Creació d'Aplicacions Interactives amb servidor Shiny

Curs R Avançat Equips - Sessió 5

Avui

Creació d'Aplicacions Interactives amb servidor Shiny, i conèixer l'existència d'algunes les altres opcions en R també.

1. Introducció a R Shiny

Aplicacions dinàmiques i interactives, amb components reactius, que requereixen de servidor shiny on col.locar les aplicacions.

- Shiny: https://www.rstudio.com/products/shiny/[1]

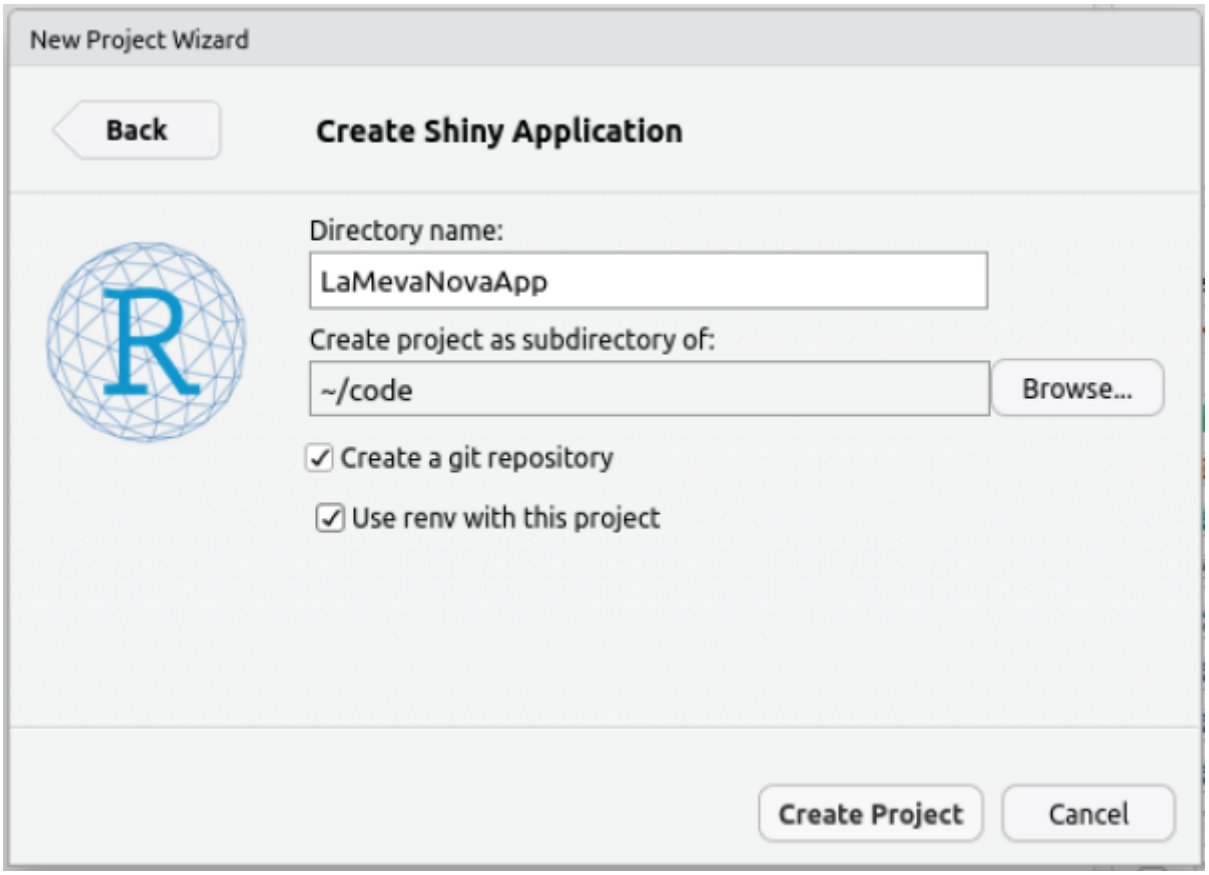
Veure alguna aplicació feta amb Shiny (requereix servidor de R i de Shiny), del tipus:

- Exemple senzill: https://dades.ajuntament.barcelona.cat/estadistiques-cens-comercial/[2]
- Exemple avançat: https://dades.ajuntament.barcelona.cat/la-ciutat-al-dia/[3]

Treball pràctic de creació d'un dashboard amb Shiny i modificació.
Es pot aprendre com fer-ne apps Shiny, a poc a poc, a través de:
https://mastering-shiny.org/basic-app.html[4]

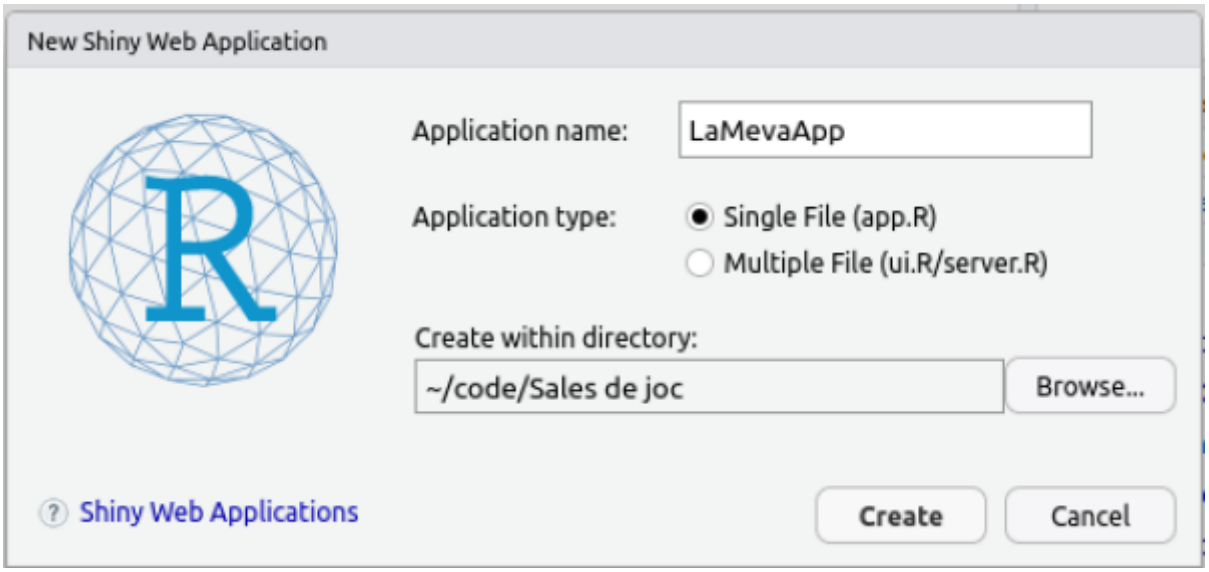
Es pot començar emprant una plantilla base que proporciona RStudio en fer un projecte nou de tipus Shiny App:

