# Conformal prediction with missing values

Margaux Zaffran

MIA Seminar

February 9, 2023

**Aymeric Dieuleveut**
Ecole Polytechnique
*Paris*

**Julie Josse**
INRIA IDESP
*Montpellier*

**Yaniv Romano**
Technion - Israel Institute of Technology
*Haifa*

# Motivation: critical medical care

# TraumaBase®: decision support for trauma patients

- More than 30 000 trauma patients
- 30 hospitals
- 4 000 new patients per year
- 250 continuous and categorical variables
  $\hookrightarrow$ Many useful statistical tasks

# TraumaBase®: decision support for trauma patients

- More than 30 000 trauma patients
- 30 hospitals
- 4 000 new patients per year
- 250 continuous and categorical variables
  $\hookrightarrow$ Many useful statistical tasks

Predict the level of platelets upon arrival at hospital, given 7 covariates chosen by medical doctors.

# TraumaBase®: decision support for trauma patients

- More than 30 000 trauma patients
- 30 hospitals
- 4 000 new patients per year
- 250 continuous and categorical variables
  $\hookrightarrow$ Many useful statistical tasks

Predict the level of platelets upon arrival at hospital, given 7 covariates chosen by medical doctors.

These covariates are not always observed.

# Missing values: ubiquitous in data science pratice

| $Y$ | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ |
|---|---|---|---|---|---|---|
| 22.42 | 0.55 | 0.67 | 0.03 | 0.75 | 0.05 | 0.05 |
| 8.26 | 0.72 | 0.18 | 0.55 | 0.05 | 0.73 | 0.50 |
| 19.41 | 0.60 | 0.58 | NA | NA | NA | 0.40 |
| 19.75 | 0.54 | 0.43 | 0.96 | 0.77 | 0.06 | 0.66 |
| 7.32 | NA | 0.19 | NA | 0.02 | 0.83 | 0.04 |
| 13.55 | 0.65 | 0.69 | 0.50 | 0.15 | NA | 0.87 |
| 20.75 | 0.43 | 0.74 | 0.61 | 0.72 | 0.52 | 0.35 |
| 9.26 | 0.89 | NA | 0.84 | 0.01 | 0.73 | NA |
| 9.68 | 0.963 | 0.45 | 0.65 | 0.04 | 0.06 | NA |

# Missing values: ubiquitous in data science pratice

| $Y$ | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ |
|---|---|---|---|---|---|---|
| 22.42 | 0.55 | 0.67 | 0.03 | 0.75 | 0.05 | 0.05 |
| 8.26 | 0.72 | 0.18 | 0.55 | 0.05 | 0.73 | 0.50 |
| ~~19.41~~ | ~~0.60~~ | ~~0.58~~ | ~~NA~~ | ~~NA~~ | ~~NA~~ | ~~0.40~~ |
| 19.75 | 0.54 | 0.43 | 0.96 | 0.77 | 0.06 | 0.66 |
| ~~7.32~~ | ~~NA~~ | ~~0.19~~ | ~~NA~~ | ~~0.02~~ | ~~0.83~~ | ~~0.04~~ |
| ~~13.55~~ | ~~0.65~~ | ~~0.69~~ | ~~0.50~~ | ~~0.15~~ | ~~NA~~ | ~~0.87~~ |
| 20.75 | 0.43 | 0.74 | 0.61 | 0.72 | 0.52 | 0.35 |
| ~~9.26~~ | ~~0.89~~ | ~~NA~~ | ~~0.84~~ | ~~0.01~~ | ~~0.73~~ | ~~NA~~ |
| ~~9.68~~ | ~~0.963~~ | ~~0.45~~ | ~~0.65~~ | ~~0.04~~ | ~~0.06~~ | ~~NA~~ |

[1] Zhu et al. (2019), *High-dimensional PCA with heterogeneous missingness*, JRSS B

# Missing values: ubiquitous in data science pratice

| $Y$ | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ |
|---|---|---|---|---|---|---|
| 22.42 | 0.55 | 0.67 | 0.03 | 0.75 | 0.05 | 0.05 |
| 8.26 | 0.72 | 0.18 | 0.55 | 0.05 | 0.73 | 0.50 |
| ~~19.41~~ | ~~0.60~~ | ~~0.58~~ | ~~NA~~ | ~~NA~~ | ~~NA~~ | ~~0.40~~ |
| 19.75 | 0.54 | 0.43 | 0.96 | 0.77 | 0.06 | 0.66 |
| ~~7.32~~ | ~~NA~~ | ~~0.19~~ | ~~NA~~ | ~~0.02~~ | ~~0.83~~ | ~~0.04~~ |
| ~~13.55~~ | ~~0.65~~ | ~~0.69~~ | ~~0.50~~ | ~~0.15~~ | ~~NA~~ | ~~0.87~~ |
| 20.75 | 0.43 | 0.74 | 0.61 | 0.72 | 0.52 | 0.35 |
| ~~9.26~~ | ~~0.89~~ | ~~NA~~ | ~~0.84~~ | ~~0.01~~ | ~~0.73~~ | ~~NA~~ |
| ~~9.68~~ | ~~0.963~~ | ~~0.45~~ | ~~0.65~~ | ~~0.04~~ | ~~0.06~~ | ~~NA~~ |

If each entry has a probability 0.01 of being missing:

$$d = 6 \rightarrow \approx 94\% \text{ of rows kept}$$

$$d = 300 \rightarrow \approx 5\% \text{ of rows kept}$$

[1] Zhu et al. (2019), *High-dimensional PCA with heterogeneous missingness*, JRSS B

# Missing values: ubiquitous in data science pratice

| $Y$ | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ |
|---|---|---|---|---|---|---|
| 22.42 | 0.55 | 0.67 | 0.03 | 0.75 | 0.05 | 0.05 |
| 8.26 | 0.72 | 0.18 | 0.55 | 0.05 | 0.73 | 0.50 |
| ~~19.41~~ | ~~0.60~~ | ~~0.58~~ | ~~NA~~ | ~~NA~~ | ~~NA~~ | ~~0.40~~ |
| 19.75 | 0.54 | 0.43 | 0.96 | 0.77 | 0.06 | 0.66 |
| ~~7.32~~ | ~~NA~~ | ~~0.19~~ | ~~NA~~ | ~~0.02~~ | ~~0.83~~ | ~~0.04~~ |
| ~~13.55~~ | ~~0.65~~ | ~~0.69~~ | ~~0.50~~ | ~~0.15~~ | ~~NA~~ | ~~0.87~~ |
| 20.75 | 0.43 | 0.74 | 0.61 | 0.72 | 0.52 | 0.35 |
| ~~9.26~~ | ~~0.89~~ | ~~NA~~ | ~~0.84~~ | ~~0.01~~ | ~~0.73~~ | ~~NA~~ |
| ~~9.68~~ | ~~0.963~~ | ~~0.45~~ | ~~0.65~~ | ~~0.04~~ | ~~0.06~~ | ~~NA~~ |

If each entry has a probability 0.01 of being missing:

$$d = 6 \rightarrow \; \approx 94\% \text{ of rows kept}$$

$$d = 300 \rightarrow \; \approx 5\% \text{ of rows kept}$$

*One of the* **ironies of Big Data** *is that missing data play an ever more significant role.*[1]

[1] Zhu et al. (2019), *High-dimensional PCA with heterogeneous missingness*, JRSS B

# Handling missing values depends on pattern and mechanism

- $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables.
- $M \in \{0, 1\}^d$ is defined as $M_j = 1 \Leftrightarrow X_j$ is missing.
  $M$ is called the **mask** or the **missing pattern**.

[2] Rubin (1976), *Inference and missing data*, Biometrika

# Handling missing values depends on pattern and mechanism

- $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables.
- $M \in \{0,1\}^d$ is defined as $M_j = 1 \Leftrightarrow X_j$ is missing.
  $M$ is called the **mask** or the **missing pattern**.

**Example**

We observe $(\texttt{NA}, 6, 2)$. Then $m = (1, 0, 0)$.

[2] Rubin (1976), *Inference and missing data*, Biometrika

## Handling missing values depends on pattern and mechanism

- $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables.
- $M \in \{0,1\}^d$ is defined as $M_j = 1 \Leftrightarrow X_j$ is missing.
  $M$ is called the **mask** or the **missing pattern**.

**Example**

We observe $(-1, \texttt{NA}, 2)$. Then $m = (0, 1, 0)$.

[2] Rubin (1976), *Inference and missing data*, Biometrika

# Handling missing values depends on pattern and mechanism

- $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables.
- $M \in \{0, 1\}^d$ is defined as $M_j = 1 \Leftrightarrow X_j$ is missing.
  $M$ is called the **mask** or the **missing pattern**.

**Example**

We observe $(-1, \texttt{NA}, \texttt{NA})$. Then $m = (0, 1, 1)$.

[2] Rubin (1976), *Inference and missing data*, Biometrika

# Handling missing values depends on pattern and mechanism

- $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables.
- $M \in \{0, 1\}^d$ is defined as $M_j = 1 \Leftrightarrow X_j$ is missing.
  $M$ is called the **mask** or the **missing pattern**.

**Example**

We observe $(-1, \texttt{NA}, \texttt{NA})$. Then $m = (0, 1, 1)$.

There are $2^d$ **patterns** (statistical and computational challenges).

[2] Rubin (1976), *Inference and missing data*, Biometrika

# Handling missing values depends on pattern and mechanism

- $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables.
- $M \in \{0, 1\}^d$ is defined as $M_j = 1 \Leftrightarrow X_j$ is missing.
  $M$ is called the **mask** or the **missing pattern**.

**Example**

We observe $(-1, \texttt{NA}, \texttt{NA})$. Then $m = (0, 1, 1)$.

There are $2^d$ **patterns** (statistical and computational challenges).

- Three **mechanisms**[2] can generate missing values.

