HMS 520 Final Project Initial Documentation

Waiting Time Distribution Package

6th December 2024

Waiting Time Distribution (WTD) Function

Overview

The **Waiting Time Distribution (WTD)** project is designed to calculate the 80th percentile of the non-parametric waiting time distribution for medication use. This serves as a measure of the duration of exposure when the exact number of days prescribed is unavailable. The package includes two key functions:

1. **wtd()**: For calculating the weighted time distribution and summarizing results.
2. **wtdplot():** A plotting function for generating visual representations of the WTD (e.g., eCDF or histogram).

Function 1: wtd()

This main function computes the WTD using three key parameters and provides flexibility through additional arguments for customization.

Arguments

- **id**:
  - **Description**: The patient identifier, provided as an atomic vector, to distinguish individual users.
- **Date**:
  - **Description**: A vector of dates or date-like characters used to calculate the WTD.
- **Percent**:
  - **Description**: The empirical cumulative distribution percentile used as the duration of exposure (default = 80th percentile).
- **Strata**:
  - **Description**: A vector of patient characteristics for estimating substrata-specific WTDs (e.g., age groups or comorbidities). Default is FALSE (no stratification).
- **Random**:
  - **Description**: Allows the use of random dates for the start of follow-up at the individual level instead of a fixed date (e.g., January 1st). Default is FALSE.
- **Prevalence**:
  - **Description**: Filters patients to calculate the WTD only for prevalent users. By default, 0 includes all patients. This can be increased (e.g., 1, 2, 3, etc.) to include only patients with a specific number of prescriptions in the year prior to the estimation.
- **Start**:
  - **Description**: Specifies the start date for follow-up (default = January 1st).

Output

(i) **Atomic Vector**: Returns the WTD as a numeric double vector.
(ii) **Summary Table**: Outputs a table summarizing the WTD parameters, including:

- Drug name
- Percentile (e.g., 80th)
- Stratified summaries (if Strata is used).

Additional Features

- Handles NA values seamlessly.
- Ensures compatibility and proper handling of mismatched lengths among arguments (e.g., id, Date, and medications).

Function 2: wtdplot()

The second function generates visualizations of the WTD results.

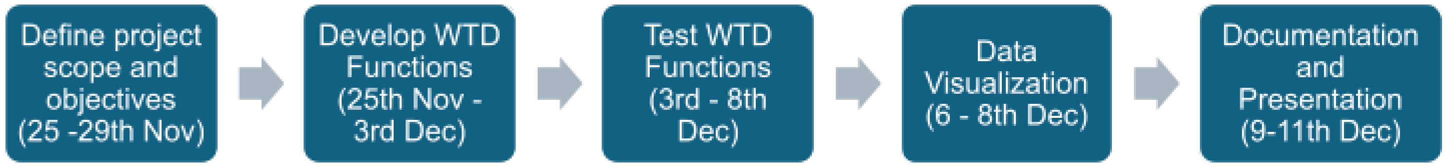
Features

- **Plot Types**:
  - Empirical Cumulative Distribution Function (eCDF).
  - Histogram of WTD days.
- **Input**: Uses the object created by the wtd() function.
- **Customization**: Leverages ggplot2 for visually appealing and customizable plots.

How WTD is Calculated Using Non-Parametric Methods

1. **Arbitrary Start Date**: A start date for follow-up is chosen (typically, January 1st of a given year).
2. **First Prescription Date**: For each patient, the first date of a filled prescription after the start date is identified.
3. **Waiting Time Calculation**: The time in days between the start date and the first prescription date is computed for each patient.
4. **eCDF Estimation**: Using an empirical cumulative distribution function (eCDF), the desired percentile (default = 80th) is calculated as the WTD.

