

# Retrieving Monthly Record Counts (rc)

Mark White | [markhwhiteii@gmail.com](mailto:markhwhiteii@gmail.com)

5/8/2017

This document describes R functions that will create reports summarizing how many records a feed or facility reported in a given month. Each row represents a day in the month, each column represents a feed or facility, and the cells represent how many records were reported. These reports can be generated for visit records (i.e., how many records were reported for *patient visits* that happened during that month?) and/or arrival records (i.e., how many records *arrived at the CDC* during that month?). Two functions will be discussed: `feedrc`, which reports record counts by feed, and `facilityrc`, which reports them by facility.

To use these functions, you will need to first run the code in the `rc.R` document. (*Note:* For those interested in editing the code, I discuss it line-by-line in the `rcExplainer.pdf` document).

## Preparing to use the `feedrc` and `facilityrc` functions

First, you will have to set up a connection using the `odbcConnect` function in the `RODBC` package. Load this package using the `library(RODBC)` command. An example of this code looks like:

```
channel <- odbcConnect("Biosense_Platform", "USERNAME", "*****")
```

Second, you will have to install the `lubridate`, `data.table`, `dplyr`, and `Hmisc` packages before using these functions, since the functions discussed here depend on those packages (as well as the `RODBC` package). After you have this set up, you are ready to run the functions.

## The `feedrc` function

This function will return a data frame that summarizes records by feed (the columns) and day (the rows).

This function has five arguments:

1. `channel` = the channel you set up through `odbcConnect`.
2. `visit` = a logical argument (TRUE or FALSE) for whether or not you want to retrieve counts for visits happening during that month.
3. `arrived` = a logical argument (TRUE or FALSE) for whether or not you want to retrieve counts for records that arrived during that month.
4. `month` = the month you want the report for (i.e., "01", "02", ..., "12").
5. `year` = the year you want the report for (e.g., "2017").

*Note:* The arguments for month and year must be put between quotation marks.

## The `facilityrc` function

This function will return a data frame that summarizes records by facility (the columns) and day (the rows). This function has the same five arguments as the `feedrc` function.

## Examples

Let's say I wanted to generate a report detailing how many records arrived at the CDC in April 2017 for each feed. I would enter:

```
feedarrivals <- feedrc(channel=channel, visit=FALSE, arrived=TRUE,
month="04", year="2017")
```

This data frame will be too big for this document, so I will display only the first six rows and the first five feeds:

```
kable(feedarrivals[c(1:6),c(1:6)])
```

Arrived_Date	kansashin	KNMercy	KSAnderson	KSCommunityMemorial	KSGeary
2017-04-01	4517	371	7	30	223
2017-04-02	4311	229	11	41	337
2017-04-03	5967	284	39	39	387
2017-04-04	5191	301	0	0	326
2017-04-05	5680	387	0	0	331
2017-04-06	5041	337	21	0	265

This shows us that 4,517 kansashin records arrived on April 1st, 2017, for example. But no records arrived on the 4th, 5th, or 6th from KSCommunityMemorial. Is this because there were no visits? We can check that, too:

```
feedvisits <- feedrc(channel=channel, visit=TRUE, arrived=FALSE, month="04",
year="2017")
```

Again, I will only display the first few rows and cases:

```
kable(feedvisits[c(1:6),c(1:6)])
```

Visit_Date	kansashin	KNMercy	KSAnderson	KSCommunityMemorial	KSGeary
2017-04-01	5261	397	12	28	358
2017-04-02	5273	262	33	56	329
2017-04-03	5015	214	19	52	338
2017-04-04	4365	314	9	12	251
2017-04-05	4781	349	23	28	281
2017-04-06	4450	379	9	17	297

It looks like, although KSCommunityMemorial did not report anything on those days, they had 59 records for visits during these days. What this could mean is that their reporting system was down for a few days, but they managed to get their system back up and report records that happened on those days at a later time. One could also look at arrivals and visits in the same table. This time, let's look at the month of March:

```
feedboth <- feedrc(channel=channel, visit=TRUE, arrived=TRUE, month="03",
year="2017")
kable(feedboth[c(1:6),c(1,4:7)])
```