[2] Rubin (1976), *Inference and missing data*, Biometrika

# Handling missing values depends on pattern and mechanism

- $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables.
- $M \in \{0,1\}^d$ is defined as $M_j = 1 \Leftrightarrow X_j$ is missing.
  $M$ is called the **mask** or the **missing pattern**.

**Example**

We observe $(-1, \texttt{NA}, \texttt{NA})$. Then $m = (0, 1, 1)$.

There are $2^d$ **patterns** (statistical and computational challenges).

- Three **mechanisms**[2] can generate missing values.
  $\hookrightarrow$ **Missing Completely At Random** (MCAR):
  $\mathbb{P}(M = m|X) = \mathbb{P}(M = m)$ for all $m \in \{0,1\}^d$.

[2] Rubin (1976), *Inference and missing data*, Biometrika

# Handling missing values depends on pattern and mechanism

- $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables.
- $M \in \{0,1\}^d$ is defined as $M_j = 1 \Leftrightarrow X_j$ is missing.
  $M$ is called the **mask** or the **missing pattern**.

**Example**

We observe $(-1, \texttt{NA}, \texttt{NA})$. Then $m = (0, 1, 1)$.

There are $2^d$ **patterns** (statistical and computational challenges).

- Three **mechanisms**[2] can generate missing values.
  $\hookrightarrow$ **Missing Completely At Random** (MCAR):
  $\mathbb{P}(M = m|X) = \mathbb{P}(M = m)$ for all $m \in \{0,1\}^d$. $M \perp\!\!\!\perp X$,
  missingness does not depend on the variables.

[2] Rubin (1976), *Inference and missing data*, Biometrika

# Supervised learning with missing values

Impute-then-regress procedures are widely used (Le Morvan et al., 2021).

---

Le Morvan et al. (2021), *What's a good imputation to predict with missing values?*, NeurIPS

# Supervised learning with missing values

Impute-then-regress procedures are widely used (Le Morvan et al., 2021).

1. Replace `NA` using an imputation function (e.g. the mean), noted $\phi$.

Le Morvan et al. (2021), *What's a good imputation to predict with missing values?*, NeurIPS

Impute-then-regress procedures are widely used (Le Morvan et al., 2021).

1. Replace NA using an imputation function (e.g. the mean), noted $\phi$.

$$\begin{array}{c}x^{(1)}\\x^{(2)}\\x^{(3)}\\x^{(4)}\end{array}\begin{array}{|c|c|c|c|}\hline -1 & -10 & 6 & 0 \\ \hline 4 & \texttt{NA} & -2 & 2 \\ \hline 5 & 1 & 2 & \texttt{NA} \\ \hline 0 & \texttt{NA} & \texttt{NA} & 1 \\ \hline \end{array} \xrightarrow{\quad\phi\quad} \begin{array}{c}u^{(1)}\\u^{(2)}\\u^{(3)}\\u^{(4)}\end{array}\begin{array}{|c|c|c|c|}\hline -1 & -10 & 6 & 0 \\ \hline 4 & -4.5 & -2 & 2 \\ \hline 5 & 1 & 2 & 1 \\ \hline 0 & -4.5 & 3 & 1 \\ \hline \end{array}$$

# Supervised learning with missing values

Impute-then-regress procedures are widely used (Le Morvan et al., 2021).

1. Replace `NA` using an imputation function (e.g. the mean), noted $\phi$.
2. Train your algorithm (Random Forest, Neural Nets, etc.) on the imputed data: $\left\{ \underbrace{\phi\left(x^{(k)}_{\mathrm{obs}(m^{(k)})}, m^{(k)}\right)}_{\text{imputed } x^{(k)}}, y^{(k)} \right\}_{k=1}^{n}$.

Le Morvan et al. (2021), *What's a good imputation to predict with missing values?*, NeurIPS

# Supervised learning with missing values

Impute-then-regress procedures are widely used (Le Morvan et al., 2021).

1. Replace `NA` using an imputation function (e.g. the mean), noted $\phi$.
2. Train your algorithm (Random Forest, Neural Nets, etc.) on the imputed data: $\left\{ \underbrace{\phi\left(x^{(k)}_{\mathrm{obs}(m^{(k)})}, m^{(k)}\right)}_{\text{imputed } x^{(k)}}, y^{(k)} \right\}_{k=1}^{n}$.

Le Morvan et al. (2021) show that for **any deterministic imputation** and **universal learner** this procedure is **Bayes-consistent**.

---

Le Morvan et al. (2021), *What's a good imputation to predict with missing values?*, NeurIPS

# Back to predicting the levels of platelets

- **Challenging task:** Jiang et al. (2022) achieved an average relative prediction error ($\|\hat{y} - y\|^2 / \|y\|^2$) no lower than 0.23

# Back to predicting the levels of platelets

- **Challenging task:** Jiang et al. (2022) achieved an average relative prediction error ($\|\hat{y} - y\|^2 / \|y\|^2$) no lower than 0.23
- **Crucial task:** high-stakes decision-making problem

$\hookrightarrow$ High need for quantifying the predictive uncertainty.

# Beyond point prediction?

# Objective

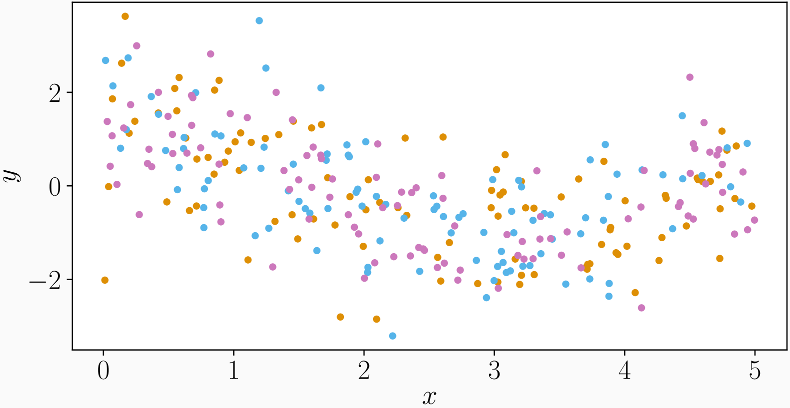
- Predict an unseen point $Y^{(n+1)}$ at $X^{(n+1)}$ with **confidence**
- Miscoverage level $\alpha \in [0, 1]$

▶ Build a predictive interval $\mathcal{C}_\alpha$ such that:

$$\mathbb{P}\left\{Y^{(n+1)} \in \mathcal{C}_\alpha\left(X^{(n+1)}\right)\right\} \geq 1 - \alpha, \tag{1}$$

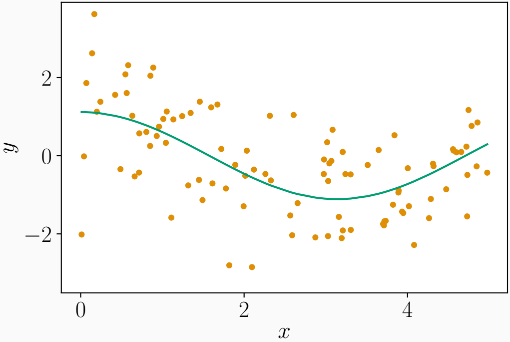
and $\mathcal{C}_\alpha$ should be as small as possible, in order to be informative.

# Split conformal prediction[1,2,3]: toy example




[1] Vovk et al. (2005), *Algorithmic Learning in a Random World*
[2] Papadopoulos et al. (2002), *Inductive Confidence Machines for Regression*, ECML
[3] Lei et al. (2018), *Distribution-Free Predictive Inference for Regression*, JRSS B

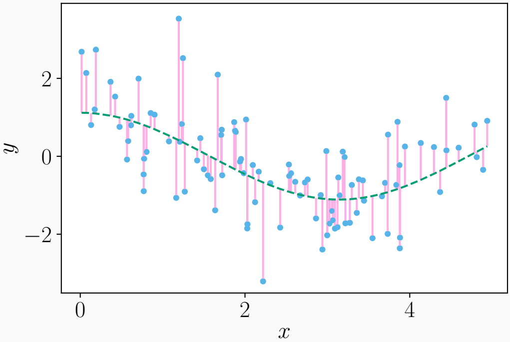
# Split conformal prediction[1,2,3]: training step




- Learn $\hat{\mu}$

[1]Vovk et al. (2005), *Algorithmic Learning in a Random World*
[2]Papadopoulos et al. (2002), *Inductive Confidence Machines for Regression*, ECML
[3]Lei et al. (2018), *Distribution-Free Predictive Inference for Regression*, JRSS B

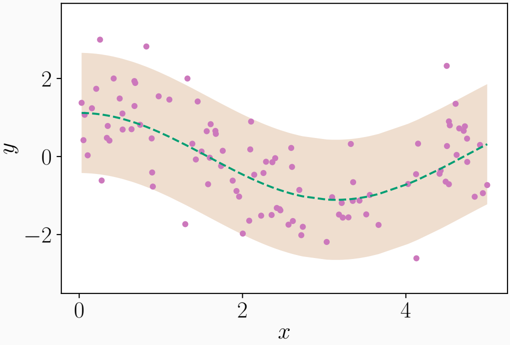
# Split conformal prediction[1,2,3]: calibration step




- Predict with $\hat{\mu}$
- Get the residuals $\hat{\varepsilon}^{(k)}$
- Compute the $(1-\alpha)\times(1+\frac{1}{\#\mathrm{Cal}})$ empirical quantile of the $\left|\hat{\varepsilon}^{(k)}\right|$, noted $Q_{1-\tilde{\alpha}}\left(\left|\hat{\varepsilon}^{(k)}\right|\right)$

[1]Vovk et al. (2005), *Algorithmic Learning in a Random World*
[2]Papadopoulos et al. (2002), *Inductive Confidence Machines for Regression*, ECML
[3]Lei et al. (2018), *Distribution-Free Predictive Inference for Regression*, JRSS B

