Package: hcruR (via r-universe)

October 28, 2026

Title Estimate, Compare, and Visualize Healthcare Resource Utilization for Real-World Evidence

Version 1.0.1

Description Tools to estimate, compare, and visualize healthcare resource utilization using data derived from electronic health records or real-world evidence sources. The package supports pre index and post index analysis, patient cohort comparison, and customizable summaries and visualizations for clinical and health economics research. Methods implemented are based on Scott et al. (2022) <doi:10.1080/13696998.2022.2037917> and Xia et al. (2024) <doi:10.14309/ajg.0000000000002901>.

Depends R (>= 4.2.0)

Imports checkmate, dplyr, ggplot2, glue, gtsummary, purrr, rlang

Suggests covr, devtools, knitr, pkgdown, remotes, rmarkdown, testthat (>= 3.0.0), tibble

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Language en-US

RoxygenNote 7.3.2

VignetteBuilder knitr, rmarkdown

BugReports https://github.com/mumbarkar/hcruR/issues

Config/testthat/edition 3

Config/Needs/website pkgdown

URL https://github.com/mumbarkar/hcruR

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev

Repository https://mumbarkar.r-universe.dev

Date/Publication 2026-08-29 18:57:42 UTC

RemoteUrl https://github.com/mumbarkar/hcrur

RemoteRef HEAD

RemoteSha f2e918cd0cd7857561065a09a2047bee8ace974b

Contents

estimate_hcru 2
hcru_sample_data 4
plot_hcru 4
preproc_hcru_fun 6
summarize_descriptives 7
summarize_descriptives_gt 8

Index **11**

`estimate_hcru` *estimate_hcru*

Description

This function calculates estimates of healthcare resource utilization (HCRU) from electronic health record data across various care settings (e.g., IP, OP, ED/ER). It provides descriptive summaries of patient counts, encounters, costs, length of stay, and readmission rates for pre- and post-index periods.

Usage

```
estimate_hcru(
  data,
  cohort_col = "cohort",
  patient_id_col = "patient_id",
  admit_col = "admission_date",
  discharge_col = "discharge_date",
  index_col = "index_date",
  visit_col = "visit_date",
  encounter_id_col = "encounter_id",
  setting_col = "care_setting",
  cost_col = "cost_usd",
  readmission_col = "readmission",
  time_window_col = "period",
  los_col = "length_of_stay",
  custom_var_list = NULL,
  pre_days = 180,
  post_days = 365,
  readmission_days_rule = 30,
  group_var_main = "cohort",
  group_var_by = "care_setting",
  test = NULL,
  timeline = "Pre",
  gt_output = TRUE
)
```

Arguments

`data`	A dataframe specifying the health care details.
`cohort_col`	A character specifying the name of the cohort column.
`patient_id_col`	A character specifying the name of the patient identifier column.
`admit_col`	A character specifying the name of the date of admission column.
`discharge_col`	A character specifying the name of the date of discharge column.
`index_col`	A character specifying the name of the index date or diagnosis column.
`visit_col`	A character specifying the name of the date of visit/claim column.
`encounter_id_col`	A character specifying the name of the encounter/claim column.
`setting_col`	A character specifying the name of the HCRU setting column e.g. IP, ED, OP, etc.
`cost_col`	A character specifying the name of cost column.
`readmission_col`	A character specifying the name of readmission column.
`time_window_col`	A character specifying the name of time window column.
`los_col`	A character specifying the name of length of stay column.
`custom_var_list`	A character vector providing the list of additional columns.
`pre_days`	Number of days before index (default 180 days).
`post_days`	Number of days after index (default 365 days).
`readmission_days_rule`	Rule for how many days can be permissible to define readmission criteria in AP setting (default 30 days).
`group_var_main`	A character specifying the name of the main grouping column.
`group_var_by`	A character specifying the name of the secondary grouping column.
`test`	An optional named list of statistical tests (e.g., `list(age = "wilcox.test")`).
`timeline`	A character specifying the timeline window (default "Pre").
`gt_output`	Logical; if `TRUE`, also returns output formatted using **gtsummary** (default is `TRUE`).

Value

A list containing one or two summary data frames:

Summary by settings using dplyr A descriptive summary of HCRU metrics by cohort, setting, and time window.

Summary by settings using gtsummary (optional) Formatted summary statistics using **gtsummary**, if `gt_output = TRUE`.