- File > New Project > New Directory > Shiny Web Application



O si es vol afegir dins un projecte de RStudio pre-existent:

- File > New File > Shiny Web App > New Shiny Web Application.



O es pot emprar una plantilla base a través del **ShinyUIEditor**, que empra a la seva vegada una disposició versàtil de tipus gridlayout
Build Shiny application UIs by dragging-and-dropping. Generates clean and proper code as you build.

1.1. Estructura d'una app shiny

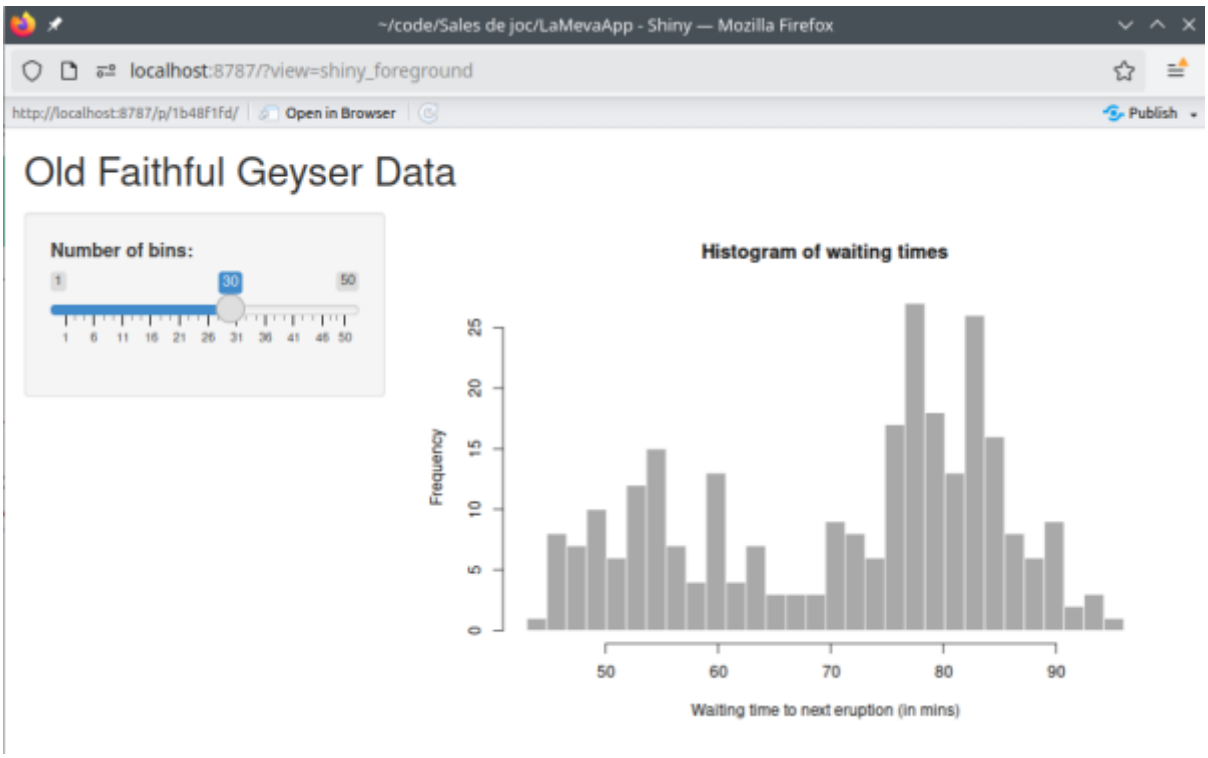
En crear una app shiny a partir d'un arxiu nou dins un projecte de Rstudio pre-existent, se'ns crea aquest arxiu `app.R`:

Contingut de l'arxiu app.R

```
#
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
#

library(shiny)

# Define UI for application that draws a histogram
ui <- fluidPage(

    # Application title
    titlePanel("Old Faithful Geyser Data"),

    # Sidebar with a slider input for number of bins
    sidebarLayout(
        sidebarPanel(
            sliderInput("bins",
                        "Number of bins:",
                        min = 1,
                        max = 50,
                        value = 30)
        ),

        # Show a plot of the generated distribution
        mainPanel(
           plotOutput("distPlot")
        )
    )
)

# Define server logic required to draw a histogram
server <- function(input, output) {

    output$distPlot <- renderPlot({
        # generate bins based on input$bins from ui.R
        x    <- faithful[, 2]
```

```
        bins <- seq(min(x), max(x), length.out = input$bins + 1)

        # draw the histogram with the specified number of bins
        hist(x, breaks = bins, col = 'darkgray', border = 'white',
             xlab = 'Waiting time to next eruption (in mins)',
             main = 'Histogram of waiting times')
    })
}

# Run the application
shinyApp(ui = ui, server = server)
```

En clicar al botó de "RunApp" se'ns aixeca una finestra emergent amb aquest contingut:



A partir d'aquí, podríem anar retocant el codi de l'app.R per fer evolucionar la app.

