

# Déployer des applications web R-Shiny

# Retour d'expérience démarche développement R et R Shiny ARS BFC

Volonté depuis plusieurs années de mettre à disposition des données utiles au pilotage :

- Mieux valoriser et mobiliser une donnée sous utilisée.
- Avoir une approche de la donnée moins figée et s'adapter aux enjeux et questions posées.
- Disposer d'une plateforme de diffusion de la donnée qui propose des méthodes d'analyse, modélisation, simulation.
- Automatiser la réponse à certaines demandes récurrentes pour redonner aux équipes statistiques du temps pour l'analyse.
- Rendre les développements et analyses statistiques réutilisables et d'automatiser leur mise à jour pour capitaliser sur les réalisations ponctuelles.

Des outils susceptibles d'apporter des réponses à ces besoins existent (R-Studio et Shiny) mais :

- Nécessitent, outre la formation de l'équipe, l'investissement dans une infrastructure (serveur).
- L'hébergement des outils et données doit se faire en environnement sécurisé et avec une gestion des droits d'accès.

## Quelques exemples de réalisations

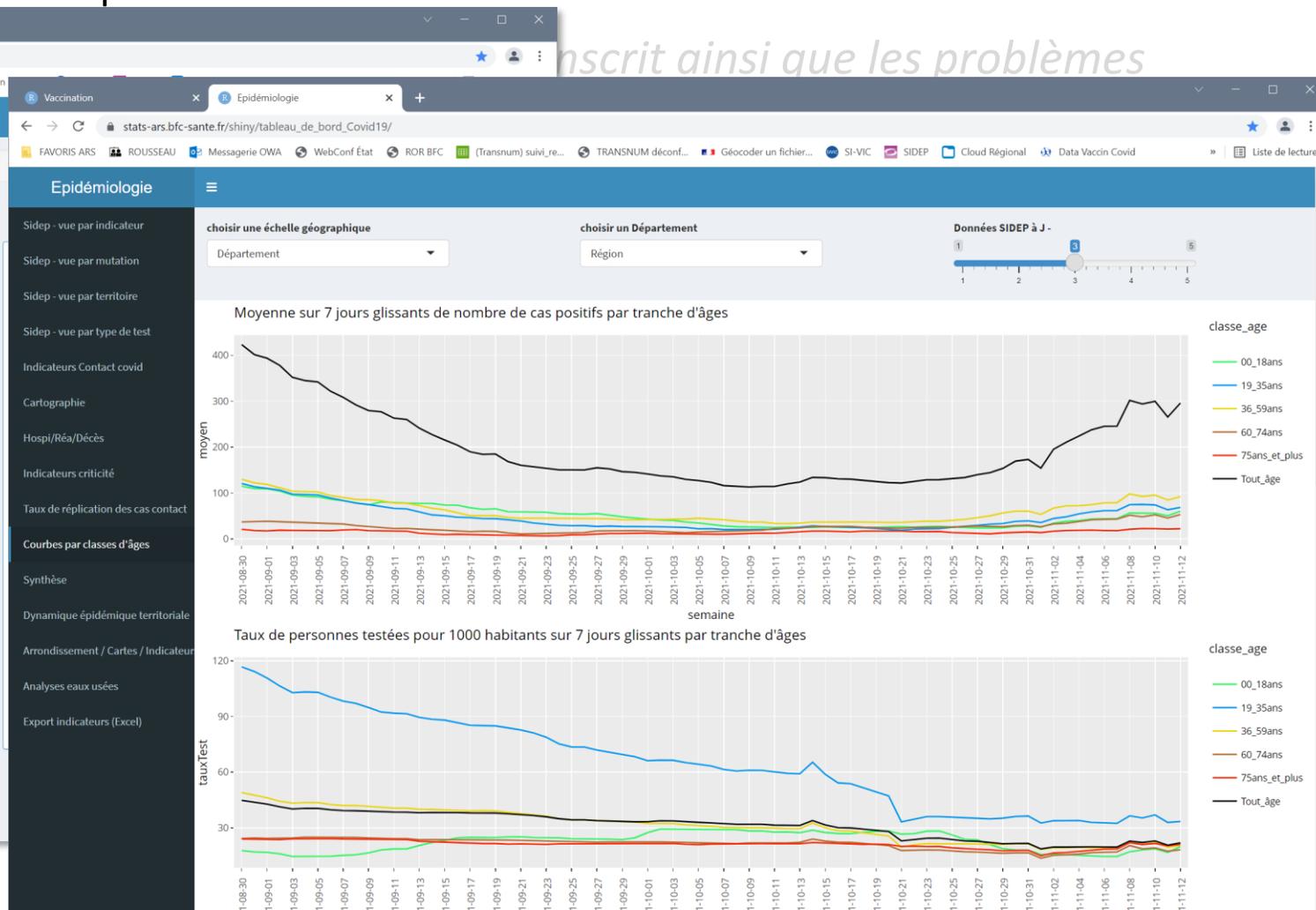
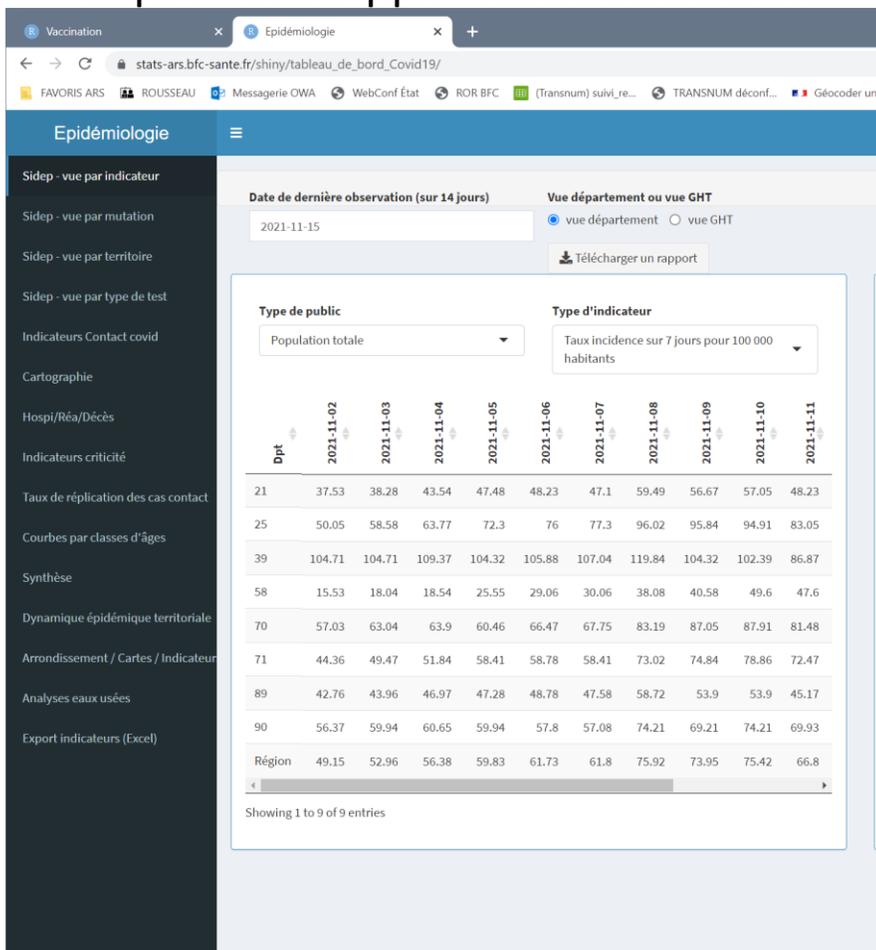
Exemple 1 : développement d'outils de suivi de l'épidémie de Covid-19 et de la vaccination :

Besoin pour l'ARS et les préfectures d'outils de suivi :

- à jour en temps réel,
- s'adaptant à différentes phases de l'épidémie ayant chacune ses enjeux évoluant,
- prenant en charge des données issues de systèmes d'information eux-mêmes évolutifs,
- répondants aux besoins de différents métiers : du contact-tracing aux décideurs,
- permettant l'export de tableaux, diaporamas, synthèses,
- sans mobiliser l'équipe statistique pour l'alimentation ou des traitements récurrents.

