



Le réseau
de transport
d'électricité

Synthèse hebdomadaire de la consommation électrique française

12/01/2023

Meetup R Addicts

Valentin Cadoret et Victor Perrier
Direction Statistiques et Valorisation des Données

Copyright RTE – 2023. Ce document est la propriété de RTE. Toute communication, reproduction, publication même partielle est interdite sauf autorisation écrite du Gestionnaire du Réseau de Transport d'Électricité (RTE)

Contexte

Pourquoi et pour qui
créer une synthèse de la
consommation ?

Ecosystème

Quels sont les outils et
source de données à
notre disposition ?
Détails du processus mis
en place

Focus

Détails sur le code et les
packages utilisés pour
produire la synthèse
finale

Contexte et objectifs

- La synthèse hebdomadaire de la consommation électrique française, de quoi s'agit-il ?
- Pourquoi publier une telle synthèse ?
- À qui s'adresse-t-elle ?
- Quand est-elle publiée ?



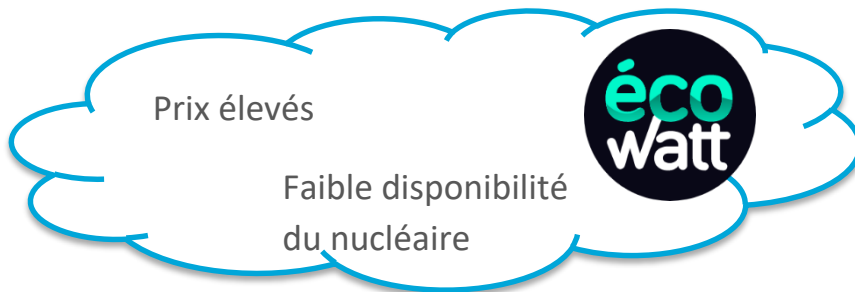
La synthèse

Publiée chaque mardi sur le site RTE :

<https://www.rte-france.com/synthese-hebdomadaire-consommation-electrique-francaise>

Pourquoi ?

- Hiver sous haute vigilance
- Besoin de communication / pédagogie
- En appui de la publication mensuelle « [Passage de l'Hiver](#) »



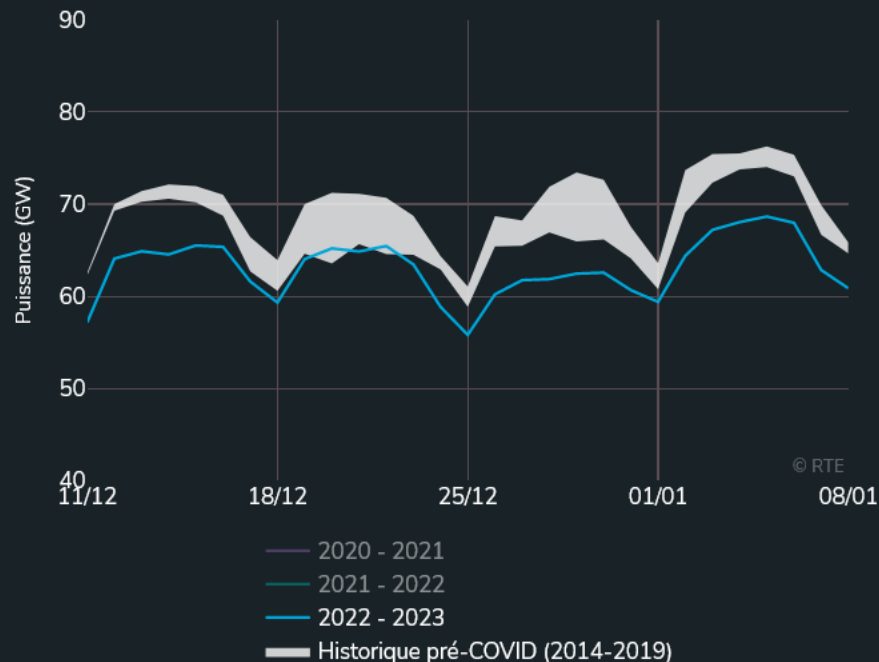
La synthèse est issue d'une application shiny de suivi interne

Consommation d'électricité

Il s'agit de la consommation qui aurait eu lieu si les températures avaient été alignées sur les températures normales pour la période.