O bé, podem començar a partir del `shinyuieditor`, que ens permetrà començar ja amb una interfícies shiny molt més avançada per ajustar-se al que volíem per a la nostra app Shiny.

1.2. ShinyUIEditor

https://rstudio.github.io/shinyuieditor/[5]

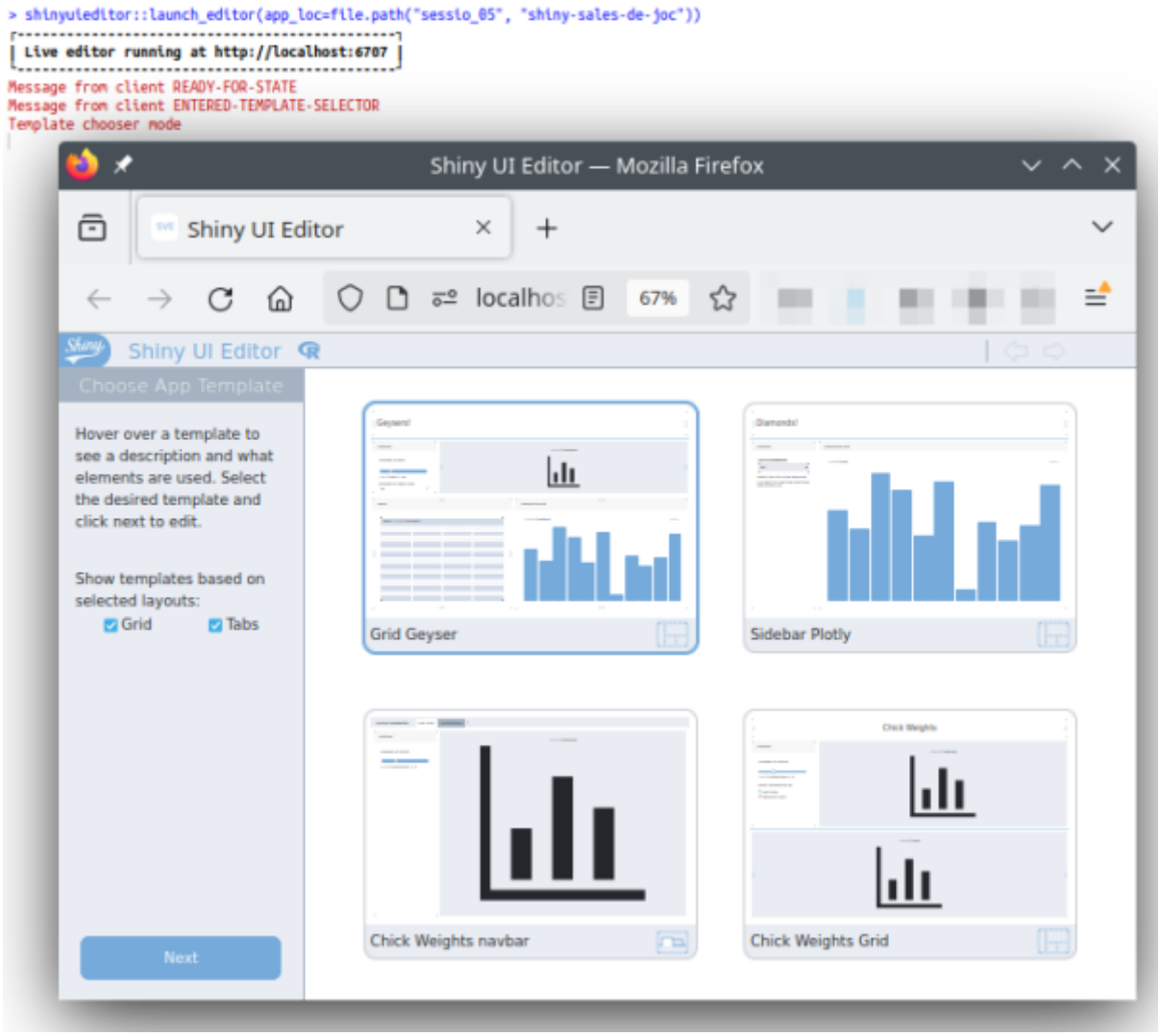
1.3. App demo shiny-sales-de-joc

S'ha creat una app demo de shiny, emprant les plantilles de shinyuieditor.