# Quelques exemples de réalisations

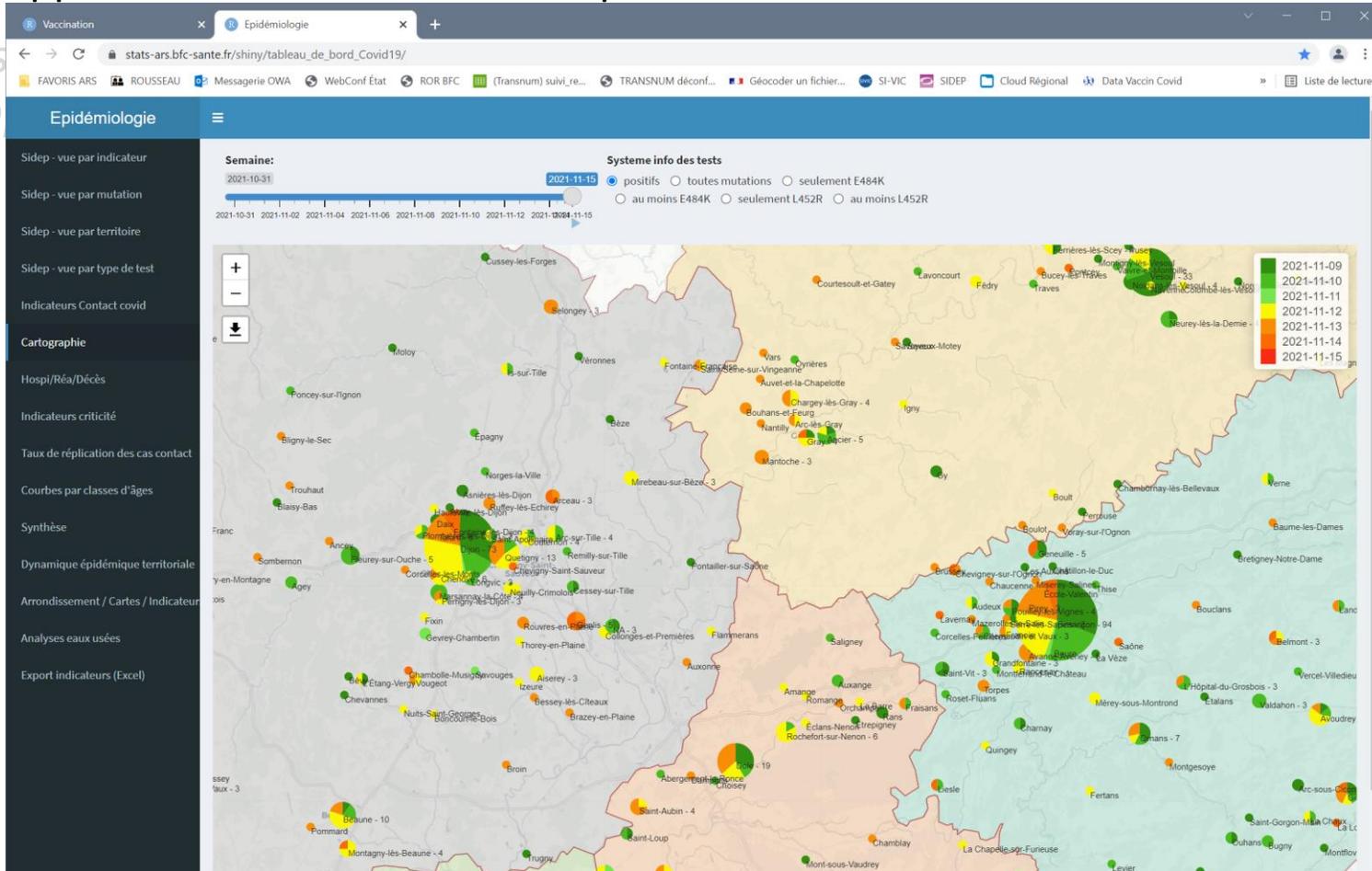
Exemple : développement d'outils de suivi de l'épidémie de Covid-19 et de la vaccination



# Quelques exemples de réalisations

Exemple : développement d'outils de suivi de l'épidémie de Covid-19 et de la vaccination

