Medicin Macro

A matemathical formulation of the algorithm, an efficient implementation and a new R interface

Helene Charlotte Rytgaard

University of Copenhagen, Section of Biostatistics

December 8, 2016

Register data preprocessing steps

Input:

`pop`, `lmdb`, `lpr`, `indh`, `dod`, `sogne` ...

Data preprocessing

code from hell, medicin macro, ...

Output:

Time 1	Time 2	Status	Age	Sex	Comorbidity in interval	Exposure in interval

Existing SAS Interface

Input data

Drug purchases:

	atc	eksd	pnr	strnum	apk	packsize
1	1a	2020-04-08	1000	75	3	20
2	1a	2011-10-30	1000	125	2	5
3	1a	2020-01-23	1000	125	1	5
4	1a	2015-09-17	1000	50	2	10
5	2a	2012-05-04	1000	50	3	5
6	2a	2011-10-11	1000	75	2	9
7	2a	2015-12-20	1000	125	1	9
8	2a	2015-10-27	1000	100	3	18

Admission dates:

	pnr	max_indl	inddto1	uddto1	inddto2	uddto2	inddto3	uddto3	inddto4	uddto4
1:	1000	3	2012-04-12	2012-04-25	2015-01-30	2015-02-14	2017-05-28	2017-05-31	2021-02-21	2021-03-08
2:	2000	2	2012-12-10	2012-12-15	2015-05-31	2015-06-05	NA	NA	NA	NA
3:	4000	1	2011-05-14	2011-05-23	NA	NA	NA	NA	NA	NA
4:	5000	4	2011-10-11	2011-10-13	2015-01-25	2015-02-01	2017-03-15	2017-03-23	2020-04-08	2020-04-21
5:	6000	1	2011-04-25	2011-04-30	NA	NA	NA	NA	NA	NA
6:	7000	1	2013-03-16	2013-03-21	NA	NA	NA	NA	NA	NA
7:	8000	4	2011-06-30	2011-07-02	2014-02-02	2014-02-07	2016-03-10	2016-03-23	2019-12-21	2019-12-28
8:	9000	4	2012-09-01	2012-09-14	2015-04-18	2015-05-03	2019-01-30	2019-02-12	2021-02-21	2021-02-27

Current interface (medicin macro)


```
%x_recepter(recept_data, /* forventes at indeholde variable - skulle gerne passe med DST-standarder:
                            pnr - cpr/patientidentifikation
                            atc - ATC kode
                            eksd - udleveringsdato som sas-dato
                            strnum - numerisk styrke
                            apk - antal udleverede pakker
                            packsize - antal piller i hver pakke*/

                    datoer, /* Et produkt af medicin-hjælpe-macro eller andet program som ordner ALLE indlæggelser pr PNR på
                            EEN record med fortløbende inddto uddto */


                    out, /* tabel over behandlingsperioder - navn på SAS datasæt valgt af brugeren*/
                    1a, /* atc kode - den behandling som der skal beregnes på*/
                    5, /* antal recepter der indgår i beregninger - testet med 5, altså op til 2 før og 2 efter interesserecept */
                    50,  75, 100, 125, /* Doser svarende til de følgende variable - det er pillestørrelser
                           - her og de følgende variable skal ALLE have en værdi. Hvis der findes færre skal der blot gentages*/
                    10, 50, 25, 50,    /* Mindst accepterede dosis af lægemidler på hver pillestyrke*/
                    75, 200, 150, 150,    /* Max accepterede dosis*/
                    50, 100,  75, 100,    /* Typiske doser - en slags "default" dosis - og startdosis altid ved left_only */
                    10, /* Maximum sktørrelse af "restdosis" som kan overføres til følgende receptperioder. Denne giver mulighed fo
                     at forhindre excessiv ophobning hvis små antagelser om maxdosis medfører til tiltagende stort depot
                     Max_depot er piller*styrke - Hvis der højst må gemmes 100 piller a 10 mg, så er max_depot 1000
                     */
                    '01sep12'd, /* første og sidste dato som har interesse kan angives som en "SAS-dato" eller med konventionen
                       'ddmmyy'd     */
                    '02may20'd,
                    1, /* Hvis værdien er 1 så kommer der tracking udskrift i loggen - hvis nul, så ikke. Tilsvarende slettes en ræ
                      temporære datasæt hvis værdien er 0 */
                    1, /* Hvis værdien er 1 så kommer der grafer */
                    test, /* præfix på generede variable som kan benyttes til at skelne fra lignende variable genereret i andre tri
                    1 /* danner tabeller "l_" hvor doser og sluttider KUN regnes bagud*/
                    );
```