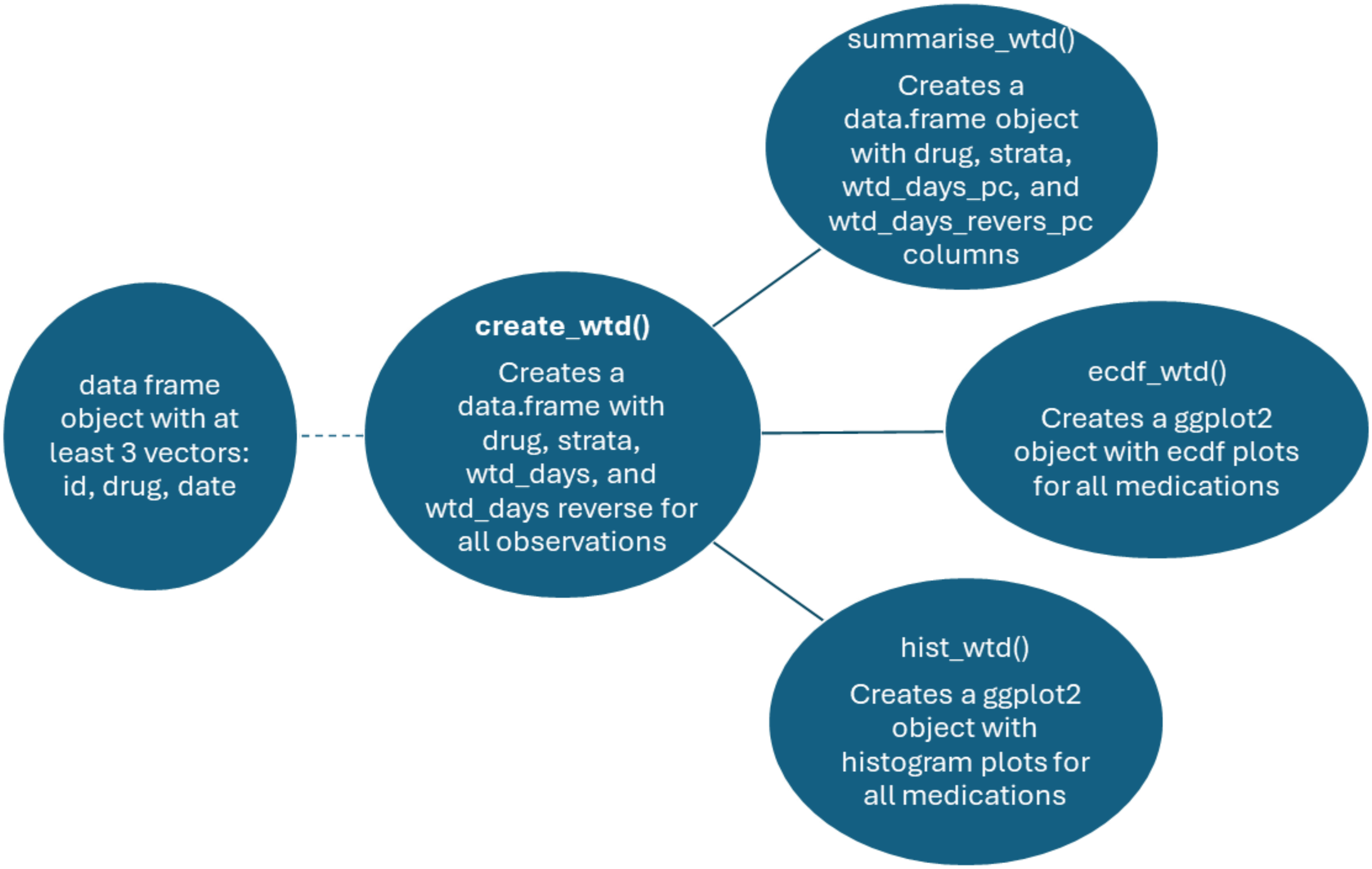
Timeline


Define project scope and objectives (25 -29th Nov)
Develop WTD Functions (25th Nov - 3rd Dec)
Test WTD Functions (3rd - 8th Dec)
Data Visualization (6 - 8th Dec)
Documentation and Presentation (9-11th Dec)

