Prévision probabiliste et génération de trajectoires

Jean Thorey, équipe Data Science IA

16/11/2023

PRODUCTION

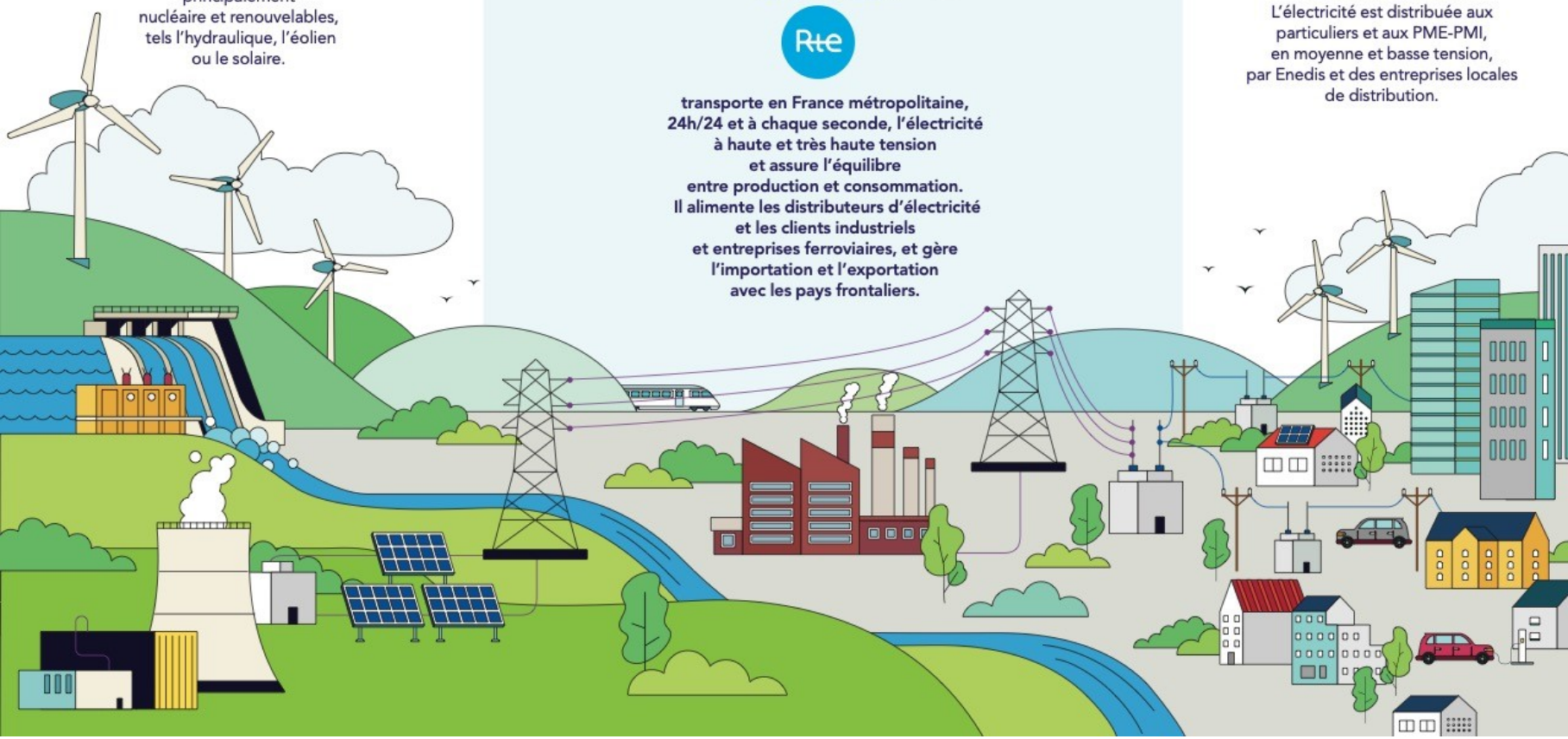
L'électricité est produite par différentes sources d'énergie, principalement nucléaire et renouvelables, tels l'hydraulique, l'éolien ou le solaire.

TRANSPORT

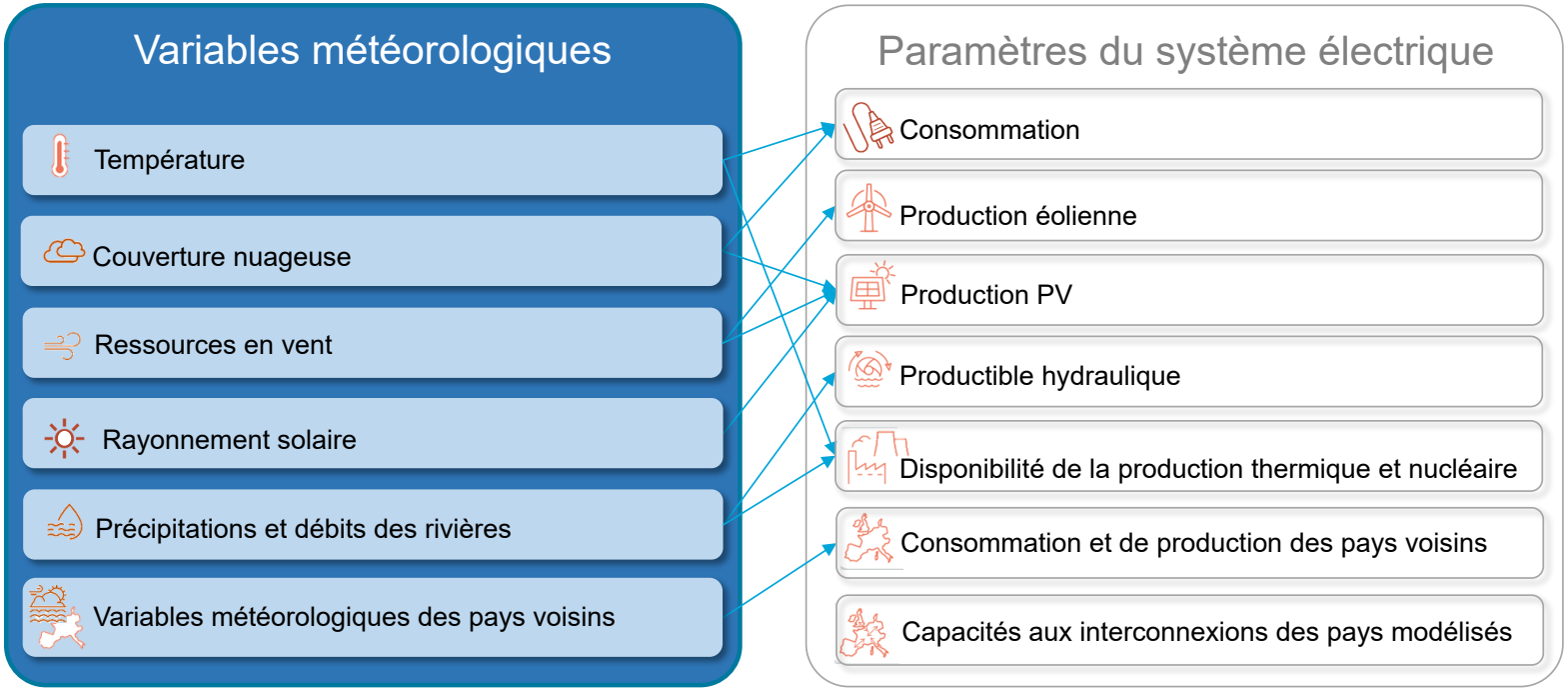
transporte en France métropolitaine, 24h/24 et à chaque seconde, l'électricité à haute et très haute tension et assure l'équilibre entre production et consommation. Il alimente les distributeurs d'électricité et les clients industriels et entreprises ferroviaires, et gère l'importation et l'exportation avec les pays frontaliers.

DISTRIBUTION

L'électricité est distribuée aux particuliers et aux PME-PMI, en moyenne et basse tension, par Enedis et des entreprises locales de distribution.