Output data

Continuing example:

	pnr	dosis	startdag	slutdag
1	1000	50	17SEP15	06OCT15
2	1000	100	23JAN20	28JAN20
3	1000	100	08APR20	22MAY20
4	2000	20	15MAY13	05AUG13
5	2000	75	04NOV15	16NOV15
6	2000	100	15MAR17	21MAY17
7	3000	100	16MAR13	21MAR13
8	3000	100	26APR13	02MAY13
9	3000	50	10MAR16	08MAY16
10	3000	75	04JAN19	16JAN19
11	3000	100	14JUL19	04AUG19
12	3000	150	05AUG19	16AUG19
13	3000	75	17AUG19	22AUG19
14	3000	50	01NOV19	19NOV19
15	3000	75	20NOV19	16DEC19

Immediate limitations

- Speed
  - Each drug must be processed separately, repeating the macro call
- Lack of transparency
  - Mathematical formulas?
- Other issues:
  - Dependence on the future
  - Only possible to specify four different doses
  - Graphical checks (working?)?

New R Interface

Addressing the issues in SAS

- Speed
  - We use the `Rcpp` to integrate `C++`-code directly in `R`
- Lack of transparency
  - We have written up all steps in the macro explicitly
  - Everything is collected in a documentation article
- New interface
  - Object oriented coding for more effiency
  - Visualization tools to plot input data and output estimates
  - Accessed via `R`-package `heaven` (github)

In `R` the package is installed via:

```
devtools::install_github("tagteam/heaven")
```

Intro to `R`-packages: `http://r-pkgs.had.co.nz/`
How to use Git in Rstudio: `http://r-pkgs.had.co.nz/git.html`

New R interface

- Same input data sets as before (almost)
  - Admission dates data set long format

```
   pnr     inddto      uddto
1:   1 2003-12-20 2003-12-24
2:   1 2006-07-20 2006-09-01
3:   1 2007-04-30 2007-05-15
4:   1 2010-11-27 2011-01-02
5:   1 2013-05-11 2013-05-16
```

...

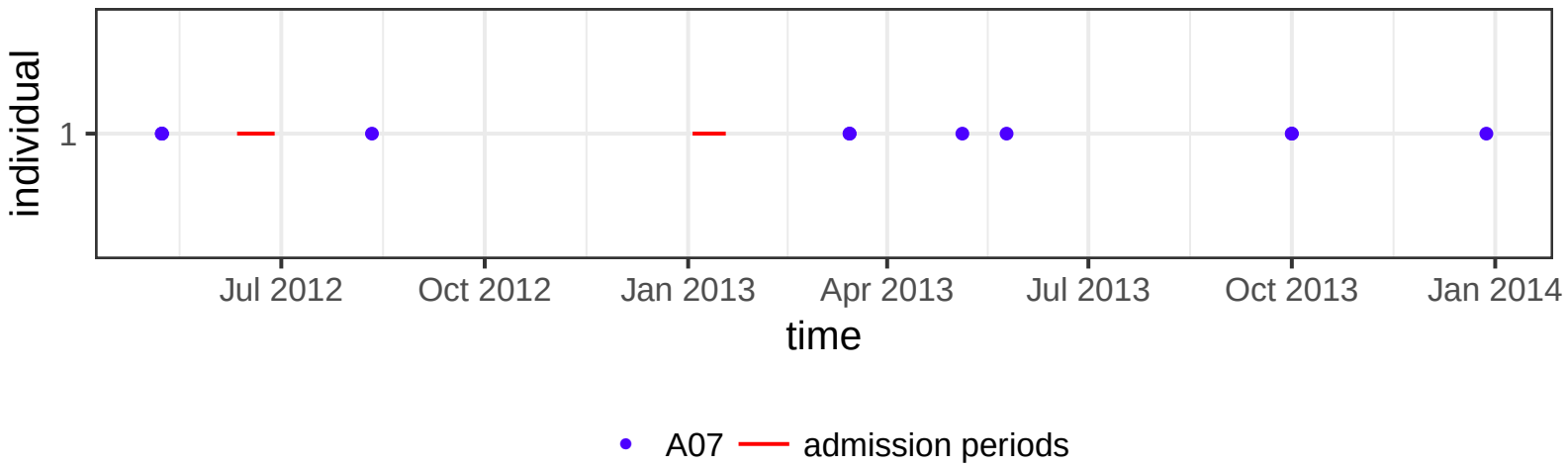
- Attach relevant data (more user details in a moment)

```
drugdb(d) <- drugdata
admdb(d) <- admdata
```