Landmark WTD Studies for Reference

1. Hallas J, Gaist D, Bjerrum L. The Waiting Time Distribution as a Graphical Approach to Epidemiologic Measures of Drug Utilization. Epidemiology. 1997;8(6).
2. Pottegård A, Hallas J. Assigning exposure duration to single prescriptions by use of the waiting time distribution. Pharmacoepidemiol Drug Saf. 2013 Aug;22(8):803–9.
3. Støvring H, Pottegård A, Hallas J. Determining prescription durations based on the parametric waiting time distribution: Determining Prescription Durations. Pharmacoepidemiol Drug Saf. 2016 Dec;25(12):1451–9.
4. Bødkergaard K, Selmer RM, Hallas J, Kjerpeseth LJ, Pottegård A, Skovlund E, et al. Using the waiting time distribution with random index dates to estimate prescription durations in the presence of seasonal stockpiling. Pharmacoepidemiol Drug Saf. 2020 Sep;29(9):1072–8.

How does this package work



data frame object with at least 3 vectors: id, drug, date

create_wtd()

Creates a data.frame with drug, strata, wtd_days, and wtd_days reverse for all observations

summarise_wtd()

Creates a data.frame object with drug, strata, wtd_days_pc, and wtd_days_revers_pc columns

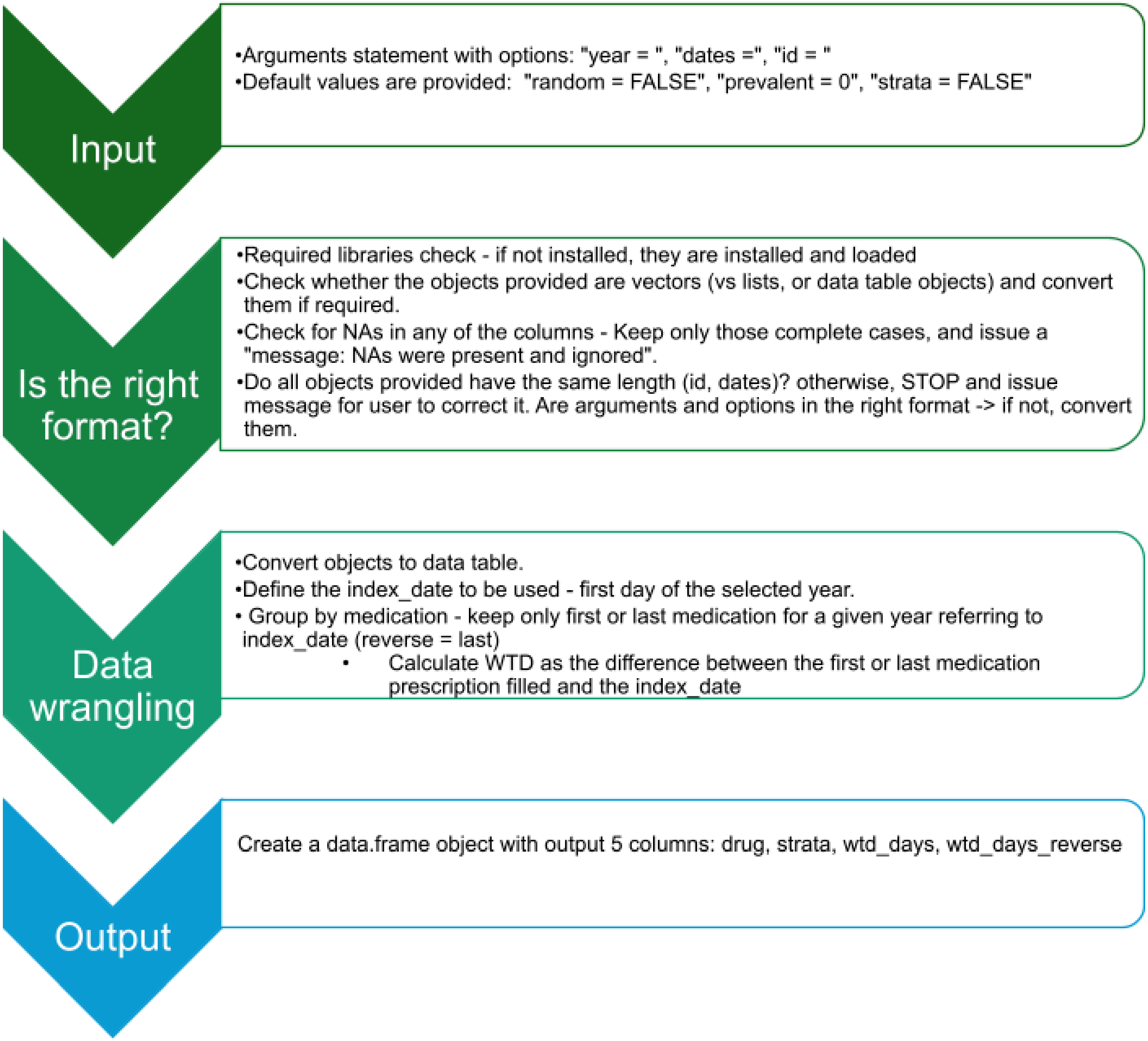
ecdf_wtd()

