

# Travaux sous tension : traitement d'un volume important de données et outils de visualisation

RTE : S. Issad, R&D

Datastorm : B. Thieurmél, Data Scientist

30 juin 2017



Le réseau  
de transport  
d'électricité



DataStorm



GENES

ENSAE

ENSAI

CREST

CEPE

CASO C

## **Filiale d'EDF, Responsable du réseau public de transport d'électricité haute tension en France**

- Exploitation, Entretien
- Développement des lignes et stations associées
- Garant de l'équilibre du système électrique

**Achemine l'électricité depuis les unités de production vers des clients industriels et le réseau de distribution et gère les interconnexions avec les autres pays Européen**

105 000 km de lignes

# Le projet TST (Travaux sous tension)



## Motivations

- Aider l'entité en charge des TST à dimensionner ses outils pour répondre à la demande de maintenance
- Limiter les consignations <sup>a</sup> et l'indisponibilité des ouvrages concernés. Les TST sont un atout essentiel pour l'exploitant

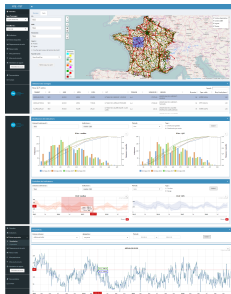
---

<sup>a</sup>Mise hors tension pour intervention de maintenance

## Les enjeux :

- Optimiser des outils tels que les shunts et les dispositifs de maintien
- Faciliter les travaux de maintenance en TST pour les opérateurs
- Rendre les TST attractifs

Développement d'une web-application **shiny** destinée aux métiers afin de leur donner des outils statistiques d'aide à la décision



## Motivations

- Accès moderne, rapide, pratique, et ergonomique aux données et aux résultats
- Filtrage patrimoniale, temporelle et spatial
- Visualisation dynamique
- Exportation des résultats / rapport

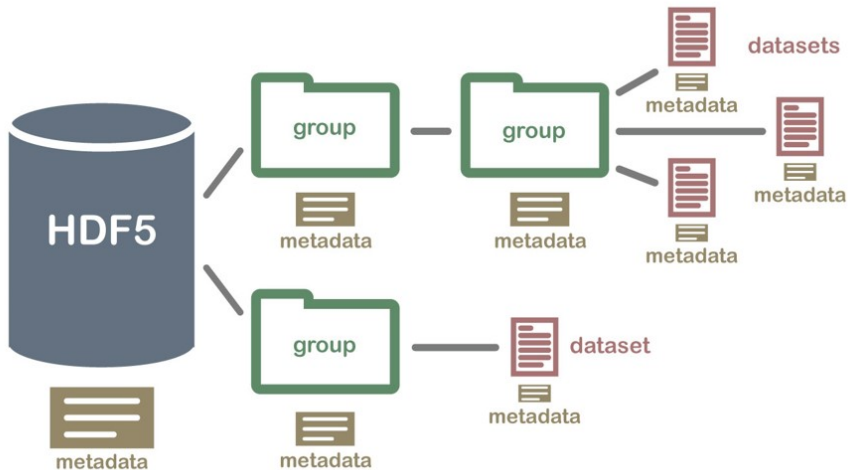
Gestion de projet agile

## Les données disponibles

- **4 ans** d'historique des lignes HT et THT, **pas de temps 5 minutes**
  - 1 lignes et 1 mois : 8930 mesures
  - 1 lignes et 4 ans : 430 000 mesures
  - **12K** lignes, **5 Milliard** de mesures
  - une centaine de fichiers plats, **60 Go**
- 
- Formatage et stockage au **format hdf5**
  - Utilisation du package **rhdf5**

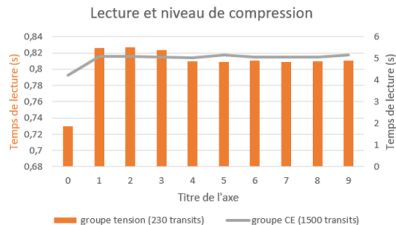
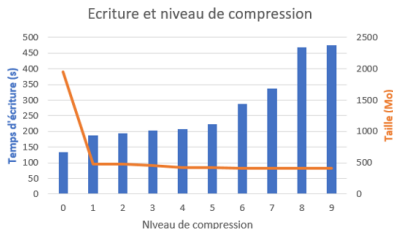
## Fichiers mensuels

- Gain de place (**facteur 5**)
- Performance dans le requêtage d'un sous-ensemble de données
- Parallélisation



<https://support.hdfgroup.org/HDF5/>

## Exemple d'écriture et de lecture d'un fichier mensuel. 12K transits pour plus de 1.8 Go de fichiers .txt.



Niveau de compression	Temps d'écriture (s)	Taille du fichier .h5
0	133,792	1.9 GiB
1	188,3	478.0 MiB
2	194,251	476.3 MiB
3	203,056	458.9 MiB
4	206,369	423.4 MiB
5	222,091	419.9 MiB
6	286,701	412.4 MiB
7	337,245	410.1 MiB
8	467,608	408.6 MiB
9	475,748	408.6 MiB

Niveau de compression	un transit	Grp. 250 transits	Grp. 1500 transits
0	0,008	0,73	4,214
1	0,007	0,826	5,095
2	0,008	0,827	5,093
3	0,007	0,824	5,068
4	0,007	0,81	5,019
5	0,007	0,809	5,171
6	0,007	0,811	5,061
7	0,007	0,809	5,055
8	0,008	0,81	5,051
9	0,007	0,811	5,151

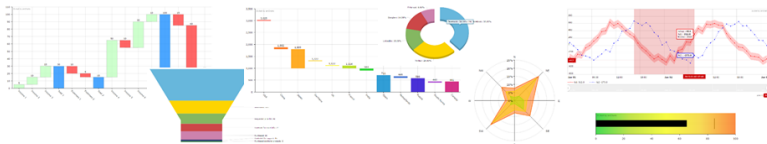
# Visualisation des résultats : le package rAmCharts

## Proposer à l'utilisateur des fonctions simples pour la création de graphiques interactifs sous R

- Développé par les équipes de *Datastorm* sur leur temps de R&D
- Disponible sur le CRAN
- [http://datastorm-open.github.io/introduction\\_ramcharts/](http://datastorm-open.github.io/introduction_ramcharts/)

### Les +

- Richesse et simplicité (dont des représentations "marketing")
- Exportation (jpeg, pdf, json, ...), annotations
- Basé sur la librairie gratuite **amCharts** (<https://www.amcharts.com/>)





# rAmCharts : les améliorations motivées par ce projet

## Synchronisation des séries temporelles

- Fonction *amTimeSeries*
- Calqué sur **dygraphs**, avec l'utilisation simple du paramètre *group*

## Exportation côté server des graphiques

- Via un module **shiny**, et la fonction *rAmChartsExportServer*
- Sauvegarder un ensemble de graphiques / éviter de recoder les graphiques javascripts en **ggplot2** pour les inclure dans un rapport, ....

## Visualisation de séries temporelles de grande dimension

- communication entre **shiny** et la librairie javascript pour agréger plus ou moins finement les données en fonction du zoom utilisateur
- utilisation de **data.table** pour une exécution optimisée
- Disponible avec le module *rAmChartsTimeSeriesServer*

# DEMONSTRATION