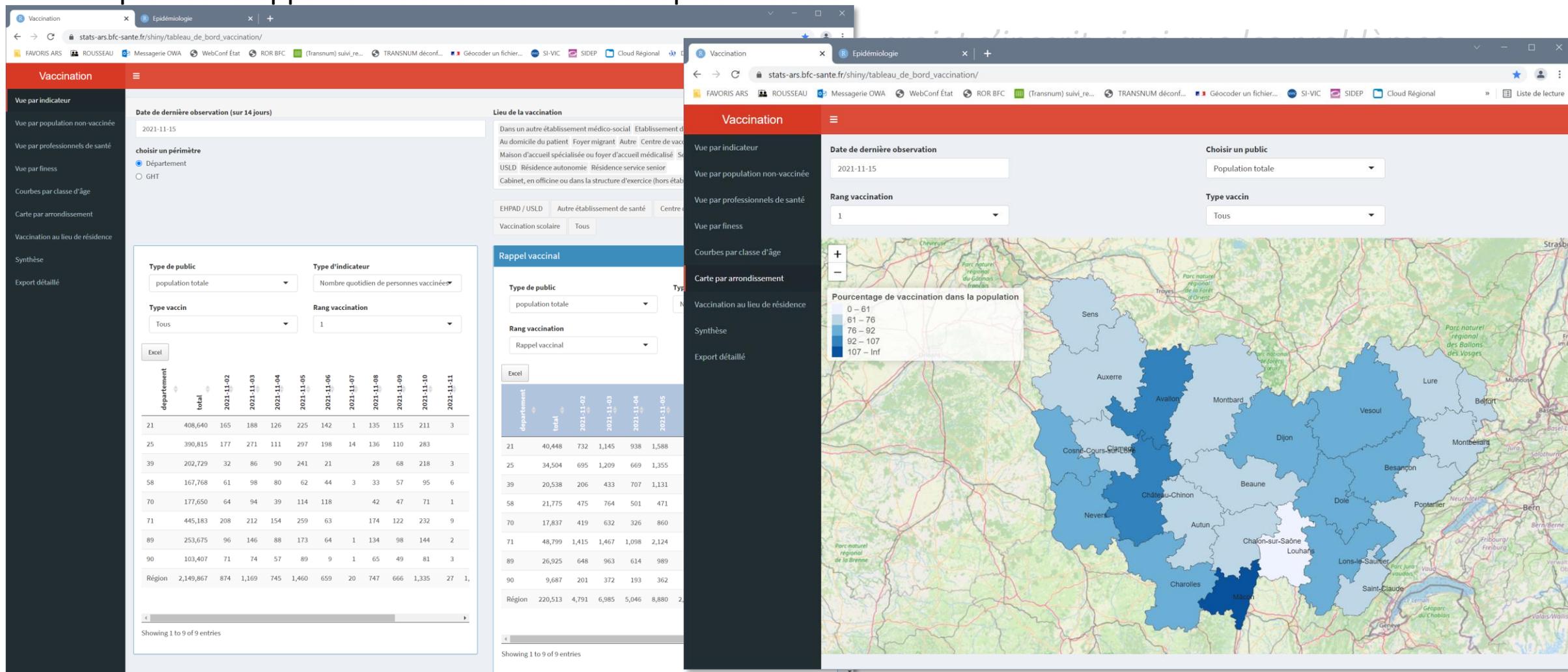
*Décrivez-nous  
auxquels il ap*



*les problèmes*

# Quelques exemples de réalisations

