

# Package ‘daySupply’

October 13, 2022

**Type** Package

**Title** Calculating Days' Supply and Daily Dose of Prescriptions

**Version** 0.1.0

**Maintainer** Abdollah Safari <a.saffari72@gmail.com>

**Description** Allows clinicians and researchers to compute daily dose (and subsequently days' supply) for prescription refills using the following methods: Fixed window, fixed tablet, defined daily dose (DDD), and Random Effects Warfarin Days' Supply (RE-WarDS). Daily dose is the computed dose that the patient takes every day. For medications with fixed dosing (e.g. direct oral anticoagulants) this is known and does not need to be estimated. For medications with varying dose such as warfarin, however, the daily dose should be assumed or estimated to allow measurement of drug exposure. Days' supply is the number of days that patients' supply of medication will last after each prescription fill. Estimating days' supply is necessary to calculate drug exposure. The package computes days' supply and daily dose at both the prescription and patient levels. Results at the prescription level are denoted with “-Rx-” and those at patient level are denoted with “-Pt-”.

**Depends** R (>= 3.5.0)

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Imports** dplyr, lme4, rlang, magrittr

**NeedsCompilation** no

**Author** Abdollah Safari [aut, cre],  
Shahzad Salmasi [aut],  
Peter Loewen [aut]

**Repository** CRAN

**Date/Publication** 2021-04-29 07:50:02 UTC

## R topics documented:

definedDailyDose . . . . .	2
fixedTablet . . . . .	4
fixedWindow . . . . .	5
REWarDS . . . . .	7

---

definedDailyDose	<i>Defined daily dose</i>
------------------	---------------------------

---

### Description

Uses the World Health Organization's (WHO) defined daily dose method to compute the daily dose and days' supply for prescriptions. This method assumes an average daily consumption of a fixed dose, the defined daily dose (DDD), specified by WHO on their website at [https://www.whooc.no/atc\\_ddd\\_index/?code=B01A](https://www.whooc.no/atc_ddd_index/?code=B01A)

### Usage

```
definedDailyDose(
  data,
  WHO_ddd,
  dspd_qty,
  strength,
  id,
  tot_dose_disp = NULL,
  Pt_level = FALSE
)
```

### Arguments

data	Sample simulated data. Data may have multiple rows per person (one row per prescription fill). Required columns include: 1. ID: Patient's unique identification number 2. ServDate: Date on which each prescription was filled. 3. DSPD_QTY: Dispensed quantity: Number of tablets dispensed to patient at each prescription fill. 4. strength: Strength of the tablets dispensed.
WHO_ddd	The defined daily dose in mg as specified by the WHO.
dspd_qty	Dispensed quantity: Number of the dispensed tablets to the patient at each prescription fill.
strength	Strength of the tablet dispensed in milligrams.
id	Unique patient identification number.
tot_dose_disp	Total dose dispensed: dispensed quantity x strength of the tablets dispensed for each prescription fill.
Pt_level	When TRUE, the estimated daily dose and days' supply are averaged for the patient.

### Details

The DDD method can be used for any medication. However, its accuracy has been shown to differ between drug classes.

**Value**

definedDailyDose returns a dataset called "DDD\_result". This data set includes all the variables originally in the data, plus the following:

DDD\_Rx\_dose: Daily dose for prescription.

DDD\_Rx\_DS: Days' supply for prescription.

DDD\_Pt\_dose: Average daily dose for patient.

DDD\_Pt\_DS: Average days' supply for patient.

**Examples**

```
#Patient collects 100 tablets of 5 mg warfarin on January 3rd,
#and 100 tablets of 7 mg warfarin on February 1st.

#Generate a simulated dataset

library(dplyr)
n_patients <- 10
n_records <- 80
data <- data.frame(ID = rep(c(1 : n_patients), each = n_records))
data %>%
  group_by(ID) %>%
  mutate(ServDate = as.Date('2020/01/01') + abs(round(rnorm(n = 80, 700, 330))),
         DSPD_QTY = abs(round(rnorm(n = 80, 43, 28))),
         strength = abs(round(rnorm(n = 80, 4, 1)))) -> data
data <- as.data.frame(data)

# Using 1 unit DDD:

data_new <- definedDailyDose (data, WHO_ddd = 7.5, Pt_level = TRUE,
                             id = "ID", dspd_qty = "DSPD_QTY", strength = "strength",
                             tot_dose_disp = NULL)

#WHO_ddd is set as 7.5 mg as that is the defined daily dose set by WHO for warfarin.

#tot_dose_disp: 500mg on January 3rd and 700mg for February 1st.
#DDD_Rx_dose: 7.5 mg for each prescription fill
#DDD_Rx_DS is: For Jan 3rd: 500/7.5 = 66.66 day;
#              For Feb 1st: 700/7.5=93.33 days

#Pt_level can be set as TRUE to get mean values for each patient
#DDD_Pt_dose: (7.5+ 7.5)/2 = 7.5 mg
#DDD_Pt_DS: (66.66+ 93.33)/2 = 79.99 days
```