# Split conformal prediction[1,2,3]: prediction step




- Predict with $\hat{\mu}$
- Build $\widehat{C}_\alpha(x)$:
  $[\hat{\mu}(x) \pm Q_{1-\tilde{\alpha}}\left(|\hat{\varepsilon}^{(k)}|\right)]$

[1] Vovk et al. (2005), *Algorithmic Learning in a Random World*
[2] Papadopoulos et al. (2002), *Inductive Confidence Machines for Regression*, ECML
[3] Lei et al. (2018), *Distribution-Free Predictive Inference for Regression*, JRSS B

# Theory on SCP

Papadopoulos et al. (2002); Lei et al. (2018) prove that:

- given any regression function $\hat{\mu}$
- for any (**finite**) sample size $n$
- if the $(X^{(k)}, Y^{(k)})$ are **exchangeable**

then:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha(X)\right) \geq 1 - \alpha.$$

# Theory on SCP

Papadopoulos et al. (2002); Lei et al. (2018) prove that:

- given any regression function $\hat{\mu}$
- for any (**finite**) sample size $n$
- if the $(X^{(k)}, Y^{(k)})$ are **exchangeable**

then:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha(X)\right) \geq 1 - \alpha.$$

If additionally the scores $|\hat{\varepsilon}_k|$ are almost surely distinct:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha(X)\right) \leq 1 - \alpha + \frac{1}{1 + \#\mathrm{Cal}}.$$

# Split conformal prediction: summary

Split conformal prediction is simple to compute and works:

- any regression algorithm (neural nets, random forest...);
- distribution-free as long as the data is exchangeable;

- finite sample.

# Split conformal prediction: summary

Split conformal prediction is simple to compute and works:

- any regression algorithm (neural nets, random forest...);
- distribution-free as long as the data is exchangeable;
  - $\hookrightarrow$ the scores need to be exchangeable (but then it would not work with any regression algorithm)
- finite sample.

# Split conformal prediction: summary

Split conformal prediction is simple to compute and works:

- any regression algorithm (neural nets, random forest...);
- distribution-free as long as the data is exchangeable;
  - $\hookrightarrow$ the scores need to be exchangeable (but then it would not work with any regression algorithm)
- finite sample.

The theoretical guarantee is **marginal** over the joint distribution of $(X, Y)$, and **not conditional**. No guarantee that for any $x \in \mathbb{R}$:

$$\mathbb{P}\left\{Y^{(n+1)} \in \widehat{C}_\alpha\left(X^{(n+1)}\right) | X^{(n+1)} = x\right\} \geq 1 - \alpha.$$

# Split conformal prediction: summary

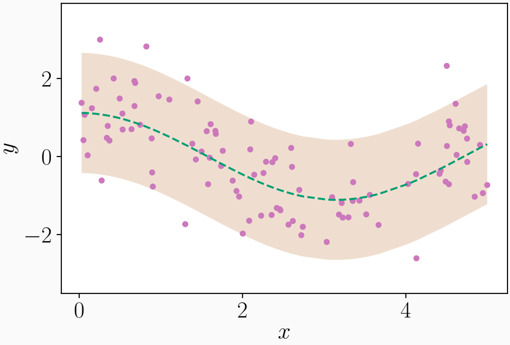
Split conformal prediction is simple to compute and works:

- any regression algorithm (neural nets, random forest...);
- distribution-free as long as the data is exchangeable;
  - $\hookrightarrow$ the scores need to be exchangeable (but then it would not work with any regression algorithm)
- finite sample.

The theoretical guarantee is **marginal** over the joint distribution of $(X, Y)$, and **not conditional**. No guarantee that for any $x \in \mathbb{R}$:

$$\cancel{\mathbb{P}\left\{ Y^{(n+1)} \in \widehat{C}_\alpha\left(X^{(n+1)}\right) | X^{(n+1)} = x \right\} \geq 1 - \alpha.}$$

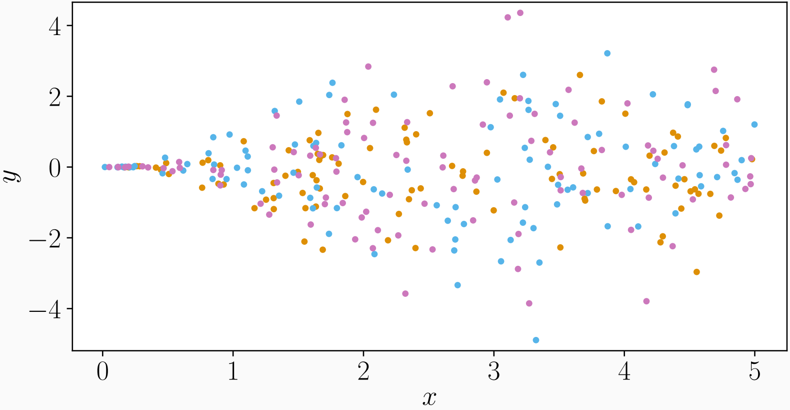
# Split conformal prediction[1,2,3]: prediction step




- Predict with $\hat{\mu}$
- Build $\widehat{C}_{\alpha}(x)$: $[\hat{\mu}(x) \pm Q_{1-\tilde{\alpha}}\left(|\hat{\varepsilon}^{(k)}|\right)]$

[1] Vovk et al. (2005), *Algorithmic Learning in a Random World*
[2] Papadopoulos et al. (2002), *Inductive Confidence Machines for Regression*, ECML
[3] Lei et al. (2018), *Distribution-Free Predictive Inference for Regression*, JRSS B

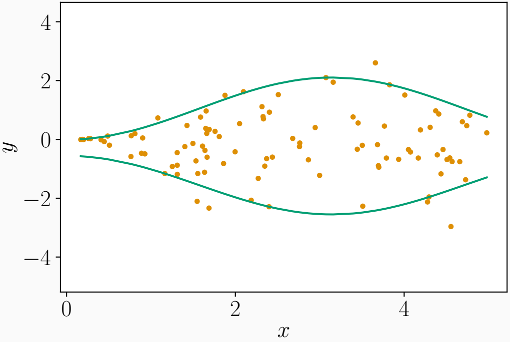
# Conformalized Quantile Regression (Romano et al., 2019)




Randomly split the data to obtain a proper training set and a calibration set. Keep the test set.

Romano et al. (2019), *Conformalized Quantile Regression*, NeurIPS

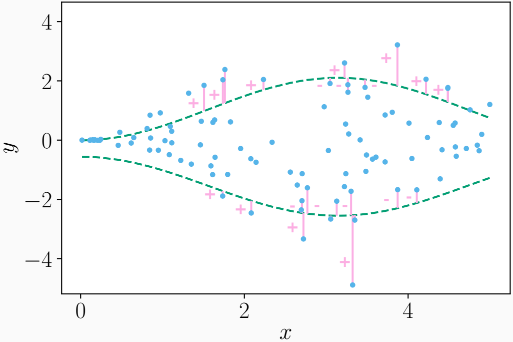
# Conformalized Quantile Regression (Romano et al., 2019)




- Learn $\hat{q}_{\mathsf{inf}}$ and $\hat{q}_{\mathsf{sup}}$

Romano et al. (2019), *Conformalized Quantile Regression*, NeurIPS

# Conformalized Quantile Regression (Romano et al., 2019)

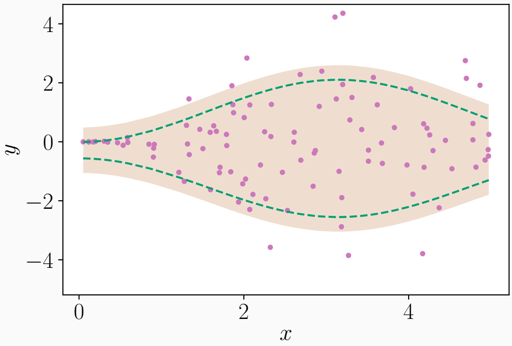



- Predict with $\hat{q}_{\text{inf}}$ and $\hat{q}_{\text{sup}}$
- Get the scores $e^{(k)}$
- Compute the $(1-\alpha) \times (1 + \frac{1}{\#\text{Cal}})$ empirical quantile of the $e^{(k)}$, noted $Q_{1-\tilde{\alpha}}(e)$

$$\hookrightarrow \quad e^{(k)} := \max\left\{\hat{q}_{\text{inf}}\left(x^{(k)}\right) - y^{(k)}, y^{(k)} - \hat{q}_{\text{sup}}\left(x^{(k)}\right)\right\}$$

Romano et al. (2019), *Conformalized Quantile Regression*, NeurIPS

# Conformalized Quantile Regression (Romano et al., 2019)




- Predict with $\hat{q}_{\text{inf}}$ and $\hat{q}_{\text{sup}}$
- Build $\hat{C}_\alpha(x)$:

$$\left[\hat{q}_{\text{inf}}(x) - Q_{1-\tilde{\alpha}}\left(e\right), \hat{q}_{\text{sup}}(x) + Q_{1-\tilde{\alpha}}\left(e\right)\right]$$

Romano et al. (2019), *Conformalized Quantile Regression*, NeurIPS

# Theory on CQR

Romano et al. (2019) prove that:

- given any quantile regression functions $\hat{q}_{\mathrm{inf}}$ and $\hat{q}_{\mathrm{sup}}$
- for any (**finite**) sample size $n$
- if the $(X^{(k)}, Y^{(k)})$ are **exchangeable**

then:

$$\mathbb{P}\left(Y \in \hat{C}_{\alpha}(X)\right) \geq 1 - \alpha$$

Romano et al. (2019), *Conformalized Quantile Regression*, NeurIPS

# Theory on CQR

Romano et al. (2019) prove that:

- given any quantile regression functions $\hat{q}_{\text{inf}}$ and $\hat{q}_{\text{sup}}$
- for any (**finite**) sample size $n$
- if the $(X^{(k)}, Y^{(k)})$ are **exchangeable**

then:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha(X)\right) \geq 1 - \alpha$$

If additionally the scores $e^{(k)}$ are almost surely distinct:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha(X)\right) \leq 1 - \alpha + \frac{1}{1 + \#\mathrm{Cal}}.$$

Romano et al. (2019), *Conformalized Quantile Regression*, NeurIPS

# Conformal prediction with missing values

# Impute-then-regress+conformalization is marginally valid

To apply conformal prediction we need **exchangeable** data.

