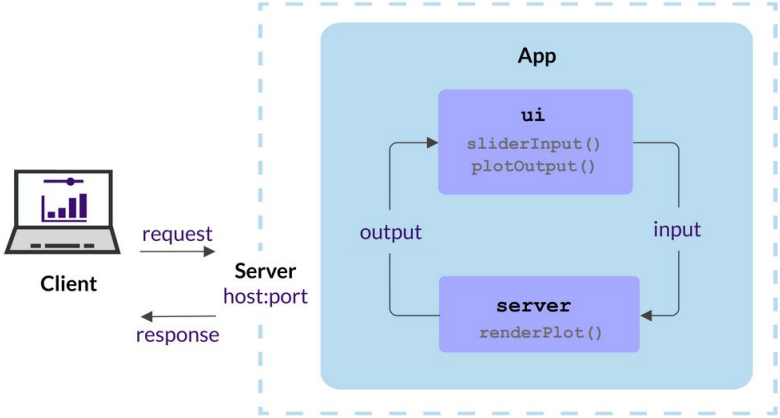

Rensselaer
why not change the world?®
Shiny
Ready for Prime Time: How to Deploy Shiny Apps!
Wednesday, 12 Mar 2025
RPIrates: The RPI R Users Group
The Rensselaer Institute for Data exploration and Applictions
Rensselaer Polytechnic Institute
IDEA
Rensselaer Institute for Data Exploration and Applications
RPIrates
The RPI R Users Group

Tall

What is a Shiny App?

- Web apps natively coded in R or (since 2022) Python
- An easy way to make your data analytics code interactive and accessible



https://shiny.posit.co/

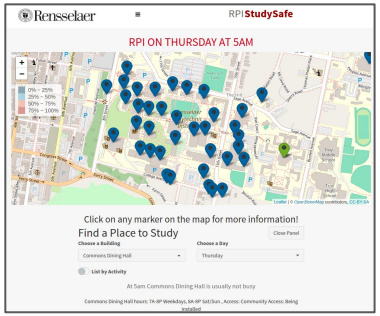
"Static" code:
Libraries…
Data initialization…
Static function definitions…

UI code:
Layout
Input widgets
Output placement
(text, plots, tables, maps)

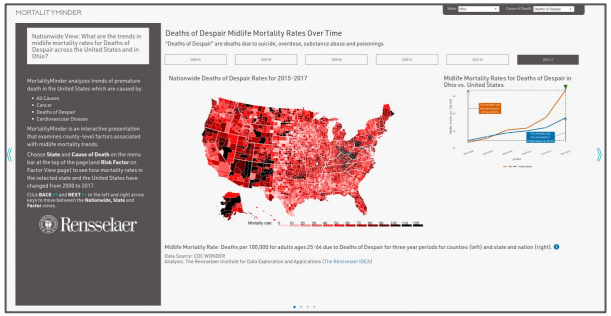
"Server" code:
Input processing
Reactive data processing
Reactive output generation

