  $X_1, \ldots, X_j$ )

# Constant (mean) imputation is consistent

Constant imputed entry  $x' = (x'_1, x_2, ..., x_d)$ :  $x'_1 = x_1 \mathbb{1}_{M_1=0} + \alpha \mathbb{1}_{M_1=1}$  **Theorem. (J. et al. 2019)**   $f^*_{impute}(x') = \mathbb{E}[Y|X_2 = x_2, ..., X_d = x_d, M_1 = 1]$   $\mathbb{1}_{x'_1=\alpha} \mathbb{1}_{\mathbb{P}[M_1=1|X_2=x_2,...,X_d=x_d]>0}$   $+ \mathbb{E}[Y|X = x'] \mathbb{1}_{x'_1=\alpha} \mathbb{1}_{\mathbb{P}[M_1=1|X_2=x_2,...,X_d=x_d]=0}$  $+ \mathbb{E}[Y|X_1 = x_1, X_2 = x_2, ..., X_d = x_d, M_1 = 0] \mathbb{1}_{x'_1\neq\alpha}.$ 

Prediction with mean is equal to the Bayes function almost everywhere

$$f^{\star}_{impute}(X') = f^{\star}(\tilde{X}) = \mathbb{E}[Y|\tilde{X} = \tilde{x}]$$

Rq: pointwise equality if using a constant out of range.

 $\Rightarrow$  Learn on the mean-imputed training data, impute the test set with the same means and predict is optimal if the missing data are MAR and the **learning algorithm is universally consistent** 

# Consistency of supervised learning with NA: Rationale

- Specific value, systematic like a code for missing
- The learner detects the code and recognizes it at the test time
- With categorical data, just code "Missing"
- With continuous data, any constant:
- Need a lot of data (asymptotic result) and a super powerful learner



Mean imputation not bad for prediction; it is consistent; despite its drawbacks for estimation - Useful in practice!

Empirically good results for MNAR

# Consistency of supervised learning with NA: Rationale

- Specific value, systematic like a code for missing
- The learner detects the code and recognizes it at the test time
- With categorical data, just code "Missing"
- With continuous data, any constant: out of range
- Need a lot of data (asymptotic result) and a super powerful learner



Mean imputation not bad for prediction; it is consistent; despite its drawbacks for estimation - Useful in practice!

Empirically good results for MNAR

### CART (Breiman, 1984)

Built recursively by splitting the current cell into two children: Find the feature  $j^*$ , the threshold  $z^*$  which minimises the (quadratic) loss

$$(j^{\star}, z^{\star}) \in \underset{(j,z)\in\mathcal{S}}{\operatorname{arg\,min}} \mathbb{E}\Big[ \left(Y - \mathbb{E}[Y|X_j \leq z]\right)^2 \cdot \mathbb{1}_{X_j \leq z} + \left(Y - \mathbb{E}[Y|X_j > z]\right)^2 \cdot \mathbb{1}_{X_j > z}\Big].$$



root

## CART (Breiman, 1984)

Built recursively by splitting the current cell into two children: Find the feature  $j^*$ , the threshold  $z^*$  which minimises the (quadratic) loss

$$(j^{\star}, z^{\star}) \in \underset{(j,z)\in\mathcal{S}}{\operatorname{arg\,min}} \mathbb{E}\Big[ \left(Y - \mathbb{E}[Y|X_j \leq z]\right)^2 \cdot \mathbb{1}_{X_j \leq z} + \left(Y - \mathbb{E}[Y|X_j > z]\right)^2 \cdot \mathbb{1}_{X_j > z} \Big].$$



10

### CART (Breiman, 1984)

Built recursively by splitting the current cell into two children: Find the feature  $j^*$ , the threshold  $z^*$  which minimises the (quadratic) loss

$$(j^{\star}, z^{\star}) \in \underset{(j,z)\in\mathcal{S}}{\operatorname{arg\,min}} \mathbb{E}\Big[ \left(Y - \mathbb{E}[Y|X_j \leq z]\right)^2 \cdot \mathbb{1}_{X_j \leq z} + \left(Y - \mathbb{E}[Y|X_j > z]\right)^2 \cdot \mathbb{1}_{X_j > z}\Big].$$