# Impute-then-regress+conformalization is marginally valid

To apply conformal prediction we need **exchangeable** data.

**Lemma (Exchangeability after imp., Zaffran et al., 2023)**

*Assume* $\left(X^{(k)}, M^{(k)}, Y^{(k)}\right)_{k=1}^{n}$ *are i.i.d. (or exchangeable).*

*Then, for any missing mechanism, for almost all imputation function* $\phi$*:*

$\left(\phi\left(X^{(k)}_{obs(M^{(k)})}, M^{(k)}\right), Y^{(k)}\right)_{k=1}^{n}$ *are* **exchangeable**.

# Impute-then-regress+conformalization is marginally valid

To apply conformal prediction we need **exchangeable** data.

**Lemma (Exchangeability after imp., Zaffran et al., 2023)**

*Assume* $\left(X^{(k)}, M^{(k)}, Y^{(k)}\right)_{k=1}^{n}$ *are i.i.d. (or exchangeable).*

*Then, for any missing mechanism, for almost all imputation function* $\phi$*:*

$\left(\phi\left(X^{(k)}_{obs(M^{(k)})}, M^{(k)}\right), Y^{(k)}\right)_{k=1}^{n}$ *are* ***exchangeable****.*

$\Rightarrow$ Conformal prediction applied on an imputed data set still enjoys marginal guarantees:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha\left(X_{\text{obs}(M)}, M\right)\right) \geq 1 - \alpha.$$

# Impute-then-regress+conformalization is marginally valid

To apply conformal prediction we need **exchangeable** data.

**Lemma (Exchangeability after imp., Zaffran et al., 2023)**

*Assume* $\left(X^{(k)}, M^{(k)}, Y^{(k)}\right)_{k=1}^{n}$ *are i.i.d. (or exchangeable).*

*Then, for* **any missing mechanism**, *for almost all imputation function* $\phi$*:*

$\left(\phi\left(X^{(k)}_{obs(M^{(k)})}, M^{(k)}\right), Y^{(k)}\right)_{k=1}^{n}$ *are* **exchangeable**.

$\Rightarrow$ Conformal prediction applied on an imputed data set still enjoys marginal guarantees:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha\left(X_{\text{obs}(M)}, M\right)\right) \geq 1 - \alpha.$$

Even if the imputation is not accurate, the guarantee will hold.

# CQR performances on an illustrative example

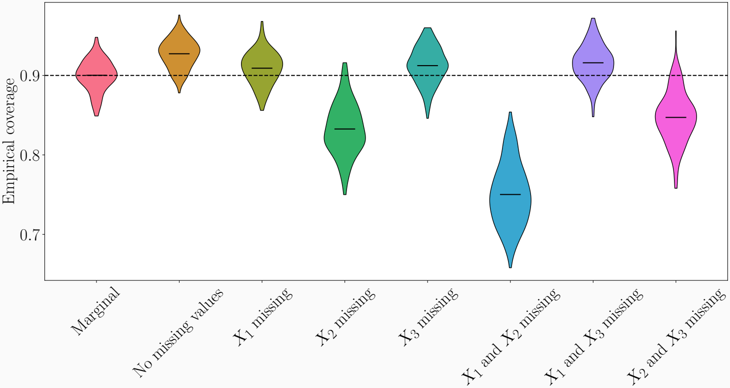
$$Y = \beta^T X + \varepsilon,$$

with $\beta = (1, 2, -1)^T$, $\varepsilon \perp\!\!\!\perp X$ and $X$ and $\varepsilon$ are Gaussian.

# CQR performances on an illustrative example

$$Y = \beta^T X + \varepsilon,$$

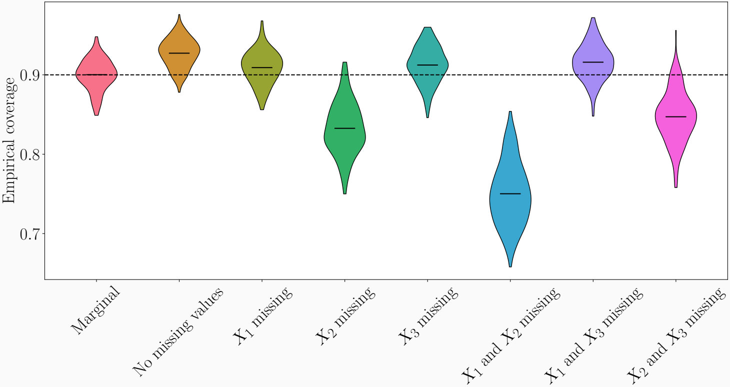
with $\beta = (1, 2, -1)^T$, $\varepsilon \perp\!\!\!\perp X$ and $X$ and $\varepsilon$ are Gaussian.



# CQR performances on an illustrative example

$$Y = \beta^T X + \varepsilon,$$

with $\beta = (1, 2, -1)^T$, $\varepsilon \perp\!\!\!\perp X$ and $X$ and $\varepsilon$ are Gaussian.




Warning: the predictive intervals cover properly **marginally**, but suffer from high **disparities depending on the missing patterns**.

# Missing values induce heteroskedasticity

Theoretical study of the Gaussian linear model ($Y = \beta^T X + \varepsilon$) generalizes:

# Missing values induce heteroskedasticity

Theoretical study of the Gaussian linear model ($Y = \beta^T X + \varepsilon$) generalizes:

**Proposition (Oracle intervals under the Gaussian lin. mod.)**

$$\mathcal{L}_\alpha^*(m) = 2 \times q_{1-\alpha/2}^{\mathcal{N}(0,1)} \times \sqrt{\beta_{\mathrm{mis}(m)}^T \Sigma_{\mathrm{mis}|\mathrm{obs}}^m \beta_{\mathrm{mis}(m)} + \sigma_\varepsilon^2}.$$

# Missing values induce heteroskedasticity

Theoretical study of the Gaussian linear model ($Y = \beta^T X + \varepsilon$) generalizes:

**Proposition (Oracle intervals under the Gaussian lin. mod.)**

$$\mathcal{L}_\alpha^*(m) = 2 \times q_{1-\alpha/2}^{\mathcal{N}(0,1)} \times \sqrt{\beta_{\mathrm{mis}(m)}^T \Sigma_{\mathrm{mis|obs}}^m \beta_{\mathrm{mis}(m)} + \sigma_\varepsilon^2}.$$

- Even with an homoskedastic noise, missingness generates heteroskedasticity

# Missing values induce heteroskedasticity

Theoretical study of the Gaussian linear model ($Y = \beta^T X + \varepsilon$) generalizes:

**Proposition (Oracle intervals under the Gaussian lin. mod.)**

$$\mathcal{L}_\alpha^*(m) = 2 \times q_{1-\alpha/2}^{\mathcal{N}(0,1)} \times \sqrt{\beta_{\mathrm{mis}(m)}^T \Sigma_{\mathrm{mis}|\mathrm{obs}}^m \beta_{\mathrm{mis}(m)} + \sigma_\varepsilon^2}.$$

- Even with an homoskedastic noise, missingness generates heteroskedasticity
- The uncertainty increases when missing values are associated with larger regression coefficients (i.e. the most predictive variables)

# Missing values induce heteroskedasticity

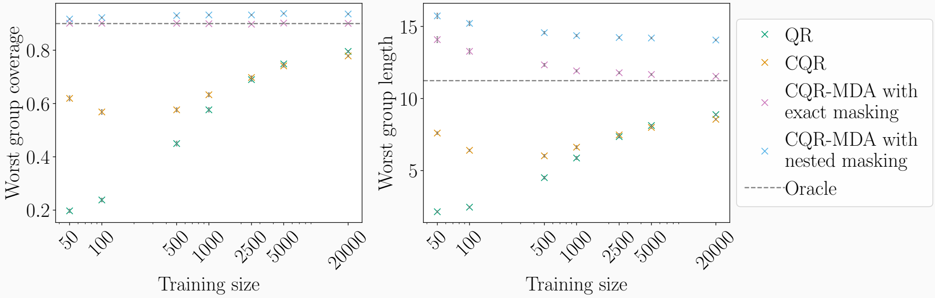
Theoretical study of the Gaussian linear model ($Y = \beta^T X + \varepsilon$) generalizes:

**Proposition (Oracle intervals under the Gaussian lin. mod.)**

$$\mathcal{L}^*_\alpha(m) = 2 \times q_{1-\alpha/2}^{\mathcal{N}(0,1)} \times \sqrt{\beta^T_{\mathrm{mis}(m)} \Sigma^m_{\mathrm{mis|obs}} \beta_{\mathrm{mis}(m)} + \sigma^2_\varepsilon}.$$

- Even with an homoskedastic noise, missingness generates heteroskedasticity
- The uncertainty increases when missing values are associated with larger regression coefficients (i.e. the most predictive variables)
- The uncertainty increases when there are more missing values