Examples

```
df <- hcru_sample_data[sample(nrow(hcru_sample_data), 10), ]
estimate_hcru(data = df)
```

hcru_sample_data *Sample Cohort Data*

Description

A sample dataset representing a patient cohort with index dates.

Usage

```
hcru_sample_data
```

Format

A data frame with columns:

patient_id Unique patient identifier

cohort Cohort identifier (e.g., treatment group)

index_date Index date (as Date)

encounter_id encounter/claim identifier (e.g., claim number)

care_setting HCRU domain types (e.g., IP, OP, ER, etc.)

visit_date Visit date (as Date)

admission_date Admission date (as Date)

discharge_date Discharge date (as Date)

encounter_date Encounter/Claim date (as Date)

period period (e.g., Pre/Post)

cost_usd Cost of utilization of health resources

Source

Simulated data

plot_hcru *plot_hcru*

Description

This function provides the visualization of the events of the settings grouped by cohort and time window.

Usage

```
plot_hcru(
  summary_df,
  x_var = "time_window",
  y_var = "Cost",
  cohort_col = "cohort",
  facet_var = "care_setting",
  facet_var_n = 3,
  title = "Average total cost by domain and cohort",
  x_label = "Healthcare Setting (Domain)",
  y_label = "Average total cost",
  fill_label = "Cohort"
)
```

Arguments

summary_df	Output from estimate_hcru()
x_var	A character specifying column name to be plotted on x-axis
y_var	A character specifying column name to be plotted on y-axis
cohort_col	A character specifying cohort column name
facet_var	A character specifying column name to generate faceted plots
facet_var_n	A numeric specifying number of columns for facet output
title	A character specifying the plot title
x_label	A character specifying x-axis label
y_label	A character specifying y-axis label
fill_label	A character specifying fill legend label

Details

Plot HCRU Event Summary

Value

ggplot object

Examples

```
df <- data.frame(
  time_window = rep(c("Pre", "Post"), each = 2),
  cohort = rep(c("A", "B"), 2),
  care_setting = rep("Setting1", 4),
  Cost = c(100, 120, 110, 130)
)
plot_hcru(
  summary_df = df,
  x_var = "time_window",
  y_var = "Cost",
```

```
  cohort_col = "cohort",
  facet_var = "care_setting",
  facet_var_n = 1,
  title = "Example Plot",
  x_label = "Time Window",
  y_label = "Cost",
  fill_label = "Cohort"
)
```

`preproc_hcru_fun` *preproc_hcru_fun*

Description

This function helps to pre-process the heath care resource utilization (HCRU) for a given electronic health record data for a given set of settings e.g. IP, OP, ED/ER, etc.

Usage

```
preproc_hcru_fun(
  data,
  cohort_col = "cohort",
  patient_id_col = "patient_id",
  admit_col = "admission_date",
  discharge_col = "discharge_date",
  index_col = "index_date",
  visit_col = "visit_date",
  encounter_id_col = "encounter_id",
  setting_col = "care_setting",
  pre_days = 180,
  post_days = 365,
  readmission_days_rule = 30
)
```

Arguments

`data`	A dataframe specifying the health care details
`cohort_col`	A character specifying the name of the cohort column
`patient_id_col`	A character specifying the name of the patient identifier column
`admit_col`	A character specifying the name of the date of admission column
`discharge_col`	A character specifying the name of the date of discharge column
`index_col`	A character specifying the name of the index date or diagnosis column
`visit_col`	A character specifying the name of the date of visit/claim column
`encounter_id_col`	A character specifying the name of the encounter/claim column

setting_col	A character specifying the name of the HCRU setting column e.g. IP, ED, OP, etc.
pre_days	Number of days before index (default 180 days)
post_days	Number of days after index (default 365 days)
readmission_days_rule	Rule for how many days can be permissible to define readmission criteria in AP setting (default 30 days)

Value

dataframe with HCRU estimates.

Examples

```
preproc_hcru_fun(data = hcru_sample_data)
```

---

summarize_descriptives

Generate Detailed Descriptive Statistics

Description

Generate Detailed Descriptive Statistics

Usage

```
summarize_descriptives(
  data,
  patient_id_col = "patient_id",
  setting_col = "care_setting",
  cohort_col = "cohort",
  encounter_id_col = "encounter_id",
  cost_col = "cost_usd",
  los_col = "length_of_stay",
  readmission_col = "readmission",
  time_window_col = "time_window"
)
```