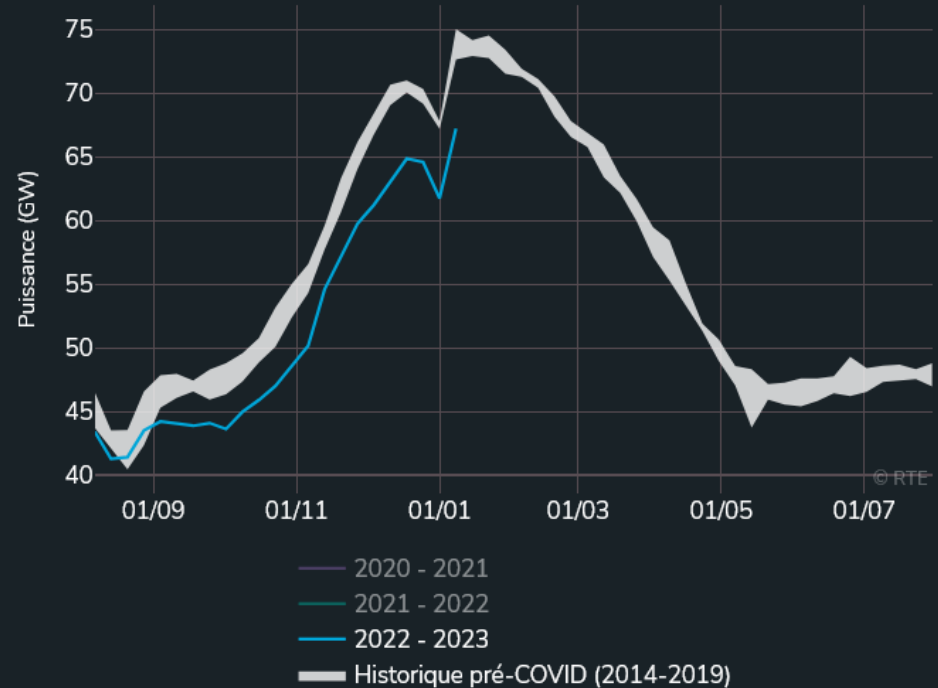
Comparaison jour par jour, par exemple :

- Noël
- Mercredi de la 1^{ère} semaine de l'année



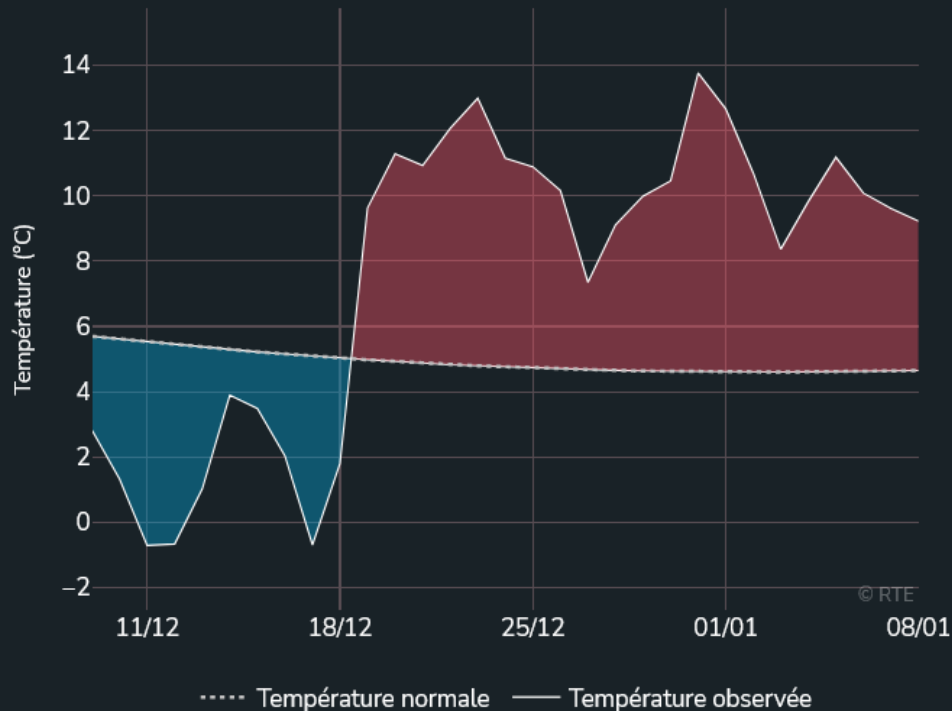
Évolution depuis le début de l'année

- Analyse hebdomadaire, sur jours ouvrés uniquement
- Enveloppe : min/max des années de référence (2014-2019)



