Ma science sur le web

Développer des applications web en recherche à destination de différents publics

Corentin M. Barbu

INRAE et Ecole Doctorale ABIES (réseau ADUM)

9 avril 2025

Introduction

Objectifs

- Voir les outils de publications sur internet et comprendre leurs spécificités

Programme

- présentations des possibilités de mise en ligne suivant la complexité du projet
- application minimaliste fonctionnant en local
- mise en ligne de l'application minimaliste sur un serveur

Note: les exemples seront tous en R mais les mêmes outils sont souvent aussi disponibles sous Python

3 GRANDS TYPES DE MISES EN LIGNE

SITE INTERNET (PLUS OU MOINS INTERACTIF)

- Une collection de documents fixes : Quarto (Rmarkdown) . . .
  ex: https://mocoriba.fr/
- . . . mais des graphiques peuvent être interactifs (plotly) . . .
  ex: https://plotly.com/r/getting-started/
- . . . la page peut même contenir de mini applications (shiny)
  ex: https://quarto.org/docs/interactive/shiny/

UNE APPLICATION WEB

- Une interface qui communique avec un serveur . . .
  . . . pour un contenu interactif. ex: https://ccexplorer.eu

UNE API (PLUMBER)

- Permettre la consultation de bases de données, plus ou moins traitées : ex: Swagger MoCoRiBA:
  https://mocoriba.fr/mocoribaAPI/__docs__/

Section 1

SPÉCIFICITÉS DES TROIS SOLUTIONS

Quarto

Note: Si Rmarkdown (.Rmd) existe encore, il est progressivement remplacé par Quarto (.qmd)

Comment ça se présente

- Quarto en ligne de commande
- Quarto dans R (python)
- aide de quarto

Avantages et inconvénients

- qmd $\rightarrow$ pdf ou html
- Organisation générale
- tables des matières automatiques
- Recherche dans les billets de blog et navigation blog facilitées

$\times$ les html générés sont très lourds

Shiny

Comment ça se présente

- Un fichier app.R (dans le cas le plus simple)
- 2 fonctions : ui et server
- D'autres choses avant qui définissent tout ce que l'on veut en R simple
- possibilité d'inclure:
  - html et javascript
  - appel à des API externes

Avantages et inconvénients

- Extrêmement flexible
- Relativement simple pour une application web (ui/server)
- × Assez lent
- × Plus complexe que R habituel, surtout sensible pour le débuggage

Plumber

Comment ça se présente

- Un fichier .R chargé dans R (voir plumber.R)
- Génération automatique d'une documentation en ligne et interactive (Swagger)

Avantages et inconvénients

- Extrêmement flexible
- Très simple à prendre en main
- Très rapide et relativement simple à débugger

Section 2

Les programmes compagnons

La gestion des codes

Le protocole/programme ubiquitaire: git

- facilite la sauvegarde
- permet la restauration
- permet le développement en parallèle de plusieurs “branches”

La gestion des données

- Fichiers ouverts directement par R (rds/csv/etc.), data/
- Base clé-valeur : ex. REDIS
- Base de données relationelle : ex: PostgresSQL

Section 3

Le déploiement

Mise en ligne et passage à l'échelle

Solutions de mise en ligne

- Solutions d'hébergement d'applications (ex: SK8 https://sk8.inrae.fr/ ou Shinyapps.io)
- Hébergement cloud/"local" (ex: OVH)

  ⇒ serveur en ligne (linux ?)
  - nom de domaine libre (ovh, inrae)
  - installation de shiny server
    → sudo apt install shiny-server
  - placer son application au bon endroit
  - proxy (ex:nginx)
    ex: redirection ccexplorer

Passage à l'échelle

- hébergement d'application : payer
- serveur géré ex: docker swarm

Passage à l'échelle avec Docker et Traefik

- docker
  - permet la conteneurisation : “mini machines virtuelles”
  - un conteneur est de tailles beaucoup plus faible qu'une machine virtuelle
  - installations dans le conteneur indépendantes du système
  - déploiement ultra-rapide sur de nouveaux serveurs
- docker swarm
  - Permet la gestion de conteneurs multiples
- traefik
  - Permet de maintenir un utilisateur sur un même conteneur si besoin
  - Un conteneur additionnel par lequel passent toutes les requêtes

SÉCURISATION INTERNE À L'APPLICATION

- "purification" des chaînes de caractères entrées
  ex: `entreePropre<-gsub('[^a-zA-Z0-9]','',entree)`
- Voir plus généralement les recommendations OPSWAT
  https://www.opswat.com
- authentification/gestion de droits par token

Sécurisation serveur

Les classiques

- pare-feu : ufw (linux) + généralement fourni par hébergeur cloud en amont
- https (letsencrypt/certbot)
- Antivirus (inrae: https://www.withsecure.com/)
- fail2ban : surveillance de logs
- authentification (Basic Auth)

Subsection 1

Installation et mise à jour

LOGICIELS DE SYNCHRONISATION

- Graphique : ex: FileZilla
- en ligne de commande linux : rsync
  rsync -avz repSource/* user@server:repDest/

Section 4

Atelier

UN SITE DE DÉMONSTRATION

https://github.com/cbarbu/WebMyScience

Exercices supplémentaires possibles

- Générer un pdf avec Quarto
- Faire un blog Quarto (contents: posts dans l'en-tête)
- Faire une base de site Quarto (fichier _quarto.yml)
- ajouter un graphique plotly dans l'application
- ajouter une entrée texte pour requêter l'API
- Ajouter une fonction dans l'API : résultat de prédiction d'un modèle basé sur le jeu de données iris
- Démonstration de mise en ligne sur un serveur (avec serveur OVH et Nginx)