**Fertility Statistics Auto-visualization Tool**

**Maintenance Guide (version 1.0)**

**13th September 2024**

Yuting Li; [yutili@utu.fi](mailto:yutili@utu.fi)

# **1. Project Overview**

This Fertility Statistics Auto-visualization Tool application is a visualization tool for fertility-related data across countries over long-term period, with data sourced from the **Human Fertility Database (HFD)** and built by Rstudio (R-4.4.1; RStudio 2024.04.2) and Shiny 1.9.1. Users can interact with the app to select variables, adjust country filters, set X and Y axis ranges, and modify the graphical style. Additional features include an interactive plotting, downloading the generated plots (in formats JPG, PNG, EPS, and PDF), switching between English and Finnish language modes, and toggling a night mode (dark mode) for the UI.

The app provides two methods to update the HFD data:

1. **Automated API request** to fetch the latest data from HFD (packaged in an Rscript: *“Update data from HFD web.R”*).
   * **Note!!** The API request process requires a **username and password for HFD**. Currently, these credentials are coded into the script. This practice poses potential risk as the username and password are exposed in the code.
   * **Action Required:** ensure that only authorized personnel have access to the script.
2. **Manual update** by downloading the compressed HFD dataset locally (packaged in an R script: *“Update data from local.R”*).

# **2. Project Structure**

Understanding the directory and file structure is critical for maintaining and updating the app.

1. **01\_data:** A folder containing the “HFD/” folder that manually downloaded and web-scraped data from HFD that the app uses.
2. **Rsconnect/:** A folder related to RStudio Connect, where Shiny applications are deployed.
3. **HFD\_visualization.Rproj**: This is the RStudio project file, which helps organize your workspace and files for this Shiny app project.
4. **HFD\_auto\_v3.R**: An R script that contains the UI and Sever of the application.
5. **MaintenanceGuide\_v1.docx**: A file contains instructions for maintaining the Shiny app.
6. **Update data from HFD web.R**: An R script that handles the process of updating data from the HFD directly via the web through an API developed by [@Timothy L. M. Riffe](https://github.com/timriffe/TR1).
7. **Update data from local.R**: An R script that manages updating data from locally downloaded files (e.g., HFD data packages).
8. **update\_log.txt**: A text file containing logs of updates made to the data, tracking changes over time.

# **3. Data Update Process**

There are two ways to update the app’s dataset. **Web request is the recommended method**, but the app also supports manual updates as a fallback.

**Method 1: Web requests (recommend)**

The script *Update data from HFD web*.R fetches the latest data from the Human Fertility Database.

* **Step-by-step instructions**:
  1. Run the script: “Update data from HFD web.R”.
  2. The script will automatically scrape, process, and update the dataset.
  3. The API request may take between 3 to 7 minutes, depending on network speed and server load at HFD. **Note:** If you encounter delays or errors due to server congestion, please try the request again after a short wait.
  4. Verify the new dataset has been correctly processed by checking the *update\_log*.txt file and restarting the app and checking the plots.

**Method 2: Manual Local Update**

The script *Update data from local*.R is used if web scraping is not possible.

* **Step-by-step instructions**:
  1. Download the latest zipped data file (click the button “All types of HFD data”) from [Human Fertility Database](https://www.humanfertility.org/Data/ZippedDataFiles).
  2. Place the decompressed file in the /01\_data folder.
  3. Run the script: “Update data from local.R”.
  4. This script will process the local file and update the app’s dataset.

# **4. Key Functionalities in UI**

The app provides several user interactions and customizations:

* **Variable Selection**: Allows the user to select different fertility-related variables for plotting.
* **Country Selection**: Users can filter the dataset by country.
* **X and Y Axis Adjustments**: Slider inputs allow users to adjust the ranges of both axes.
* **Plot Style**: Users can select from different ggplot2 themes, which are dynamically applied to the generated plot.
* **Interactive plotting feature:** When the user hovers the mouse over a data point in the graph, additional information about the specific data point is displayed in a tooltip. This includes the x and y-axis values, along with other relevant information like country and variable type, depending on the graph.
* **Zoom and Pan:** Users can zoom into particular areas of the plot by selecting regions, as well as pan across the plot by clicking and dragging.
* **Download Options**: Users can download plots in multiple formats: JPG, PNG, EPS, and PDF.
* **Language Selection**: The app supports a bilingual UI, switching between English and Finnish.
* **Dark Mode**: Users can toggle a dark mode for the app interface.

# **5. Plot Customization**

The plots in the app are created using **ggplot2** and **plotly**, and there are several customization options emphasized here, which can be applied to improve the visualizations:

* **Themes**: The plot themes come from ggplot2 themes (Classic, Gray, Light, etc.), but developer can add more (or remove) from [ggthemes](https://yutannihilation.github.io/allYourFigureAreBelongToUs/ggthemes/) package.
* **Axis Range and Labels**: The axis ranges are dynamically updated based on the minimum and maximum values of specific variables. The axis labels are dynamically updated according to language selection (English or Finnish).
* **Country-fixed Colors for Line Plot:**
  + In line plots, each country is assigned a fixed color to maintain consistency in visual comparisons. These are **colorblind-friendly colors** generated by <https://medialab.github.io/iwanthue/>
  + **Nordic countries** (e.g., Finland, Sweden, Norway, Denmark, Iceland) have been assigned the most distinctive colors—red, blue, green, and purple—to enhance the clarity and separation of their data lines in multi-country plots.
  + The color scheme has been predefined and **should be modified** in the backend code **if new countries are added** or existing colors need to be adjusted.
* **Dark Mode:** If dark mode is enabled through input\_dark\_mode, the app will automatically switch to a dark-themed plot style.

**Note:** To modify the appearance or behavior of the plots, update the ggplot code in *HFD\_auto\_v3*.R

# **6. Adding New Features**

If additional functionality needs to be added, such as new plot types or data sources, follow these guidelines:

* **UI Modifications**: Update the UI part of *HFD\_auto\_v3*.R file, to include new input controls or output elements.
* **Data Processing**: Modify the*Update data from HFD web*.R and *Update data from local*.R to handle new data transformations or filters.
* **Plotting**: Adjust the ggplot sections in Server part of *HFD\_auto\_v3*.R to incorporate new visual elements or modify existing ones.
* **Language Support**: Ensure that any new features support both English and Finnish by updating the lang\_dict object in the *HFD\_auto\_v3*.R file, which handles text translations.

# **8. Troubleshooting**

* **Web request error or delay**: The API request may take between 3 to 7 minutes, depending on network speed and server load at HFD. If you encounter delays or errors due to server congestion, please try the request again after a short wait.
* **Data issues**: If the plots fail to update after running the data update scripts, check the update log or path to dataset is in place. Ensure that no errors occurred during the script execution.
* **UI rendering issues**: If the UI doesn’t render correctly after a new feature is added, ensure that the corresponding inputs/outputs have been correctly linked in both UI and Serve parts.
* **Language switch**: If language switching is not functioning as expected, verify that the lang\_dict object contains the correct translations for any new text elements added.