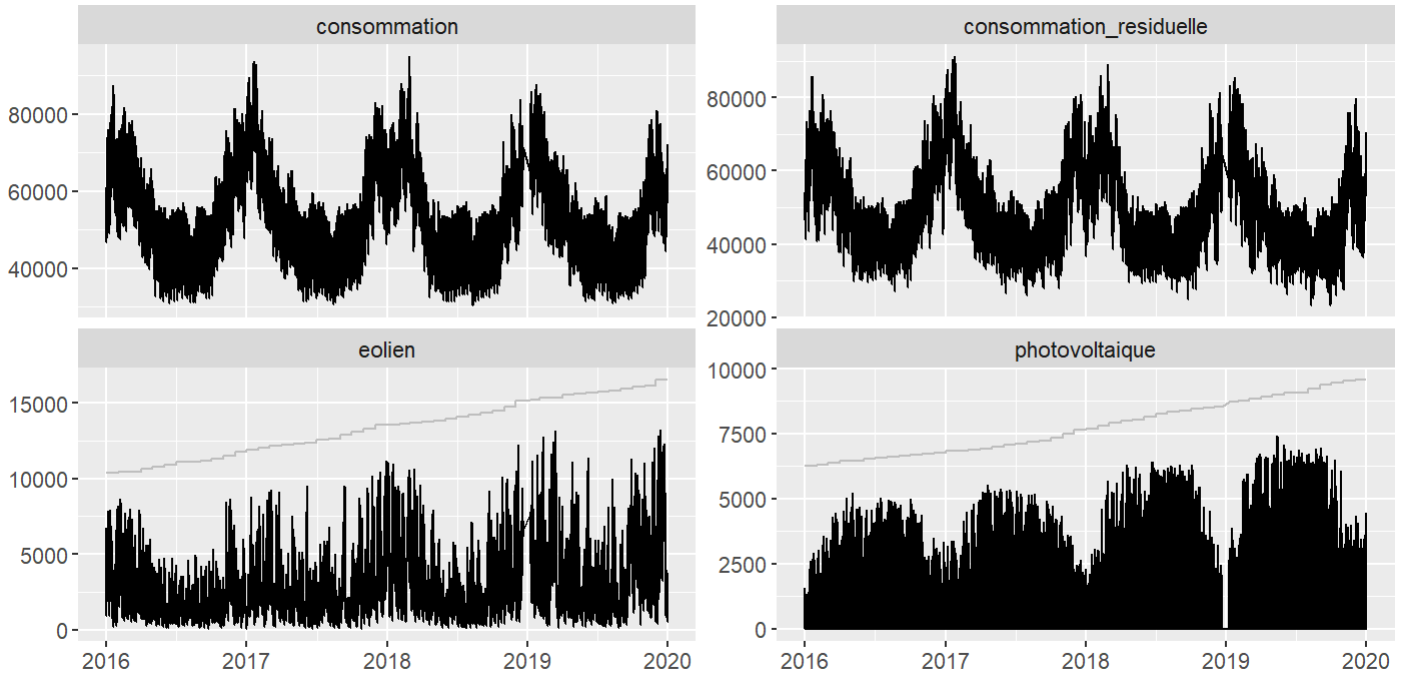
Équilibre offre/demande et météo



La prévision des marges

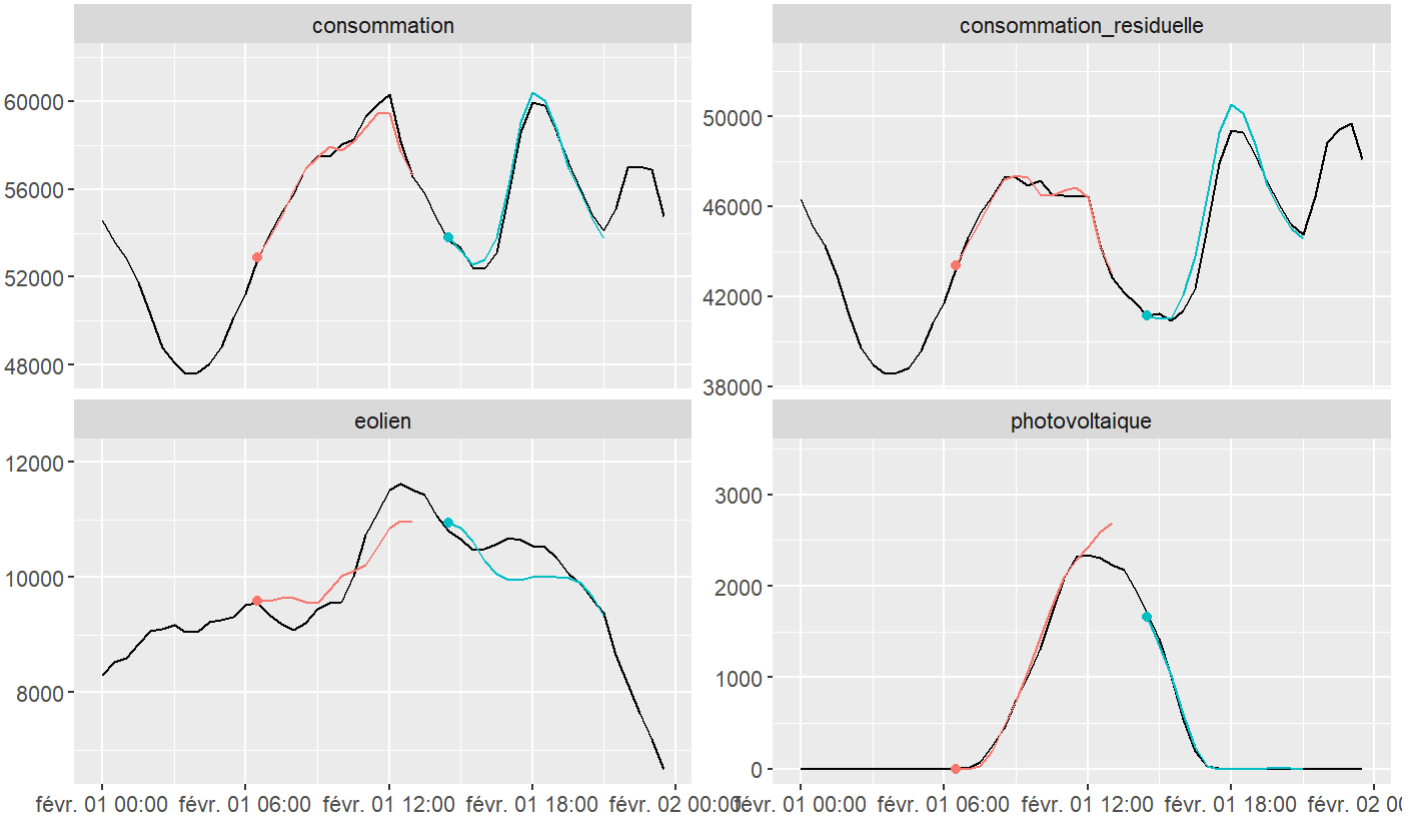
Notre cas : prévision lancée toutes les heures pour +30min/1h/1h30/.../7h sur

- Eolien
- PV
- Consommation
- **Consommation résiduelle = consommation - PV - éolien**



Prévisions déterministes

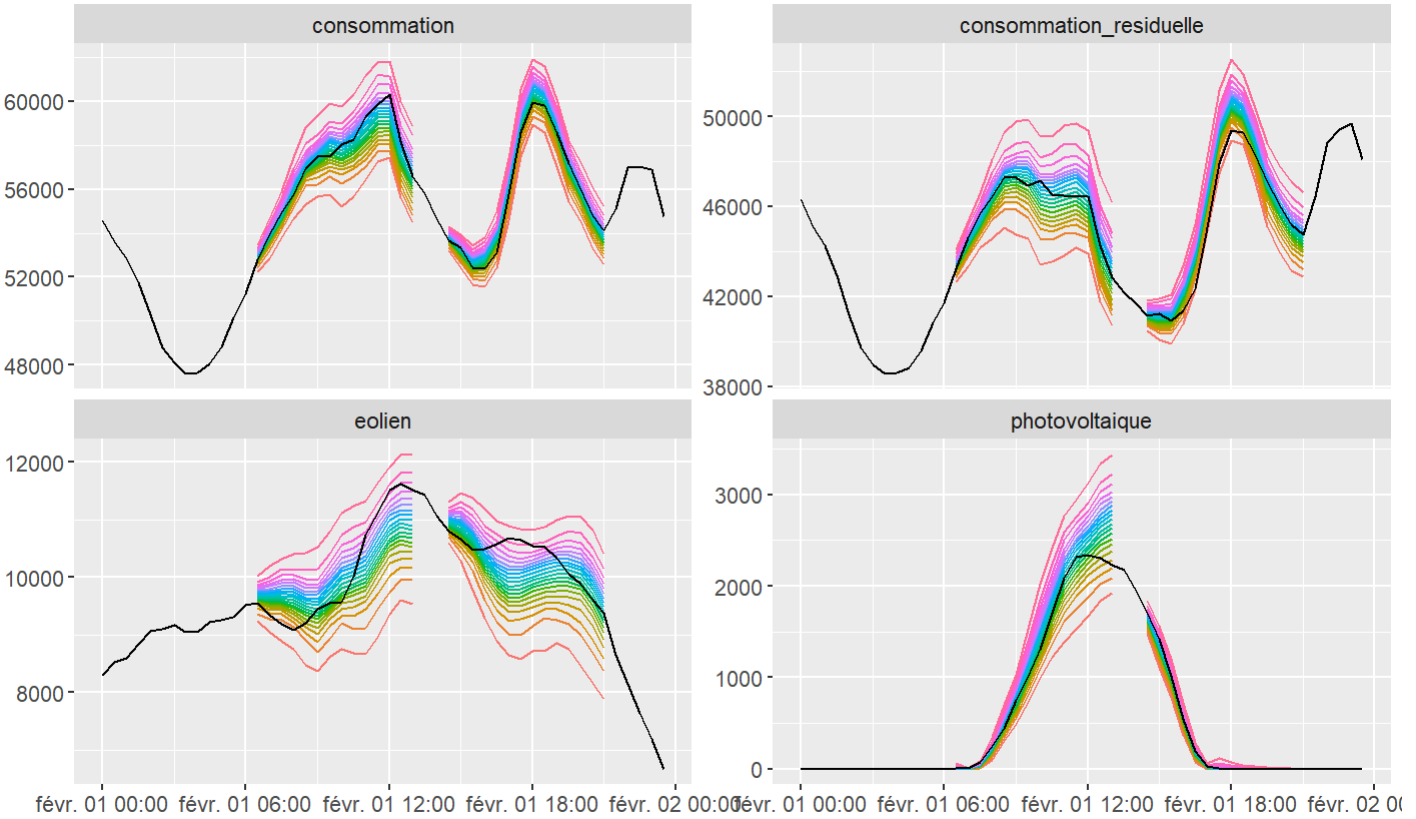
Observations + prévision déterministe : 2020-02-01.



Variables explicatives : prévision météo + observations récentes + données calendaires.