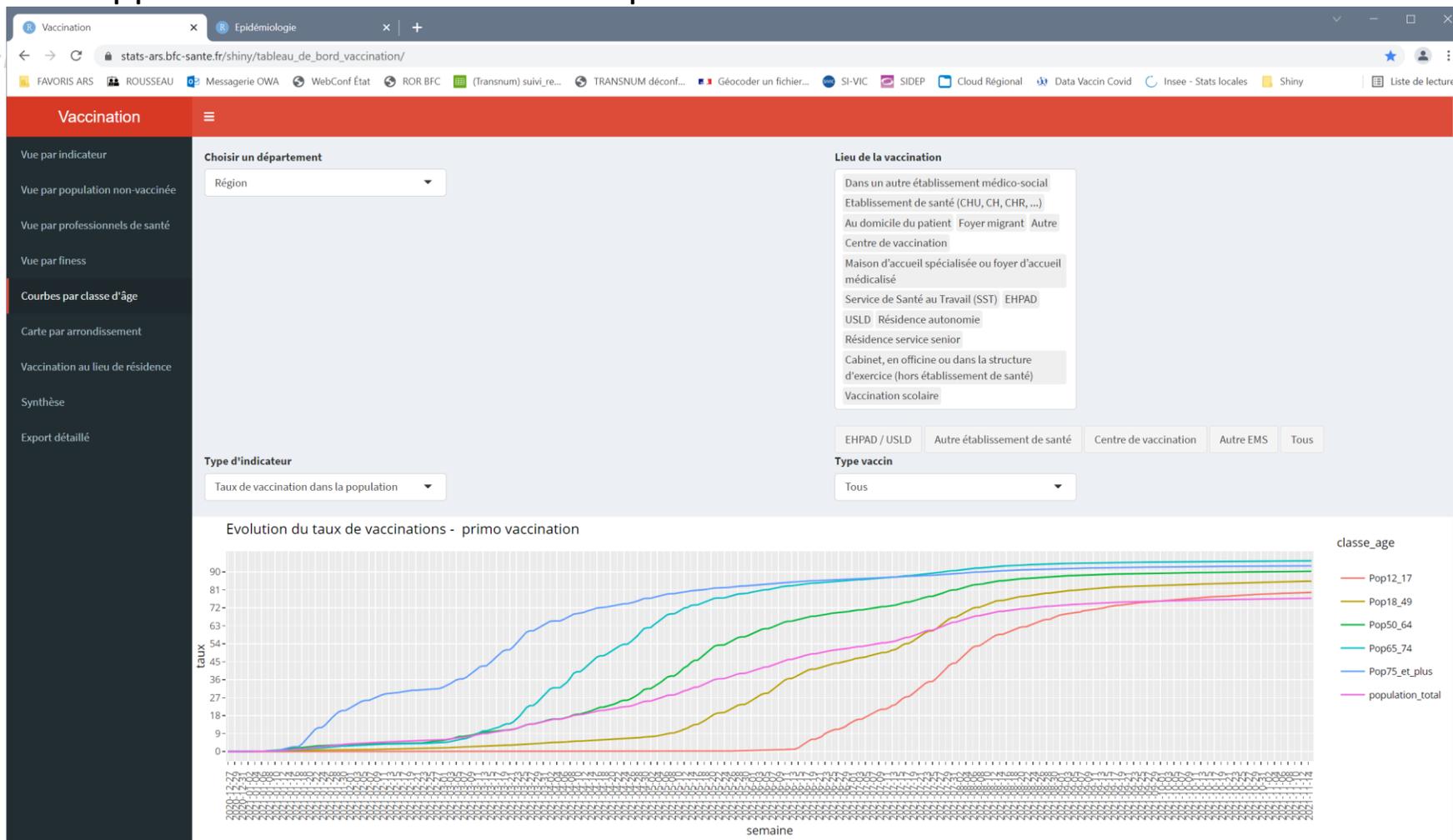
Exemple : développement d'outils de suivi de l'épidémie de Covid-19 et de la vaccination