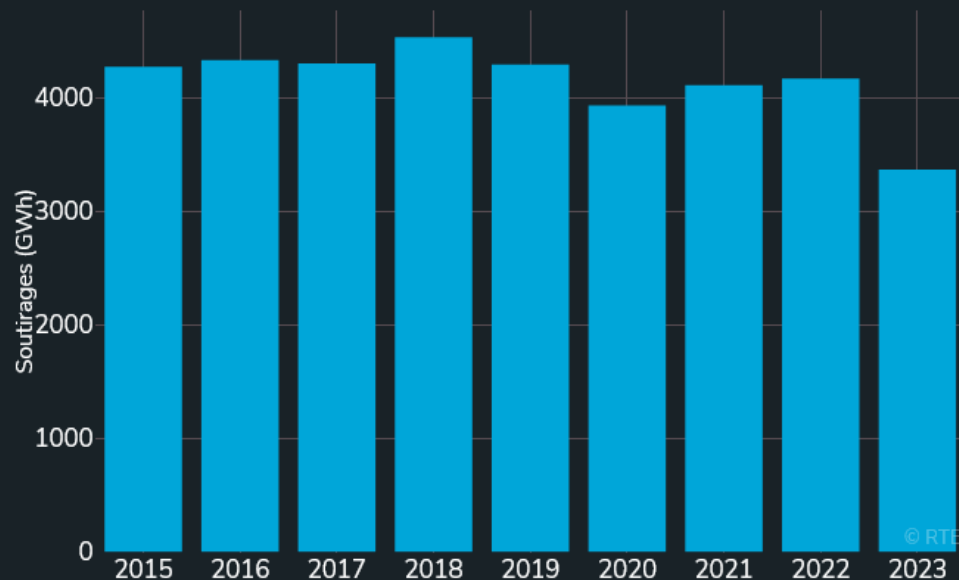
Conditions météorologiques sur la période

Écart de la température observée à la température normale



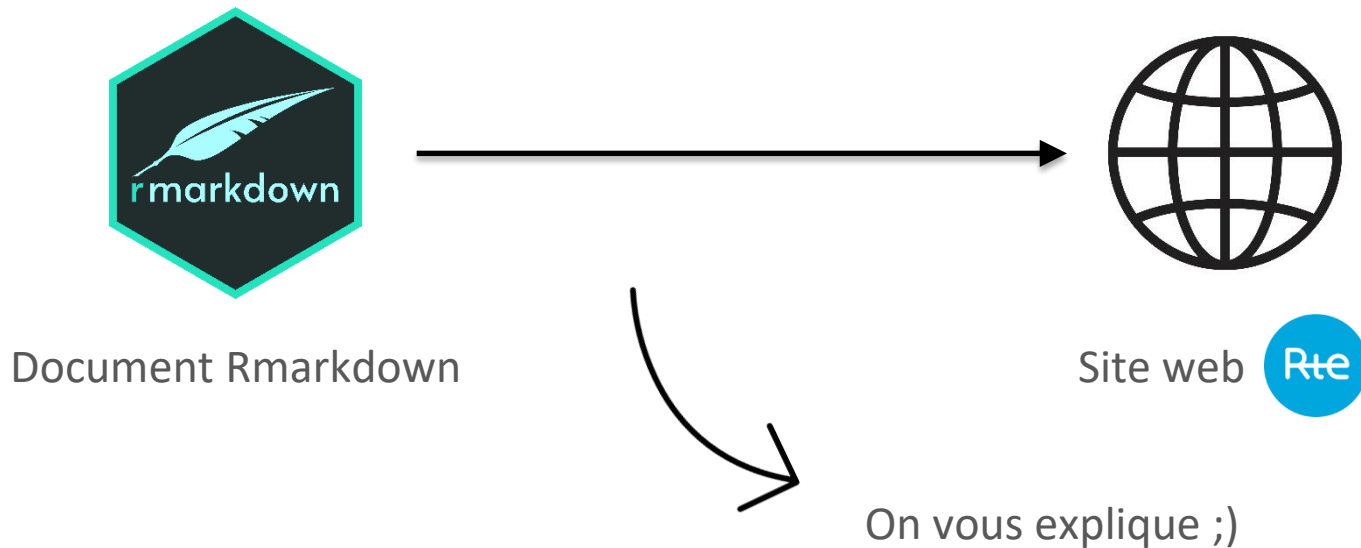
Analyse sectorielle

- Comparaison des consommations par secteur
- Actuellement, uniquement les gros consommateurs (industriels et SNCF)



Comment cela marche ?

Tout est fait en  en suivant le workflow suivant :



#2

Ecosystème

- Environnement de développement
- Source des données
- Outils développés
- Worklow mis en place



Environnement de développement

Data Lab équipé de Rstudio (Posit) Workbench et d'un GitLab interne :



RStudio Server Pro

Interface Rstudio accessible sur un serveur mutualisé



RStudio Connect

Serveur de publication (applications Shiny, documents Rmarkdown...)