Creates a ggplot2 object with ecdf plots for all medications

hist_wtd()

Creates a ggplot2 object with histogram plots for all medications

Data Coding Process: create_wtd()



Input

- Arguments statement with options: "year = ", "dates =", "id = "
- Default values are provided: "random = FALSE", "prevalent = 0", "strata = FALSE"

Is the right format?

- Required libraries check - if not installed, they are installed and loaded
- Check whether the objects provided are vectors (vs lists, or data table objects) and convert them if required.
- Check for NAs in any of the columns - Keep only those complete cases, and issue a "message: NAs were present and ignored".
- Do all objects provided have the same length (id, dates)? otherwise, STOP and issue message for user to correct it. Are arguments and options in the right format -> if not, convert them.

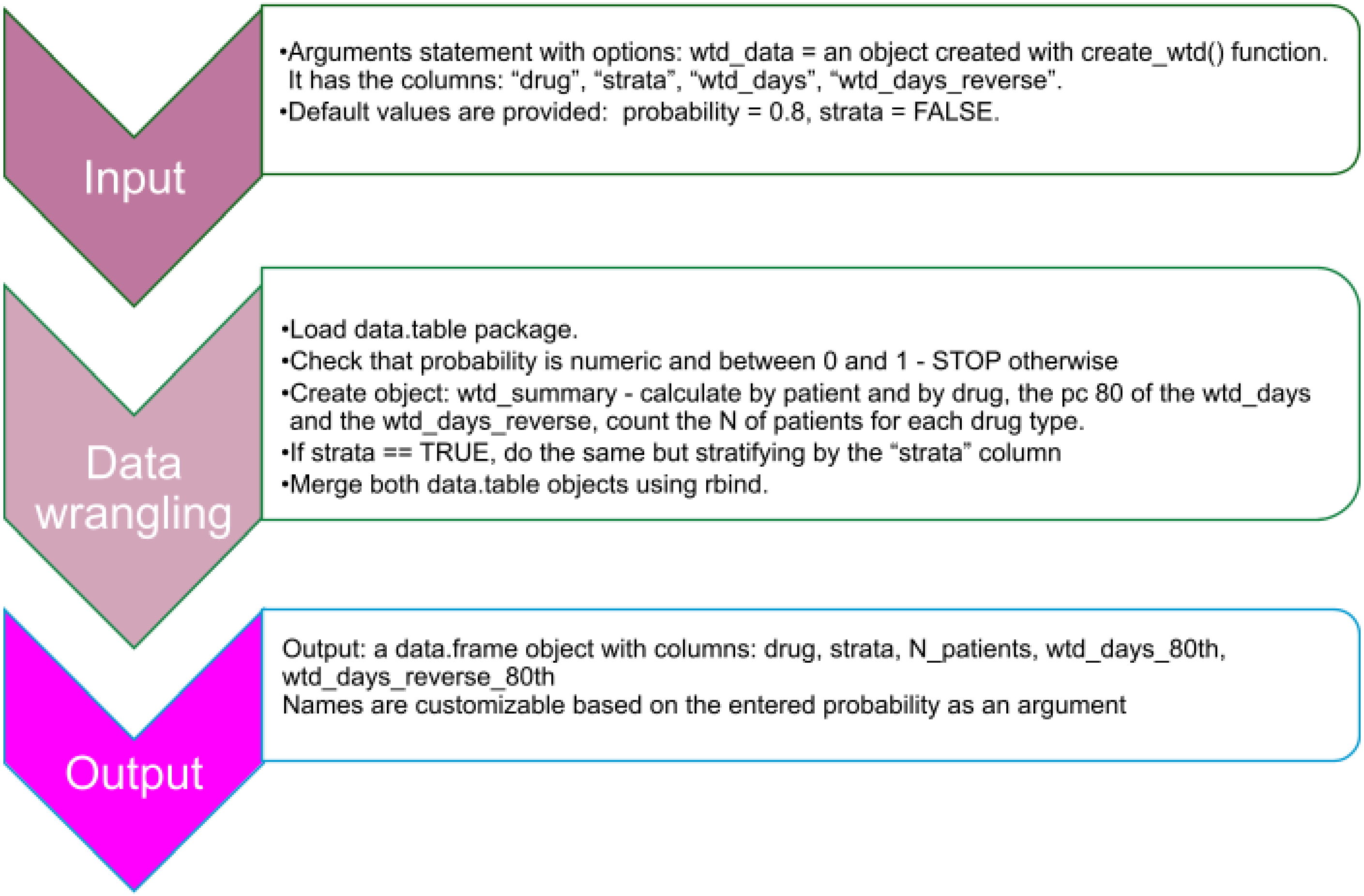
Data wrangling

- Convert objects to data table.
- Define the index_date to be used - first day of the selected year.
- Group by medication - keep only first or last medication for a given year referring to index_date (reverse = last)
  - Calculate WTD as the difference between the first or last medication prescription filled and the index_date

Output

Create a data.frame object with output 5 columns: drug, strata, wtd_days, wtd_days_reverse

Data Coding Process: summarise_wtd()



Input

- Arguments statement with options: wtd_data = an object created with create_wtd() function. It has the columns: “drug”, “strata”, “wtd_days”, “wtd_days_reverse”.
- Default values are provided: probability = 0.8, strata = FALSE.