---

 fixedTablet
 

---

*Fixed tablet***Description**

Computes the daily dose and days' supply for prescriptions by assuming an average daily consumption of a fixed number of tablets (usually 1) per day by the patient.

**Usage**

```
fixedTablet(
  data,
  tablet = 1,
  dspd_qty,
  strength,
  id,
  serv_date,
  tot_dose_disp = NULL,
  Pt_level = FALSE
)
```

**Arguments**

data	Sample simulated data. Data may have multiple rows per person (one row per prescription fill). Required columns include: 1. ID: Patient's unique identification number 2. ServDate: Date on which each prescription was filled. 3. DSPD_QTY: Dispensed quantity: Number of tablets dispensed to patient at each prescription fill. 4. strength: Strength of the tablets dispensed.
tablet	Number of tablets assumed to be consumed by the patient per day. Default=1.
dspd_qty	Dispensed quantity: Number of the dispensed tablets to the patient at each prescription fill.
strength	Strength of the tablet dispensed in milligrams.
id	Unique patient identification number.
serv_date	Date of the prescription fill.
tot_dose_disp	Total dose dispensed: dispensed quantity x strength of the tablets dispensed for each prescription fill.
Pt_level	When TRUE, the estimated daily dose and days' supply are averaged for the patient.

**Details**

The fixed tablet method can be used for any medication. However, its accuracy has been shown to differ between drug classes.

**Value**

fixedTablet returns a dataset called "fixedTablet\_result". This data set includes all the variables originally in the data, plus the following:

tot\_dose\_disp: Total dose dispensed at prescription fill: dispensed quantity x strength of the tablet dispensed.

fixed\_1\_tab\_Rx\_dose: Daily dose for prescription.

fixed\_1\_tab\_Rx\_DS: Days' supply for prescription.

fixed\_1\_tab\_Pt\_dose: Average daily dose for patient.

fixed\_1\_tab\_Pt\_DS: Average days' supply for patient.

**Examples**

```
#Patient collects 100 tablets of 5 mg warfarin on January 3rd,
#and 100 tablets of 7 mg warfarin on February 1st.

#Generate a simulated dataset

library(dplyr)
n_patients <- 10
n_records <- 80
data <- data.frame(ID = rep(c(1 : n_patients), each = n_records))
data %>%
  group_by(ID) %>%
  mutate(ServDate = as.Date('2020/01/01') + abs(round(rnorm(n = 80, 700, 330))),
         DSPD_QTY = abs(round(rnorm(n = 80, 43, 28))),
         strength = abs(round(rnorm(n = 80, 4, 1)))) -> data
data <- as.data.frame(data)

#Assuming consumption of 1 tablet per day:

data_new <- fixedTablet(data, tablet = 1, Pt_level = FALSE, id = "ID",
                       dspd_qty = "DSPD_QTY", strength = "strength",
                       serv_date = "ServDate", tot_dose_disp = NULL)

#tot_dose_disp: 500mg on January 3rd and 700mg for February 1st.
#fixed_1_tab_Rx_dose: 5 mg for the prescription refill on Jan 3rd, 7 mg for prescription
# refill on Feb 1st.
#fixed_1_tab_Rx_DS is: For Jan 3rd: 500/5= 100 day; For Feb 1st: 700/7= 100 days

#pt_level can be set as TRUE to get mean values for each patient
#DDD_1_Pt_dose: (5+ 7)/2 = 6 mg
#DDD_1_Pt_DS: (100+100)/2 = 100 days
```

**Description**

Computes the daily dose and days' supply for prescriptions by assuming a fixed number of days of exposure (usually 90 days) for all patients, reflecting the medication supply policies of most medication insurance plans.

