



Meetup R : le template shiny EDF

Mise en place de bonnes pratiques

Mai 2018 – Arnaud Gaborit

SOMMAIRE

1.

Contexte : pourquoi un template ?

2.

Etape 1 : Utilisation de l'addin

3.

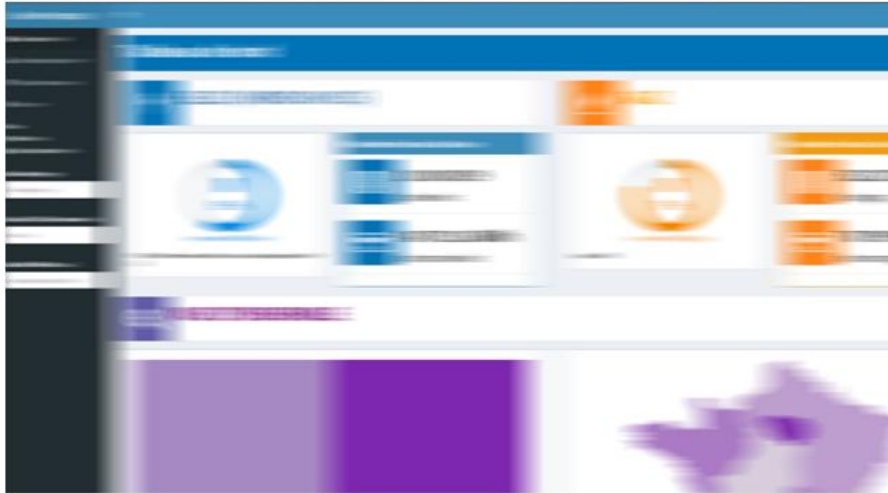
Etape 2 : modifier les scripts récupérés

4.

Perspectives

Contexte : pourquoi un template ?

Uniformisation des applis shiny



Contexte : pourquoi un template ?

Partage de bonnes pratiques



Est-ce que la date des données est affichée ?

Une page d'informations ?

Présence du lien vers le portail des autres app shiny ?



Est-ce que les scripts ui.R et server.R sont décomposés ?

Mise en place d'une programmation modulaire ?

Contexte : pourquoi un template ?



Gain de temps

Mise à disposition d'une galerie d'exemples pour faciliter la prise en main



Etape 1 : Utilisation de l'addin

Cancel Choisir des fonctionnalités pour un shiny Done

Saisir un nom de projet → **Nom du projet pour l'entête des fichiers**

Saisir votre prénom et votre nom → **Nom prénom pour l'entête des fichiers**

Graphique de base Graphique en JS Graphique avec highcharter Rendu de tables avec DT Graphique avec dygraphs



Cartographie avec leaflet Programmation modulaire



Sélectionner les exemples à retenir

Répertoire du shiny :
 → **Répertoire où va être stocké le template généré (le crée si absent, complète le dossier sinon)**

→ **Valider pour enregistrer les fichiers**

Choix

Une application shiny fonctionnelle
















The screenshot shows a web browser window displaying a Shiny application. The browser's address bar shows the URL. The application has a blue header with the EDF logo and the text 'Titre à compléter'. A sidebar menu on the left contains several items: 'Accueil', 'Histogramme', 'Graphique avec logo', 'Barchart', 'Visualisation table', 'Serie temporelle', 'Cartographie', 'Module', 'Informations', and 'Retour portail IDée'. A 'Bookmark...' button is also visible. The main content area features a histogram titled 'Exemple histogramme' with the subtitle 'Titre du graphique'. The histogram has a y-axis labeled 'Frequency' ranging from 0 to 25 and an x-axis labeled 'Label de l'abscisse' with values 50, 60, 70, 80, and 90. The bars are colored in a gradient from orange to black. A notification icon in the top right corner of the application shows a '1'. Red arrows point from text labels to various elements: 'Un double sidebar Menu et Filtre' points to the sidebar; 'La date des données' points to the notification icon; 'Une page informations' points to the 'Informations' menu item; 'Un retour au portail IDée' points to the 'Retour portail IDée' menu item; 'Pour obtenir l'URL de l'application en état' points to the browser's address bar; and 'La charte graphique EDF' points to the histogram.



La démo !

Etape 2 : modifier les scripts récupérés → On commence par les choses simples !