# Quelques exemples de réalisations

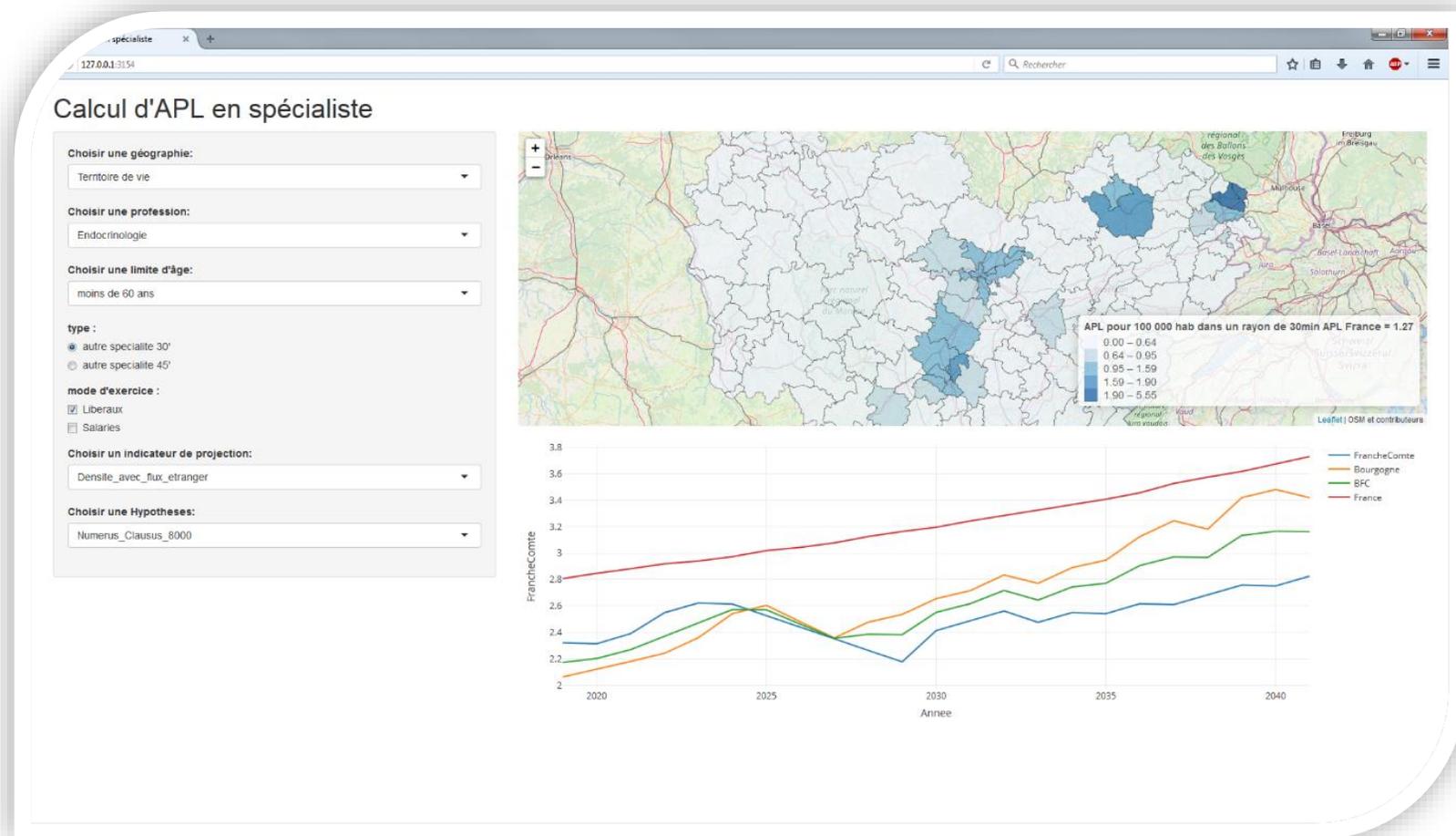
Exemple : développement d'outils de suivi de l'épidémie de Covid-19 et de la vaccination

