



# **EXERCICES SHINY**



## Avant de commencer...

Créer un répertoire de travail pour l'après-midi.

Y extraire le fichier « formation\_shiny\_ingestat.zip » envoyé.

Tester les commandes du fichier prerequis.R (en changeant le chemin)



Aide dans R de la fonction « `machin` » : `?machin`

R est sensible à la casse

`??Machin`



## Exercice 0 : Créez votre première application

- ❖ Ouvrir Rstudio
- ❖ File > New File > Shiny Web App
- ❖ Application Name : **App0** (cela créera un dossier)  
Application Type : Multiple File  
dans votre répertoire de travail
- ❖ Cliquer sur Run App
- ❖ Et voilà !
- ❖ Observer la structure du fichier ui correspondant à l'application et comprendre les composantes.




## Exercice 0 : Créez votre première application

- ❖ On va travailler avec nos données : charger les données `datasncf_prepare.csv` et changer le fichier appelé dans `server.ui`.


 `dt<-read.csv("../datasncf_prepare.csv")`

- ❖ Comment être sûr que j'appelle la bonne variable ?
- ❖ Changer le type de graphique. Ex. nuage de points entre deux variables

 `plot`



## Exercice 1 : Modification de l'interface

- ❖ Fermer les fichiers `server.R` et `ui.R` puis ouvrir les fichiers `server.R` et `ui.R` dans `/App1`
- ❖ Modifier l'application pour permettre à l'utilisateur de changer le titre du graphique.  
 `textInput`  
`plot(x, y, main="mon super titre")`
- ❖ Ajouter un `selectInput` pour choisir la variable `y`.



## Exercice 2 : On complique (un peu)...

- ❖ Fermer les fichiers `server.R` et `ui.R` puis ouvrir les fichiers `server.R` et `ui.R` dans `/App2`
- ❖ Ajouter un slider pour choisir l'année des données dans le graphique.




`sliderInput`  
`subset`



## Exercice 3 : On complique encore...

- ❖ Fermer les fichiers `server.R` et `ui.R` puis ouvrir les fichiers `server.R` et `ui.R` dans `/App3`
- ❖ Ajouter dans l'interface un nouvel onglet dans lequel on affichera les données correspondant à l'année choisie.

 `tabPanel`  
`tableOutput`

+ en extra : essayer `dataTableOutput`



## Exercice 4 :

### C'est pas très optimal tout ça !

- ❖ Fermer les fichiers `server.R` et `ui.R` puis ouvrir les fichiers `server.R` et `ui.R` dans `/App4`
- ❖ Utiliser l'environnement `reactive` pour ne "créer" qu'une fois le sous-jeu de données correspondant à l'année.



`reactive`


l'environnement `reactive` retourne une fonction...





## Exercice 5 : Et ce ui bien fainéant...

- ❖ Fermer les fichiers `server.R` et `ui.R` puis ouvrir les fichiers `server.R` et `ui.R` dans `/App5`
- ❖ Utiliser l'environnement `observe` pour charger automatiquement les noms de colonnes dans le `selectInput` pour choisir la variable `y`.

 `observe`  
`colnames`  
`updateSelectInput`

Attention à l'écriture du `shinyServer(function...`



## Mise en ligne

`library(rsconnect)`

- ❖ Je me connecte à mon compte sur <https://www.shinyapps.io/>
- ❖ Suivre les instructions sur la page : <https://www.shinyapps.io/admin/#/dashboard>  
1 et 2 à ne faire que la première fois.
- ❖ Je m'assure que les données et les codes ui.R et server.R sont dans le même dossier.  
Sous-dossiers acceptés., fichiers css, js, ... acceptés dans un sous-dossier www

`rsconnect::deployApp('chemin/App7_publication')`

- ❖ J'attends que ça tourne et j'envoie le lien à des gens :  
[https://elisemaigne.shinyapps.io/app7\\_publication/](https://elisemaigne.shinyapps.io/app7_publication/)
- ❖ (attention c'est public)