Prévisions probabilistes

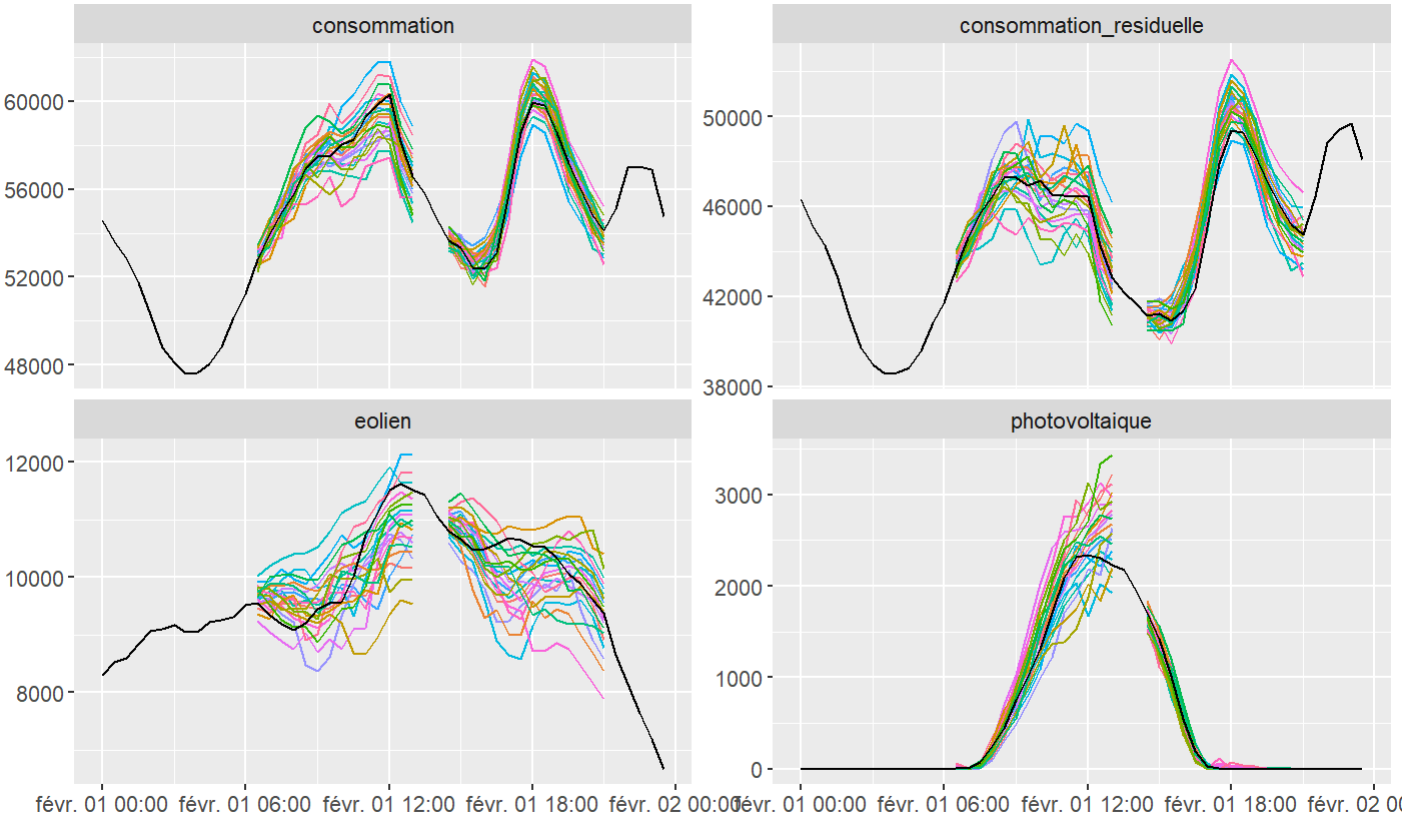
Observations + prévision quantile : 2020-02-01.



Variables explicatives : prévision déterministe + données calendaires.

Prévisions probabilistes

Observations + prévision trajectoire : 2020-02-01.



Variables explicatives : prévision déterministe + données calendaires.

Pour cette présentation

Un peu de théorie sur la prévision probabiliste. Orienté météo/climat.

En pratique : prévision probabiliste

- Statistiques : GAM multiparamètres, QGAM.
- IA : réseaux de neurones (NN).

En pratique : génération de trajectoires

- Statistiques : copules empiriques.
- IA : réseaux de neurones.

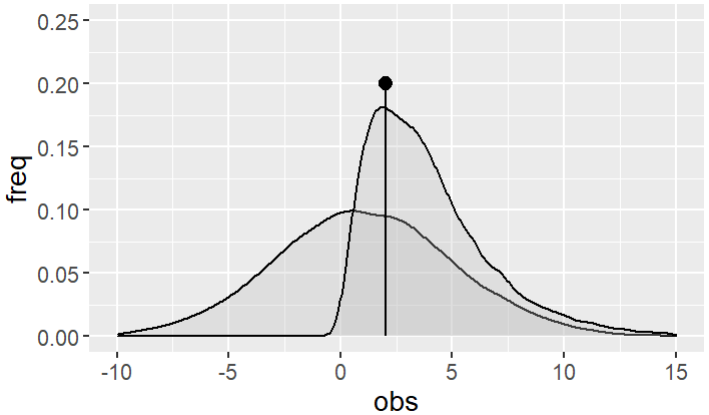
Non abordé :

- prévision conforme, boosting et forêts.
- GAN, Normalizing flows, Diffusion models.
- quantiles extrêmes (2.5%<q<97.5%).

$\rightarrow$ Un aperçu de ce qu'il est raisonnablement facile d'obtenir, sans optimisation intensive d'hyperparamètres ou d'architecture.

Prévision probabiliste

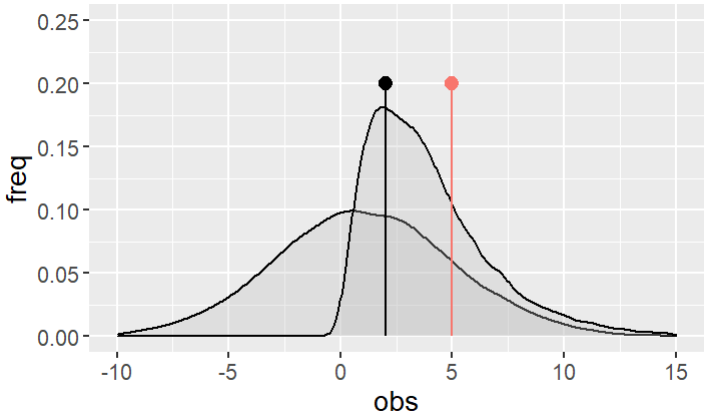
Quelle est la meilleure des 3 distributions prévues ?



Comment on valide une distribution (prévue) avec une observation unique ?

Prévision probabiliste

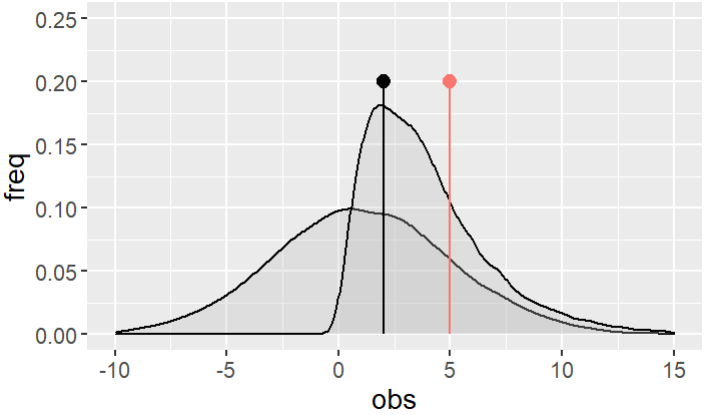
Quelle est la meilleure des 3 distributions prévues pour l'obs y = 5 ?



Comment on valide une distribution (prévue) avec une observation unique ?

Prévision probabiliste

Quelle est la meilleure des 3 distributions prévues pour l'obs y = 5 ?



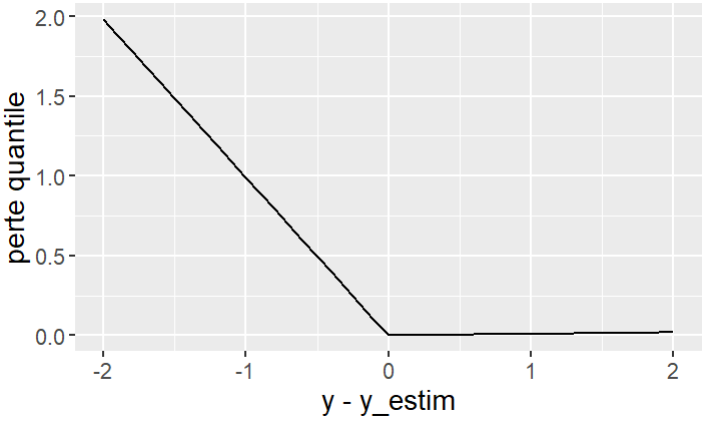
Comment on valide une distribution (prévue) avec une observation unique ? avec plusieurs pas de temps.

- **Score ponctuel** : perte quantile, CRPS...
- **Fiabilité/calibration** : cette semaine, on prévoit $P(conso < 50) = 0.9$. Problème de fiabilité si la conso est toujours inférieure à 50 GW.
- **Relation incertitude-erreur** (spread-skill) : les prévisions avec une incertitude élevée doivent correspondre aux larges erreurs.
- **Résolution/discrimination** : incertitude faible (à 50 MW près svp...).

> *"maximize sharpness subject to calibration."*

Bonus biblio sur l'évaluation des prévisions dynamiques : Thèse Thibault Modeste.