RStudio Package Manager

Gestion des packages (CRAN + packages internes)



Gestionnaire de version pour le code

Outils développés



- Mise à jour des données : Rmarkdown schedulé sur RStudio Connect

- Application shiny de suivi interne



- Mise à jour de la synthèse :

- 1- Lecture des données pré-enregistrées
- 2- Calcul des indicateurs et génération des graphiques
- 3- Déploiement sur Rstudio Connect pour relecture
- 4- Déploiement sur le site RTE



- Mise à disposition des données sur [ODRE](#)

Données

Source des données :



dremio



Serveur FTP: fichiers plats



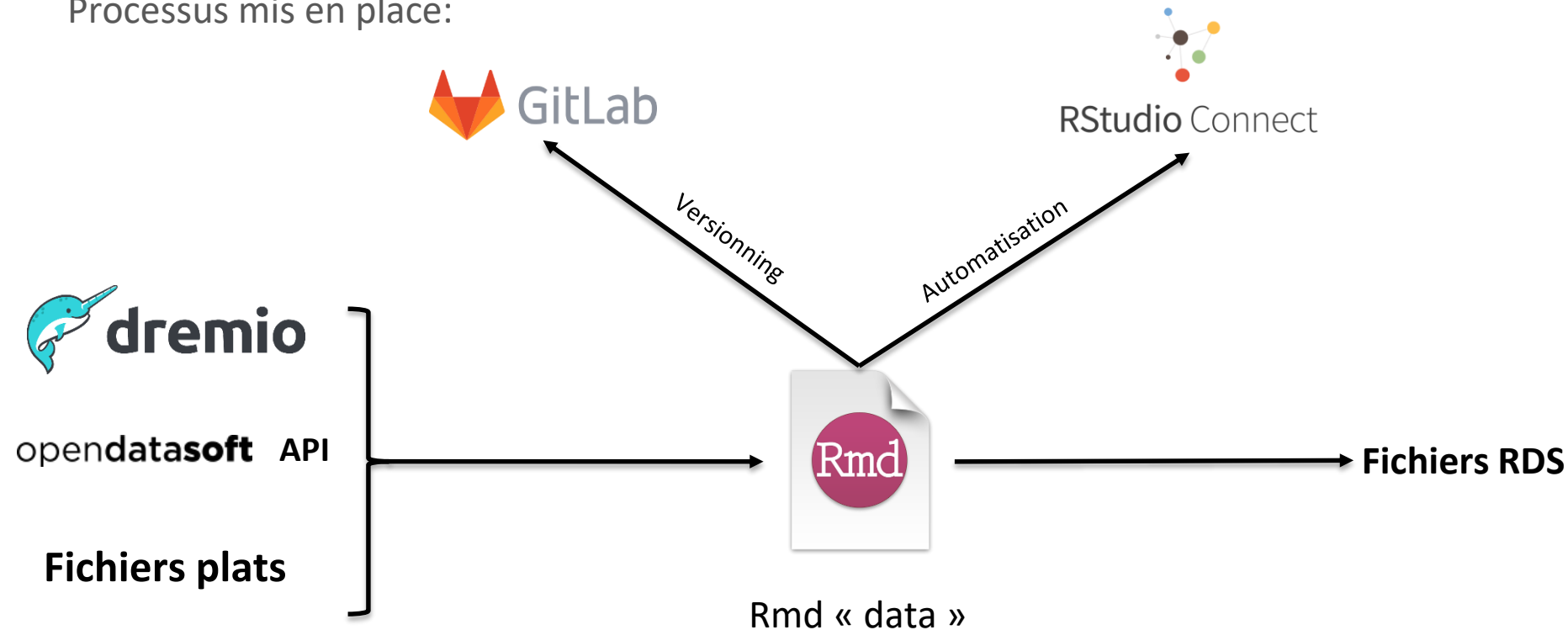
opendatasoft API



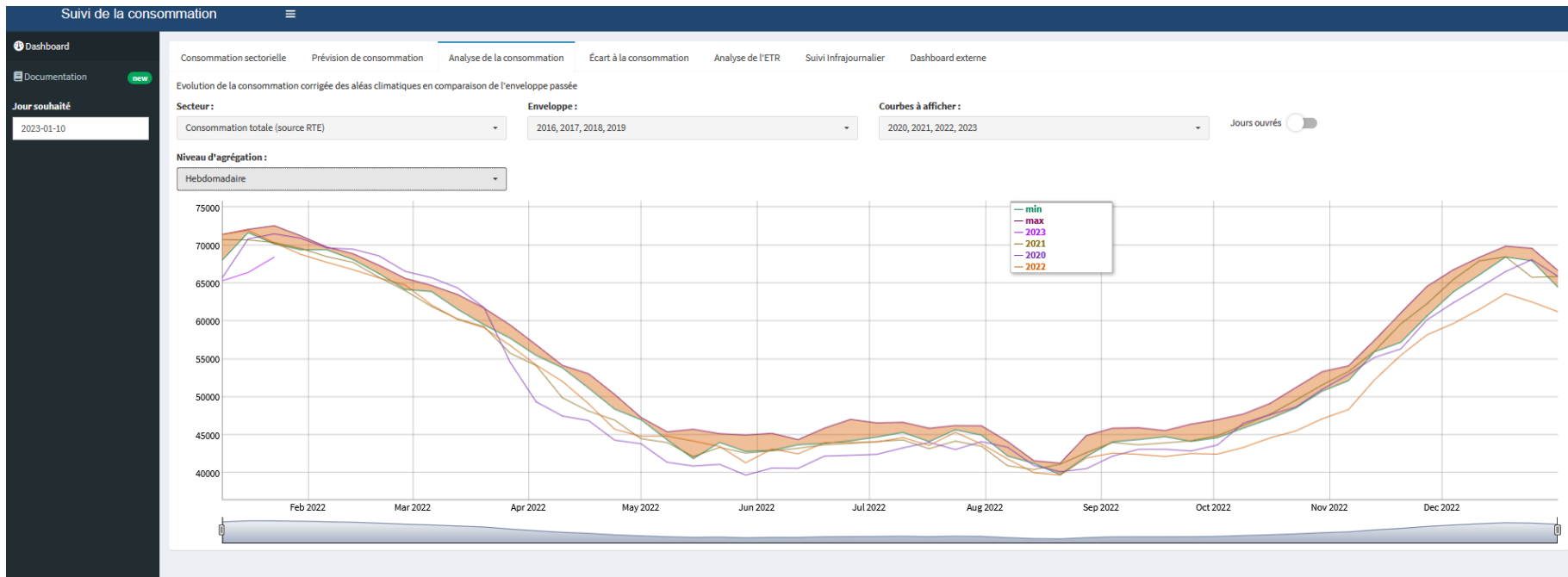
{httr2}

Préparation des données et automatisation

Processus mis en place:

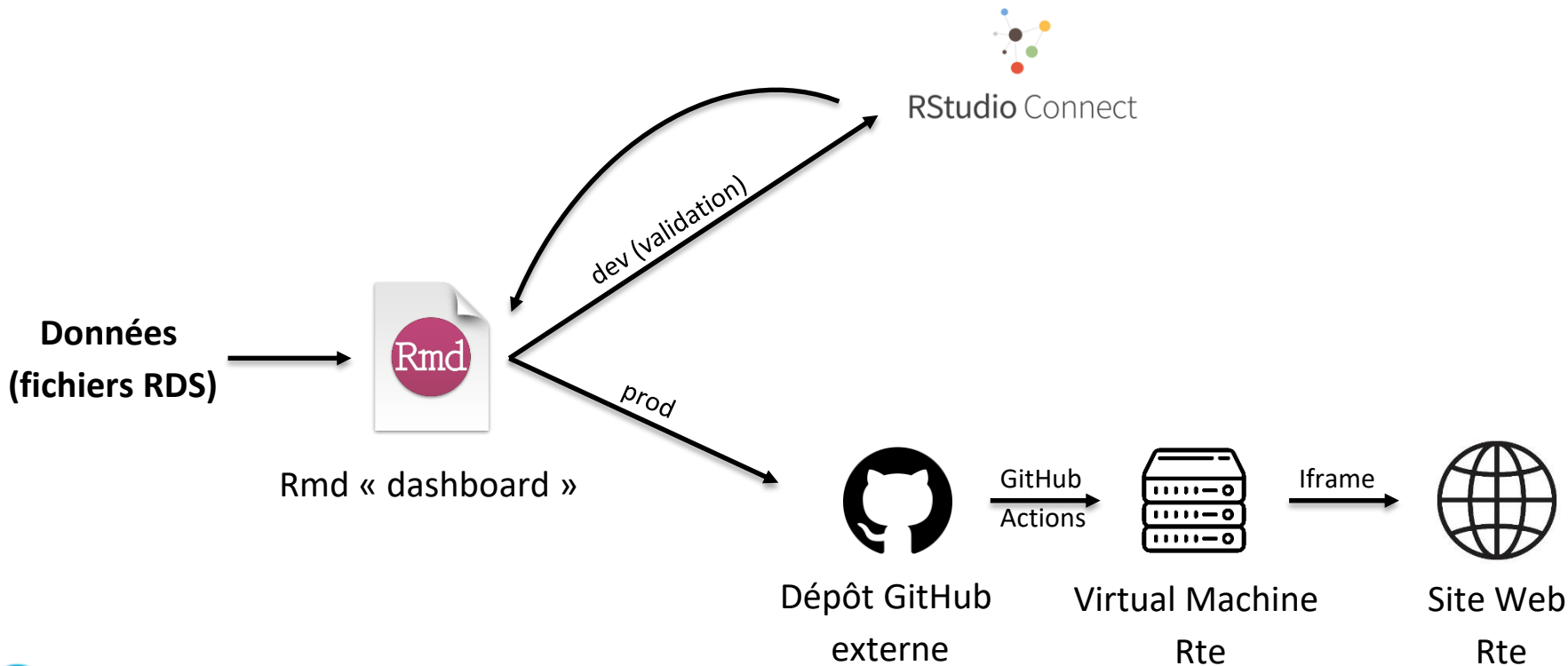


Une app Shiny d'analyse technique



Génération et publication

Processus mis en place:



