

Modèle d'IA avec R et Shiny

1. Présentation de Shiny
2. R vs Python

A. Interface Graphique

```
#install.packages("shiny")
library(shiny)
library(shinythemes)
library(ggplot2)

ui <- fluidPage(theme = shinytheme("cyborg"),

  headerPanel("Prédiction des prix des biens immobiliers à Paris"),

  sidebarPanel(

    textInput("hasYard", "Dispose d'une cour : ", ""),
    textInput("hasPool", "Dispose d'une piscine : ", ""),
    textInput("hasGuestRoom", "Dispose d'une chambre d'amis : ", ""),
    textInput("garage", "Dispose d'un garage : ", ""),
    textInput("hasStorageRoom", "A une pièce de stockage : ", ""),

    actionButton('go', "Prediction")

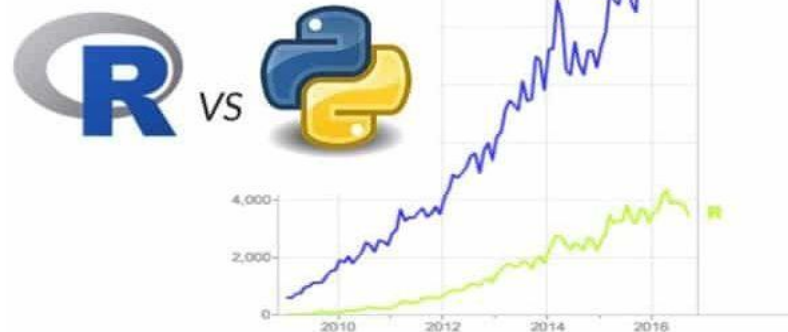
  ),

  mainPanel(
    sidebarPanel( width = 20,
      headerPanel("Prix estimé du bien immobilier: "),
      textOutput("value"),
```

B. Serveur

```
server <- function(input, output) {  
  output$distPlot <- renderPlot({  
    x <- data$made  
    bins <- seq(min(x), max(x), length.out = input$bins + 1)  
    hist(x, breaks = bins, col = '#E13131', border = "white",  
         xlab = "Année de construction",  
         main = "Evolution des construction entre 1990 et 2021")  
  })  
  
  #Plot Graph corrélation des prix  
  output$plot <- renderPlot({  
    plot(data$floors, data$price)  
  })  
  
  shinyApp(ui, server)
```

2. R vs Python



- Avec des fonctions telles que **lm**, **predict**, **R** permet de faire la plupart du travail ;
- **R** a plus de fonction 'built-in' pour analyser des données, **Python** s'appuie sur des librairies ;
- **Python** exécute du code **C** pour économiser les ressources mémoire ;
- Il est généralement plus **simple** de faire des tâches non statistiques en **Python** (**BeautifulSoup**, **requests...**) ;
- **R** est plus **fonctionnel**, **Python** est plus **orienté objet**.