Titanic Survival - ShinyApp

Daria Bartkowiak

February 2026

1 Description of the Shiny Application

1.1 Dataset Used

The application uses the built-in `Titanic` dataset available in the standard R library. The dataset contains aggregated information about passengers divided into the following variables:

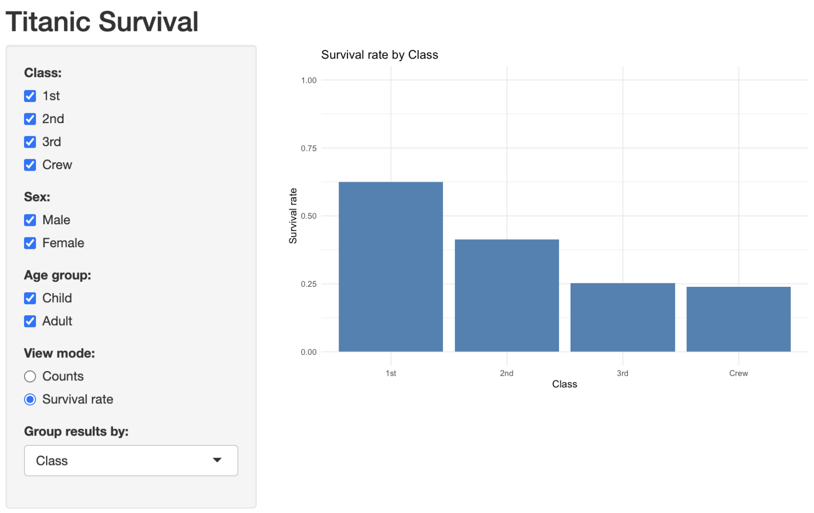
- **Class** – passenger class (First, Second, Third, Crew)
- **Sex** – passenger sex (Male, Female)
- **Age** – age group (Child, Adult)
- **Survived** – survival status (Yes, No)
- **Freq** – number of passengers in a given category

1.2 Application Architecture

The application was built using the client–server architecture characteristic of the Shiny framework, which consists of two main components:

- **UI (User Interface)** – defines the structure and elements of the user interface
- **Server** – contains the reactive logic and generates the outputs

The user interface uses the `sidebarLayout`, which divides the screen into a sidebar panel with controls and a main panel with visualization.
Figure 1 shows the overall layout of the application, including the sidebar and the plot area.



Titanic Survival

Class:

- ☑ 1st
- ☑ 2nd
- ☑ 3rd
- ☑ Crew

Sex:

- ☑ Male
- ☑ Female

Age group:

- ☑ Child
- ☑ Adult

View mode:

- ○ Counts
- ● Survival rate

Group results by:

Class ▾

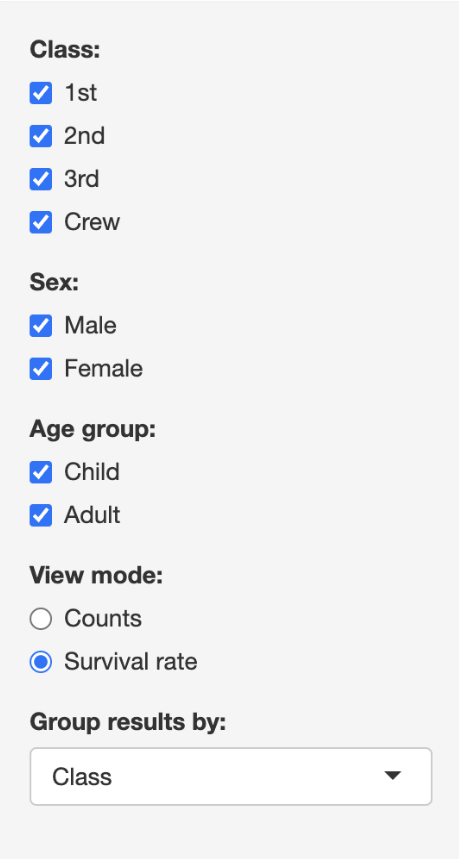
Rysunek 1: Overview of the Shiny application interface

1.3 User Interface Components

The sidebar panel contains the following reactive input elements:

- passenger class selection (checkboxGroupInput)
- passenger sex selection (checkboxGroupInput)
- age group selection (checkboxGroupInput)
- visualization mode selection (counts or survival rate)
- selection of grouping variable for the X axis (Class, Sex, Age)

The main panel contains a dynamically generated bar chart.
Figure 2 presents the sidebar panel with available filtering and visualization options.