# CQR is not enough (and spoiler)



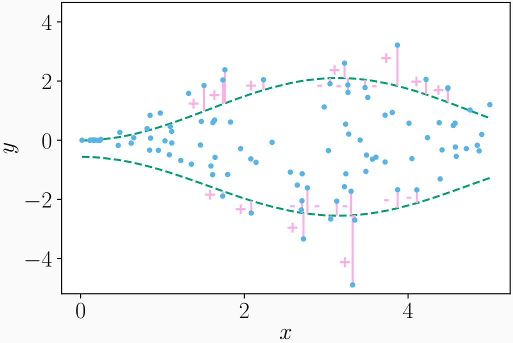
# Missing data augmentation

# Goal: validity conditionally to the mask

**Goal:** for any $m \in \mathcal{M} \subset \{0,1\}^d$:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha\left(X_{\mathrm{obs}(M)}, M\right) | M = m\right) \geq 1 - \alpha.$$

# Issue during the calibration step




- Predict with $\hat{q}_{\text{inf}}$ and $\hat{q}_{\text{sup}}$
- Get the scores $e^{(k)}$
- Compute the $(1-\alpha) \times (1 + \frac{1}{\#\text{Cal}})$ empirical quantile of the $e^{(k)}$, noted $Q_{1-\tilde{\alpha}}(e)$

# Missing data augmentation of the calibration set

Test point

| | | | |
|---|---|---|---|
| 3 | NA | NA | 1 |

Initial calibration set

| | | | | |
|---|---|---|---|---|
| $x^{(1)}$ | -1 | -10 | 6 | 1 |
| $x^{(2)}$ | 4 | NA | -2 | 2 |
| $x^{(3)}$ | 5 | 1 | 1 | NA |
| $x^{(4)}$ | 0 | NA | NA | 1 |

Calibration set used

| | | | | |
|---|---|---|---|---|
| $\tilde{x}^{(1)}$ | -1 | NA | NA | 1 |
| $\tilde{x}^{(2)}$ | 4 | NA | NA | 2 |
| $\tilde{x}^{(3)}$ | | | | |
| $\tilde{x}^{(4)}$ | 0 | NA | NA | 1 |

# Missing data augmentation of the calibration set

Test point

| | | | |
|---|---|---|---|
| 3 | NA | NA | 1 |

Initial calibration set

| | | | | |
|---|---|---|---|---|
| $x^{(1)}$ | -1 | -10 | 6 | 1 |
| $x^{(2)}$ | 4 | NA | -2 | 2 |
| $x^{(3)}$ | 5 | 1 | 1 | NA |
| $x^{(4)}$ | 0 | NA | NA | 1 |

Calibration set used

| | | | | |
|---|---|---|---|---|
| $\tilde{x}^{(1)}$ | -1 | NA | NA | 1 |
| $\tilde{x}^{(2)}$ | 4 | NA | NA | 2 |
| $\tilde{x}^{(3)}$ | | | | |
| $\tilde{x}^{(4)}$ | 0 | NA | NA | 1 |

$$e^{(k)} = \max \left\{ \hat{q}_{\text{inf}} \left( \tilde{x}^{(k)} \right) - y^{(k)}, y^{(k)} - \hat{q}_{\text{sup}} \left( \tilde{x}^{(k)} \right) \right\}$$

# CQR-MDA with exact masking in words

1. Split your training set into a `proper training set` and `calibration set`
2. Train your imputation function on the `proper training set`
3. Impute the `proper training set`
4. Train your quantile regressors on the `imputed proper training set`
5. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:

# CQR-MDA with exact masking in words

1. Split your training set into a `proper training set` and `calibration set`
2. Train your imputation function on the `proper training set`
3. Impute the `proper training set`
4. Train your quantile regressors on the `imputed proper training set`
5. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
   5.1 For each $j \in [\![1, d]\!]$ such that $m_j^{(n+1)} = 1$, set $\tilde{m}_j^{(k)} = 1$ (i.e. set $\tilde{x}_j^{(k)} = \mathtt{NA}$) for $k$ in the `calibration set` such that $m^{(k)} \subset m^{(n+1)}$

# CQR-MDA with exact masking in words

1. Split your training set into a `proper training set` and `calibration set`
2. Train your imputation function on the `proper training set`
3. Impute the `proper training set`
4. Train your quantile regressors on the `imputed proper training set`
5. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
   5.1 For each $j \in [\![1, d]\!]$ such that $m_j^{(n+1)} = 1$, set $\tilde{m}_j^{(k)} = 1$ (i.e. set $\tilde{x}_j^{(k)} = \mathtt{NA}$) for $k$ in the `calibration set` such that $m^{(k)} \subset m^{(n+1)}$
   5.2 Impute the new `calibration set`
   5.3 Compute the calibration correction
   5.4 Impute the test point
   5.5 Predict with the quantile regressors and the correction previously obtained

# Conditional validity

**Theorem (Zaffran et al., 2023)**

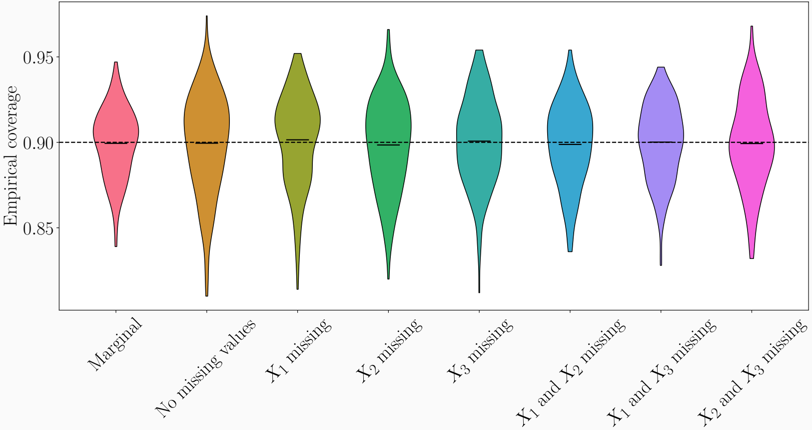
*If the data is exchangeable and MCAR, then for almost all imputation function the proposed methodology is such that for any* $m \in \mathcal{M} \subset \{0,1\}^d$*:*

$$\mathbb{P}\left(Y \in \hat{C}_\alpha\left(X_{obs(M)}, M\right) | M = m\right) \geq 1 - \alpha,$$

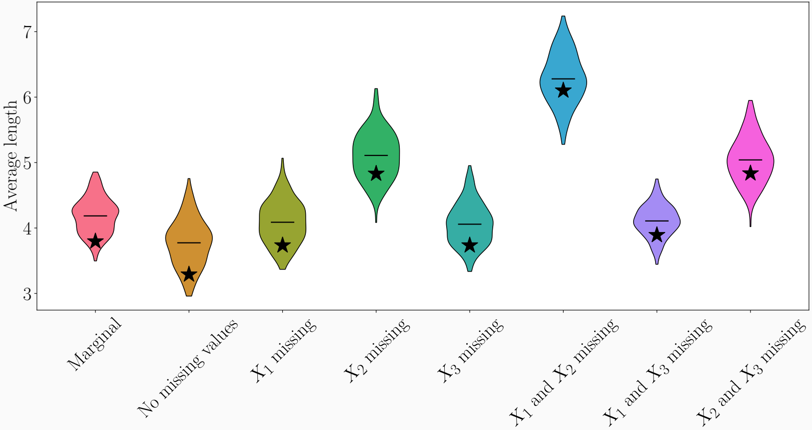
*and if additionally the scores are almost surely distinct:*

$$\mathbb{P}\left(Y \in \hat{C}_\alpha\left(X_{obs(M)}, M\right) | M = m\right) \leq 1 - \alpha + \frac{1}{1 + \#\mathrm{Cal}^{\mathrm{m}}}.$$

# Empirical coverages



# Empirical lengths



# CP-MDA-Exact reminder

Test point

| | | | |
|---|---|---|---|
| 3 | NA | NA | 1 |

Initial calibration set

| | | | | |
|---|---|---|---|---|
| $x^{(1)}$ | -1 | -10 | 6 | 1 |
| $x^{(2)}$ | 4 | NA | -2 | 2 |
| $x^{(3)}$ | 5 | 1 | 1 | NA |
| $x^{(4)}$ | 0 | NA | NA | 1 |

Calibration set used

| | | | | |
|---|---|---|---|---|
| $\tilde{x}^{(1)}$ | -1 | NA | NA | 1 |
| $\tilde{x}^{(2)}$ | 4 | NA | NA | 2 |
| $\tilde{x}^{(3)}$ | | | | |
| $\tilde{x}^{(4)}$ | 0 | NA | NA | 1 |

# What if we kept all individuals?

Test point

| | | | |
|---|---|---|---|
| 3 | NA | NA | 1 |

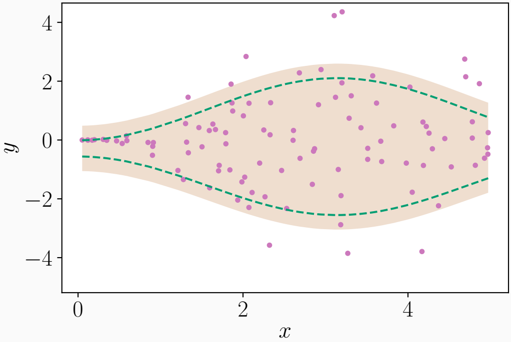
Initial calibration set

| | | | | |
|---|---|---|---|---|
| $x^{(1)}$ | -1 | -10 | 6 | 1 |
| $x^{(2)}$ | 4 | NA | -2 | 2 |
| $x^{(3)}$ | 5 | 1 | 1 | NA |
| $x^{(4)}$ | 0 | NA | NA | 1 |

Calibration set used

| | | | | |
|---|---|---|---|---|
| $\tilde{x}^{(1)}$ | -1 | NA | NA | 1 |
| $\tilde{x}^{(2)}$ | 4 | NA | NA | 2 |
| $\tilde{x}^{(3)}$ | 5 | NA | NA | NA |
| $\tilde{x}^{(4)}$ | 0 | NA | NA | 1 |