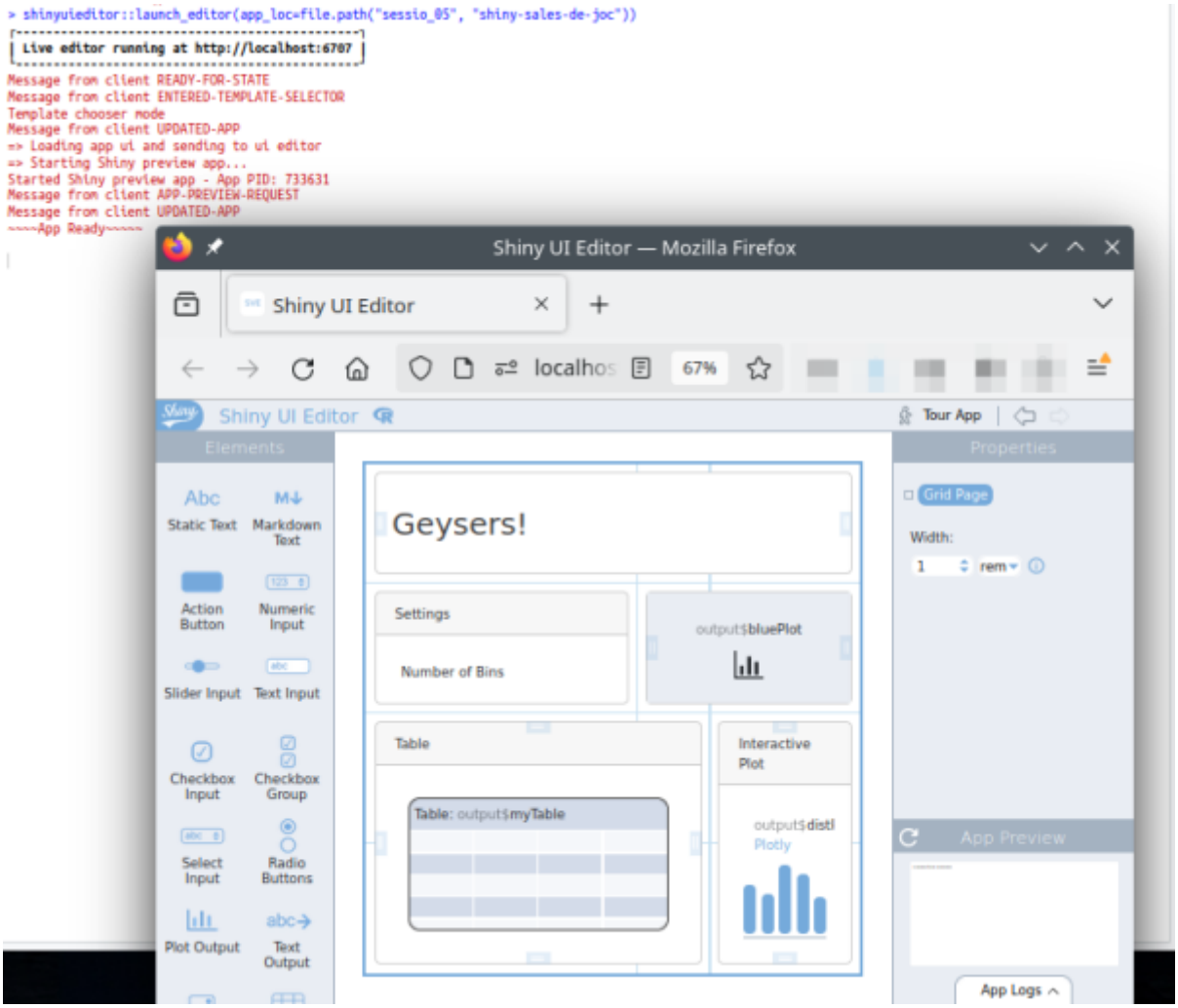
Per crear-ho des de zero, executem aquest tipus de comanda primera vegada:

```
shinyuieditor::launch_editor(app_loc=file.path("sessio_05", "shiny-sales-de-joc"))
```

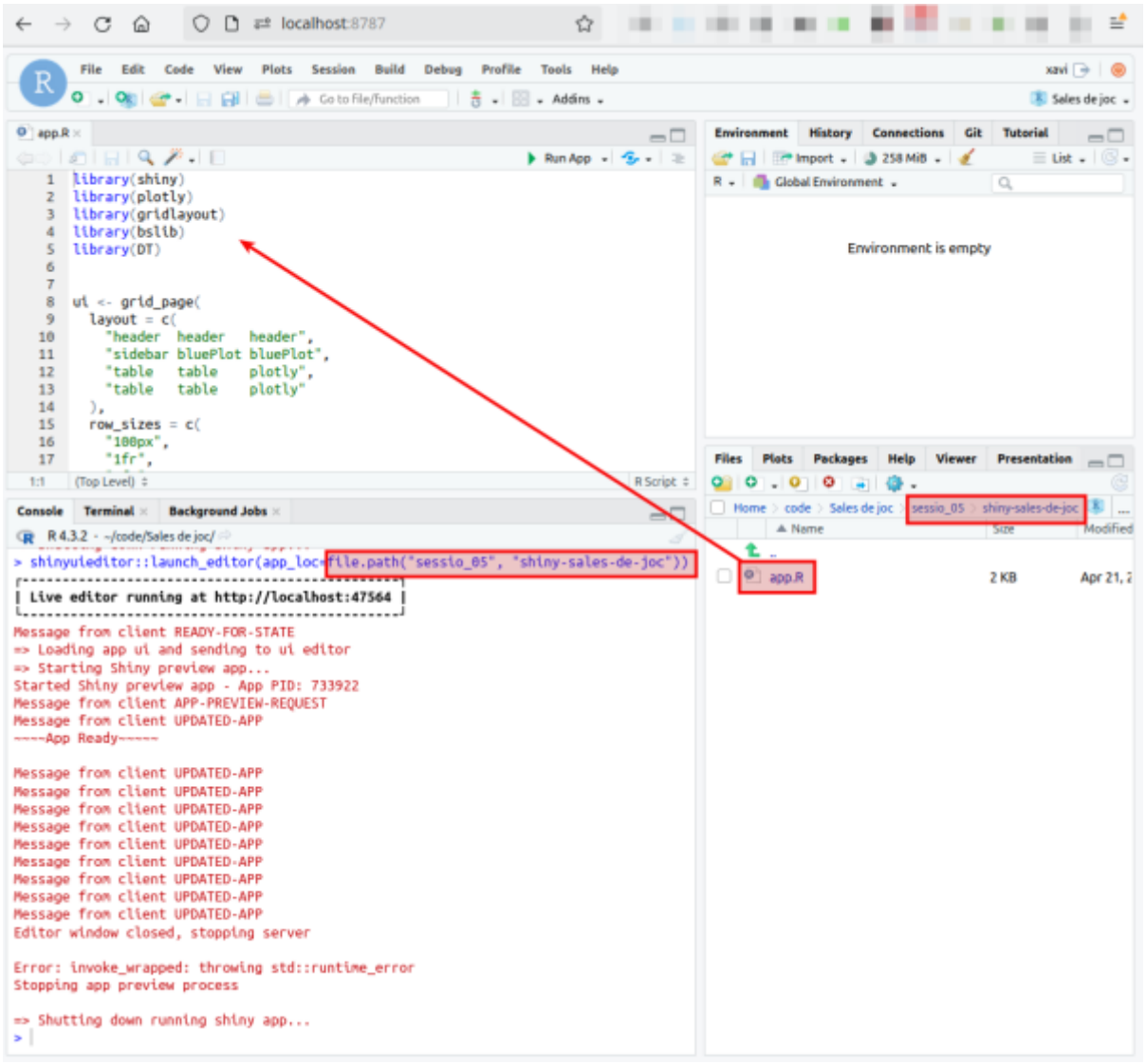
Això ens obrirà una finestra del navegador en que hi haurà l'aplicació d'edició gràfica d'interfícies d'usuari d'aplicacions shiny (shinyuieditor)