**Usage**

```
fixedWindow(
  data,
  window_length = 90,
  dspd_qty,
  strength,
  id,
  serv_date,
  tot_dose_disp = NULL,
  Pt_level = FALSE
)
```

**Arguments**

data	Sample simulated data. Data may have multiple rows per person (one row per prescription fill). Required columns include: 1. ID: Patient's unique identification number 2. ServDate: Date on which each prescription was filled. 3. DSPD_QTY: Dispensed quantity: Number of tablets dispensed to patient at each prescription fill. 4. strength: Strength of the tablets dispensed.
window_length	The number of days that patients' supply of medication is assumed to last after each prescription refill. Default= 90 days.
dspd_qty	Dispensed quantity: Number of the dispensed tablets to the patient at each prescription fill.
strength	Strength of the tablet dispensed in milligrams.
id	Unique patient identification number.
serv_date	Date of the prescription fill.
tot_dose_disp	Total dose dispensed: dispensed quantity x strength of the tablets dispensed for each prescription fill.
Pt_level	When TRUE, the estimated daily dose and days' supply are averaged for the patient.

**Value**

fixedWindow returns a dataset called "fixedWindow\_result". This data set includes all the variables originally in the data, plus the following:

tot\_dose\_disp: Total dose dispensed at prescription fill: dispensed quantity x strength of the tablet dispensed.

fixed\_window\_90\_wind\_Rx\_dose: Daily dose for prescription.

fixed\_90\_wind\_Rx\_DS: Days' supply for prescription.

fixed\_90\_wind\_Pt\_dose: Average daily dose for patient.

fixed\_90\_wind\_Pt\_DS: Average days' supply for patient.

### Examples

```
#Patient collects 100 tablets of 5 mg warfarin on January 3rd,
#and 100 tablets of 7 mg warfarin on February 1st.

#' #Generate a simulated dataset

library(dplyr)
n_patients <- 10
n_records <- 80
data <- data.frame(ID = rep(c(1 : n_patients), each = n_records))
data %>%
  group_by(ID) %>%
  mutate(ServDate = as.Date('2020/01/01') + abs(round(rnorm(n = 80, 700, 330))),
         DSPD_QTY = abs(round(rnorm(n = 80, 43, 28))),
         strength = abs(round(rnorm(n = 80, 4, 1)))) -> data
data <- as.data.frame(data)

#Assuming window length of 90 days
data_new <- fixedWindow(data, window_length = 90, id = "ID",
                        dspd_qty = "DSPD_QTY", strength = "strength",
                        serv_date = "ServDate", tot_dose_disp = NULL,
                        Pt_level = TRUE)

#tot_dose_disp = 500mg on January 3rd and 700 mg for February 1st.
#fixed_90_wind_Rx_dose : 500/90 = 5.55 mg for prescription filled on Jan 3rd;
#                          700/90=7.77 mg for prescription filled on Feb 1st.
#fixed_90_wind_Rx_DS: 90 days for all prescriptions

#pt_level can be set as TRUE to get mean values for each patient
#fixed_90_wind_Pt_dose : (5.55 + 7.77)/2 = 6.66 mg
#fixed_90_wind_Pt_DS: (90 + 90)/2 = 90
```

---

REWarDS

*REWarDS*

---

### Description

Estimates patients' individualized average daily dose and subsequently, days' supply, by fitting a random effects linear regression model to patients' cumulative dose over time. Model parameters include a minimal universally-available set of variables from prescription records.

### Usage

```
REWarDS(
  data,
```

```

    dspd_qty,
    strength,
    id,
    gap_handling = "none",
    permissible_gap = NULL,
    serv_date,
    tot_dose_disp = NULL,
    Pt_level = TRUE
)