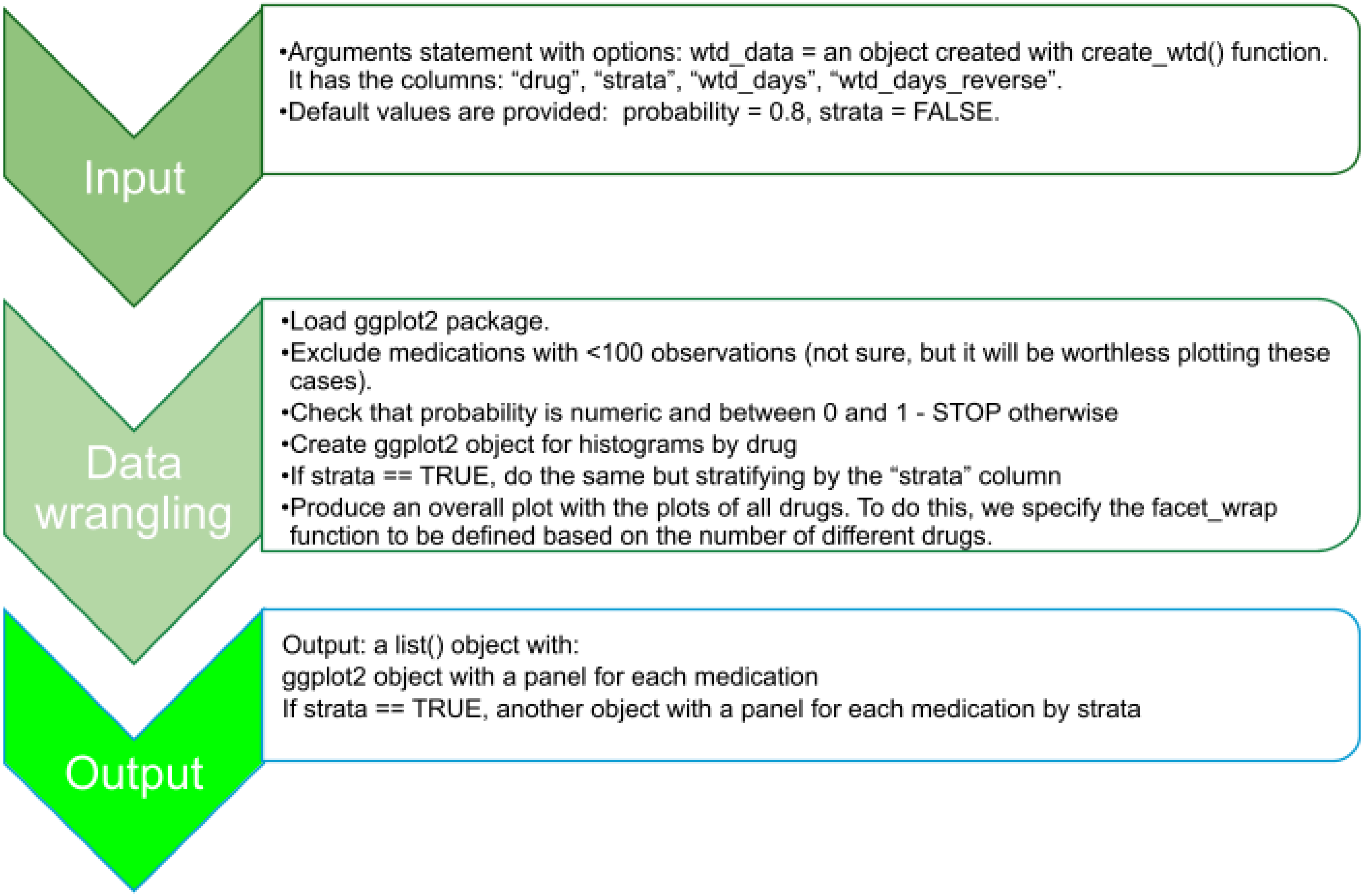
Data wrangling

- Load data.table package.
- Check that probability is numeric and between 0 and 1 - STOP otherwise
- Create object: wtd_summary - calculate by patient and by drug, the pc 80 of the wtd_days and the wtd_days_reverse, count the N of patients for each drug type.
- If strata == TRUE, do the same but stratifying by the “strata” column
- Merge both data.table objects using rbind.

Output

Output: a data.frame object with columns: drug, strata, N_patients, wtd_days_80th, wtd_days_reverse_80th
Names are customizable based on the entered probability as an argument

Data Coding Process: hist_wtd()



Input

- Arguments statement with options: wtd_data = an object created with create_wtd() function. It has the columns: “drug”, “strata”, “wtd_days”, “wtd_days_reverse”.
- Default values are provided: probability = 0.8, strata = FALSE.