# What if we kept all individuals?




- Predict with $\hat{q}_{\mathsf{inf}}$ and $\hat{q}_{\mathsf{sup}}$
- Build $\hat{C}_{\hat{\alpha}}(x)$:

$$\left[\hat{q}_{\mathsf{inf}}(x) - Q_{1-\tilde{\alpha}}\left(e\right), \hat{q}_{\mathsf{sup}}(x) + Q_{1-\tilde{\alpha}}\left(e\right)\right]$$

# Idea: modify the test point accordingly

Test point

| | | | |
|---|---|---|---|
| 3 | NA | NA | 1 |

Initial calibration set

| | | | | |
|---|---|---|---|---|
| $x^{(1)}$ | -1 | -10 | 6 | 1 |
| $x^{(2)}$ | 4 | NA | -2 | 2 |
| $x^{(3)}$ | 5 | 1 | 1 | NA |
| $x^{(4)}$ | 0 | NA | NA | 1 |

Calibration set used

| | | | | |
|---|---|---|---|---|
| $\tilde{x}^{(1)}$ | -1 | NA | NA | 1 |
| $\tilde{x}^{(2)}$ | 4 | NA | NA | 2 |
| $\tilde{x}^{(3)}$ | 5 | NA | NA | NA |
| $\tilde{x}^{(4)}$ | 0 | NA | NA | 1 |

and

Temporary test points

| | | | |
|---|---|---|---|
| 3 | NA | NA | 1 |
| 3 | NA | NA | 1 |
| 3 | NA | NA | NA |
| 3 | NA | NA | 1 |

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
   1.1 For each $j \in [\![1, d]\!]$ such that $m_j^{(n+1)} = 1$, set $\tilde{m}_j^{(k)} = 1$ (i.e. set $\tilde{x}_j^{(k)} = \mathtt{NA}$) for $k$ in the `calibration set` such that $m^{(k)} \subset m^{(n+1)}$

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
    1.1 For each $j \in [\![1, d]\!]$ such that $m_j^{(n+1)} = 1$, set $\tilde{m}_j^{(k)} = 1$ (i.e. set $\tilde{x}_j^{(k)} = \mathtt{NA}$) for $k$ in the `calibration set`

## CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
   1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`

## CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
    1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`
    1.2 Impute the new `calibration set`

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
    1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`
    1.2 Impute the new `calibration set`
    1.3 For each augmented calibration point $k$:

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
    1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`
    1.2 Impute the new `calibration set`
    1.3 For each augmented calibration point $k$:
        1.3.1 Get its score $e^{(k)}$

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
    1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`
    1.2 Impute the new `calibration set`
    1.3 For each augmented calibration point $k$:
        1.3.1 Get its score $e^{(k)}$
        1.3.2 Impute-then-predict on the augmented test point $(x^{(n+1)}, \tilde{m}^{(k)})$, giving: $\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k})$ and $\hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k})$

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
   1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`
   1.2 Impute the new `calibration set`
   1.3 For each augmented calibration point $k$:
      1.3.1 Get its score $e^{(k)}$
      1.3.2 Impute-then-predict on the augmented test point $(x^{(n+1)}, \tilde{m}^{(k)})$, giving: $\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k})$ and $\hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k})$
      1.3.3 Compute the corrected prediction interval: $[\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k}) - e^{(k)}; \hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k}) + e^{(k)}] := \left[z_{\text{inf}}^{(k)}; z_{\text{sup}}^{(k)}\right]$

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
    1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`
    1.2 Impute the new `calibration set`
    1.3 For each augmented calibration point $k$:
        1.3.1 Get its score $e^{(k)}$
        1.3.2 Impute-then-predict on the augmented test point $(x^{(n+1)}, \tilde{m}^{(k)})$, giving: $\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k})$ and $\hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k})$
        1.3.3 Compute the corrected prediction interval: $[\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k}) - e^{(k)}; \hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k}) + e^{(k)}] := \left[z_{\text{inf}}^{(k)}; z_{\text{sup}}^{(k)}\right]$
    1.4 Compute the quantiles $Q_{\tilde{\alpha}}(\{z_{\text{inf}}^{(k)}\}_{k\in\mathrm{Cal}})$ and $Q_{1-\tilde{\alpha}}(\{z_{\text{sup}}^{(k)}\}_{k\in\mathrm{Cal}})$

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
   1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`
   1.2 Impute the new `calibration set`
   1.3 For each augmented calibration point $k$:
      1.3.1 Get its score $e^{(k)}$
      1.3.2 Impute-then-predict on the augmented test point $(x^{(n+1)}, \tilde{m}^{(k)})$, giving: $\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k})$ and $\hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k})$
      1.3.3 Compute the corrected prediction interval: $[\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k}) - e^{(k)}; \hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k}) + e^{(k)}] := \left[z_{\text{inf}}^{(k)}; z_{\text{sup}}^{(k)}\right]$
   1.4 Compute the quantiles $Q_{\tilde{\alpha}}(\{z_{\text{inf}}^{(k)}\}_{k\in\mathrm{Cal}})$ and $Q_{1-\tilde{\alpha}}(\{z_{\text{sup}}^{(k)}\}_{k\in\mathrm{Cal}})$
   1.5 Predict $[Q_{\tilde{\alpha}}(\{z_{\text{inf}}^{(k)}\}_{k\in\mathrm{Cal}}); Q_{1-\tilde{\alpha}}(\{z_{\text{sup}}^{(k)}\}_{k\in\mathrm{Cal}})]$

# CQR-MDA with nested masking in words

1. For a test point $\left(x^{(n+1)}, m^{(n+1)}\right)$:
   1.1 Set $\tilde{m}^{(k)} = \max(m^{(k)}, m^{(n+1)})$ for $k$ in the `calibration set`
   1.2 Impute the new `calibration set`
   1.3 For each augmented calibration point $k$:
      1.3.1 Get its score $e^{(k)}$
      1.3.2 Impute-then-predict on the augmented test point $(x^{(n+1)}, \tilde{m}^{(k)})$, giving: $\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k})$ and $\hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k})$
      1.3.3 Compute the corrected prediction interval: $[\hat{q}_{\text{inf}}(\tilde{x}^{(n+1),k}) - e^{(k)}; \hat{q}_{\text{sup}}(\tilde{x}^{(n+1),k}) + e^{(k)}] := \left[z_{\text{inf}}^{(k)}; z_{\text{sup}}^{(k)}\right]$
   1.4 Compute the quantiles $Q_{\tilde{\alpha}}(\{z_{\text{inf}}^{(k)}\}_{k\in\mathrm{Cal}})$ and $Q_{1-\tilde{\alpha}}(\{z_{\text{sup}}^{(k)}\}_{k\in\mathrm{Cal}})$
   1.5 Predict $[Q_{\tilde{\alpha}}(\{z_{\text{inf}}^{(k)}\}_{k\in\mathrm{Cal}}); Q_{1-\tilde{\alpha}}(\{z_{\text{sup}}^{(k)}\}_{k\in\mathrm{Cal}})]$

Test point

| | | | |
|---|---|---|---|
| 3 | NA | NA | 1 |

Initial calibration set

| | | | | |
|---|---|---|---|---|
| $x^{(1)}$ | -1 | -10 | 6 | 1 |
| $x^{(2)}$ | 4 | NA | -2 | 2 |
| $x^{(3)}$ | 5 | 1 | 1 | NA |
| $x^{(4)}$ | 0 | NA | NA | 1 |

CP-MDA with exact masking:

calibration set

| | | | | |
|---|---|---|---|---|
| $\tilde{x}^{(1)}$ | -1 | NA | NA | 1 |
| $\tilde{x}^{(2)}$ | 4 | NA | NA | 2 |
| $\tilde{x}^{(3)}$ | | | | |
| $\tilde{x}^{(4)}$ | 0 | NA | NA | 1 |

---

CP-MDA with nested masking:

calibration set

| | | | | |
|---|---|---|---|---|
| $\tilde{x}^{(1)}$ | -1 | NA | NA | 1 |
| $\tilde{x}^{(2)}$ | 4 | NA | NA | 2 |
| $\tilde{x}^{(3)}$ | 5 | NA | NA | NA |
| $\tilde{x}^{(4)}$ | 0 | NA | NA | 1 |

and

temporary test points

| | | | |
|---|---|---|---|
| 3 | NA | NA | 1 |
| 3 | NA | NA | 1 |
| 3 | NA | NA | NA |
| 3 | NA | NA | 1 |

# Towards asymptotic individualized coverage

# Consistency of a universal quantile learner after imputation

Let $\Phi$ be an imputation function chosen by the user.

Denote

$$g^*_{\beta,\Phi} \in \underset{g:\mathbb{R}^d \to \mathbb{R}}{\operatorname{argmin}} \ \mathbb{E}\left[\rho_\beta(Y - g \circ \Phi(X_{\mathrm{obs(M)}}, M))\right] := \mathcal{R}_{\beta,\phi}(g).$$

# Consistency of a universal quantile learner after imputation

Let $\Phi$ be an imputation function chosen by the user.

Denote

$$g^*_{\beta,\Phi} \in \underset{g:\mathbb{R}^d \to \mathbb{R}}{\operatorname{argmin}} \; \mathbb{E}\left[\rho_\beta(Y - g \circ \Phi(X_{\mathrm{obs(M)}}, M))\right] := \mathcal{R}_{\beta,\phi}(g).$$

Comparison with: $\underset{f}{\operatorname{argmin}} \; \mathbb{E}\left[\rho_\beta(Y - f(X_{\mathrm{obs(M)}}, M))\right]$ *(informal)*.

# Consistency of a universal quantile learner after imputation

Let $\Phi$ be an imputation function chosen by the user.