Date	KNMercy_A	KNMercy_V	KSAnderson_A	KSAnderson_V
2017-03-01	351	333	34	20
2017-03-02	277	250	28	26
2017-03-03	249	259	34	27
2017-03-04	387	514	9	28
2017-03-05	522	491	21	35
2017-03-06	423	307	38	34

Note that the columns have now changed slightly. Each feed name is suffixed with either "\_A" or "\_V" for arrival or visit, respectively. For KSAnderson on March 4th, 9 records arrived to the CDC, but 28 happened on that date.

The exact same reports that we just generated for feed can also be generated for facility. The column names represent the C\_Biosense\_Facility\_ID, whereas the labels of the columns represent the Facility\_Name from the MFT dataset:

```
facilityarrivals <- facilityrc(channel=channel, visit=FALSE, arrived=TRUE,
month="04", year="2017")
facilityvisits <- facilityrc(channel=channel, visit=TRUE, arrived=FALSE,
month="04", year="2017")
facilityboth <- facilityrc(channel=channel, visit=TRUE, arrived=TRUE,
month="03", year="2017")
kable(facilityarrivals[c(1:6),c(1:6)])
```

Arrived_Date	3782	3788	3791	3792	3793
2017-04-01	10	19	282	4	415
2017-04-02	6	16	252	6	444
2017-04-03	38	25	200	16	515
2017-04-04	26	7	318	7	502
2017-04-05	13	36	250	2	466
2017-04-06	8	36	241	7	466

```
kable(facilityvisits[c(1:6),c(1:6)])
```

Visit_Date	3782	3788	3791	3792	3793
2017-04-01	23	22	234	11	393

2017-04-02	30	23	235	13	509
2017-04-03	12	10	321	5	530
2017-04-04	8	18	278	4	429
2017-04-05	8	13	185	10	468
2017-04-06	5	27	246	0	458

```
kable(facilityboth[c(1:6),c(1:7)])
```

Date	3782_A	3782_V	3788_A	3788_V	3791_A	3791_V
2017-03-01	54	29	36	23	309	256
2017-03-02	12	6	31	10	294	248
2017-03-03	18	17	21	34	270	219
2017-03-04	8	19	24	30	281	297
2017-03-05	35	35	23	32	274	272
2017-03-06	21	41	32	37	1	467

If you look at these inside RStudio using the `View()` command, the facility names will show up right below the facility ID numbers. One can list all of them in the console by using the `label()` command (for brevity, I'll only list the first 5 facility names):

```
label(facilityarrivals)[2:6]
```

```
##                                3782                                3788
##      "Rooks County Health Center"                                "Scott County Hospital"
##                                3791                                3792
## "Children's Mercy Hospital - South"                                "Sheridan County Hospital"
##                                3793
##                                "Stormont-Vail"
```

if you would like to export any of these tables, one could use one of the many `write. ...` functions, such as to a `.csv`:

```
write.csv(feedvisits, "feedvisits.csv", row.names=FALSE)
```

## Joining monthly reports

One may want to generate a report every month and simply append it to the last month's report, instead of storing them in different files. I have found that this presents an interesting issue: If a facility was not present in a month, the facility does not show up in the report at all. This means that, if we are joining month after month, sometimes the two data frames being bound together will differ in facilities (and thus differ in columns).

Imagine we want to bind reports for March (`marchfeed`) and April (`aprilfeed`). We can use the `bind_rows()` function from the `dplyr` package:

```
both <- bind_rows(marchfeed, aprilfeed)
```

Let's say that "Facility A" reported records in April, but not March. For every date in March, the new file will have an NA value for "Facility A". One could replace all of the NA values with zeroes in the new document by specifying:

```
both[is.na(both)] <- 0
```