Perte quantile de niveau $\alpha \in [0, 1]$

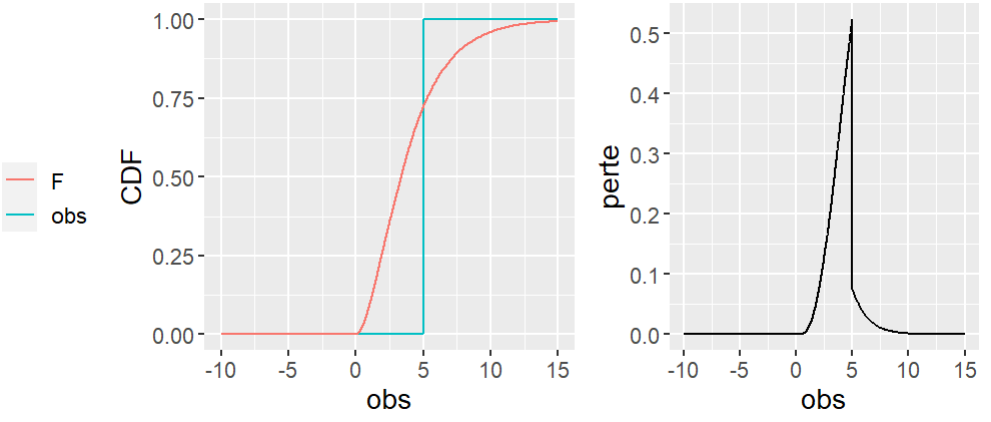


$$\rho_\alpha(y, \hat{y}_\alpha) = (\hat{y}_\alpha - y) \times (\mathbb{I}_{(y<\hat{y}_\alpha)} - \alpha)$$

Pour le quantile 99%, c'est 99 fois plus coûteux d'être inférieur à l'observation plutôt que supérieur.

Propriété : $E_y[\rho_\alpha(y, \hat{y}_\alpha)]$ est minimisée par le quantile d'ordre α de la distribution de y.

Continuous ranked probability score (CRPS)



CRPS = score quantile moyen = $\int \rho_\alpha(y, F^{-1}(\alpha))d\alpha = \int (H_y - F)^2$

- Formules analytiques pour certaines distributions paramétriques.
- Ecriture à noyau : $CRPS(y, F) = E(|X - y|) - \frac{1}{2}E(|X - X'|)$

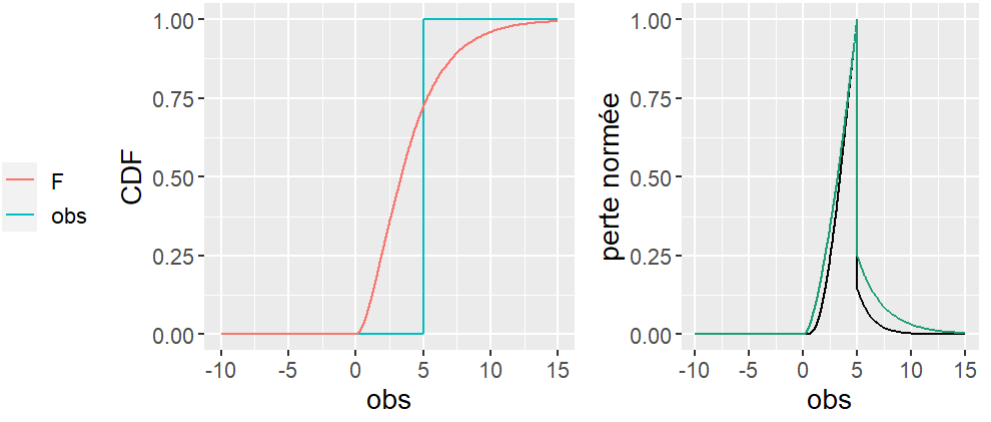
Score d'énergie :

- généralisation multivariée du CRPS : $E(||X - y||) - \frac{1}{2}E(||X - X'||)$
- écart quadratique entre les fonctions caractéristiques [Székely 2013].
- **permet de donner un score à un ensemble de trajectoires.**

> *"strictly proper scoring rules."*

$E_Y(CRPS(Y, F))$ est minimisée si $Y \sim F$.

Continuous ranked probability score (CRPS)



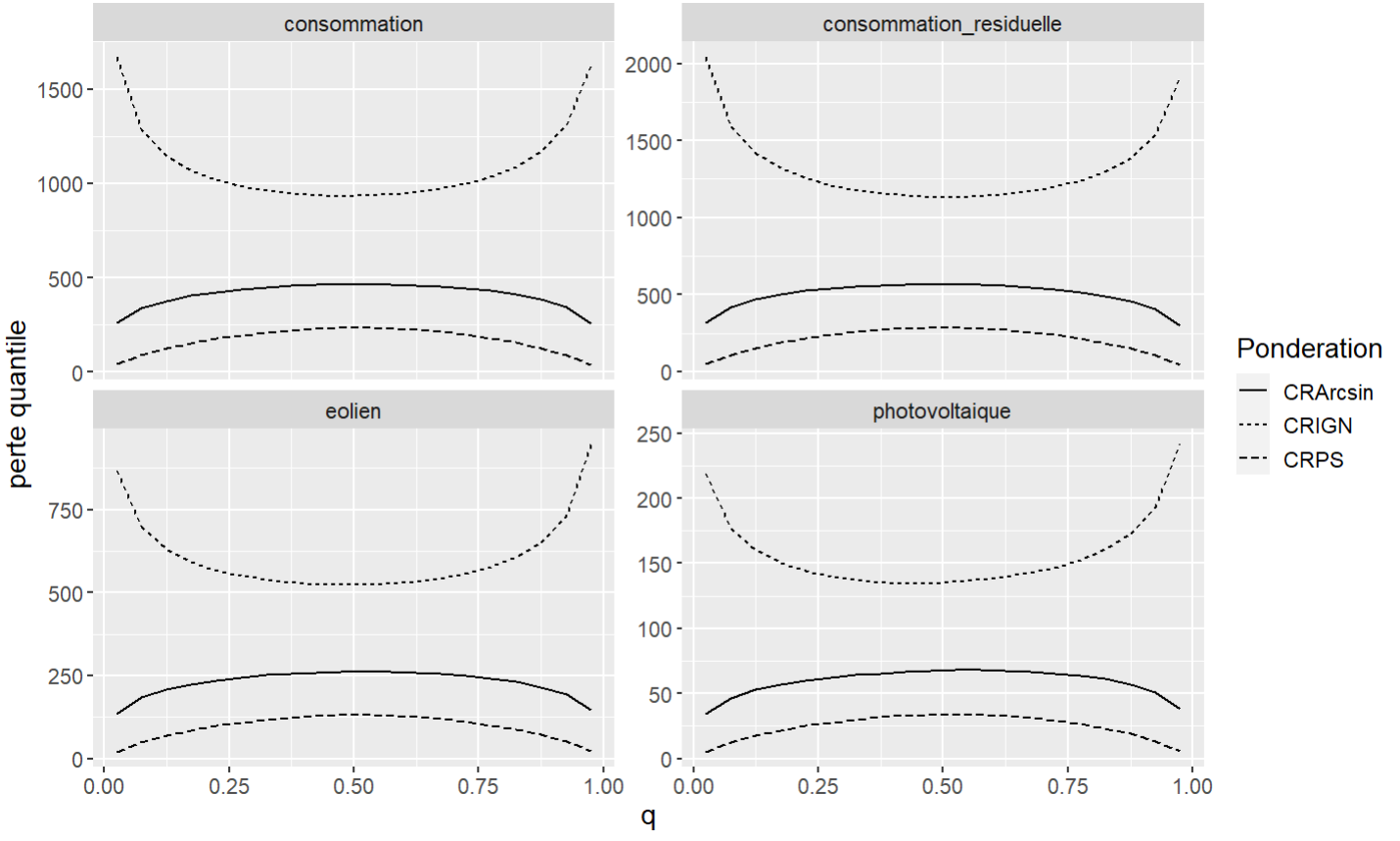
CRPS = score quantile moyen = $\int \rho_\alpha(y, F^{-1}(\alpha))d\alpha = \int (H_y - F)^2$

CRIGN = score quantile pondéré $\frac{1}{\alpha(1-\alpha)}$: $\int \frac{\rho_\alpha(y,F^{-1}(\alpha))}{\alpha(1-\alpha)}d\alpha = -\int H_y ln(F) + (1-H_y)ln(1-F)$

- Généralisations $\alpha^c(1-\alpha)^c$ [Buja 2005, Ehm 2016]
- Score local $S(f(y))$ vs. non local $S(F, y)$.
- Utilité en validation et/ou en apprentissage ?

Continuous ranked probability score (CRPS)

Pondération des quantiles : CRPS (1) vs. CRIGN $\frac{1}{\alpha(1-\alpha)}$ vs. CRArcsin $\frac{1}{\sqrt{\alpha(1-\alpha)}}$



GAM mgcv::gam

Modèle Additif Généralisé = régression linéaire + base de splines

$$y = g_1(x_1) + g_2(x_2) + g_3(x_3, x_4) + \cdots + \varepsilon\,, \qquad \text{avec } g_i(x_i) = \sum_{p=1}^{k} \beta_p^i b_p^i(x_i)$$

Résolution = maximum de vraisemblance + régularisation + calibration γ

Cas gaussien : $\widehat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_i (y_i - \hat{y}_i)^2 + \gamma \int (g'')^2$

Cas général : $\widehat{\beta} = \underset{\beta}{\operatorname{argmax}} \log p(y|\beta) - \mathrm{Pen}(\beta|\gamma)$

GAM mgcv::gam multiparamètres

Modélisation complète d'une distribution gaussienne :

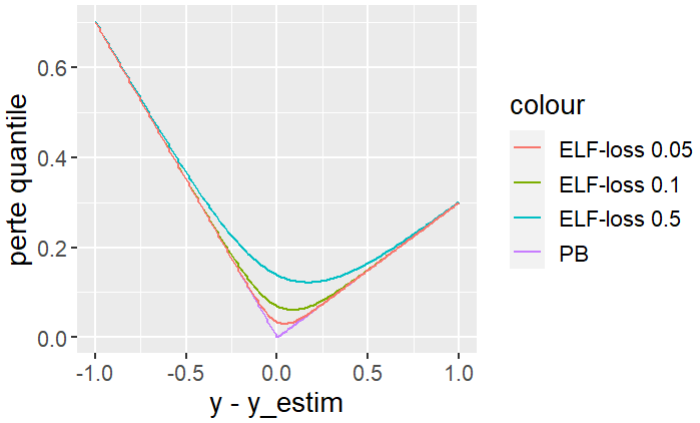
- $\mu(x) = m_1(x_1) + m_2(x_2)$
- $\log(\sigma(x)) = z_1(x_1)$

```
gam(list(y~s(x1)+s(x2), ~s(x1)), family = gaulss, data = dat)
```

Familles : Cox, Gamma, Gauss, GEV, Gumbel, Sinh-arcsinh...