# CART with missing values

root

	$X_1$	<i>X</i> <sub>2</sub>	Υ
1			
2	NA		
3	NA		
4			

#### CART with missing values



1) Select variable and threshold on observed values (1 & 4 for  $X_1$ )  $\mathbb{E}\Big[(Y - \mathbb{E}[Y|X_j \le z, M_j = 0])^2 \cdot \mathbb{1}_{X_j \le z, M_j = 0} + (Y - \mathbb{E}[Y|X_j > z, M_j = 0])^2 \cdot \mathbb{1}_{X_j > z, M_j = 0}\Big].$ 

#### CART with missing values



1) Select variable and threshold on observed values (1 & 4 for  $X_1$ )  $\mathbb{E}\Big[(Y - \mathbb{E}[Y|X_j \le z, M_j = 0])^2 \cdot \mathbb{1}_{X_j \le z, M_j = 0} + (Y - \mathbb{E}[Y|X_j > z, M_j = 0])^2 \cdot \mathbb{1}_{X_j > z, M_j = 0}\Big].$ 

2) Propagate observations (2 & 3) with missing values?

• Probabilistic split:  $Bernoulli(\frac{\#L}{\#L+\#R})$  (Rweeka)

• Block: Send all to a side by minimizing the error (xgboost, lightgbm)

• Surrogate split: Search another variable that gives a close partition (rpart)

One step: select the variable, the threshold and propagate missing values

1. 
$$\{\widetilde{X}_j \leq z \text{ or } \widetilde{X}_j = \mathbb{N}\mathbb{A}\} \text{ vs } \{\widetilde{X}_j > z\}$$
  
2.  $\{\widetilde{X}_j \leq z\} \text{ vs } \{\widetilde{X}_j > z \text{ or } \widetilde{X}_j = \mathbb{N}\mathbb{A}\}$   
3.  $\{\widetilde{X}_i \neq \mathbb{N}\mathbb{A}\} \text{ vs } \{\widetilde{X}_i = \mathbb{N}\mathbb{A}\}.$ 

- The splitting location z depends on the missing values
- Missing values treated like a category (well to handle  $\mathbb{R} \cup NA$ )
- Good for informative pattern (*M* explains *Y*)

Targets one model per pattern:

$$\mathbb{E}\left[Y\Big|\tilde{X}\right] = \sum_{m \in \{0,1\}^d} \mathbb{E}\left[Y|X_{obs(m)}, M = m\right] \mathbb{1}_{M=m}$$

• Implementation <sup>3</sup>: grf package, scikit-learn, partykit

 $\Rightarrow$  Extremely **good performances** in practice **for any mechanism**.

 $<sup>^3</sup>$  implementation trick, J. Tibshirani, duplicate the incomplete columns, and replace the missing entries once by  $+\infty$  and once by  $-\infty$ 

# Consistency: 40% missing values MCAR



# Linear regression with missing values - MLP

#### Explicit Bayes predictor with missing values

#### Linear model:

$$Y = \beta_0 + \langle X, \beta 
angle + \varepsilon, \quad X \in \mathbb{R}^d, \ \varepsilon \text{ gaussian}.$$

#### Bayes predictor for the linear model:

$$f^{\star}(\tilde{X}) = \mathbb{E}[Y|\tilde{X}] = \mathbb{E}[\beta_0 + \beta^{\mathsf{T}}X \mid M, X_{obs(M)}]$$
  
=  $\beta_0 + \beta^{\mathsf{T}}_{obs(M)}X_{obs(M)} + \beta^{\mathsf{T}}_{mis(M)} \mathbb{E}[X_{mis(M)} \mid M, X_{obs(M)}]$ 

#### Assumptions on covariates and missing values

Gaussian pattern mixture model (PMM):  $X \mid (M = m) \sim \mathcal{N}(\mu_m, \Sigma_m)$ Gaussian assumption  $X \sim \mathcal{N}(\mu, \Sigma)$  + MCAR and MAR