A Gallery of Shiny Apps

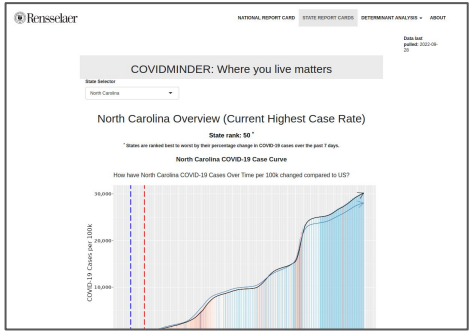
StudySafe



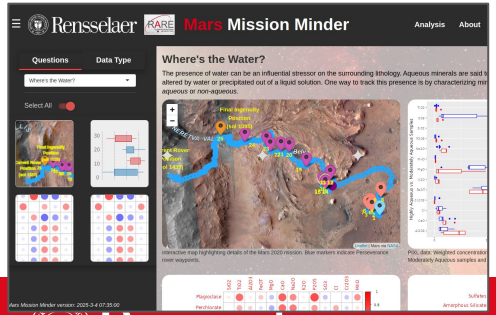
MortalityMinder



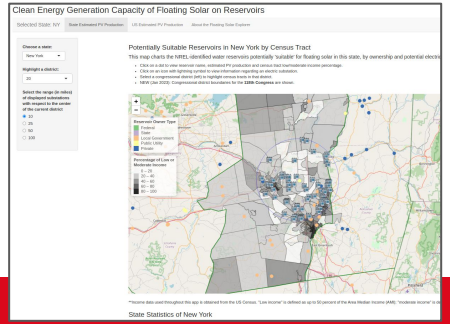
COVIDMINDER



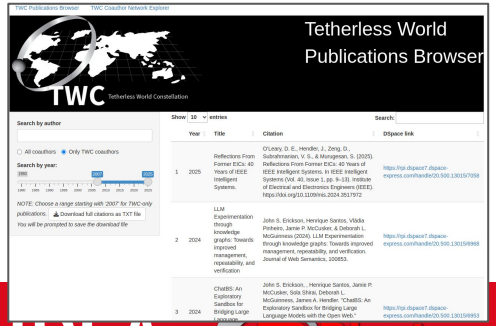
Mars Mission Minder



Floating Solar Explorer



TWC Pubs Browser



R-based Shiny

https://shiny.posit.co/

```
library(shiny)
library(bslib)
library(dplyr)
library(ggplot2)
library(ggExtra)

penguins_csv <-
"https://raw.githubusercontent.com/jcheng5/simplepenguins.R/main/penguins.csv"

df <- readr::read_csv(penguins_csv)
# Find subset of columns that are suitable for scatter plot
df_num <- df |> select(where(is.numeric), -Year)

ui <- page_sidebar(
  sidebar = sidebar(
    varSelectInput("xvar", "X variable", df_num, selected = "Bill Length (mm)"),
    varSelectInput("yvar", "Y variable", df_num, selected = "Bill Depth (mm)"),
    checkboxGroupInput(
      "species", "Filter by species",
      choices = unique(df$Species),
      selected = unique(df$Species)
    ),
    hr(), # Add a horizontal rule
    checkboxInput("by_species", "Show species", TRUE),
    checkboxInput("show_margins", "Show marginal plots", TRUE),
    checkboxInput("smooth", "Add smoother"),
  ),
  plotOutput("scatter")
)
```

```
server <- function(input, output, session) {
  subsetted <- reactive({
    req(input$species)
    df |> filter(Species %in% input$species)
  })

  output$scatter <- renderPlot({
    p <- ggplot(subsetted(), aes(!!input$xvar, !!input$yvar)) + list(
      theme(legend.position = "bottom"),
      if (input$by_species) aes(color = Species),
      geom_point(),
      if (input$smooth) geom_smooth()
    )

    if (input$show_margins) {
      margin_type <- if (input$by_species) "density" else "histogram"
      p <- ggExtra::ggMarginal(p, type = margin_type, margins = "both",
        size = 8, groupColour = input$by_species, groupFill = input$by_species)
    }

    p
  }, res = 100)
}

 shinyApp(ui, server)
```