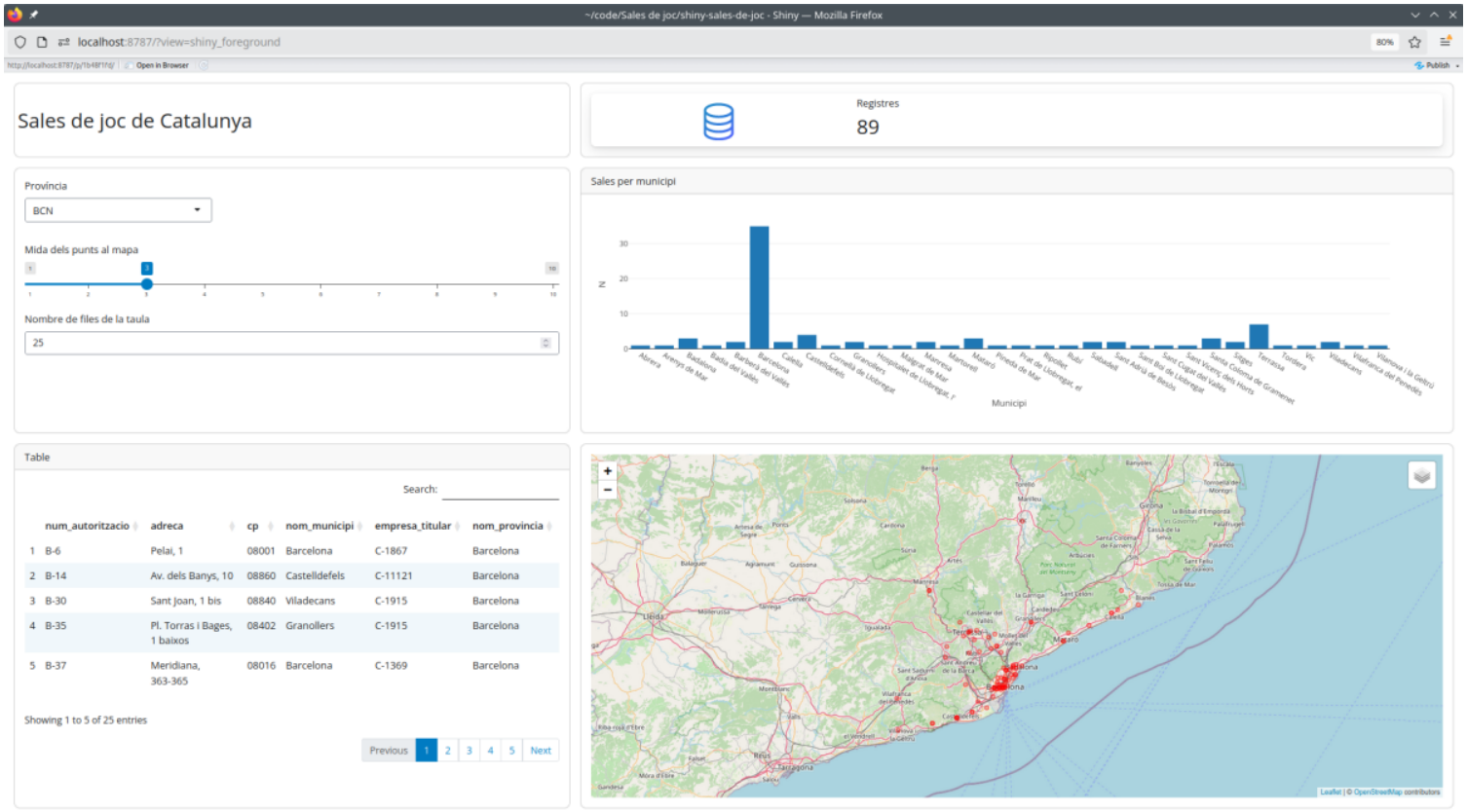
Escollim quina plantilla volem agafar de partida (per exemple, "Grid Geyser"), i cliquem al botó "Next" de sota a l'esquerra.
Llavors ja podem fer els canvis que necessitem a la nostra interfície.



En quant tanquem la finestra del navegador on hi havia el shinyuieditor, ens trobarem que tenim actualitzat l'arxiu "app.R" a la carpeta que havíem indicat inicialment, amb el codi actualitzat necessari per produir aquesta interfície d'usuari de Shiny.



Ara podem continuar canviant els conjunts de dades que venen per omissió sobre geishers (dataframe "**faithful**" al codi), i les columnes/variables d'interès, per les nostres.
I associar els elements nous de dades (dins la funció server()) als controls de la interfície que calgui, per adaptar la nostra nova aplicació shiny a les nostres necessitats.