#3

Focus

- Packages utilisés
- Zoom sur quelques fonctionnalités
- Evolutions à venir
- Anatomie du Rmd

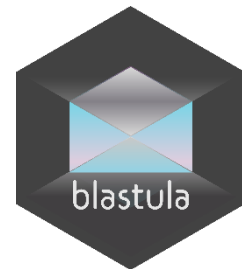


Packages utilisés



- Mais aussi {plotly}, {R.utils} et une librairie interne pour l'accès aux données

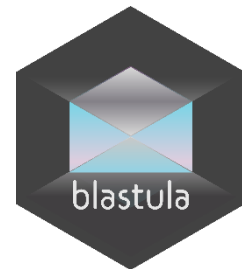
Envoi d'email en cas d'erreur



```
library(blastula)

envoyer_email <- function(mon_contenu, objet_email = "", destinataires_email = NULL){
  email <- compose_email(
    body = md(mon_contenu),
    footer = md(
      c("Email envoyé le ", format(lubridate::with_tz(Sys.time(), "Europe/Paris"), "%d/%m/%Y à
%H:%M:%S")
    )
  )
  email %>%
  smtp_send(
    to = destinataires_email,
    from = "erreur@ne-pas-repondre.com",
    subject = objet_email,
    credentials = creds_envvar(host = "url_host", port = "port", use_ssl = F)
  )
}
```

Envoi d'email en cas d'erreur



A partir d'un script R / Cron :

à mettre dans le script :

```
table_email <- readr::read_csv2("email_developpeur_push.csv")
options(error = emailOnce(destinataires_email = table_email$email))
```

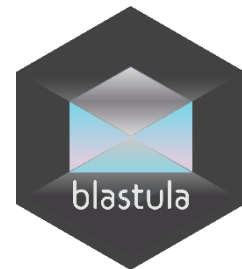
Et la fonction
emailOnce() :

```
emailOnce <- function(message_error, destinataires_email = NULL) {
  old <- getOption("error")
  function() {
    options(error = old)

    # Get script path (cf https://stackoverflow.com/questions/1815606/determine-path-of-the-executing-script)
    initial.options <- commandArgs(trailingOnly = FALSE)
    file.arg.name <- "--file="
    script_erreur <- sub(file.arg.name, "", initial.options[grep(file.arg.name, initial.options)])

    envoyer_email(mon_contenu = paste("Attention, il y a eu une erreur dans le script suivant
:\n",script_erreur),
                  objet_email = paste("[cronR] erreur script",
sub(dirname(script_erreur),",",script_erreur)),
                  destinataires_email = destinataires_email)
  }
}
```

Envoi d'email en cas d'erreur



A partir de Rmarkdown :

```
liste_email <- get_emails_dev()
knitr::knit_hooks$set(error = function(x, options) {
  if (!email_envoye) {
    envoyer_email(
      "Une erreur est survenue lors de l'exécution, visitez la version en ligne pour plus de détails.",
      objet_email = "Mise à jour données dashboard",
      destinataires_email = liste_email
    )
  }
  email_envoye <- TRUE
  paste0("<pre style=\"color: red; font-weight: bold;\"><code>", x, "</code></pre>")
})
```

Anatomie d'un Rmd

Configuration du Rmd produisant la synthèse :

```
---  
title: Conso Hebdo  
author: Rte  
output:  
  html_document:  
    self_contained: false  
    lib_dir: assets/libs  
    template: templates/simple.html # navbar.html ou simple.html  
    css: ["assets/styles.css", "assets/tabsets.css", "assets/regles.scss"]  
    pandoc_args: [  
      "--strip-comments"  
    ]  
    theme:  
      version: 5  
      bg: "#10181d"  
      card_bg: "#192227"  
      fg: "#FFF"  
      primary: "#00A6D9"  
      base_font: "Nunito Sans"  
      font_weight_bolder: 900  
      grid-breakpoints: '(xs: 0, sm: 576px, md: 811px, lg: 992px, xl: 1200px, xxl: 1400px)'  
---
```

→ Template Pandoc

→ Règles CSS + SCSS

→ Thème Bootstrap 5
avec {bslib}

Anatomie d'un Rmd

Utilisation du template HTML de Pandoc pour plus de flexibilité :