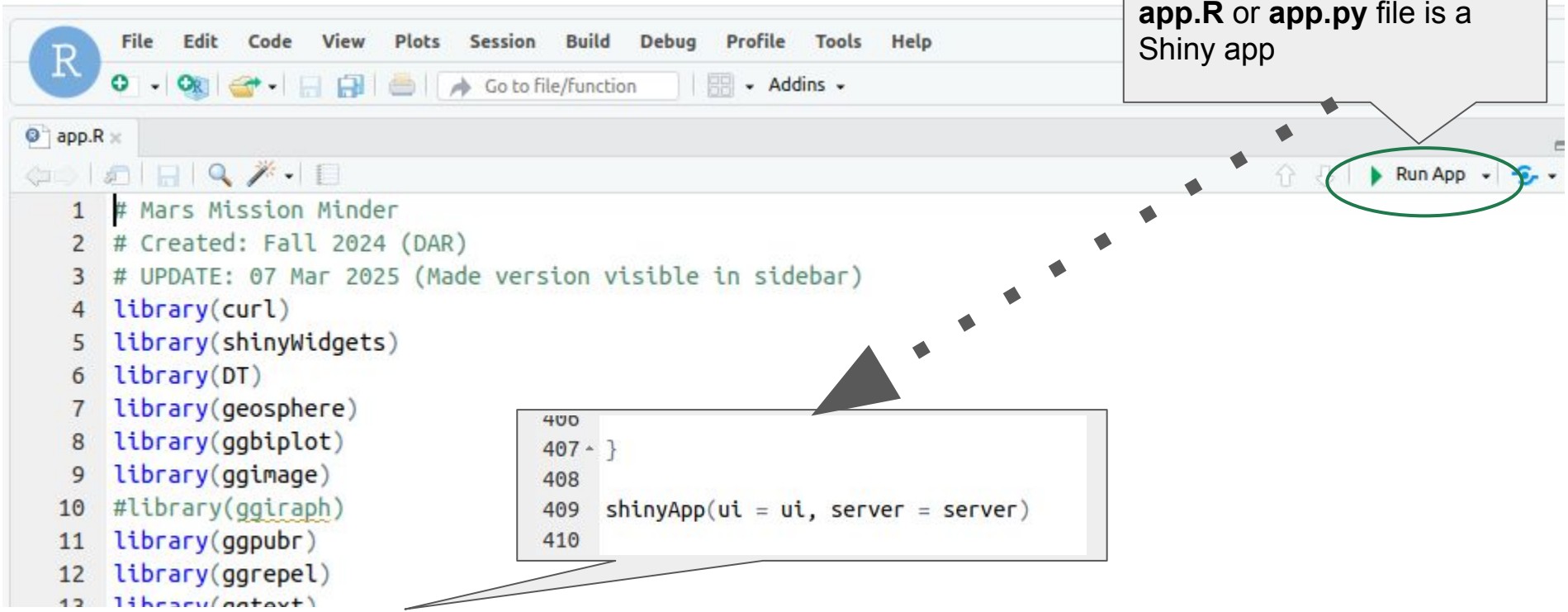
Python-based Shiny

https://shiny.posit.co/

```
from pathlib import Path

import pandas as pd
import seaborn as sns

from shiny import App, Inputs, Outputs, Session, reactive, render, req, ui

sns.set_theme()

# https://raw.githubusercontent.com/jcheng5/simplepenguins.R/main/penguins.csv

df = pd.read_csv(Path(__file__).parent / "penguins.csv", na_values="NA")
numeric_cols = df.select_dtypes(include=["float64"]).columns.tolist()
species = df["Species"].unique().tolist()
species.sort()

app_ui = ui.page_sidebar(
    ui.sidebar(
        ui.input_selectize(
            "xvar", "X variable", numeric_cols, selected="Bill Length (mm)"
        ),
        ui.input_selectize(
            "yvar", "Y variable", numeric_cols, selected="Bill Depth (mm)"
        ),
        ui.input_checkbox_group(
            "species", "Filter by species", species, selected=species
        ),
        ui.hr(),
        ui.input_switch("by_species", "Show species", value=True),
        ui.input_switch("show_margins", "Show marginal plots", value=True),
    ),
    ui.card(
        ui.output_plot("scatter"),
    ),
)
```

```
def server(input: Inputs, output: Outputs, session: Session):
    @reactive.Calc
    def filtered_df() -> pd.DataFrame:
        """Returns a Pandas data frame that includes only the desired rows"""

        # This calculation "req"uires that at least one species is selected
        req(len(input.species()) > 0)

        # Filter the rows so we only include the desired species
        return df[df["Species"].isin(input.species())]

    @output
    @render.plot
    def scatter():
        """Generates a plot for Shiny to display to the user"""

        # The plotting function to use depends on whether margins are desired
        plotfunc = sns.jointplot if input.show_margins() else sns.scatterplot

        plotfunc(
            data=filtered_df(),
            x=input.xvar(),
            y=input.yvar(),
            hue="Species" if input.by_species() else None,
            hue_order=species,
            legend=False,
        )


app = App(app_ui, server)
```

Running from RStudio (built-in)



"Deploying" on the FOCI Cluster: Shiny Server

- A simple way to test apps and share within the RPI network
- Create individual app directories in your **ShinyApps** sub-directory
  - Your "app" must be named **app.R** (but can be a symbolic link)
  - App directory can be a symbolic link to another directory
- "Danger Will Robinson!" Not multi-user; shared process for all clients!
- URLs are a variant of your RCS ID and app directory name:

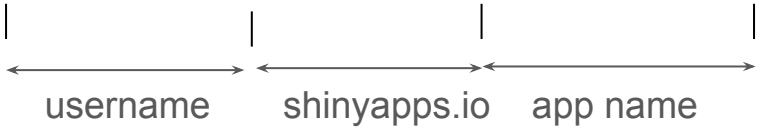
https://lp01.idea.rpi.edu/shiny/**erickj4**/**MarsMissionMinder-S25**/

Deploying via Shinyapps.io

- Utilizes **rsconnect** package:
  - deployApp() <- app name defaults to current working directoy
  - deployApp(appName = "myAppName")
- Account required; free for individual use (limited apps)
- App size limit
- URLs are a variant of your username and app name:

https://**olyerickson**.shinyapps.io/**wordler**/

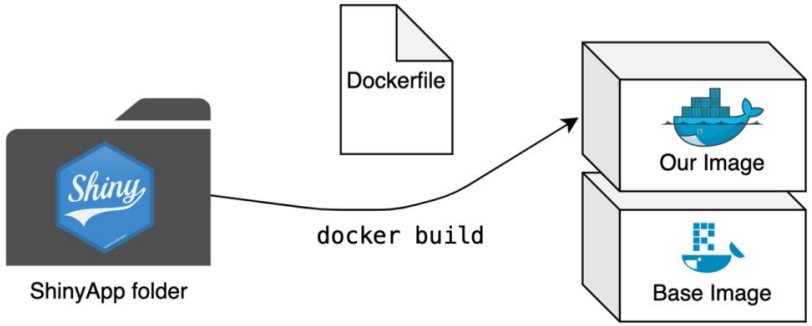
https://**olyerickson**.shinyapps.io/**cohoespvapp**/