On modifie le fichier config.yml

-  config → On modifie le fichier config.yml
-  data → On enregistre les tables de données
-  fonctions → Les fonctions utilisées dans l'application
-  inputs
-  modules
-  outputs
-  server
-  ui
-  www → Les fichiers png, js et css insérées dans l'appli
-  gestionnaire_habilitation.R
-  global.R → On complète la liste des packages à charger
-  information.md → On modifie la page d'informations qui est présente dans l'appli
-  restart.txt
-  server.R → Fichier chapeau server.R (pas de modification à faire)
-  ui.R → Fichier chapeau ui.R (pas de modification à faire)

```
# 1. Titre de l'application (affiché en haut) -----#
# -----#
titre: Titre à compléter

# 2. Identifiant interne de l'application (pour gestion des habilitations) -----#
# -----#
id_application: id_a_modifier

# 3. Gestion des répertoires (à modifier) -----#
# -----#
repertoire:
  base:      .
  data:     data
  functions: fonctions
  modules:  modules
  server:   server
  ui:      ui
  outputs: outputs
  inputs:  inputs
  www:     www

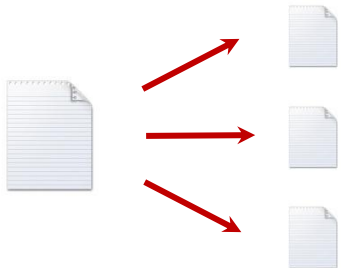
# 4. Date des données (à modifier) -----#
# -----#
date_donnees: 26/01/2018
```

Etape 2 : modifier les scripts récupérés → On modifie le corps de l'appli



Etape 2 : modifier les scripts récupérés

Les bonnes pratiques



Eclatement du ui et du server pour :

- Une meilleure organisation dans le code de l'appli
- Travailler plus facilement à plusieurs sur le même projet via git

Côté ui : un fichier par page de l'appli

Côté server : objets d'une même page dans le même fichier / tables reactive dans un même fichier