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="$lang$" xml:lang="$lang$" $if(dir)$
dir="$dir$" $endif$>
<head>
  <meta charset="utf-8" />
  <meta name="generator" content="pandoc" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes" />
  $for(author-meta)$
  <meta name="author" content="$author-meta$" />
  $endif$
  $if(date-meta)$
  <meta name="dcterms.date" content="$date-meta$" />
  $endif$
  $if(keywords)$
  <meta name="keywords" content="$for(keywords)$ $keywords$ $sep$, $endif$" />
  $endif$
  $if(description-meta)$
  <meta name="description" content="$description-meta$" />
  $endif$
  <title>$if(title-prefix)$ $title-prefix$ – $endif$ $pagetitle$ </title>
  $for(css)$
  <link rel="stylesheet" href="$css$" />
  $endif$
  $for(header-includes)$
  $header-includes
  $endif$
  ...

```

- Suppression des éléments non utilisés (affichage du code, tables des matières, ...)
- Ajout de markup HTML propre à nos besoins
- Ajout potentiel d'un bandeau en haut et pied de page

Anatomie d'un Rmd

Règles SCSS pour modifier des éléments à l'aide des classes utilitaires de Bootstrap :

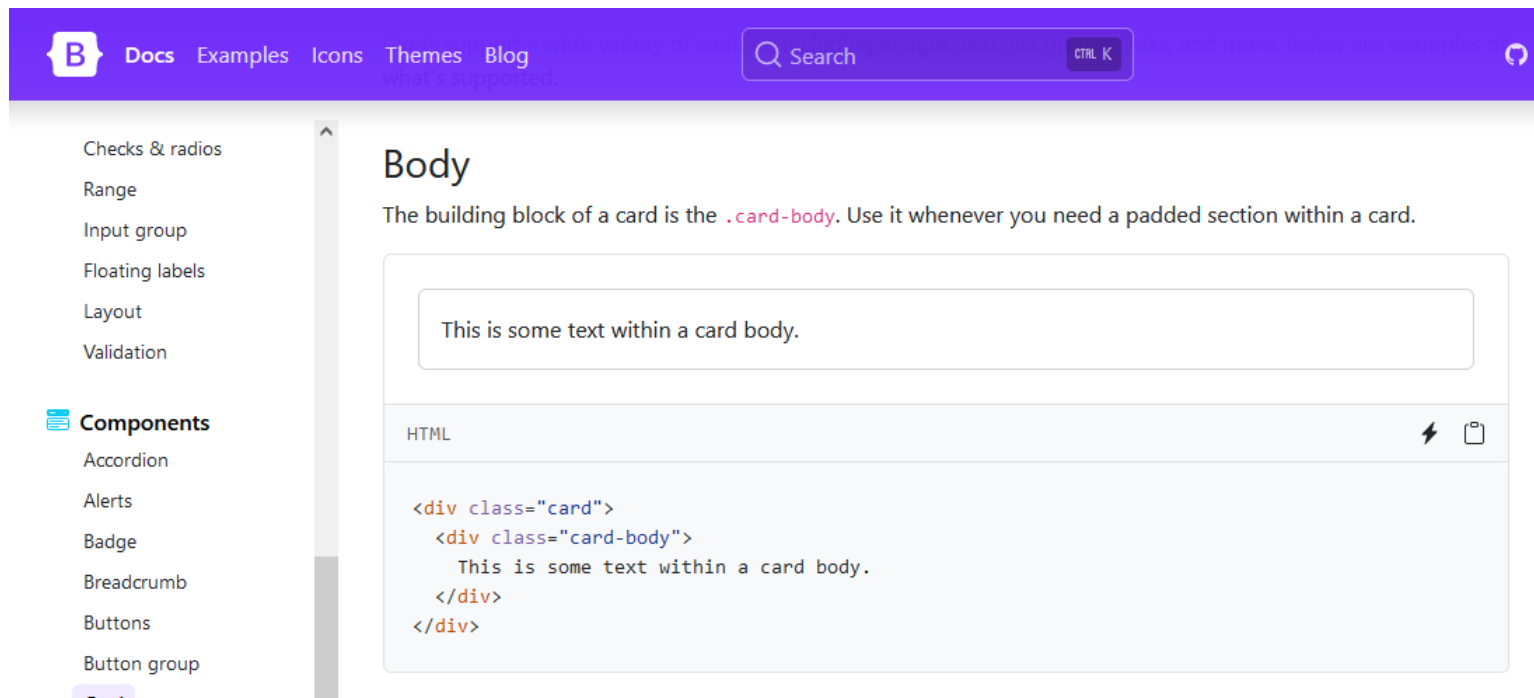
```
/* Regles SCSS spécifiques */