Dockerizing

https://colinfay.me/docker-r-reproducibility/

- A robust way to create a stand-alone, shareable image of your app
- Publicly shareable via DockerHub
- Anyone with Docker installed can download and run your app locally
- **rocker** repository contains selection of pre-built "base" images



https://hub.docker.com/u/rocker

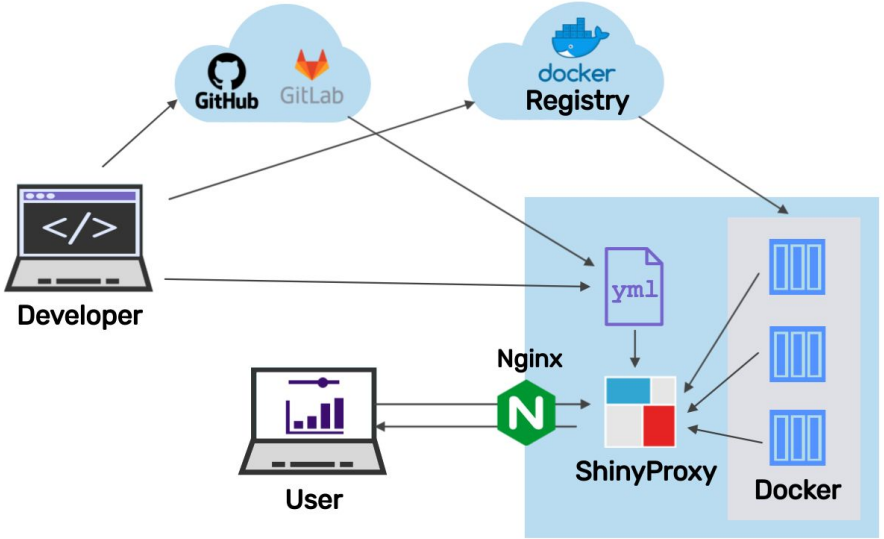
A Dockerfile Example [R]

```
FROM rstudio/r-base:4.4.1-noble

LABEL maintainer="John Erickson <erickj4@rpi.edu>"

# system libraries of general use
RUN apt-get update && apt-get install -y --no-install-recommends \
    sudo \
    cmake \
    libharfbuzz-dev \
    libfreetype6-dev \
    libcurl4-gnutls-dev \
    libcairo2-dev \
    libxt-dev \
    libssl-dev \
    libssh2-1-dev \
    libxml2-dev \
    libproj-dev \
    libgdal-dev \
    libudunits2-dev \
    librdf0 \
    librdf0-dev \
    && rm -rf /var/lib/apt/lists/*
```

```
## Regular R packages
RUN R -e 'install.packages("dplyr",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("DT",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("shiny",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("remotes",repos="https://cloud.r-project.org/")'
RUN R -e 'remotes::install_github("irudnyts/openai", ref = "r6")'
RUN R -e 'install.packages("RCurl",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("rlist",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("jsonlite",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("tidyr",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("stringr",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("purrr",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("bslib",repos="https://cloud.r-project.org/")'
RUN R -e 'install.packages("shinyjs",repos="https://cloud.r-project.org/")'

# copy the app to the image
RUN mkdir /root/ctsuggest
COPY . /root/ctsuggest

COPY Rprofile.site /usr/lib/R/etc/

EXPOSE 1824

CMD ["R", "-e", "shiny::runApp('/root/ctsuggest')"]
```

Deploying via ShinyProxy (general)

- Open-source framework for hosting publicly hosting many apps for many users
- Based on Docker: one container instance per user
- One or many apps per host, depending upon load and other requirements



© Analythium

Deploying via ShinyProxy (at RPI)

- VM exposes app to RPI internal network
- "inciteprojects" proxy (safely) publishes to the World
- Until recently, one app per VM

Internal:
http://**chatbs**.idea.rpi.edu:8080/**chatbs**/app/**chatbs**/

External (through proxy):
https://inciteprojects.idea.rpi.edu/**chatbs**/app/**chatbs**/