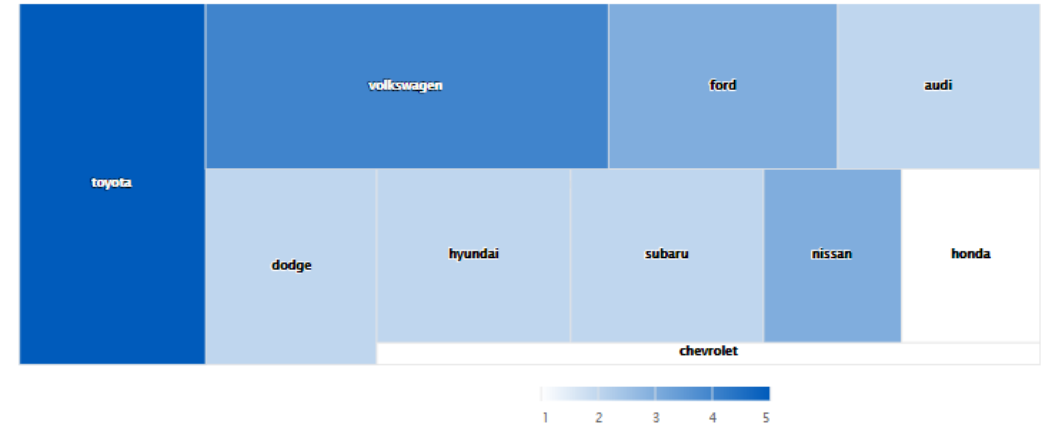
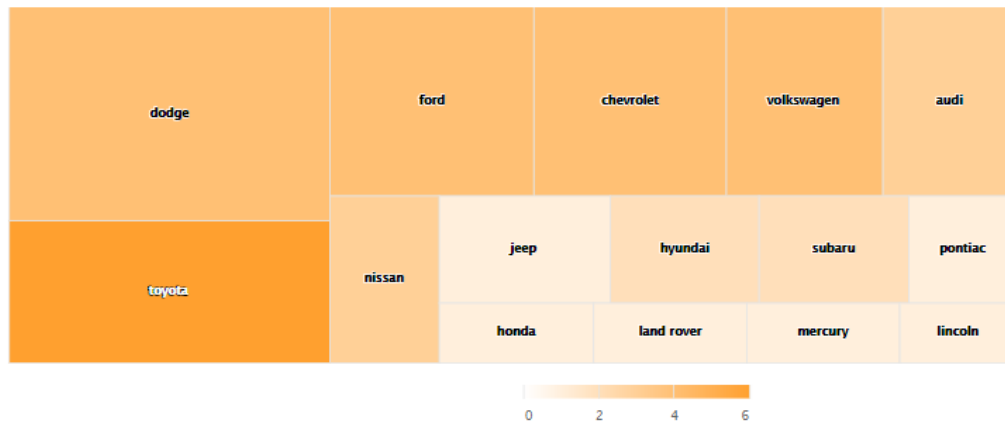
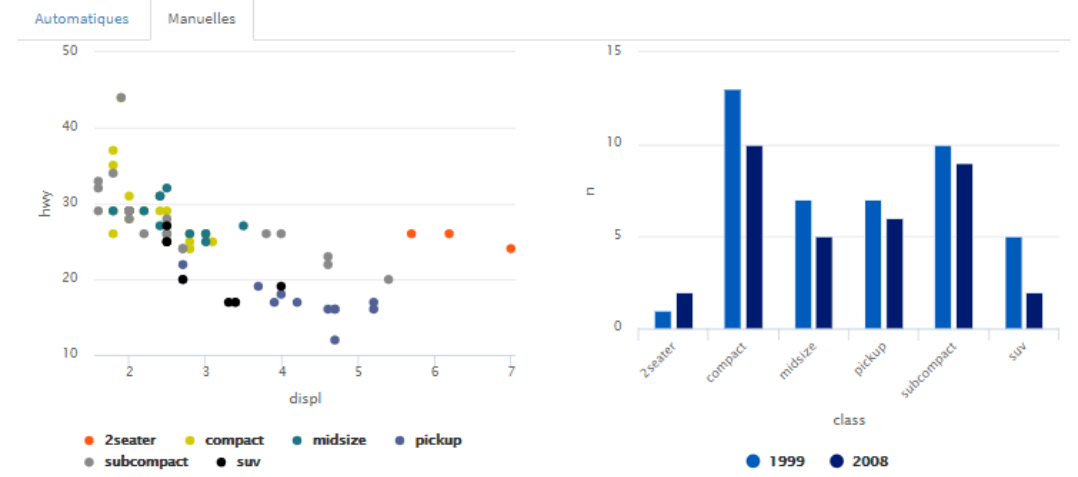
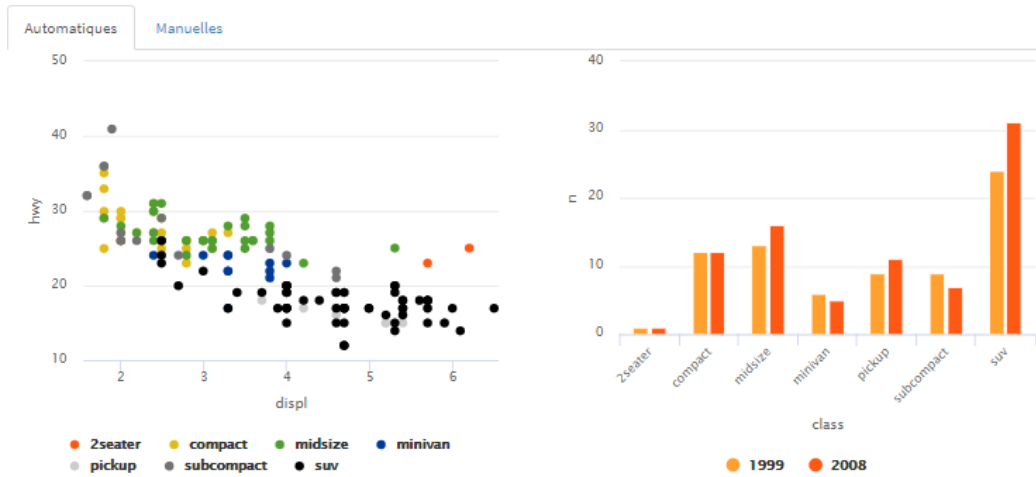
Vérifier l'unicité des inputs et des outputs

rpackage
HIGHCHARTER

Utiliser highcharter et non rCharts

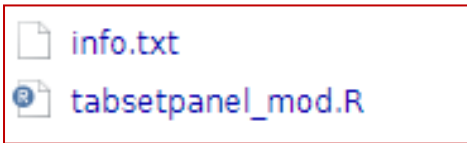
Etape 2b (UTILISATION AVANCEE) : Shiny Module 1/3

But : définir un ensemble d'éléments réutilisables dans l'application



Etape 2b : Shiny Module 2/3 → Modification des 2 fonctions (partie UI et partie SERVER)