GAM qgam::qgam

QGAM = GAM + perte quantile lissée ρ + distribution spécifique $\propto e^{-\rho}$



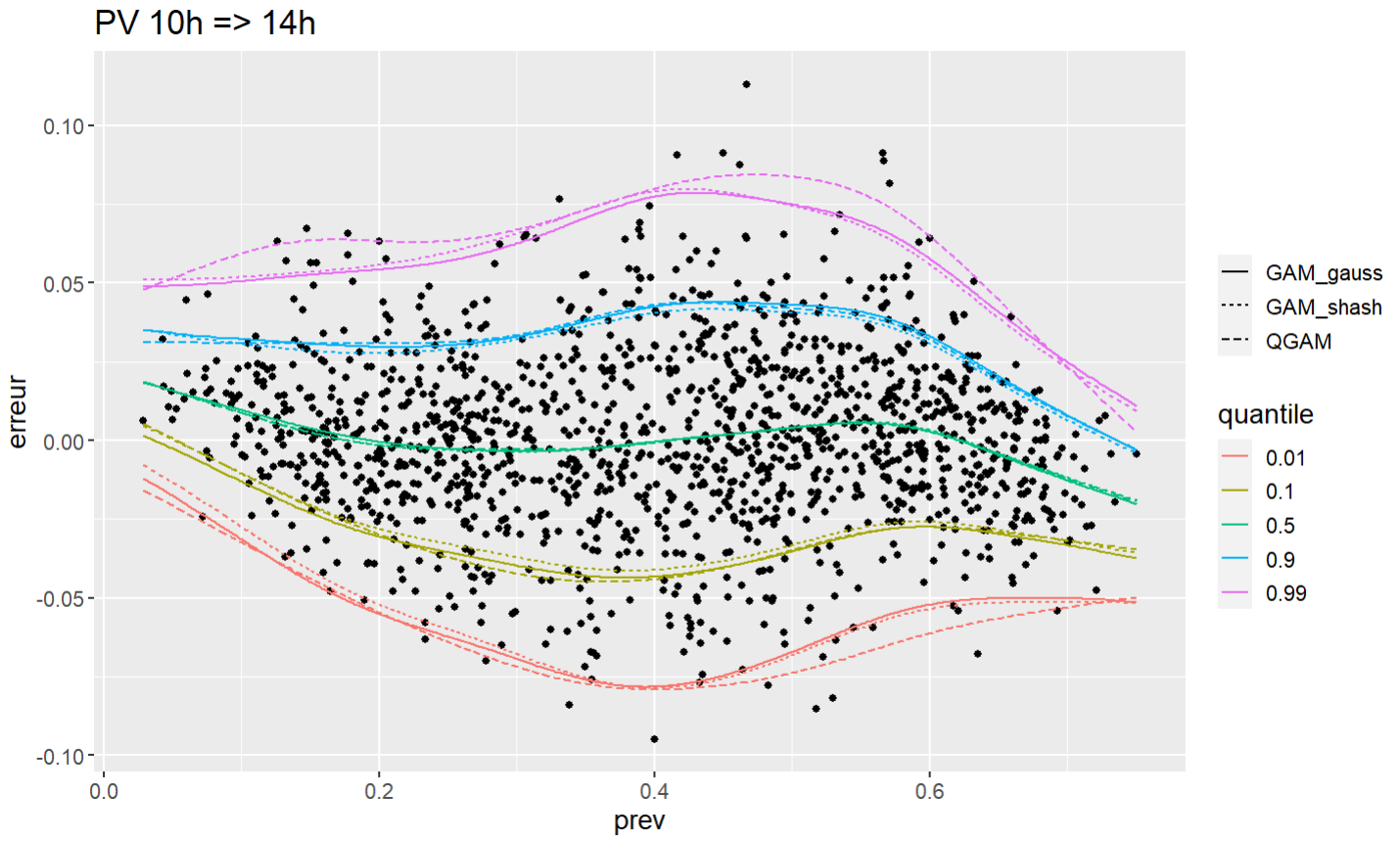

```
gam(y~s(x1)+s(x2)+s(x3, x4), family = elf(co = 0.1, qu = 0.9), data = dat)

qgam(y~s(x1)+s(x2)+s(x3, x4), qu = 0.9, data = dat)
```

- Calibration de γ et du lissage de la perte quantile.

$\rightarrow$ qGAM un modèle par quantile vs. GAM multiparamètres.

GAM : qGAM et multiparamètres

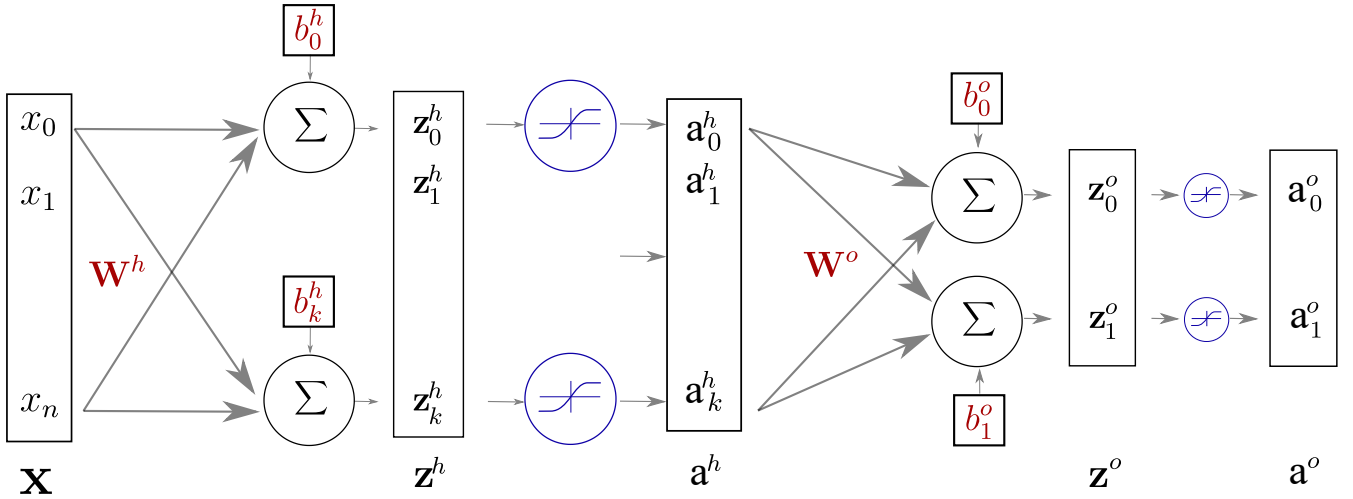
1 modèle par heure et par échéance sur les erreurs de prévision.



- Simplification du modèle pour les graphiques (pas de position dans l'année).

→ Facile à prendre en main.

Réseaux de neurones



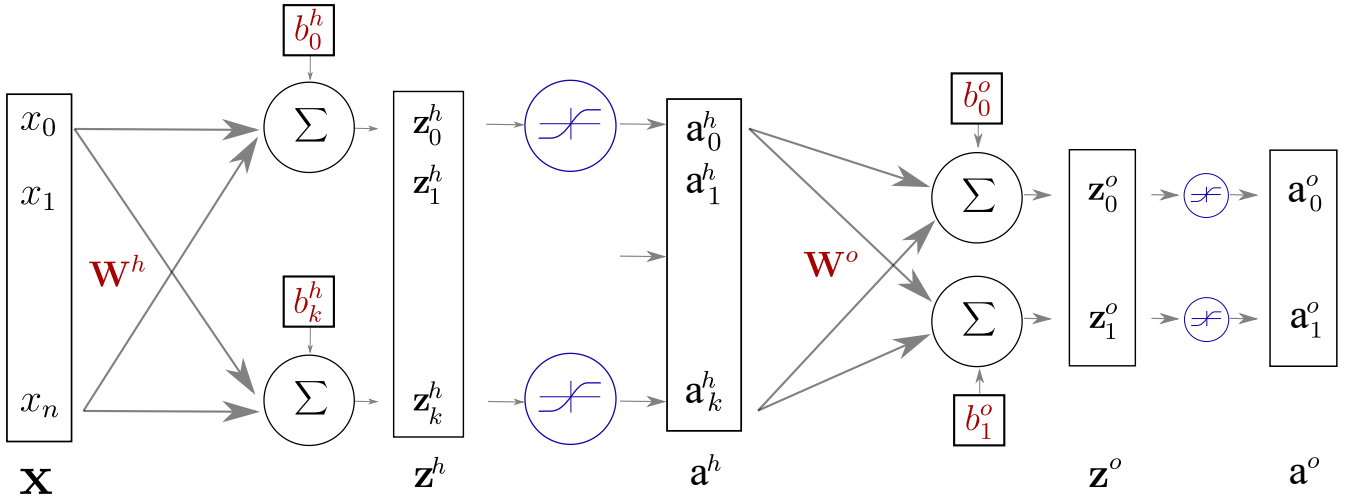
Sortie au choix :

- multivariée : 1 sortie par quantile.
- distributionnelle : sortie = μ, σ, κ.