Contingut de app.R

```
library(shiny)
library(plotly)
library(gridlayout)
library(bslib)
library(DT)
library(sf)
library(readxl)
library(dplyr)
library(stringr)
library(janitor)
library(leaflet)
library(leaflet.providers)

ui <- grid_page(
  layout = c(
    "header  header     indicator ",
    "sidebar sidebar    plotly  ",
    "table   table      mymap   ",
    "table   table      mymap   "
  ),
  row_sizes = c(
    "120px",
    "1.5fr",
    "1fr",
    "1fr"
```

```
  ),
  col_sizes = c(
    "290px",
    "0.59fr",
    "1.41fr"
  ),
  gap_size = "1rem",
  grid_card(
    area = "sidebar",
#    card_header("Paràmetres"),
    card_body(
      selectInput(
        inputId = "myProvincies",
        label = "Província",
        choices = list(
          "BCN" = "Barcelona",
          "GRN" = "Girona",
          "LLD" = "Lleida",
          "TCN" = "Tarragona"
        ),
        selected = "Barcelona"
      ),
      sliderInput(
        inputId = "myradius",
        label = "Mida dels punts al mapa",
        min = 1,
        max = 10,
        value = 3,
        width = "100%"
      ),
      numericInput(
        inputId = "numRows",
        label = "Nombre de files de la taula",
        value = 5,
        min = 1,
        step = 1,
        width = "100%"
      )
    )
  ),
  grid_card_text(
    area = "header",
    content = "Sales de joc de Catalunya",
    alignment = "start",
    is_title = FALSE
  ),
  grid_card(
    area = "table",
    card_header("Table"),
```

```
    card_body(DTOutput(outputId = "myTable", width = "100%"))
  ),
  grid_card(
    area = "mymap",
    card_body(leafletOutput("mymap"))
  ),
  grid_card(
    area = "indicator",
#    card_header("Nombre de Sales de Joc"),
    card_body(
      value_box(
        title = "Registres",
        value = textOutput(outputId = "indicator"),
        showcase = bsicons::bs_icon("database")
      )
    )
  ),
  grid_card(
    area = "plotly",
    card_header("Sales per municipi"),
    card_body(
      plotlyOutput(
        outputId = "distPlot",
        width = "100%",
        height = "100%"
      )
    )
  )
)


server <- function(input, output) {
  # Carreguem les dades dins una expressió reactiva (només s'executa una vegada,
fins que canviïn les dades)
  # Importa les dades
  dataset <- reactive({
    muni_cat <- rio::import(file.path("..", "muni_cat.csv"))
    st_as_sf(
      read_xlsx(file.path("..", "..", "curs-r-avancat-equips", "sessio_02",
"salons_de_joc_en_la_web.xlsx")),
      coords = c("longitud", "latitud"), crs = 4326, agr = "constant"
    ) %>% # Retoquem noms per a que encaixin tots al join de sota
    mutate(
        nom_municipi = case_when(
          cp == "08904" ~ "Cornellà de Llobregat",
          TRUE ~ nom_municipi),
        nom_municipi = str_replace_all(nom_municipi, fixed("La Jonquera"),
"Jonquera, la"),
        nom_municipi = str_replace_all(nom_municipi, fixed("L'Hospitalet de
```

```
Llobregat"), "Hospitalet de Llobregat, l'"),
        nom_municipi = str_replace_all(nom_municipi, fixed("El Prat de Llobregat"),
"Prat de Llobregat, el"),
        nom_municipi = str_replace_all(nom_municipi, fixed("Sant Vicenç del
Horts"), "Sant Vicenç dels Horts"),
        nom_municipi = str_replace_all(nom_municipi, fixed("Castelló d'Empúries"),
"Castelló d'Empúries"),
        nom_municipi = str_replace_all(nom_municipi, fixed("Castell-Platja d'Aro"),
"Castell d'Aro, Platja d'Aro i s'Agaró"),
        nom_municipi = str_replace_all(nom_municipi, fixed("El Vendrell"),
"Vendrell, el")
      )  %>%
      left_join(
        muni_cat %>% select(nom, nom_provincia),
        by=c("nom_municipi"="nom")
        )
    })

  output$distPlot <- renderPlotly({
    # generate bins based on input$bins from ui.R
    plot_ly(x = ~ dataset() %>% filter(nom_provincia %in% input$myProvincies) %>%
pull(nom_municipi), type = "histogram") %>%
      layout(xaxis = list(title = 'Municipi'),
             yaxis = list(title = 'N'))
  })
  output$mymap <- renderLeaflet({
    leaflet(width="100%", height="100%") %>%

      # OS map layer
      addProviderTiles(providers$Esri.WorldImagery, group="ESRI Satellite",
                       options=leafletOptions(maxNativeZoom=19,maxZoom=100)) %>%
      addProviderTiles("OpenStreetMap",
                       options=leafletOptions(maxNativeZoom=19,maxZoom=100)) %>%

      # Sample points
      # generate radius based on input$myradius from ui.R
      addCircleMarkers(data=dataset() %>% filter(nom_provincia %in%
input$myProvincies),
                       radius=input$myradius, weight=2, color="red") %>%

      # Add layer control elements
      addLayersControl(baseGroups = c("OpenStreetMap", "ESRI Satellite"),
                       options = layersControlOptions(collapsed = TRUE,
                                                      autoZIndex = F))

  })
  output$myTable <- renderDT({
    head(dataset() %>% tibble() %>%
           filter(nom_provincia %in% input$myProvincies) %>%
```

```
            select(-arxiu_origen,-ordre, -starts_with("municipi"), -cp_num, -
geometry),
          input$numRows)
  },
  options = list(dom='ftip',
                 lengthMenu = list(c(5, 10, -1), c('5', '10', 'All'))
                 )
  )
  output$indicator <- renderText(
    nrow(dataset() %>% tibble() %>% filter(nom_provincia %in% input$myProvincies))
  )
}

shinyApp(ui, server)
```

1.4. Publicar l'app a shinyapps.io

Pots emprar el teu compte de posit.cloud a https://www.shinyapps.io/[6]
O bé, fer-te un de nou de franc.

I segueix les instruccions que t'hi donaràn allà per tal de publicar la teva app shiny local al seu servidor.

