

# Package: patientcounter (via r-universe)

September 4, 2024

**Title** Count Hospital Patients Quickly

**Version** 0.1.0

**Description** Fast, flexible census of patient occupancy in multiple locations in hospital setting. Obtain the net count of patients by interval, accounting for those admitted prior to specific intervals, those who move during intervals, and those not yet discharged, without requiring a database or knowledge of 'SQL'.

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 2.10)

**Imports** data.table, lubridate, magrittr

**LazyData** true

**URL** <https://github.com/johnmackintosh/patientcounter>

**BugReports** <https://github.com/johnmackintosh/patientcounter/issues>

**RoxygenNote** 7.1.1

**Suggests** testthat (>= 2.1.0)

**Repository** <https://johnmackintosh.r-universe.dev>

**RemoteUrl** <https://github.com/johnmackintosh/patientcounter>

**RemoteRef** HEAD

**RemoteSha** 9cddc4d91f461da137ab8e0f8282c8116d9d90b8

## Contents

beds . . . . .	2
interval_census . . . . .	2

<b>Index</b>	<b>5</b>
--------------	----------

---

beds	<i>#' 10 observations showing patient time in and out.</i>
------	--

---

### Description

A dataset containing ten sets of time in and out by bed. This simulates patients moving in and out of departments or hospitals.

### Usage

```
beds
```

### Format

A data frame with 10 rows and 4 variables:

**bed** a grouping variable, representing a specific bed

**patient** individual patients

**start\_time** time each patient moved into their current bed

**end\_time** time each patient moved out of their current bed

---

interval_census	<i>Count the number of patients by interval</i>
-----------------	---

---

### Description

Counts the number of patients in each location, by the specified interval, for the duration of the patient admission. Results can be returned as a grand totals, grouped totals, or individual patient level per interval.

### Usage

```
interval_census(
  df,
  identifier,
  admit,
  discharge,
  group_var = NULL,
  time_unit = "1 hour",
  time_adjust_period = NULL,
  time_adjust_value = NULL,
  results = c("patient", "group", "total"),
  uniques = TRUE
)
```

**Arguments**

<code>df</code>	dataframe, tibble or data.table.
<code>identifier</code>	Unique patient identifier.
<code>admit</code>	Datetime of admission as POSIXct.
<code>discharge</code>	Datetime of discharge as POSIXct.
<code>group_var</code>	Optional unique character vector to identify specific patient location or responsible clinician at each interval, or at time of a change in location / responsible clinician during the interval.
<code>time_unit</code>	Character string to denote time intervals to count by e.g. "1 hour", "15 mins".
<code>time_adjust_period</code>	Optional argument which allows the user to obtain a snapshot at a specific time of day by making slight adjustments to the specified interval. Possible values are "start_sec", "start_min", "end_sec", or "end_min". For example, you may specify hourly intervals, but adjust these to 1 minute past the hour with "start_min", or several seconds before the end with "end_sec".
<code>time_adjust_value</code>	Optional. An integer to adjust the start or end of each period in minutes or seconds, depending on the chosen <code>time_adjust_period</code> (if specified).
<code>results</code>	A character string specifying the granularity of the results. 'patient' returns one row per patient, <code>group_var</code> and interval. The results can be input to external tools for further analysis or visualisation. 'group' provides an overall grouped count of patients by the specified time interval. 'total' returns the grand total of patients by each unique time interval.
<code>uniques</code>	Logical. Specifies how to deal with patients who move during an interval, and subsequently have two or more records per interval. Set "uniques" to TRUE to get a distinct count of patients per interval. To be clear, TRUE will count patients only once per interval. Setting "uniques" to FALSE will count each patient entry per interval. If a patient moves during an interval then at least two rows will be returned for that patient for that particular interval. This is useful if you want to count occupied beds, or track moves or transfers between departments. In general, if you use a grouping variable, set "uniques" to FALSE.

**Value**

data.table showing the patient identifier, the specified group variable, and the count by the relevant unit of time. Also included are the start and end of the interval, plus the date and base hour for convenient interactive filtering of the results.

**Examples**

```
interval_census(beds, identifier = "patient", admit = "start_time",
discharge = "end_time", time_unit = "1 hour", results = "total")
```

```
interval_census(beds, identifier="patient", admit="start_time",  
discharge="end_time", time_unit="1 hour", results="patient")
```

```
interval_census(beds, identifier="patient", admit="start_time",  
discharge="end_time", group_var="bed",  
time_unit="1 hour", results="group", uniques=FALSE)
```

# Index

\* **datasets**

beds, [2](#)

beds, [2](#)

interval\_census, [2](#)