Décrivez-  
auxquels



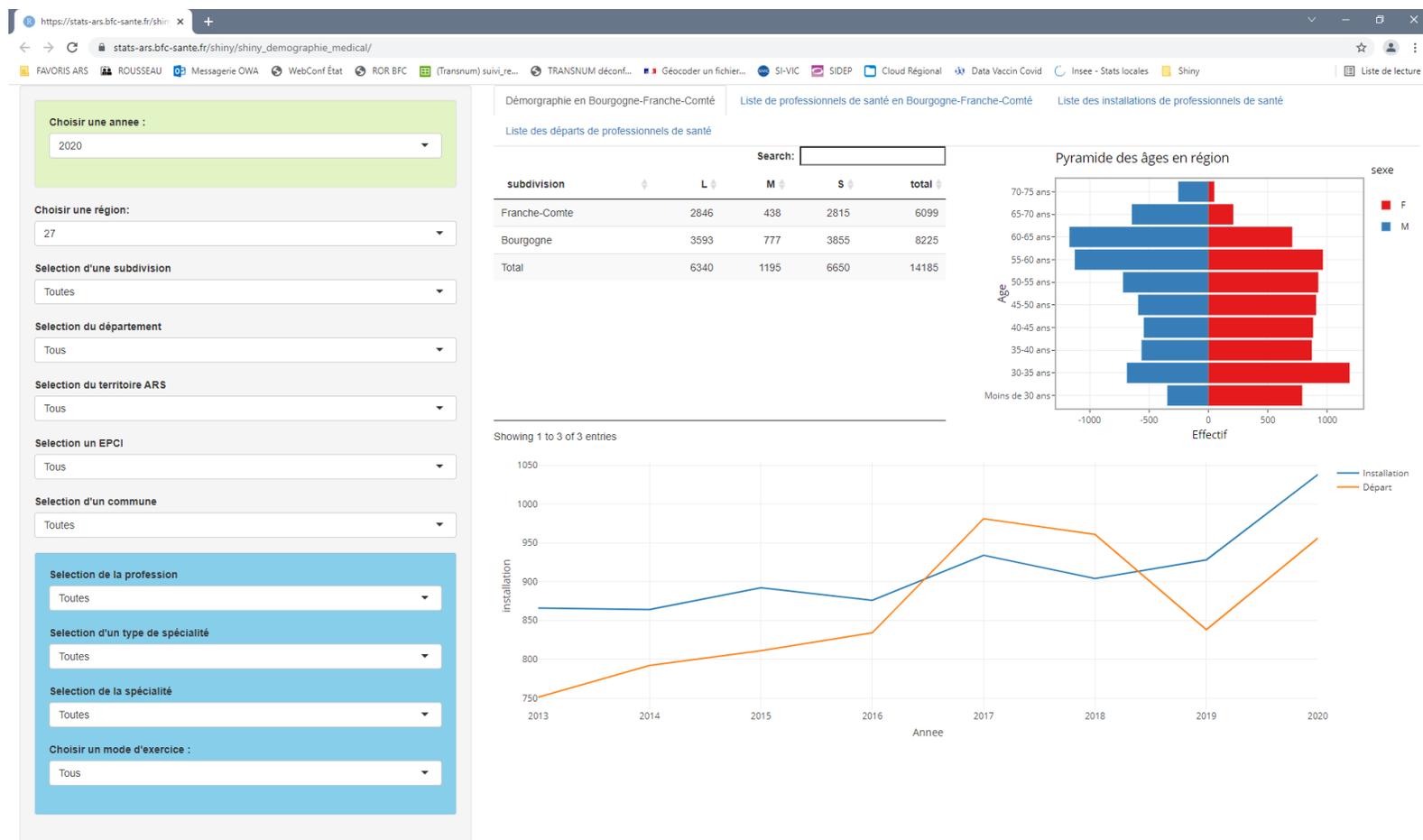
# Quelques exemples de réalisations

Exemple 2 : outil proposé aux Délégation Départementales à l'appui de leurs échanges avec les représentants des professionnels pour simuler l'effet sur l'accessibilité potentielle localisée (APL) de l'ajout de ressources en spécialistes



# Quelques exemples de réalisations

Exemple 3 : besoin en outils diagnostiques sur différentes thématiques (offre et recours aux soins, état de santé, contexte territorial)



## L'importance de la formation : la démarche de l'ARS

Former un collectif de statisticiens développeurs pour :

- De partager des bonnes pratiques et habitudes de développement au sein de l'équipe ARS puis avec d'autres partenaires.
- De mettre en commun les connaissances.
- D'avoir une progression des compétences homogène au sein de l'équipe.
- De disposer d'un pôle de compétence en développement statistique au services de projets mutualisés.

## Développement R Shiny : apparente simplicité mais...

- Très facile et rapide de créer une interface utilisateur
- Tout l'enjeu est l'articulation entre cette interface et le code de traitement des données
- Gérer la réactivité, assurer la fluidité de la réponse de l'application
- Enjeu de la maintenance du code, en particulier pour des développements en équipe : création de packages, documentation
- Envisager Git pour faciliter le travail en équipe

# La question de l'infrastructure : comment déployer une application R Shiny ?

## S'appuyer sur une plateforme d'hébergement comme Shinyapps.io de Rstudio :

Très simple

(publication directement depuis

L'IDE Rstudio)

Mais tarif dépend de la

disponibilité attendue (à évaluer)

Impossible d'héberger des données

« sensibles »

The screenshot shows the Shinyapps.io pricing page with five plans. The 'BASIC' plan is highlighted in blue. The plans are as follows:

Plan	Price	Applications	Active Hours	Key Features
FREE	\$0 /month	5 Applications	25 Active Hours	Community Support, RStudio Branding
STARTER	\$9 /month (or \$100/year)	25 Applications	100 Active Hours	Premium Email Support
BASIC	\$39 /month (or \$440/year)	Unlimited Applications	500 Active Hours	Performance Boosts, Premium Email Support
STANDARD	\$99 /month (or \$1,100/year)	Unlimited Applications	2,000 Active Hours	Authentication, Performance Boosts, Premium Email Support
PROFESSIONAL	\$299 /month (or \$3,300/year)	Unlimited Applications	10,000 Active Hours	Authentication, Account Sharing, Performance Boosts, Custom Domains, Premium Email Support

# Installer son propre serveur : ce qui est requis

Un serveur Linux (Ubuntu si possible)

Ou plusieurs : environnement production/développement

Quelques applications (solution libre, solution non libre)

Attention :

Nécessite compétences en administration Linux, administration réseau, infrastructure réseau, etc...