Class:

☑ 1st
☑ 2nd
☑ 3rd
☑ Crew

Sex:

☑ Male
☑ Female

Age group:

☑ Child
☑ Adult

View mode:

○ Counts
● Survival rate

Group results by:

Class ▾

Rysunek 2: Sidebar with user input controls

1.4 Reactive Mechanism

The central component of the application is the reactive function `filtered_data()`, which filters the dataset according to user selections. Any change in the input parameters automatically triggers recalculation of the data and updates the plot.

Additionally, the function `plot_data()` is used to compute aggregated values and survival rates according to the selected grouping variable.

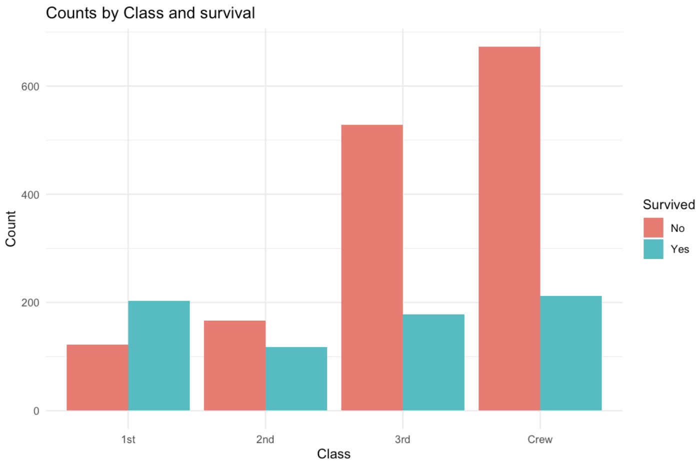
This mechanism implements a reactive dependency graph model, where outputs are automatically updated in response to changes in inputs.

1.5 Data Visualization

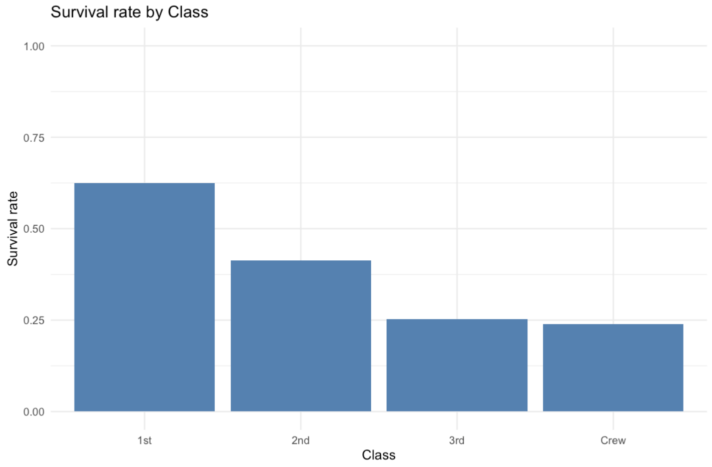
The application generates a bar chart using the `ggplot2` library. Depending on the selected mode, the chart displays either:

- the number of passengers in each group, or
- the survival rate calculated as the ratio of survivors to the total number of passengers in each group

The plot is updated in real time in response to user interaction.
Example visualizations generated by the application are shown below.



Rysunek 3: Example bar chart showing passenger counts



Rysunek 4: Example bar chart showing survival rates

2 AI usage

I used AI (ChatGPT 5.2) for translating the report from Polish to English and for generating LateX document.

3 Repository

Link to the GitHub repository: https://github.com/bartkowiakdaria/DSInMedicine