```

### Arguments

data	Sample simulated data. Data may have multiple rows per person (one row per prescription fill). Required columns include: 1. ID: Patient's unique identification number 2. ServDate: Date on which each prescription was filled. 3. DSPD_QTY: Dispensed quantity: Number of tablets dispensed to patient at each prescription fill. 4. strength: Strength of the tablets dispensed.
dspd_qty	Dispensed quantity: Number of tablets dispensed to the patient in at each prescription fill.
strength	Strength of the dispensed tablets in milligrams.
id	Unique patient identification number.
gap_handling	Method to handle gaps between prescription fills that are more than the permissible gap. Currently, gaps can be handled in three ways: 1) The "None" method: This is the default and it ignores gaps 2)The "Initial consecutive Rx" method: Starting from the first prescription fill, patients' prescription refills are considered until the permissible gap is exceeded. REWarDS uses these initial prescription refills to estimates patient's individualized daily dose. If the permissible gap is exceeded after the first fill, there will only be one prescription for REWarDS to use, and as of now, REWarDS is unable to provide estimates of daily dose based on a single prescription. 3) The "Longest consecutive Rx" method: Looks at all periods with consecutive. prescription refills with gaps between them that do not exceed the permissible gap) during the follow up, it then counts the number of prescription fills in each period, and picks the period with the highest number of prescription fills and estimates the patient's average daily dose during that period.
permissible_gap	Gap (in days) allowed between prescription fills.
serv_date	Date of the prescription fill.
tot_dose_disp	Total dose dispensed: dispensed quantity x strength of the tablets dispensed for each prescription fill.
Pt_level	When TRUE, the estimated dose and days' supply are averaged for the patient.

### Details

REWarDS (Random Effects Warfarin Days' Supply) has been validated for warfarin. It demonstrated excellent performance that was superior to all current alternative methods for estimating days' supply of warfarin. REWarDS could potentially be used for other medications with variable



dosing regimens (e.g. tacrolimus), or in populations with high inter-individual variability in drug clearance (e.g. elderly patients). Validation with cohorts of such patients, or medications other than warfarin, has yet to be done.

### Value

REWarDS returns a dataset called "REWarDS\_result". This data set includes all the variables originally in the data, plus the following:

tot\_dose\_disp: Total dose dispensed at prescription fill: dispensed quantity x strength of the tablet dispensed.

REWarDS\_avg\_daily\_dose: Patient's individualized average daily dose.

REWarDS\_Rx\_DS: Days' supply for prescription.

REWarDS\_Pt\_DS: Average days' supply for patient.

### Examples

```
#Patient collects 100 tablets of 5 mg warfarin on January 3rd,
#and 100 tablets of 7 mg warfarin on February 1st.
```

```
#' #Generate a simulated dataset
```

```
library(dplyr)
n_patients <- 10
n_records <- 80
data <- data.frame(ID = rep(c(1 : n_patients), each = n_records))
data %>%
  group_by(ID) %>%
  mutate(ServDate = as.Date('2020/01/01') + abs(round(rnorm(n = 80, 700, 330))),
         DSPD_QTY = abs(round(rnorm(n = 80, 43, 28))),
         strength = abs(round(rnorm(n = 80, 4, 1)))) -> data
data <- as.data.frame(data)
```

```
data_new <- REWarDS(data, id = "ID", dspd_qty = "DSPD_QTY",
                  strength = "strength", serv_date = "ServDate",
                  tot_dose_disp = NULL, Pt_level = FALSE,
                  gap_handling = "none", permissible_gap = NULL)
```

```
#tot_dose_disp: 500mg on January 3rd and 700mg for February 1st.
#REWarDS_avg_daily_dose: patient's individualized average daily dose obtained
#                          from regression analysis
#REWarDS_Rx_DS: 500mg/ patient's individualized average daily dose, for Jan 3rd
#               700mg/patient's individualized average daily dose , for Feb 1st
```

```
#Pt_level can be set as TRUE to get mean values for each patient
#REWarDS_Pt_DS: average of days' supply on Jan 3rd and Feb 1st
```

```
#Gap handling method can be specified
data_new <- REWarDS(data, id = "ID", dspd_qty = "DSPD_QTY",
                  strength = "strength", serv_date = "ServDate",
                  tot_dose_disp = NULL, Pt_level = TRUE,
```

```
                gap_handling = "Longest consecutive Rx", permissible_gap = 30)
#gap: Gap in number of days between each prescription and the prescription preceding it
#Rx_count: Number of prescriptions in each period of consecutive prescriptions until
#          the permissible gap is exceeded.
```

# Index

definedDailyDose, [2](#)

fixedTablet, [4](#)

fixedWindow, [5](#)

REWARDS, [7](#)