# Installer son propre serveur : solution logicielle « libre »

- Shiny server opensource sur le serveur,
- Développement sur les postes individuels : Rstudio desktop...
- ... et/ou développement sur une plateforme centralisée Rstudio server.
- Publication du code via Git. (Gitlab gratuit jusqu'à 5 utilisateurs par projet)

## Avantages :

- Coût réduit : outils open source gratuits
- Si le développement se fait sur Rstudio server : **environnement de développement standardisé, homogène (packages installés, version...)**.

## Inconvénients :

- **Si le développement se fait sur les postes individuels l'environnement de développement n'est pas standardisé.**
- Pas d'authentification individualisée des utilisateurs des application Shiny.
- Sécurisation peut nécessiter des compétences avancées en administration réseau : reverse proxy, filtres.
- Administration serveur plus complexe qu'avec les outils « propriétaire » pour s'assurer régulièrement du bon fonctionnement du processus Shiny lancé pour chaque vue et partagé par tous les utilisateurs.

# Installer son propre serveur : solution logicielle « propriétaire »

## R studio Team standard incluant :

- **Rstudio Workbench (anciennement Rstudio server pro)**, l'IDE dans lequel les utilisateurs peuvent travailler et développer.  
La version PRO contient 2 fonctionnalités essentielles : le log de ce qui est effectué sur la plateforme d'analyse (qui fait , quoi et quand, sur quelles données) et la possibilité de choisir la version du moteur R utilisé pour chaque projet.
- **Rstudio Connect (anciennement Shiny server pro)**  
outil de déploiement de "produit de la data science" (Shiny, Rmd, JupyterNotebook, Python, API ...) : permet de déployer en un clic et de rendre disponible ce qui est diffusé de manière aisée avec une gestion fine des droit d'accès, une fois le LDAP/SSO configuré.
- **Rstudio Package manager**  
permet de gérer les packages (externes et internes), simplifie grandement la communication entre Connect et workbench.

## Avantages :

- **Environnement de développement standardisé**
- Administration simplifiée du serveur Shiny (gestion des packages simplifiée)
- **Centralisation des données et référentiels utilisés** : les développeurs n'ont pas à avoir des copies de données sur leurs postes pour préparer, tester le code.
- **Possibilité d'exécuter les traitements sur une machine puissante et disposant d'une quantité de mémoire vive importante**
- **Contrôle de l'accès à la donnée, gestion des utilisateurs** : authentification individualisée des utilisateurs des application Shiny
- **Outils d'administration et de monitoring avancés** permettant d'identifier les traitements gourmands en ressources, de suivre l'utilisation des applications mises à disposition
- **Accès à un support**

## Inconvénients :

- Coût plus important : prévoir une licence d'environ 30000€ par an pour 10 développeurs/100 utilisateurs, ou 50000€ pour 10 développeurs/500 utilisateurs.

## Ou un serveur mutualisé ?

Solution à envisager compte tenu :

- Du coût,
- Des compétences nécessaires et du temps pour l'administration
- Des contraintes de sécurité
- Et pour favoriser des synergies

# Après la mise à disposition d'applications, de données, c'est fini ?

Accompagner l'utilisateur

Comment est produite la donnée :

quelles sources, leur contraintes et limites,  
quelles méthodes

Que nous apprend le résultat, qu'est ce qu'il ne nous apprend pas ?

Boucles de rétroaction : interroger le besoin de l'utilisateur, améliorer les applications

# Pourquoi tout ceci favorise coopération

Investissements mutualisables : formation, infrastructure, logiciel (si solution non libre)

Collectif de développement : entraide, appui, développeurs interface/aspects métiers... (Git)

Si on développe des interface web c'est avec l'idée d'ouverture : interne/externe

Accompagner l'utilisateur