Està explicat també pas a pas en aquest tutorial:
https://statsandr.com/blog/how-to-publish-shiny-app-example-with-shinyapps-io/[7]

Més informació a:
https://shiny.posit.co/r/articles/share/shinyapps/[8]

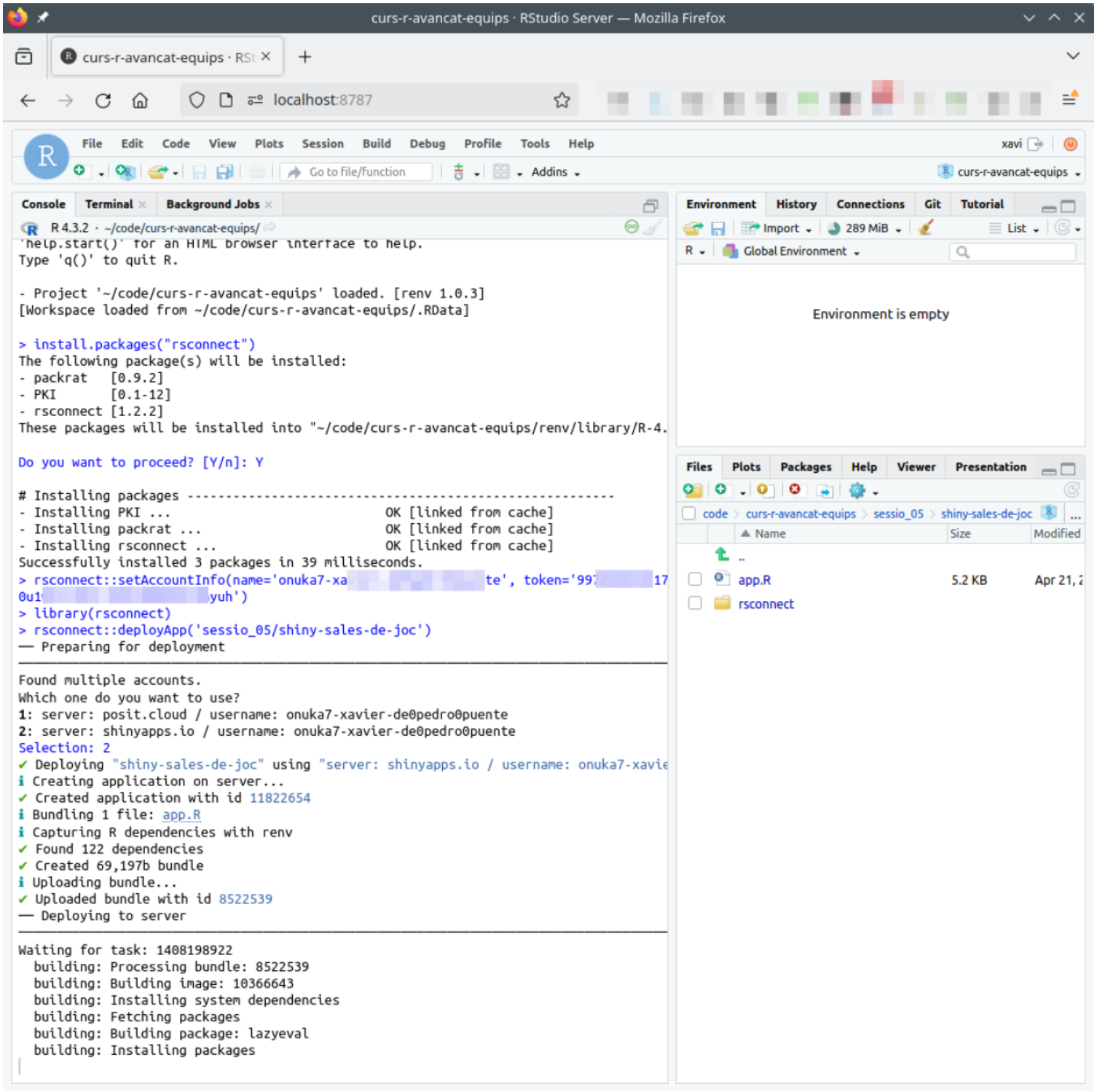
1.4.1. App Sales-de-joc

```
> install.packages("rsconnect")
The following package(s) will be installed:
- packrat   [0.9.2]
- PKI       [0.1-12]
- rsconnect [1.2.2]
These packages will be installed into "~/code/curs-r-avancat-
equips/renv/library/R-4.3/x86_64-pc-linux-gnu".

Do you want to proceed? [Y/n]: Y

# Installing packages --------------------------------------------------------
- Installing PKI ...                            OK [linked from cache]
```