#### Under Assump. the Bayes predictor is linear per pattern

 $f^{\star}(X_{obs}, M) = \beta_0^{\star} + \langle \beta_{obs}^{\star}, X_{obs} \rangle + \langle \beta_{mis}^{\star}, \mu_{mis} + \Sigma_{mis,obs} (\Sigma_{obs})^{-1} (X_{obs} - \mu_{obs}) \rangle$ 

use of obs instead of obs(M) for lighter notations

#### (Also for Gaussian assumption + MNAR self mask gaussian)

# Estimation of the bayes predictor

Under Assumpt. the Bayes predictor is linear per pattern  $f^{*}(X_{obs}, M) = \beta_{0}^{*} + \langle \beta_{obs}^{*}, X_{obs} \rangle + \langle \beta_{mis}^{*}, \mu_{mis} + \Sigma_{mis,obs}(\Sigma_{obs})^{-1}(X_{obs} - \mu_{obs}) \rangle$ 

Classical method: use Max Likelihood (EM algo) to estimate  $\Sigma$ . Issues: available implementation strugle with large *d*. <u>Most methods for MAR data</u> or few MNAR variables.

**Theorem. Bayes consistency of a MLP. Le Morvan et al. (2020)** Under linear model + Gaussian pattern mixture model, a MLP:

- with one hidden layer containing 2<sup>d</sup> hidden units
- ReLU activation functions

fed with [X ⊙ (1 − M), M] (X̃ imputed by 0 concatenated with mask)
 can achieve the Bayes rate.

Rationale: The MLP produces a prediction function piecewise affine.

Rq: reduce the model capacity by reducing the number of hidden units. 15

#### Neuman Networks to approximate the covariance matrix

#### The Bayes predictor is linear per pattern

$$f^{\star}(X_{obs}, M) = \beta_{0}^{\star} + \langle \beta_{obs}^{\star}, X_{obs} \rangle + \langle \beta_{mis}^{\star}, \mu_{mis} + \Sigma_{mis,obs} (\Sigma_{obs})^{-1} (X_{obs} - \mu_{obs}) \rangle$$

Order- $\ell$  approx of  $(\Sigma_{obs(m)}^{-1})$  for any m defined recursively:

$$S_{obs(m)}^{(\ell)} = (Id - \Sigma_{obs(m)})S_{obs(m)}^{(\ell-1)} + Id.$$

Neuman Series,  $S^{(0)} = Id$ ,  $\ell = \infty$ :  $(\Sigma_{obs(m)})^{-1} = \sum_{k=0}^{\infty} (Id - \Sigma_{obs(m)})^k$ 

#### Neuman Networks to approximate the covariance matrix

Order- $\ell$  approx of the Bayes predictor in MAR  $f_{\ell}^{\star}(X_{obs}, M) = \langle \beta_{obs}, X_{obs} \rangle + \langle \beta_{mis}, \mu_{mis} + \sum_{mis,obs} S_{obs(m)}^{(\ell)}(X_{obs} - \mu_{obs}) \rangle.$ Order- $\ell$  approx of  $(\sum_{obs(m)}^{-1})$  for any m defined recursively:  $S_{obs(m)}^{(\ell)} = (Id - \sum_{obs(m)})S_{obs(m)}^{(\ell-1)} + Id.$ Neuman Series,  $S^{(0)} = Id, \ \ell = \infty$ :  $(\sum_{obs(m)})^{-1} = \sum_{k=0}^{\infty} (Id - \sum_{obs(m)})^{k}$ 

#### $\Rightarrow$ Neural network architecture to approximate the Bayes predictor



**Figure 1:** Depth of 3,  $\bar{m} = 1 - m$ . Each weight matrix  $W^{(k)}$  corresponds to a simple transformation of the covariance matrix indicated in blue.

