

Doryan MENDY  
Timothée LONCHAMPT

## SAE 3: Sujet de proba

## Sommaire

1. Introduction
2. Présentation Shiny
3. Rapport Général
4. Conclusion

## Introduction

Bienvenue dans ce rapport dédié à Doryan MENDY et Timothée LONCHAMPT, dans le cadre de l'étude du sujet de probabilité de la SAE 3. Ce document explore les fondements de Shiny, un package R révolutionnaire conçu pour simplifier la création d'applications web interactives. Notre objectif est de fournir une compréhension approfondie des concepts essentiels de Shiny, en mettant en lumière des ressources pédagogiques spécifiques. De plus, nous examinerons un exemple pratique intégré dans le contexte de votre projet, soulignant l'utilisation de Shiny pour créer une application interactive en lien avec une plateforme de ticketing interne.

## Présentation Shiny

### 1. Shiny Lesson - Première leçon

<https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html>

- Ce lien fait un point sur les bases de Shiny, y compris la mise en place de l'interface utilisateur (UI) et du côté serveur (server) dans une application Shiny.
- Elle pourrait aborder les éléments d'interface utilisateur tels que les boutons, les sélecteurs, et les sorties pour afficher des résultats.
- Les concepts de réactivité, d'observateurs (`observeEvent`), et de valeurs réactives (`reactive`) pour gérer les interactions de l'utilisateur.

exemple de notre code:

```
observeEvent(input$simulate, {  
  output$ticket_table <- renderTable({  
    simulate_tickets(input$utilisateur, input$technicien)  
  })  
})
```

Lorsque l'utilisateur clique sur le bouton "Simuler", la fonction `simulate_tickets` est appelée avec les paramètres choisis par actionnaire des boutons de l'utilisateur et du technicien, et le résultat est rendu sous forme de tableau dans l'interface utilisateur, dans la sortie "ticket\_table". Cela permet de simuler et d'afficher des tickets en fonction des sélections de l'utilisateur.

### 2. Tutoriel Shiny - ESIEE Paris

<https://perso.esiee.fr/~courivad/R/fr-16-shiny.html>

- Ce tutoriel permet de nous fournir une introduction à Shiny avec un focus sur la création d'applications interactives
- Il pourrait inclure des explications sur les mises en page, les éléments d'interface utilisateur, et la gestion des événements.
- Des exemples pratiques pour illustrer la création d'applications interactives avec Shiny.

exemple de notre code :

```
actionButton("simulate", "Simuler"),
```

En lien avec le code juste au dessus, lorsque l'utilisateur clique sur le bouton "Simuler", les graphiques et les tableaux associés seront mis à jour en fonction des sélections d'utilisateur et de technicien.

### 3. Introduction à Shiny - TUTO VISU

[https://lrouviere.github.io/TUTO\\_VISU/shiny.html](https://lrouviere.github.io/TUTO_VISU/shiny.html)

- Ce lien permet d'explorer l'utilisation de Shiny pour créer des visualisations interactives en R.
- Notre application pourrait couvrir la création de graphiques dynamiques, la manipulation de données réactives, et la personnalisation des mises en page.
- Des informations sur le déploiement d'applications Shiny, peut-être en utilisant des services comme Shinyapps.io.

## Rapport général

### Principaux Aspect

#### 1. Architecture interface utilisateur / serveur:

- L'architecture de Shiny repose sur une distinction claire entre l'interface utilisateur (vue) et le côté serveur (contrôleur). La vue gère la présentation et l'interaction, tandis que le serveur traite la logique et les calculs.

#### 2. Éléments d'interface utilisateur:

- Shiny propose une variété d'éléments d'interface utilisateur tels que `selectInput` pour les menus déroulants, `actionButton` pour les boutons d'action, et bien d'autres. Ces éléments facilitent l'interaction avec l'utilisateur

#### 3. Réactivité:

- La réactivité est l'un des principaux avantages de Shiny. Les fonctions réactives telles que `observeEvent` permettent de réagir dynamiquement aux actions de l'utilisateur, déclenchant des mises à jour en temps réel des sorties.

## Fonctionnalité de l'application:

1. **Simulation de Tickets** : L'utilisateur peut sélectionner un utilisateur et un technicien, puis simuler les tickets associés en cliquant sur le bouton "Simuler". Les résultats sont affichés sous forme de tableaux et de graphiques en bâton. Il peut aussi choisir le mois durant lequel il a accès aux nombres de tentatives de connexion.
2. **Tentatives de Connexion par Mois** : En cliquant sur le bouton "Afficher Tentatives par Mois", l'application affiche un graphique montrant le nombre de tentatives de connexion pour le mois sélectionné.
3. **Statistiques et graphiques Dynamiques** : L'application génère des visualisations interactives telles que la distribution des tickets en fonction de l'urgence, le temps moyen de résolution par niveau d'urgence, le nombre de tickets attribués à chaque technicien, etc.

## Conclusion

En conclusion de ce rapport consacré à l'étude du sujet de probabilité dans le cadre de la SAE 3, nous avons pu voir les fondements de Shiny, un package R puissant et révolutionnaire spécialement conçu pour simplifier la création d'applications web interactives. Ce rapport possède en trois étapes principales : l'introduction générale à Shiny, la présentation de ressources pédagogiques spécifiques, et enfin, un exemple pratique intégré dans le contexte de votre projet.