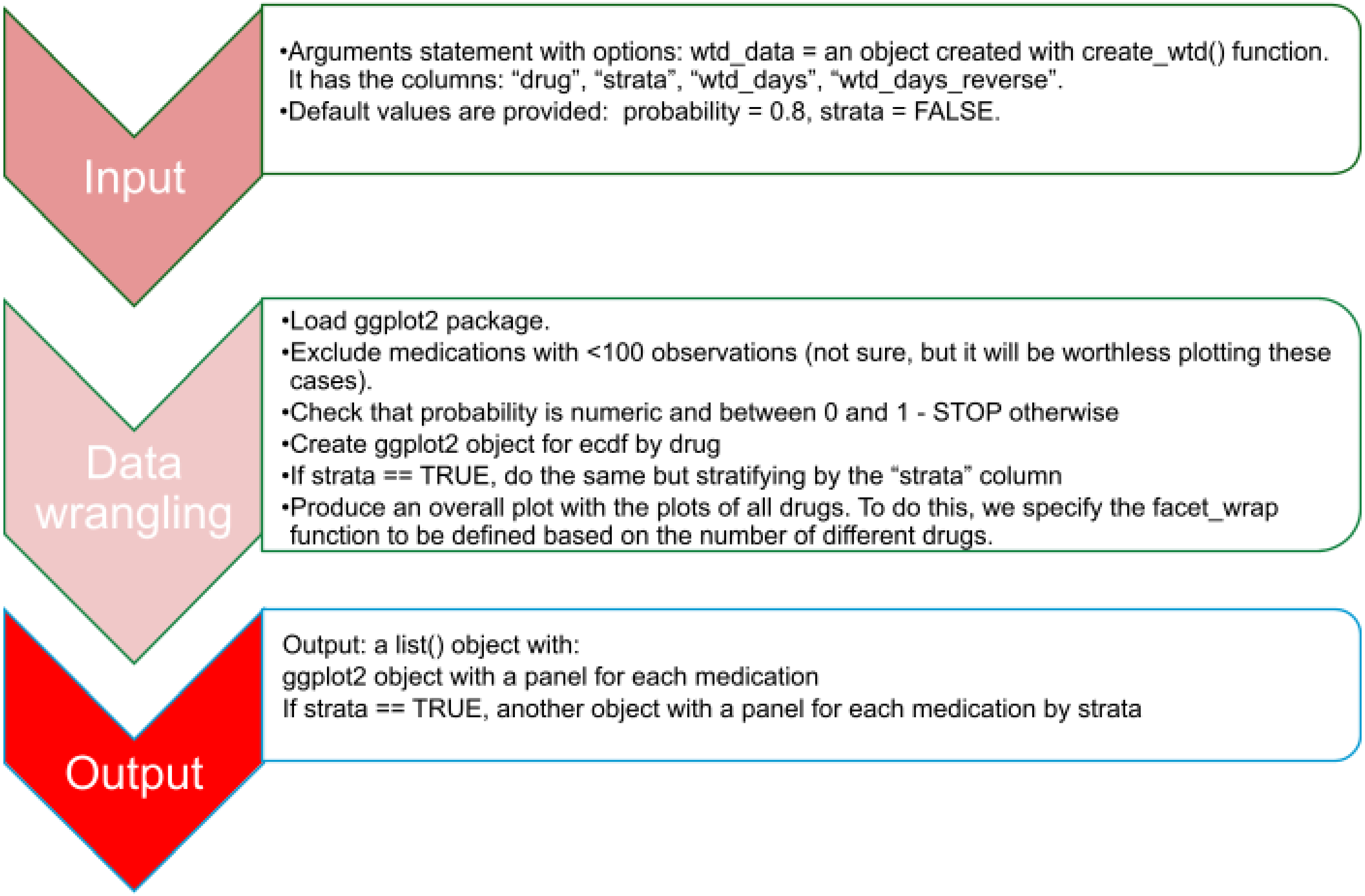
Data wrangling

- Load ggplot2 package.
- Exclude medications with <100 observations (not sure, but it will be worthless plotting these cases).
- Check that probability is numeric and between 0 and 1 - STOP otherwise
- Create ggplot2 object for histograms by drug
- If strata == TRUE, do the same but stratifying by the “strata” column
- Produce an overall plot with the plots of all drugs. To do this, we specify the facet_wrap function to be defined based on the number of different drugs.

Output

Output: a list() object with:
ggplot2 object with a panel for each medication
If strata == TRUE, another object with a panel for each medication by strata

Data Coding Process: ecdf_wtd()



Input

- Arguments statement with options: wtd_data = an object created with create_wtd() function. It has the columns: “drug”, “strata”, “wtd_days”, “wtd_days_reverse”.
- Default values are provided: probability = 0.8, strata = FALSE.

Data wrangling

- Load ggplot2 package.
- Exclude medications with <100 observations (not sure, but it will be worthless plotting these cases).
- Check that probability is numeric and between 0 and 1 - STOP otherwise
- Create ggplot2 object for ecdf by drug
- If strata == TRUE, do the same but stratifying by the “strata” column
- Produce an overall plot with the plots of all drugs. To do this, we specify the facet_wrap function to be defined based on the number of different drugs.

Output

Output: a list() object with:
ggplot2 object with a panel for each medication
If strata == TRUE, another object with a panel for each medication by strata