- config
- data
- functions
- inputs
- modules
- outputs
- server
- ui
- www
- gestionnaire_habilitation.R
- global.R
- information.md
- restart.txt
- server.R
- ui.R



```
# Partie UI -----  
Module_UI = function( id ){  
  
  # Permet de créer des ID préfixés par id_  
  ns = NS(id)  
  
  # ns() permet de préfixer le paramètre id + "_"  
  tagList( fluidRow( column( 6 , highchartOutput( ns( "graph_mod1" ) ) )  
                  , column( 6 , highchartOutput( ns( "graph_mod2" ) ) ) ) )  
            , highchartOutput( ns( "graph_mod3" ) ) )  
}
```

```
# Partie SERVER -----  
Module_SERVER = function( input, output, session, id, donnees, filtres, couleurs ){  
  
  # Fonction reactive -----  
  donnees2 = reactive({ donneesb = donnees()  
  
    # Application du filtre si non null -----  
    if( !is.null( filtres()$select_model ) )  
      donneesb = donneesb %>% filter( model %in% filtres()$select_model )  
  
    donneesb })  
  
  mpgman2 = reactive({ donnees2() %>% dplyr::count(class, year) })  
  mpgman3 = reactive({ donnees2() %>% group_by(manufacturer) %>%  
    summarise(n = n(), unique = length(unique(model))) %>%  
    arrange(-n, -unique) })  
  
  # Graphiques -----  
  output$graph_mod1 = renderHighchart2({  
    donnees2() %>%  
    hchart("scatter", hcaes(x = displ, y = hwy, group = class)) %>%  
    hc_colors( palette(length(unique(donnees2()$class))) ) })  
  
  output$ graph_mod2 = renderHighchart2({  
    mpgman2() %>%  
    hchart( "column" , hcaes(x = class, y = n , group = year )) %>%  
    hc_colors( couleurs )  
  
  })  
  
  output$ graph_mod3 = renderHighchart2({  
    mpgman3() %>%  
    hchart( "treemap", hcaes(x = manufacturer, value = n, color = unique)) %>%  
    hc_colorAxis(minColor = "#FFFFFF", maxColor = couleurs[1])  
  
  })  
}
```

Etape 2b : Shiny Module 3/3 → Appels des fonctions module

- config
- data
- functions
- inputs
- modules
- outputs
- server
- ui
- www
- gestionnaire_habilitation.R
- global.R
- information.md
- restart.txt
- server.R
- ui.R

- info.txt
- SERVER_01_histogramme.R
- SERVER_02_circle_js.R
- SERVER_03_highcharts.R
- SERVER_04_visualisation_DT.R
- SERVER_05_dygraph.R
- SERVER_06_leaflet.R
- SERVER_07_tabsetpanel_mod.R

```
# Fonctions reactivives -----
liste_filtres = reactive({ list(select_model = input$ select_model_mod ) })

mpg_auto = reactive({ mpg %>% filter( str_extract( trans, "auto" ) %in% "auto" ) })
mpg_manual = reactive({ mpg %>% filter( str_extract( trans, "manual") %in% "manual") })

# Appel des modules server -----
callModule( Module_SERVER, "Auto"
            , donnees = mpg_auto , filtres = liste_filtres, couleurs = c( palette_edf2014$orange
                                                                           , palette_edf2014$orangefonce ) )

callModule( Module_SERVER, "Manual"
            , donnees = mpg_manual, filtres = liste_filtres, couleurs = c( palette_edf2014$bleu
                                                                           , palette_edf2014$bleufonce ) )
```

- info.txt
- UI_00_filtres.R
- UI_00_menu_lateral.R
- UI_01_page1.R
- UI_02_page2.R
- UI_03_page3.R
- UI_04_page4.R
- UI_05_page5.R
- UI_06_page6.R
- UI_07_page7.R
- UI_97_page_informations.R
- UI_98_retour_portail_IDEE.R
- UI_99_bouton_bookmark.R

```
liste_tabitems_body[[1 + length(liste_tabitems_body)]] <-
  tabItem(
    tabName = "mod",
    value = "Module",
    fluidRow(
      box(
        title = "Exemple module",
        status = "info",
        width = 12,
        solidHeader = TRUE,
        collapsible = FALSE,
        tabsetPanel(
          tabPanel("Automatiques", Module_UI("Auto" )),
          tabPanel("Manuelles" , Module_UI("Manual"))
        ) ) )
```

Perspectives

Ajout dans une prochaine version



Logs pour faciliter le débogage et améliorer le tracking



Nouveaux exemples de graphiques



Export des graphiques



Améliorations suite à retours d'expérience des utilisateurs

Merci