Perte au choix :

- logarithmique : $-log(p(y))$ pour les distributions continues.
- CRPS, écriture analytique pour les distributions (mais pas pour le CRIGN).
- somme de pertes quantiles arbitrairement choisies et pondérées (CRPS et CRIGN).

Réseaux de neurones



3 variantes :

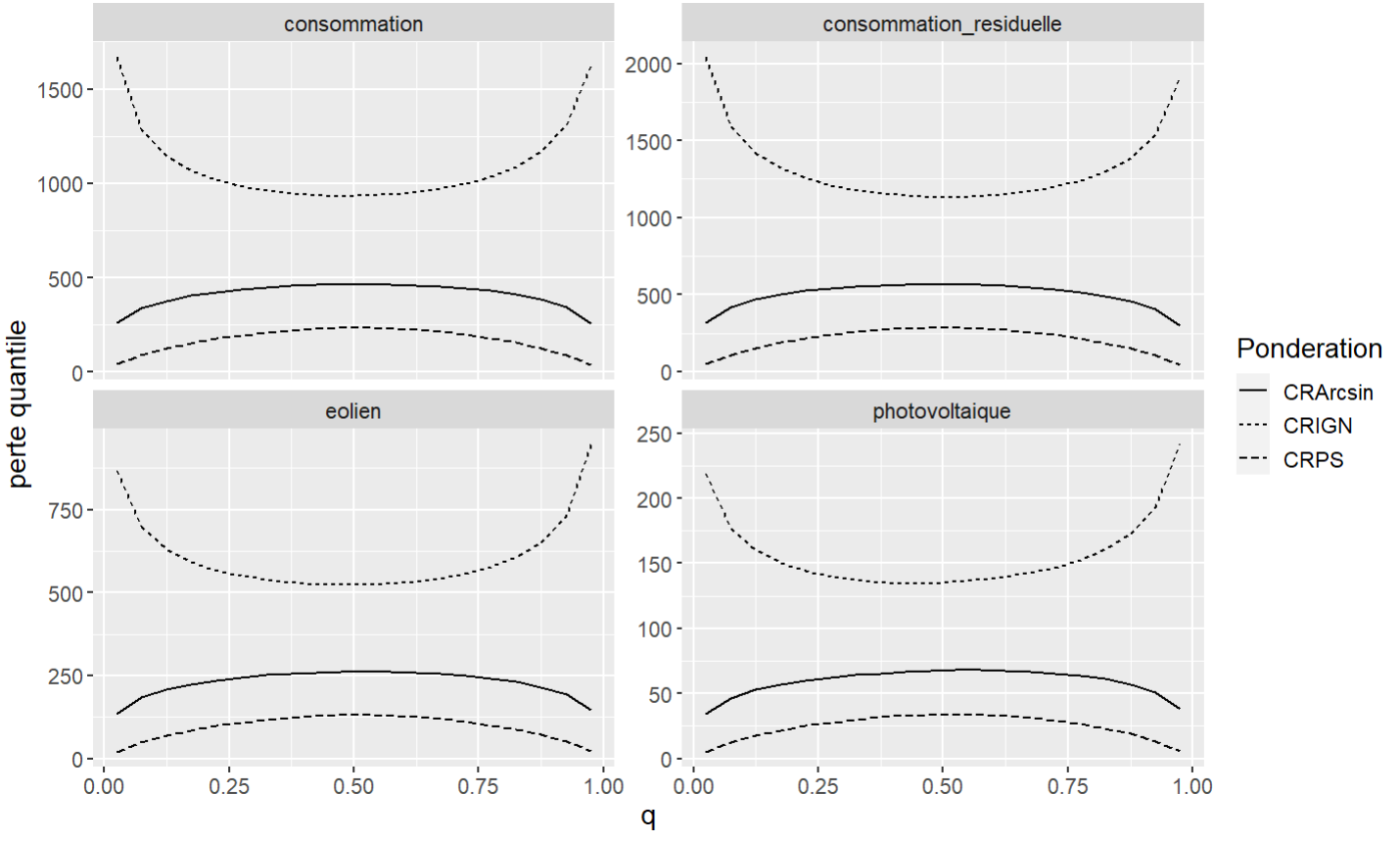
- sortie multiquantiles + CRIGN
- sortie distributionnelle (JSU) + CRIGN
- sortie distributionnelle (JSU) + log score

En pratique : un seul réseau de neurones de sortie taille $4 \times 14 \times N_p$, avec N_p le nb de paramètres des distributions ou le nombre de quantiles.

Implémentation : tensorflow, keras.

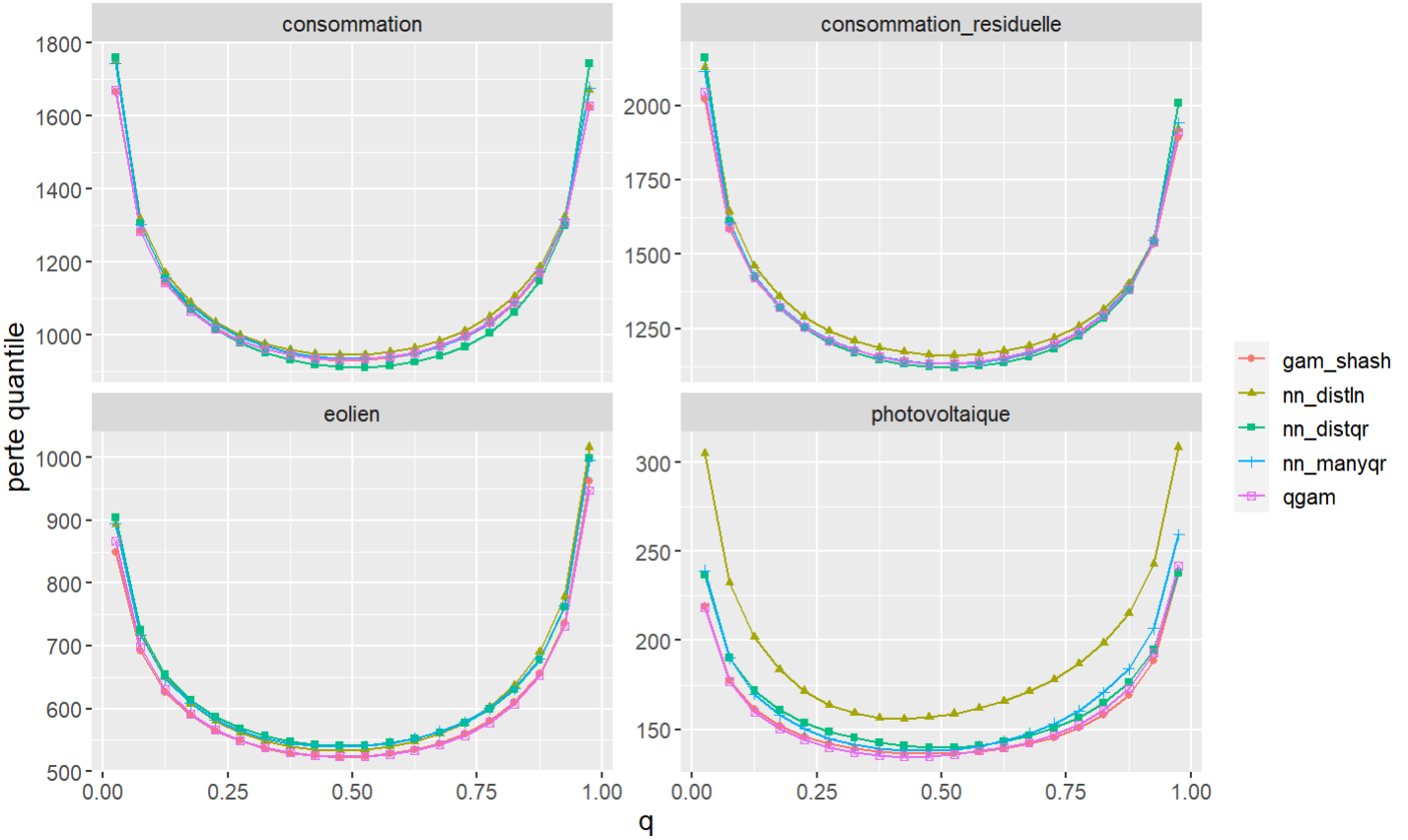
Résultats

Pondération des quantiles : CRPS (1) vs. CRIGN $\frac{1}{\alpha(1-\alpha)}$ vs. CRArcsin $\frac{1}{\sqrt{\alpha(1-\alpha)}}$