Denote

$$g^*_{\beta,\Phi} \in \underset{g:\mathbb{R}^d \to \mathbb{R}}{\operatorname{argmin}} \ \mathbb{E}\left[\rho_\beta(Y - g \circ \Phi(X_{\mathrm{obs(M)}}, M))\right] := \mathcal{R}_{\beta,\phi}(g).$$

Comparison with: $\underset{f}{\operatorname{argmin}} \ \mathbb{E}\left[\rho_\beta(Y - f(X_{\mathrm{obs(M)}}, M))\right]$ *(informal)*.

**Proposition (Pinball-consistency of an universal learner)**

For almost all $\mathcal{C}^\infty$ imputation function $\Phi$, the function $g^*_{\beta,\Phi} \circ \Phi$ is Bayes optimal for the pinball-risk of level $\beta$.

# Consistency of a universal quantile learner after imputation

Let $\Phi$ be an imputation function chosen by the user.

Denote

$$g^*_{\beta,\Phi} \in \underset{g:\mathbb{R}^d \to \mathbb{R}}{\text{argmin}} \ \mathbb{E}\left[\rho_\beta(Y - g \circ \Phi(X_{\text{obs(M)}}, M))\right] := \mathcal{R}_{\beta,\phi}(g).$$

Comparison with: $\underset{f}{\text{argmin}} \ \mathbb{E}\left[\rho_\beta(Y - f(X_{\text{obs(M)}}, M))\right]$ *(informal)*.

**Proposition (Pinball-consistency of an universal learner)**

For almost all $\mathcal{C}^\infty$ imputation function $\Phi$, the function $g^*_{\beta,\Phi} \circ \Phi$ is Bayes optimal for the pinball-risk of level $\beta$.

$\hookrightarrow$ any universally consistent algorithm for quantile regression trained on the data imputed by $\Phi$ is pinball-Bayes-consistent.

# Consistency of a universal quantile learner after imputation

Let $\Phi$ be an imputation function chosen by the user.

Denote

$$g^*_{\beta,\Phi} \in \underset{g:\mathbb{R}^d \to \mathbb{R}}{\text{argmin}} \ \mathbb{E}\left[\rho_\beta(Y - g \circ \Phi(X_{\text{obs(M)}}, M))\right] := \mathcal{R}_{\beta,\phi}(g).$$

Comparison with: $\underset{f}{\text{argmin}} \ \mathbb{E}\left[\rho_\beta(Y - f(X_{\text{obs(M)}}, M))\right]$ *(informal)*.

**Proposition (Pinball-consistency of an universal learner)**

For almost all $\mathcal{C}^\infty$ imputation function $\Phi$, the function $g^*_{\beta,\Phi} \circ \Phi$ is Bayes optimal for the pinball-risk of level $\beta$.

$\hookrightarrow$ any universally consistent algorithm for quantile regression trained on the data imputed by $\Phi$ is pinball-Bayes-consistent.

This is an extension of the result of Le Morvan et al. (2021).

# Asymptotic conditional coverage of a universal quantile learner

**Corollary**

*For any missing mechanism, for almost all $\mathcal{C}^\infty$ imputation function $\Phi$, if $F_{Y|(X_{\mathrm{obs(M)}},M)}$ is continuous, a universally consistent quantile regressor trained on the imputed data set yields asymptotic conditional coverage.*

# Asymptotic conditional coverage of a universal quantile learner

**Corollary**

*For any missing mechanism, for almost all $\mathcal{C}^\infty$ imputation function $\Phi$, if $F_{Y|(X_{\mathrm{obs(M)}},M)}$ is continuous, a universally consistent quantile regressor trained on the imputed data set yields asymptotic conditional coverage.*

$\hookrightarrow \mathbb{P}(Y \in \widehat{C}_\alpha(x) | X = x, M = m) \geq 1 - \alpha$ for any $m \in \mathcal{M}$ and any $x \in \mathbb{R}^d$, asymptotically with a super quantile learner.

# Experimental results

## Some settings

- Imputation by iterative ridge ($\sim$ conditional expectation)

# Some settings

- Imputation by iterative ridge ($\sim$ conditional expectation)
- **Concatenate the mask in the features**

# Some settings

- Imputation by iterative ridge ($\sim$ conditional expectation)
- **Concatenate the mask in the features**
- Neural network, fitted to minimize the pinball loss

## Some settings

- Imputation by iterative ridge ($\sim$ conditional expectation)
- **Concatenate the mask in the features**
- Neural network, fitted to minimize the pinball loss
- (Semi)-synthetic experiments:

## Some settings

- Imputation by iterative ridge ($\sim$ conditional expectation)
- **Concatenate the mask in the features**
- Neural network, fitted to minimize the pinball loss
- (Semi)-synthetic experiments:
  - MCAR missing values, with probability 20%

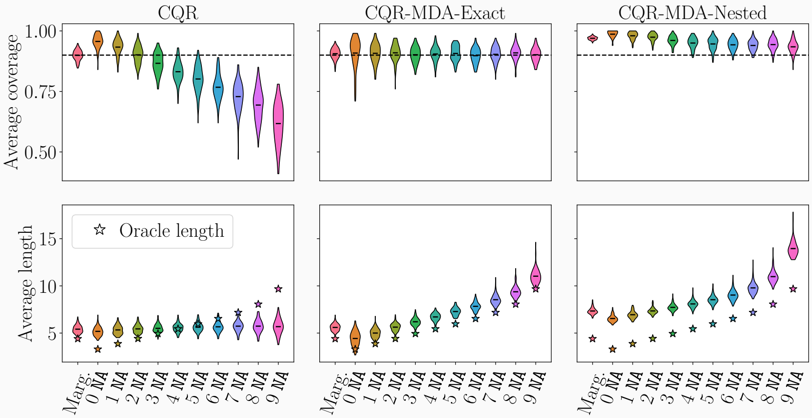
# Some settings

- Imputation by iterative ridge ($\sim$ conditional expectation)
- **Concatenate the mask in the features**
- Neural network, fitted to minimize the pinball loss
- (Semi)-synthetic experiments:
  - MCAR missing values, with probability 20%
  - 100 repetitions

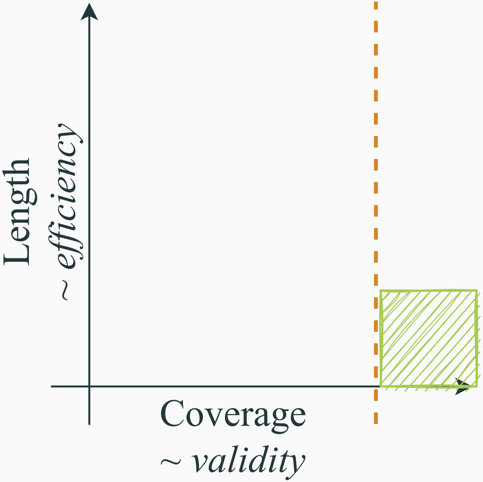
# Some settings

- Imputation by iterative ridge ($\sim$ conditional expectation)
- **Concatenate the mask in the features**
- Neural network, fitted to minimize the pinball loss
- (Semi)-synthetic experiments:
    - MCAR missing values, with probability 20%
    - 100 repetitions
    - Various test sets

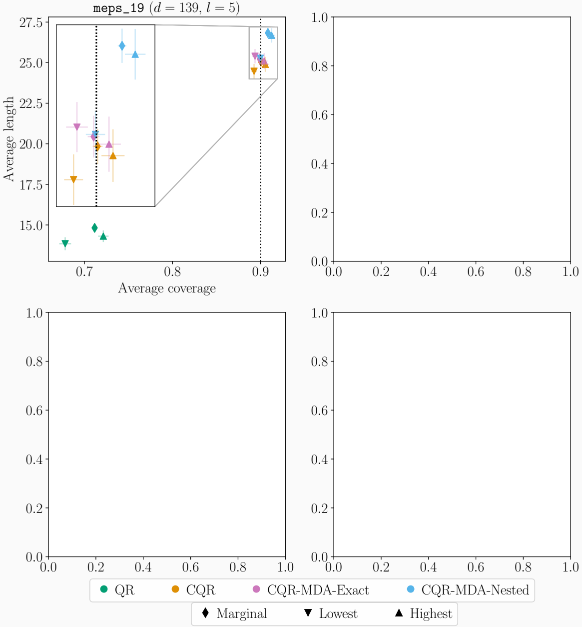
# Synthetic experiments (Gaussian linear model, $d = 10$)



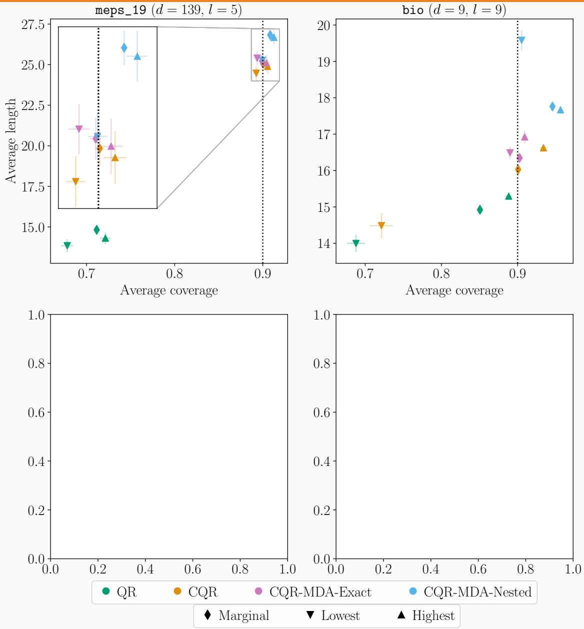
# Before more experiments, visualisation