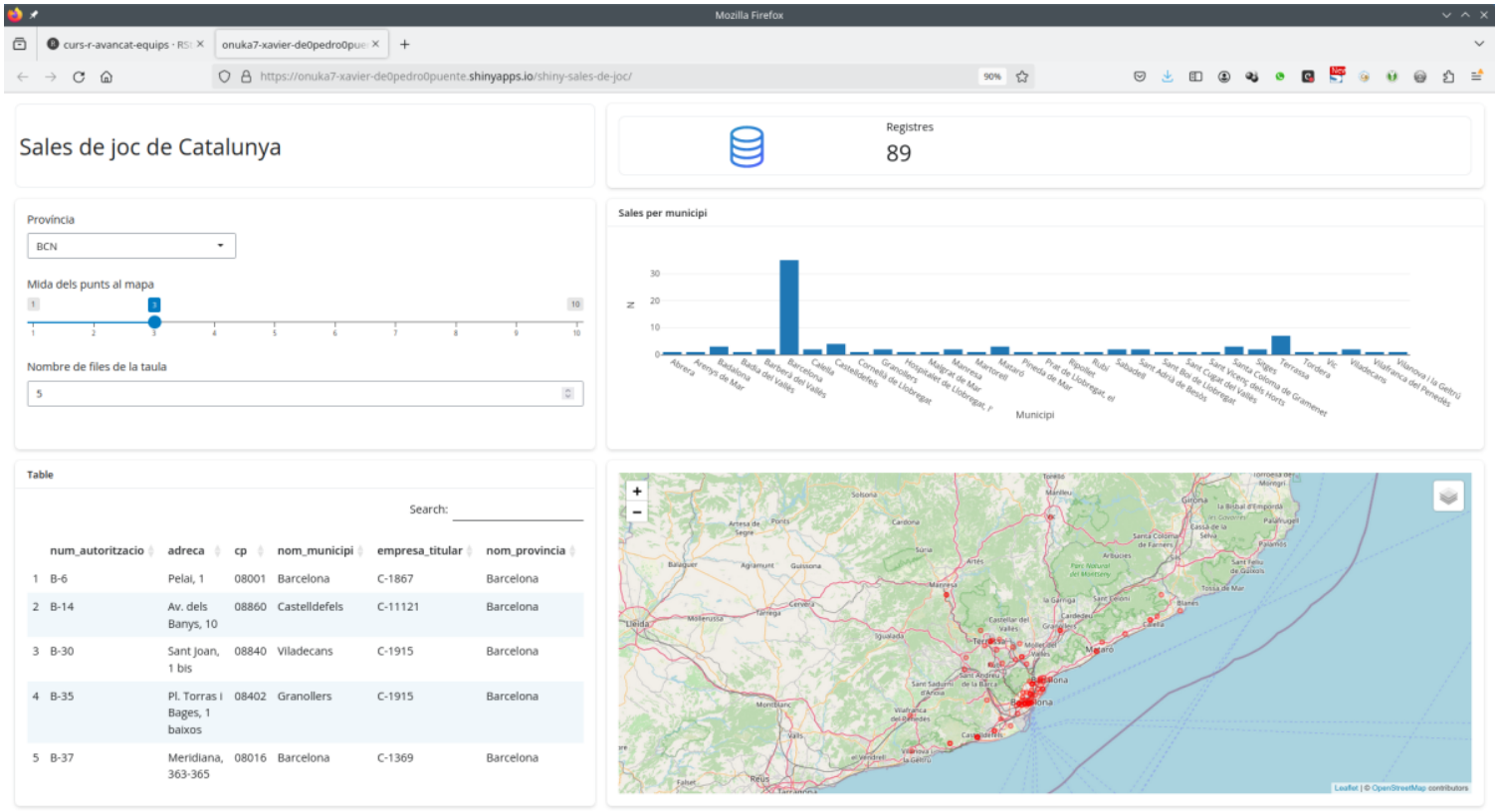
```
- Installing packrat ...                        OK [linked from cache]
- Installing rsconnect ...                      OK [linked from cache]
Successfully installed 3 packages in 39 milliseconds.
> rsconnect::setAccountInfo(name='myshinyiointernalusername', token='mytokenXXX',
secret='mysecretXXX')
> library(rsconnect)
```

Em faltaven els arxius de dades que s'agafaven de carpetes gernanes del projecte de RStudio, fora de la carpeta de l'app shiny. Els he copiat els 2 arxius dins de una nova carpeta filla "dades/", i he tornat a fer "Deploy":

```
> rsconnect::deployApp('sessio_05/shiny-sales-de-joc')
── Preparing for deployment
──────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────
✔ Re-deploying "shiny-sales-de-joc" using "server: shinyapps.io / username: onuka7-
xavier-de0pedro0puente"
 Looking up application with id "11822654"...
✔ Found application
<https://onuka7-xavier-de0pedro0puente.shinyapps.io/shiny-sales-de-joc/>
 Bundling 3 files: app.R, dades/muni_cat.csv, and
dades/salons_de_joc_en_la_web.xlsx
 Capturing R dependencies with renv
✔ Found 122 dependencies
✔ Created 98,363b bundle
 Uploading bundle...
✔ Uploaded bundle with id 8522558
── Deploying to server
──────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────
Waiting for task: 1408200276
  building: Processing bundle: 8522558
  building: Building image: 10366677
  building: Installing system dependencies
  building: Fetching packages
  building: Installing packages
  building: Installing files
  building: Pushing image: 10366677
  deploying: Starting instances
  terminating: Stopping old instances
── Deployment complete
──────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────
✔ Successfully deployed to
<https://onuka7-xavier-de0pedro0puente.shinyapps.io/shiny-sales-de-joc/>
```



URL a la app:
https://onuka7-xavier-de0pedro0puente.shinyapps.io/shiny-sales-de-joc/[9]



1.5. ShinyLive

ShinyLive (per exportar apps shiny per a que funcionin sense requerir servidor, sino tot en navegador):
https://cloud.r-project.org/web/packages/shinylive/index.html[10]

Explicació recent dels autors a Posit (2023-10):
R Shiny without a server: webR and Shinylive by George Stagg at the Shiny in Production 2023 conference.
https://www.youtube.com/watch?v=GlZKReTx8GA[11]

2. Altres: Dash per a R

Dash per a R: Create beautiful, analytic web applications in R.
https://github.com/plotly/dashR[12]

Dash Enterprise App Gallery
https://dash.gallery/Portal/[13]

3. Mostrar progrés projecte

Avançar la feina que hem estat fent del projecte per equips, i mostrar-ho als altres equips.

Noms alias d'aquest pàgina: CursRAvancatEquipsS5

[1] https://www.rstudio.com/products/shiny/

[2] https://dades.ajuntament.barcelona.cat/estadistiques-cens-comercial/

[3] https://dades.ajuntament.barcelona.cat/la-ciutat-al-dia/

[4] https://mastering-shiny.org/basic-app.html

[5] https://rstudio.github.io/shinyuieditor/

[6] https://www.shinyapps.io/

[7] https://statsandr.com/blog/how-to-publish-shiny-app-example-with-shinyapps-io/

[8] https://shiny.posit.co/r/articles/share/shinyapps/

[9] https://onuka7-xavier-de0pedro0puente.shinyapps.io/shiny-sales-de-joc/

[10] https://cloud.r-project.org/web/packages/shinylive/index.html

[11] https://www.youtube.com/watch?v=GlZKReTx8GA

[12] https://github.com/plotly/dashR

[13] https://dash.gallery/Portal/