Résultats

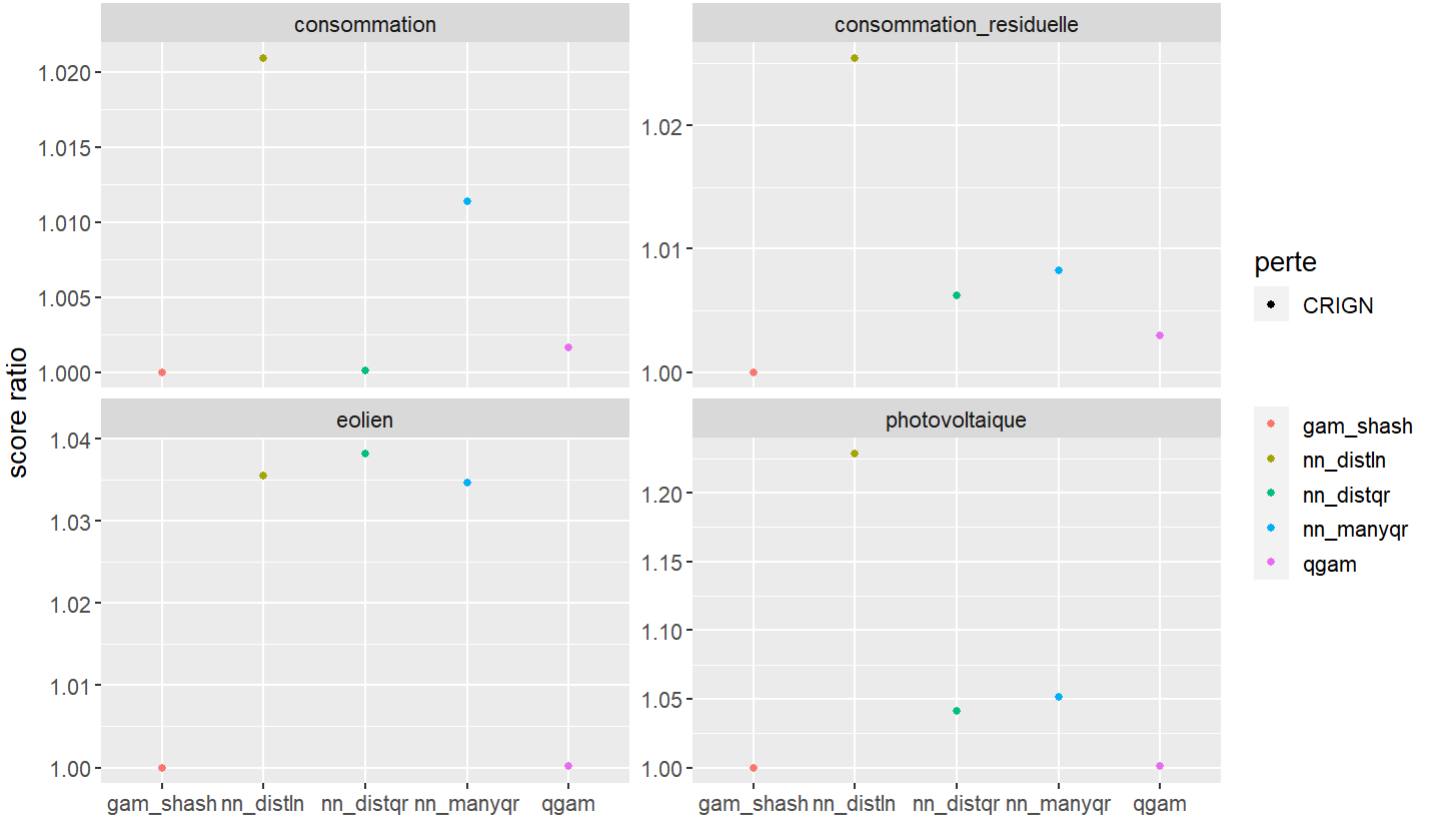
Pondération des quantiles : CRIGN $\frac{1}{\alpha(1-\alpha)}$



- Quelques ratés mais des performances comparables.

Résultats

Pondération des quantiles : CRIGN $\frac{1}{\alpha(1-\alpha)}$



- Quelques ratés mais des performances comparables.
- La métrique de validation change peu le classement.

En résumé pour la prévision probabiliste

- qGAM : perte quantile lissée / modèles par quantile
- GAM multiparamètres : maximum de vraisemblance
- NN :
  - sortie multiquantiles + CRIGN
  - sortie distributionnelle (JSU) + CRIGN
  - sortie distributionnelle (JSU) + log score

IA vs. Statistique :

- GAM : Lisibilité et performance pour une tâche précise. Choix des interactions, bases de splines.
- NN : Multivarié facilement accessible. Beaucoup plus de paramètres : dimensions, régularisation, type de couches, activation...

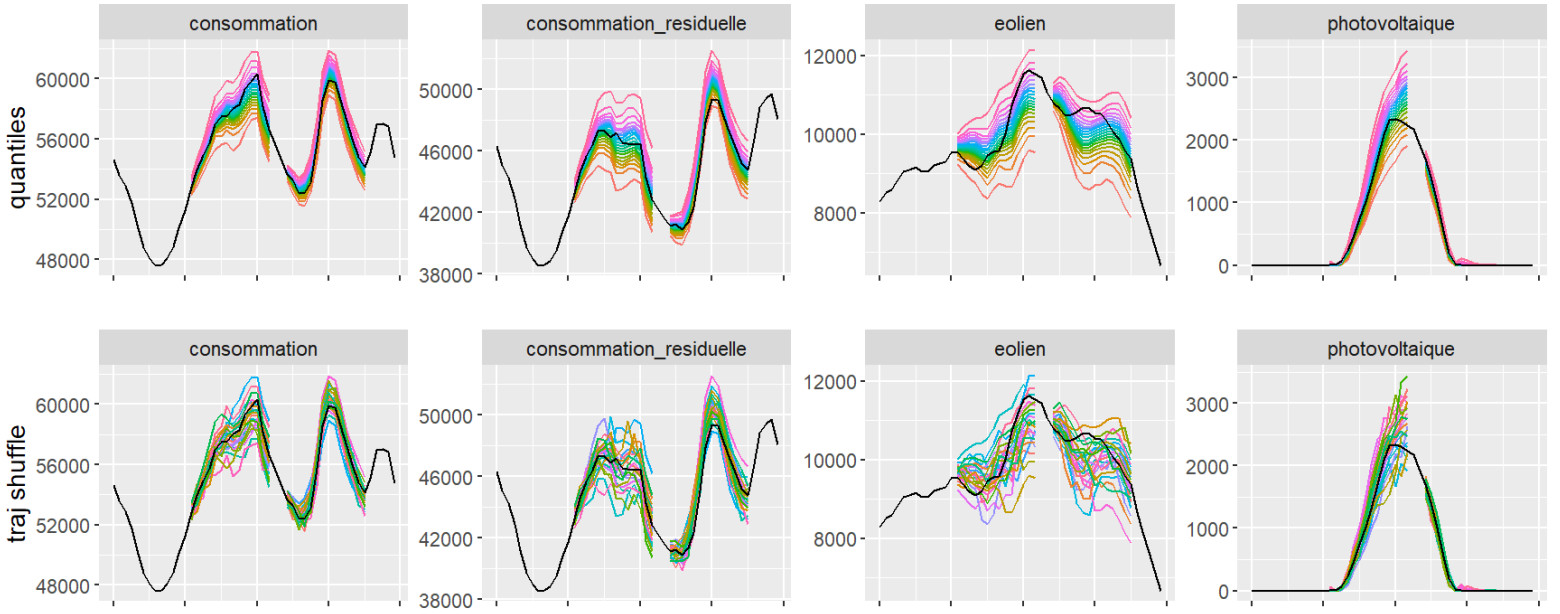
Trajectoires : copules empiriques

Copule : $F(y1, y2, y3) = C(F_1(y_1), F_2(y_2), F_3(y_3))$

Schaake shuffle = permutation des prévisions quantiles d'après les valeurs historiques :

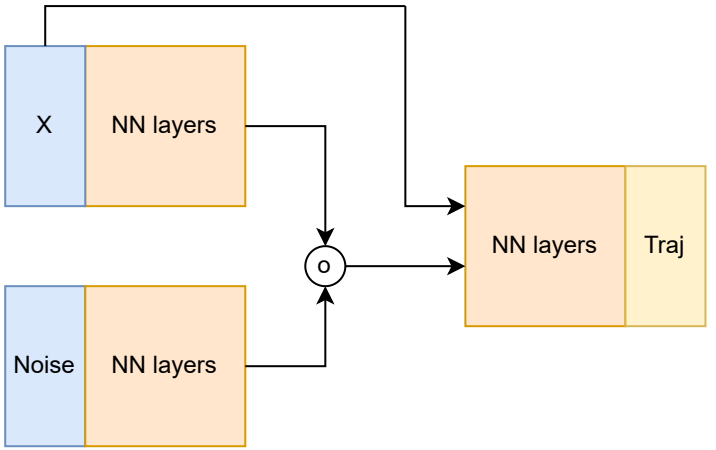
y_pred_traj = y_pred_quantile[rank(y_hist)]

- Pour 20 quantiles, on choisit 20 dates passées.
- Les permutations associent chaque trajectoire à une date passée.
- Indépendant de la dimension et peu coûteux.



Trajectoires : NN = modèle génératif

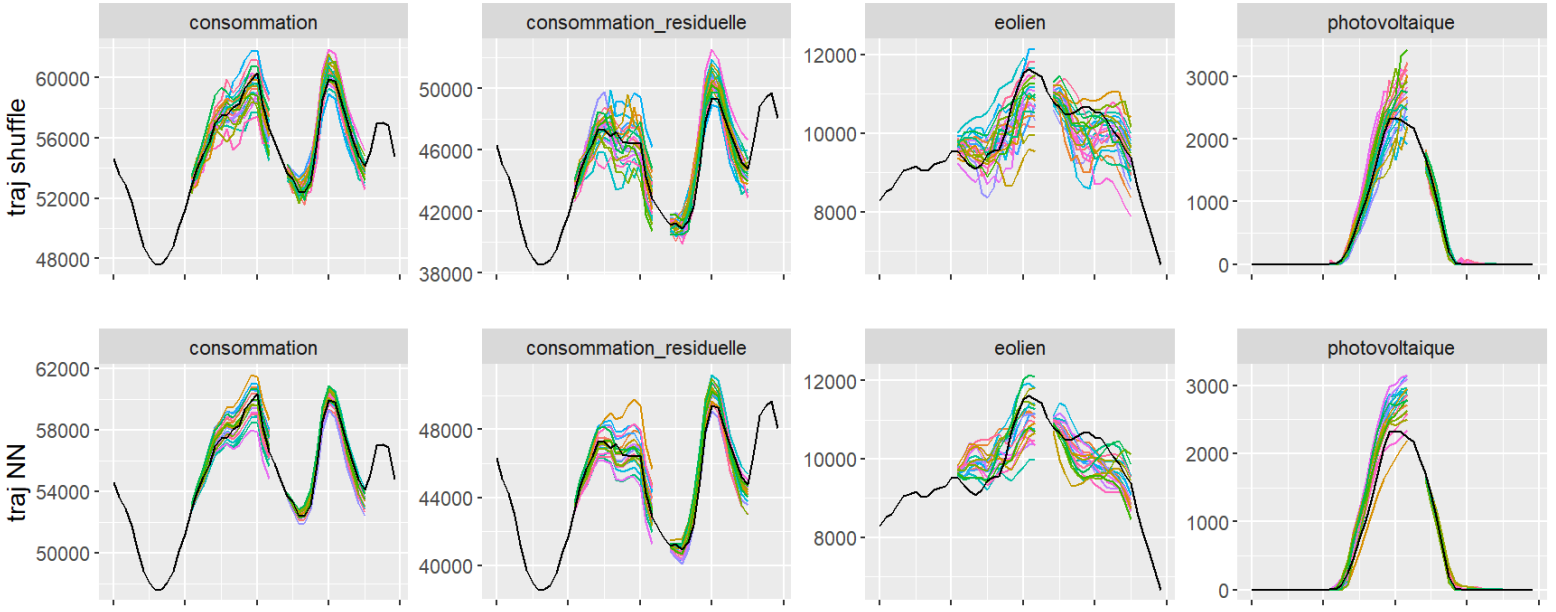
Approche simple, régulièrement reprise [Bouchacourt 2016, Chen 2022]