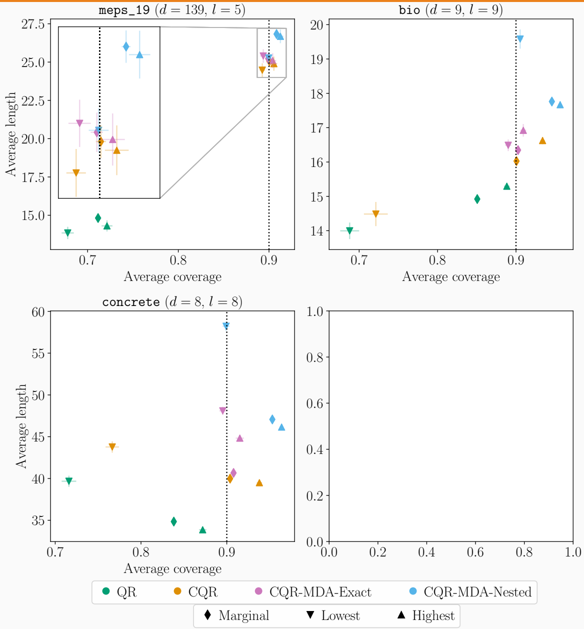
# Semi-synthetic experiments



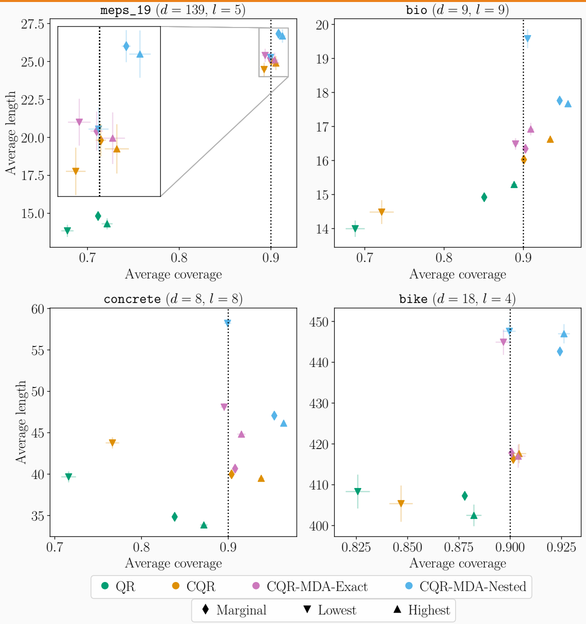
# Semi-synthetic experiments



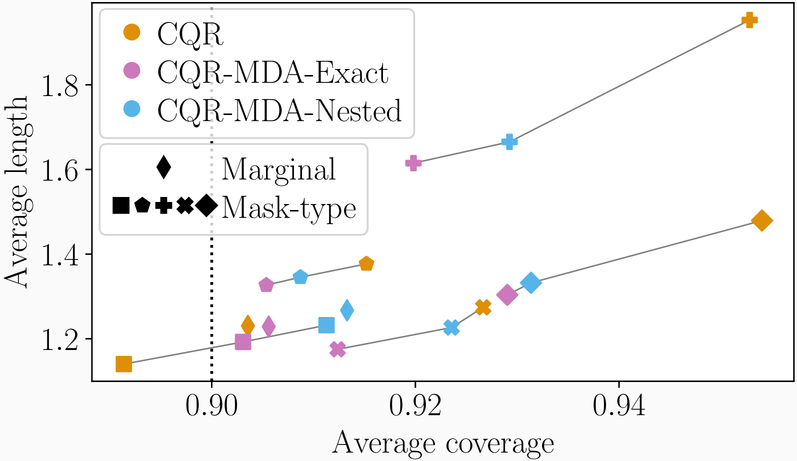
# Semi-synthetic experiments



# Semi-synthetic experiments



# Real data experiment: back to critical care medicine



# Conclusion

## Take-home-messages

- CP marginal guarantees hold on the imputed data set.

# Take-home-messages

- CP marginal guarantees hold on the imputed data set.
- Missingness introduces additional heteroskedasticity, creating a need for quantile regression based methods.

## Take-home-messages

- CP marginal guarantees hold on the imputed data set.
- Missingness introduces additional heteroskedasticity, creating a need for quantile regression based methods.
- CQR fails to attain coverage conditional on the missing pattern.

# Take-home-messages

- CP marginal guarantees hold on the imputed data set.
- Missingness introduces additional heteroskedasticity, creating a need for quantile regression based methods.
- CQR fails to attain coverage conditional on the missing pattern.
- **Missing data augmentation is the first method to output predictive intervals with missing values**.

## Take-home-messages

- CP marginal guarantees hold on the imputed data set.
- Missingness introduces additional heteroskedasticity, creating a need for quantile regression based methods.
- CQR fails to attain coverage conditional on the missing pattern.
- **Missing data augmentation is the first method to output predictive intervals with missing values**.
- Missing data augmentation attains conditional coverage with respect to the missing pattern (in MCAR setting).

**Thank you!**

Jiang, W., Bogdan, M., Josse, J., Majewski, S., Miasojedow, B., Ročková, V., and TraumaBase® Group (2022). Adaptive bayesian slope: Model selection with incomplete data. *Journal of Computational and Graphical Statistics*, 31(1):113–137.

Le Morvan, M., Josse, J., Scornet, E., and Varoquaux, G. (2021). What's a good imputation to predict with missing values? *NeurIPS*.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. (2002). Inductive Confidence Machines for Regression. In *Machine Learning: ECML 2002*. Springer.

Romano, Y., Patterson, E., and Candès, E. (2019). Conformalized Quantile Regression. *NeurIPS*.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer US.

Zhu, Z., Wang, T., and Samworth, R. J. (2019). High-dimensional principal component analysis with heterogeneous missingness.

$d = 3$

# Data generation

$(X, Y) \in \mathbb{R}^3 \times \mathbb{R}$.

$Y = \beta X + \varepsilon$

with $\varepsilon \sim \mathcal{N}(0, 1)$, $\beta = (1, 2, -1)$ and

$$(X_1, X_2, X_3) \sim \mathcal{N}\left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 1 \end{pmatrix}\right).$$

# Data generation

$(X, Y) \in \mathbb{R}^3 \times \mathbb{R}$.

$Y = \beta X + \varepsilon$

with $\varepsilon \sim \mathcal{N}(0, 1)$, $\beta = (1, 2, -1)$ and

$$(X_1, X_2, X_3) \sim \mathcal{N}\left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 1 \end{pmatrix}\right).$$

All components of $X$ each have a probability 0.2 of being missing, Completely At Random.

# Simulation settings

- Method: CQR
- Basemodel: neural network
- 200 repetitions
  - `train size` of 250 points
  - `calibration size` of 250 points
  - `test size` of 2000 points

## $d = 10$, with missing data augmentation

# Data generation

$(X, Y) \in \mathbb{R}^{10} \times \mathbb{R}$.

$Y = \beta X + \varepsilon$

with $\varepsilon \sim \mathcal{N}(0,1)$, $\beta = (1, 2, -1, 3, -0.5, -1, 0.3, 1.7, 0.4, -0.3)$

and $$(X_1, \cdots, X_{10}) \sim \mathcal{N}\left(\begin{pmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{pmatrix}, \begin{pmatrix} 1 & 0.8 & \cdots & 0.8 \\ 0.8 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0.8 \\ 0.8 & \cdots & 0.8 & 1 \end{pmatrix}\right).$$

# Data generation

$(X, Y) \in \mathbb{R}^{10} \times \mathbb{R}$.

$Y = \beta X + \varepsilon$

with $\varepsilon \sim \mathcal{N}(0, 1)$, $\beta = (1, 2, -1, 3, -0.5, -1, 0.3, 1.7, 0.4, -0.3)$

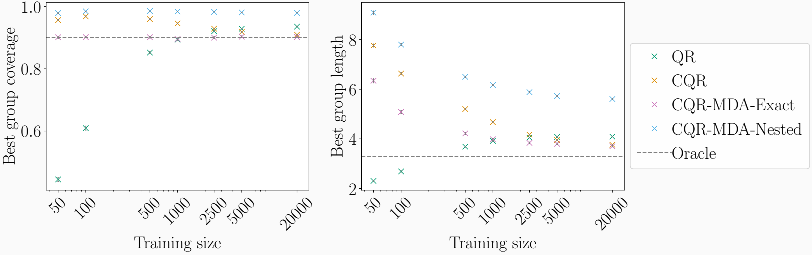
and $(X_1, \cdots, X_{10}) \sim \mathcal{N}\left(\begin{pmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{pmatrix}, \begin{pmatrix} 1 & 0.8 & \cdots & 0.8 \\ 0.8 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0.8 \\ 0.8 & \cdots & 0.8 & 1 \end{pmatrix}\right)$.

All components of $X$ each have a probability 0.2 of being missing, Completely At Random.

# Simulation settings

- Method: CQR
- Basemodel: neural network
- Imputation: iterative ($\approx$ conditional expectation)
- Mask as features: yes
- 100 repetitions
  - `train size` varies
  - `calibration size` of 1000 points
  - `test size` of 2000 points

# Results on the best group



# TraumaBase

# Data set description i

- `Age`: the age of the patient (no missing values);
- `Lactate`: the conjugate base of lactic acid, upon arrival at the hospital (17.66% missing values);
- `Delta_hemo`: the difference between the hemoglobin upon arrival at hospital and the one in the ambulance (23.82% missing values);
- `VE`: binary variable indicating if a Volume Expander was applied in the ambulance. A volume expander is a type of intravenous therapy that has the function of providing volume for the circulatory system (2.46% missing values);
- `RBC`: a binary index which indicates whether the transfusion of Red Blood Cells Concentrates is performed (0.37% missing values);

# Data set description ii

- `SI`: the shock index. It indicates the level of occult shock based on heart rate (HR) and systolic blood pressure (SBP), that is $\mathsf{SI} = \frac{\mathsf{HR}}{\mathsf{SBP}}$, upon arrival at hospital (2.09% missing values);
- `HR`: the heart rate measured upon arrival of hospital (1.62% missing values).