/* Ajout bordure aux boutons tabs */
.nav-pills>li>a { @extend .border } /* ajout bordure */
.nav-pills>li>a { @extend .border-primary } /* couleur bordure */
.nav-pills>li>a { @extend .rounded } /* arrondi coins bordure */
.nav-pills>li>a { @extend .me-1 } /* marge droite, pour ajouter un espacement entre les boutons
*/

/* miniamle hauteur unique si 2 colonnes */
@include media-breakpoint-up(md) {
  h3
    min-height: 75px;
}
}
```

Anatomie d'un Rmd

Utilisation des cartes Bootstrap pour afficher les éléments de la synthèse :



The screenshot shows the Bootstrap documentation website. The top navigation bar is purple and contains the Bootstrap logo, links for 'Docs', 'Examples', 'Icons', 'Themes', and 'Blog', a search bar with the text 'Search' and a 'CTRL K' button, and a refresh icon. The left sidebar is white and lists various components under the heading 'Components'. The main content area is white and features the heading 'Body' in a large, bold font. Below the heading is a paragraph of text: 'The building block of a card is the `.card-body`. Use it whenever you need a padded section within a card.' Below this text is a visual example of a card body, which is a white rounded rectangle with a thin gray border containing the text 'This is some text within a card body.' Below the visual example is a code block with a light gray background and a dark gray border. The code block is titled 'HTML' and contains the following code:

```
<div class="card">
  <div class="card-body">
    This is some text within a card body.
  </div>
</div>
```

Anatomie d'un Rmd

En Rmarkdown on utilise des « fenced div blocks » pour créer les balises spécifiques :

```
● ● ●

<!-- debut carte -->
::::: {.card .h-100}

<!-- debut carte contenu -->
::: {.card-body}

### Titre de la section {.card-title .mt-0}

```${r nom-chunk}
code graphique
```

Explications graphique

:::
<!-- fin carte contenu -->

:::::
<!-- fin carte -->
```

Anatomie d'un Rmd

Utilisation du package {bslib} pour personnaliser le thème Bootstrap :

```
theme:  
  version: 5  
  bg: "#10181d"  
  card_bg: "#192227"  
  fg: "#FFF"  
  primary: "#00A6D9"  
  base_font: "Nunito Sans"  
  font_weight_bolder: 900  
  grid-breakpoints: '(xs:0, sm: 576px, md: 811px, lg: 992px, xl: 1200px)'
```

- Couleur d'arrière plan
- Couleur de la police
- Couleur des cartes
- Police Google (importée via le package {gfonts})
- ...

Récupération de données via une API

Certaines données utilisées dans la synthèse proviennent de l'API Opendatasoft et sont récupérées en R via le package {httr2} :

```
telecharger_opendata <- function(service, ...) {  
  if (any("" %in% c(Sys.getenv("PROXY_ID"), Sys.getenv("PROXY_PWD")))) {  
    message("Le proxy doit être passé via les variables PROXY_ID et PROXY_PWD")  
  }  
  req <- httr2::request(paste0(service, "/api/records/1.0/download/")) %>%  
    httr2::req_proxy(  
      url = Sys.getenv("PROXY_URL", "proxy-surf.rte-france.com"),  
      port = as.integer(Sys.getenv("PROXY_PORT", 3128)),  
      username = Sys.getenv("PROXY_ID"),  
      password = Sys.getenv("PROXY_PWD")  
    ) %>%  
    httr2::req_url_query(...)  
  
  res <- httr2::req_perform(req)  
  
  donnees <- data.table::fread(text = httr2::resp_body_string(res))  
  tibble::as_tibble(donnees)  
}
```

→ Requête GET

→ Passage du proxy

→ Paramètres de la
requête

Récupération de données via une API

Interrogation de l'API Enedis, mais cela fonctionne aussi avec les autres services
Opendatasoft (RATP, SNCF, La Poste, ...)

```
# exemple
telecharger_opendata(
  service = "https://data.enedis.fr",
  dataset = "coefficients-de-profils-dynamiques-anticipes-en-j1",
  q = sprintf(
    "horodate:[%s T0 %s]",
    format(Sys.Date() - 1, format = "%Y-%m-%dT00:00:00Z"),
    format(Sys.Date() - 1, format = "%Y-%m-%dT23:59:59Z")
  )
)
```

→ URL du service

→ Nom du jeu de données

→ Filtre sur la date

Evolutions à venir

- Transformation en package
- Consolidation des fonctions (passage d'une année à l'autre, choix des années de référence, etc.)
- Utilisations des packages :
 - `{testthat}` pour protéger/vérifier les fonctionnalités développées
 - `{renv}` pour assurer la reproductibilité



Le réseau
de transport
d'électricité

Merci !

Pour toute question : Valentin (valentin.cadoret@rte-france.com) ou Victor (victor.perrier@dreamRs.fr)