Arguments

data	A dataframe with variables to summarize.
patient_id_col	A character specifying the name of patient identifier column
setting_col	A character specifying the name of HRCU setting column
cohort_col	A character specifying the name of cohort column

encounter_id_col
: A character specifying the name of encounter/claim column

cost_col
: A character specifying the name of cost column

los_col
: A character specifying the name of length of stay column

readmission_col
: A character specifying the name of readmission column

time_window_col
: A character specifying the name of time window column

Value

A table object

Examples

```
if (requireNamespace("dplyr", quietly = TRUE) &&
    requireNamespace("checkmate", quietly = TRUE)) {
  hcru_sample_data <- data.frame(
    patient_id = rep(1:10, each = 2),
    cohort = rep(c("A", "B"), 10),
    care_setting = rep(c("IP", "OP"), 10),
    admission_date = Sys.Date() - sample(1:100, 20, TRUE),
    discharge_date = Sys.Date() - sample(1:90, 20, TRUE),
    index_date = Sys.Date() - 50,
    visit_date = Sys.Date() - sample(1:100, 20, TRUE),
    encounter_id = 1:20,
    cost_usd = runif(20, 100, 1000)
  )
  df <- preproc_hcru_fun(data = hcru_sample_data)
  summary_df <- summarize_descriptives(data = df)
  # Only keep required columns for demonstration
  summary_df$LOS <- ifelse(summary_df$care_setting == "IP",
    sample(1:10, nrow(summary_df), TRUE), NA)
  summary_df$Readmission <- ifelse(summary_df$care_setting == "IP",
    sample(0:1, nrow(summary_df), TRUE), NA)
  summary_df$time_window <- "Pre"
  summary_df
}
```

---

summarize_descriptives_gt

Generate Detailed Descriptive Statistics with Custom P-Value Tests

---

Description

Generate Detailed Descriptive Statistics with Custom P-Value Tests

Usage

```
summarize_descriptives_gt(
  data,
  patient_id_col = "patient_id",
  var_list = NULL,
  group_var_main = "cohort",
  group_var_by = "care_setting",
  test = NULL,
  timeline = "Pre"
)
```

Arguments

data	A dataframe with variables to summarize from the output of the summarize_descriptives function. Kindly filter the data for timeline.
patient_id_col	A character specifying the name of patient identifier column.
var_list	Optional quoted variable list (e.g. care_setting).
group_var_main	A character specifying the name of the main grouping column.
group_var_by	A character specifying the name of the secondary grouping column.
test	Optional named list of statistical tests (e.g. age ~ "wilcox.test").
timeline	A character specifying the timeline window (default "Pre").

Value

A gtsummary table object

Examples

```
if (requireNamespace("gtsummary", quietly = TRUE) &&
    requireNamespace("dplyr", quietly = TRUE) &&
    requireNamespace("purrr", quietly = TRUE) &&
    requireNamespace("checkmate", quietly = TRUE) &&
    requireNamespace("glue", quietly = TRUE)) {
  hcru_sample_data <- data.frame(
    patient_id = rep(1:10, each = 2),
    cohort = rep(c("A", "B"), 10),
    care_setting = rep(c("IP", "OP"), 10),
    admission_date = Sys.Date() - sample(1:100, 20, TRUE),
    discharge_date = Sys.Date() - sample(1:90, 20, TRUE),
    index_date = Sys.Date() - 50,
    visit_date = Sys.Date() - sample(1:100, 20, TRUE),
    encounter_id = 1:20,
    cost_usd = runif(20, 100, 1000)
  )
  df <- preproc_hcru_fun(data = hcru_sample_data)
  summary_df <- summarize_descriptives(data = df)
  # Only keep required columns for demonstration
  summary_df$LOS <- ifelse(summary_df$care_setting == "IP",
```

```
sample(1:10, nrow(summary_df), TRUE), NA)
  summary_df$Readmission <- ifelse(summary_df$care_setting == "IP",
sample(0:1, nrow(summary_df), TRUE), NA)
  summary_df$time_window <- "Pre"
  # Run the function (should execute within 5 seconds)
  summarize_descriptives_gt(
    data = summary_df,
    patient_id_col = "patient_id",
    var_list = c("Visits", "Cost", "LOS", "Readmission"),
    group_var_main = "cohort",
    group_var_by = "care_setting",
    timeline = "Pre"
  )
}
```

Index

∗ **datasets**
    hcru_sample_data, 4

estimate_hcru, 2

hcru_sample_data, 4

plot_hcru, 4
preproc_hcru_fun, 6

summarize_descriptives, 7
summarize_descriptives_gt, 8