NN classique

- Injection de bruit.
- Taille fixe de N_p trajectoires.
- Minimisation du score d'énergie $E(||X - y||) - \frac{1}{2}E(||X - X'||)$

Trajectoires



- Plus difficile à calibrer (manque de pratique de ma part).
- Performances similaires, gain < 5%.

- NN génératif : possibilité de tirer 10000 trajectoires.
- Le Schaake shuffle conserve la calibration des prévisions quantiles.

Conclusion

IA vs. Statistique :

- GAM : Lisibilité et performance pour une tâche précise. Choix des interactions, bases de splines.
- NN : Multivarié facilement accessible. Beaucoup plus de paramètres : dimensions, régularisation, type de couches, activation...

Questions ouvertes :

- Généralisations du CRPS (quantile weighted) :
  --- Formules analytiques pour les distributions paramétriques ?
  --- Formulations multivariées ?
  --- Intérêt dans l'apprentissage et dans la validation ?
- Apprentissage des copules ?
- GAM génératif multivarié ?

Incertitudes à inclure :

- localisation des rampes et des pics.
- prévision météo (probabiliste).
- observations bruitées/partielles.

Bibliographie

Idées et code : Fasiolo, Gneiting, Lerch, Ziel, et al.

Buja, A., Stuetzle, W., & Shen, Y. (2005). Loss functions for binary class probability estimation and classification: Structure and applications. Working draft, November, 3, 13.

Székely, G. J., & Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. Journal of statistical planning and inference, 143(8), 1249-1272.

Bouchacourt, D., Mudigonda, P. K., & Nowozin, S. (2016). Disco nets: Dissimilarity coefficients networks. Advances in Neural Information Processing Systems, 29.

Ehm, W., Gneiting, T., Jordan, A., & Krüger, F. (2016). Of quantiles and expectiles: consistent scoring functions, Choquet representations and forecast rankings. Journal of the Royal Statistical Society Series B: Statistical Methodology, 78(3), 505-562.

Fasiolo, M., Wood, S. N., Zaffran, M., Nedellec, R., & Goude, Y. (2021). Fast calibrated additive quantile regression. Journal of the American Statistical Association, 116(535), 1402-1412.

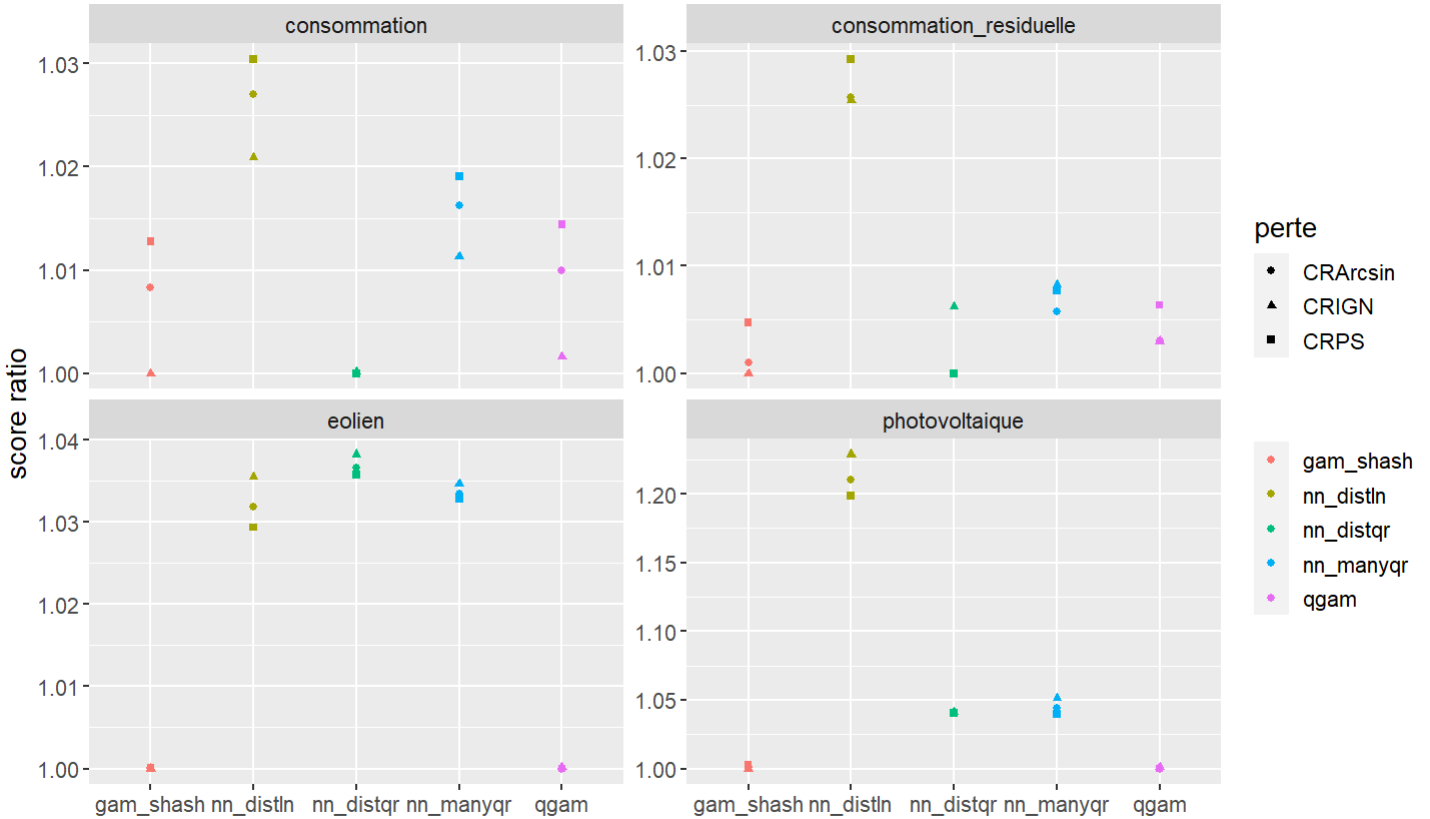
Chen, J., Janke, T., Steinke, F., & Lerch, S. (2022). Generative machine learning methods for multivariate ensemble post-processing. arXiv preprint arXiv:2211.01345.

Marcjasz, G., Narajewski, M., Weron, R., & Ziel, F. (2023). Distributional neural networks for electricity price forecasting. Energy Economics, 125, 106843.

Des questions ?

Résultats

Pondération des quantiles : CRPS (1) vs. CRIGN $\frac{1}{\alpha(1-\alpha)}$ vs. CRArcsin $\frac{1}{\sqrt{\alpha(1-\alpha)}}$



- Quelques ratés mais des performances comparables.
- La métrique de validation change peu le classement.

Exemple : pour un $t_0 \in \mathcal{T}$ fixé,

— On choisit un ensemble $\mathcal{T}' \subset \mathcal{T}$ de N_q dates t « bien choisies » (*).

1. On suppose que pour une variable $r \in \mathcal{R}$, les $N_q = 10$ quantiles de prévisions sont :

$$\mathcal{X} = (\hat{X}_{t_0,r,q})_q = (138.4, 140.2, 140.5, 141.6, 145.7, 146.2, 147., 147.3, 151.4, 154.2)$$

2. On extrait les N_q observations correspondant à la variable r pour $t \in \mathcal{T}'$:

$$Y = (X_{t,r})_{t\in\mathcal{T}'} = (Y_j)_{j\in[\![0,9]\!]} = (93.5, 117.2, 85.9, 97.3, 94.4, 80.1, 116.8, 95.8, 57.2, 106.9)$$

3. Soit σ la permutation qui permet de trier les valeurs de Y dans l'ordre croissant :

$$\mathcal{Y} = (Y_{\sigma(j)})_j = (57.2, 80.1, 85.9, 93.5, 94.4, 95.8, 97.3, 106.9, 116.8, 117.2)$$

4. On applique à $\mathcal{X}$ la permutation σ^{-1} :

$$\mathbf{X} = (\mathcal{X}_{\sigma^{-1}(j)})_j = (141.6, 154.2, 140.5, 147.0, 145.7, 140.2, 151.4, 146.2, 138.4, 147.3)$$

5. On itère ce processus pour chaque variable $r \in \mathcal{R}$

— On itère ce processus pour chaque temps $t_0 \in \mathcal{T}$