16

## Networks with missing values: $\odot M$ nonlinearity



• Implementing a network with the matrix weights  $W^{(k)} = (I - \Sigma_{obs(m)})$ masked differently for each sample can be challenging

• Masked weights is equivalent to masking input & output vector. Let v a vector,  $\overline{m} = 1 - m$ .  $(W \odot \overline{m} \overline{m}^{\top})v = (W(v \odot \overline{m})) \odot \overline{m}$ 

Classic network with multiplications by the mask nonlinearities  $\odot M$ 

#### Networks with missing values: $\odot M$ nonlinearity



• Implementing a network with the matrix weights  $W^{(k)} = (I - \Sigma_{obs(m)})$ masked differently for each sample can be challenging

• Masked weights is equivalent to masking input & output vector. Let v a vector,  $\overline{m} = 1 - m$ .  $(W \odot \overline{m} \overline{m}^{\top})v = (W(v \odot \overline{m})) \odot \overline{m}$ 

Classic network with multiplications by the mask nonlinearities  $\odot M$ 

**Proposition (equivalence MLP** - depth-0 Neumann network) A MLP with ReLU activations, one hidden layer of d hidden units, and which operates on the  $[X \odot (1 - M), M]$ , the input X imputed by 0 concatenated with the mask M, is equivalent to the 0-depth NN

- Max Likelihood: to estimate the parameters of the joint Gaussian distribution (X<sub>1</sub>, ..., X<sub>d</sub>, Y) with EM. Predict by conditional expectation of Y given X<sub>obs</sub>.
- ICE + LR: conditional imputation with an iterative imputer followed by linear regression.
- MLP: take as input the data imputed by 0 concatenated with the mask [X ⊙ (1 − M), M] with ReLU nonlinearity,
  - MLP-Wide: one hidden layer with width increased (between d & 2<sup>d</sup>)
  - MLP-Deep: 1 to 10 hidden layers of d hidden units
- **Neumann**: The Neumann architecture with the ⊙*M*, choosing the depth on a validation set.

# Results



**Figure 2: Predictive performances in various scenarios** — varying missing-value mechanisms, number of samples *n*, and number of features *d*.

 $\Rightarrow$  Best performances for MNAR scenario (50% of NA on all variables)

• More effective to increase the capacity of the Neumann network (depth) than to increase the capacity (width) of MLP Wide.

• Neumann network learn improved weights compared to Neumann iterations 19

# **Discussion** - challenges

Supervised learning different from usual inferential probabilistic models. Solutions useful in practice robust to the missing-value mechanisms but needs powerful model.

#### Powerful learner with missing values

- $\bullet$  Incomplete train and test  $\rightarrow$  same imputation model
- Single constant imputation is consistent with a powerful learner
- Tree-based models : Missing Incorporated in Attribute
- To be done: nonasymptotic results, uncertainty, distributional shift: No NA in the test? Proofs in MNAR

#### Linear regression with missing values

- The Bayes predictor is explicit under Gaussian assumptions/ MAR and gaussian self mask but high-dimensional.
- Approx include MLP which can be consistent and Neuman Network
- New architecture for network with missing data:  $\odot M$  nonlinearity.

#### <u>**R-miss-tastic**</u> https://rmisstastic.netlify.com/R-miss-tastic

- J., I. Mayer, N. Tierney & N. Vialaneix
- Project funded by the R consortium (Infrastructure Steering Committee)<sup>4</sup>

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

- list existing packages
- available literature
- tutorials
- analysis workflows on data
- main actors
- $\Rightarrow$  Federate the community

 $\Rightarrow$  Contribute!

<sup>&</sup>lt;sup>4</sup>https://www.r-consortium.org/projects/call-for-proposals

Examples:

- Lecture <sup>5</sup> General tutorial : Statistical Methods for Analysis with Missing Data (Mauricio Sadinle)
- Lecture Multiple Imputation: mice by Nicole Erler <sup>6</sup>
- Longitudinal data, Time Series Imputation (<u>Steffen Moritz</u> very active contributor of r-miss-tastic), Principal Component Methods<sup>7</sup>

multipleimputation\_2018/erler\_practical\_mice\_2018

<sup>&</sup>lt;sup>5</sup>https://rmisstastic.netlify.com/lectures/

<sup>&</sup>lt;sup>6</sup>https://rmisstastic.netlify.com/tutorials/erler\_course\_

<sup>&</sup>lt;sup>7</sup>https://rmisstastic.netlify.com/tutorials/Josse\_slides\_imputation\_PCA\_2018.pdf

# Thank you