- `plot()`-function to show purchases and admission periods

```
plot(d)
```

Input visualization tools



How to use the interface

Load package:

```
library(heaven)
```

Create empty object:

```
d <- dpp()
```

Attach relevant data:

```
drugdb(d) <- drugdata
admdb(d) <- admissiondata
```

Add treatments:

```
drug(d, "treatment1") <- atc("A12B")
drug(d, "treatment1") <- pack(c(750, 75),
                  min = c(250, 25),
                  max = c(1000, 100),
                  def = c(750, 100))
```

Specify window of prescription dates to use in calculations:

```
pwindow(d) <- 3 ## include data from up to 3 previous purchase
    dates into the calculation of the daily dosis
```

How to use the interface

When everything is specified, we perform the calculations by running:

```
process(d)
```

```
$treatment1
   id   X          B          E
1   1 100 1997-08-21 2007-11-26
2   2 100 1995-09-09 2030-02-05
3   3 100 1995-06-21 1997-08-12
4   3   0 1997-08-13 1998-02-21
5   3 100 1998-02-22 2010-02-08
6   4 100 1995-01-01 2030-08-17
7   5 100 1995-02-14 1996-02-23
8   5   0 1996-02-24 1996-04-25
9   5  75 1996-04-26 1997-08-20
10  5 100 1997-08-21 2000-03-01
11  6 100 1995-01-01 1995-03-16
12  6   0 1995-03-17 1995-09-23
13  6  25 1995-09-24 1996-05-04
14  6 100 1996-05-05 2015-01-26
15  7 100 1995-06-27 1999-09-16
16  8 100 1996-09-26 2009-08-27
17  9 100 1995-05-09 1999-06-18
18  9   0 1999-06-19 1999-11-18
19  9 100 1999-11-19 2001-06-03
20 10 100 1995-09-13 2014-04-21
```

How to use the interface

We may add treaments:

```
drug(d, "treatment2") <- atc("A07")
drug(d, "treatment2") <- pack(c(200, 400, 500),
             min = c(100, 100, 250),
             max = c(400, 500, 1000),
             def = c(300, 200, 500))
```

And then perform calculations again:

```
process(d)
```

```
$treatment1
  id   X          B          E
1  1 100 1997-08-21 2007-11-26
2  2 100 1995-09-09 2030-02-05
3  3 100 1995-06-21 1997-08-12
4  3   0 1997-08-13 1998-02-21
5  3 100 1998-02-22 2010-02-08
6  4 100 1995-01-01 2030-08-17

$treatment2
  id   X          B          E
1  1 200 1996-06-15 1996-08-13
2  1   0 1996-08-14 1997-04-13
3  1 500 1997-04-14 1997-06-12
4  1   0 1997-06-13 1998-03-22
5  1 200 1998-03-23 1998-07-20
6  1   0 1998-07-21 1998-11-04
```

How to use the interface

The function can be used treatment and/or id speficic:

```
process(d, treatment = "treatment2")
```

```
$treatment2
  id   X          B          E
1  1 200 1996-06-15 1996-08-13
2  1   0 1996-08-14 1997-04-13
3  1 500 1997-04-14 1997-06-12
4  1   0 1997-06-13 1998-03-22
5  1 200 1998-03-23 1998-07-20
6  1   0 1998-07-21 1998-11-04
```

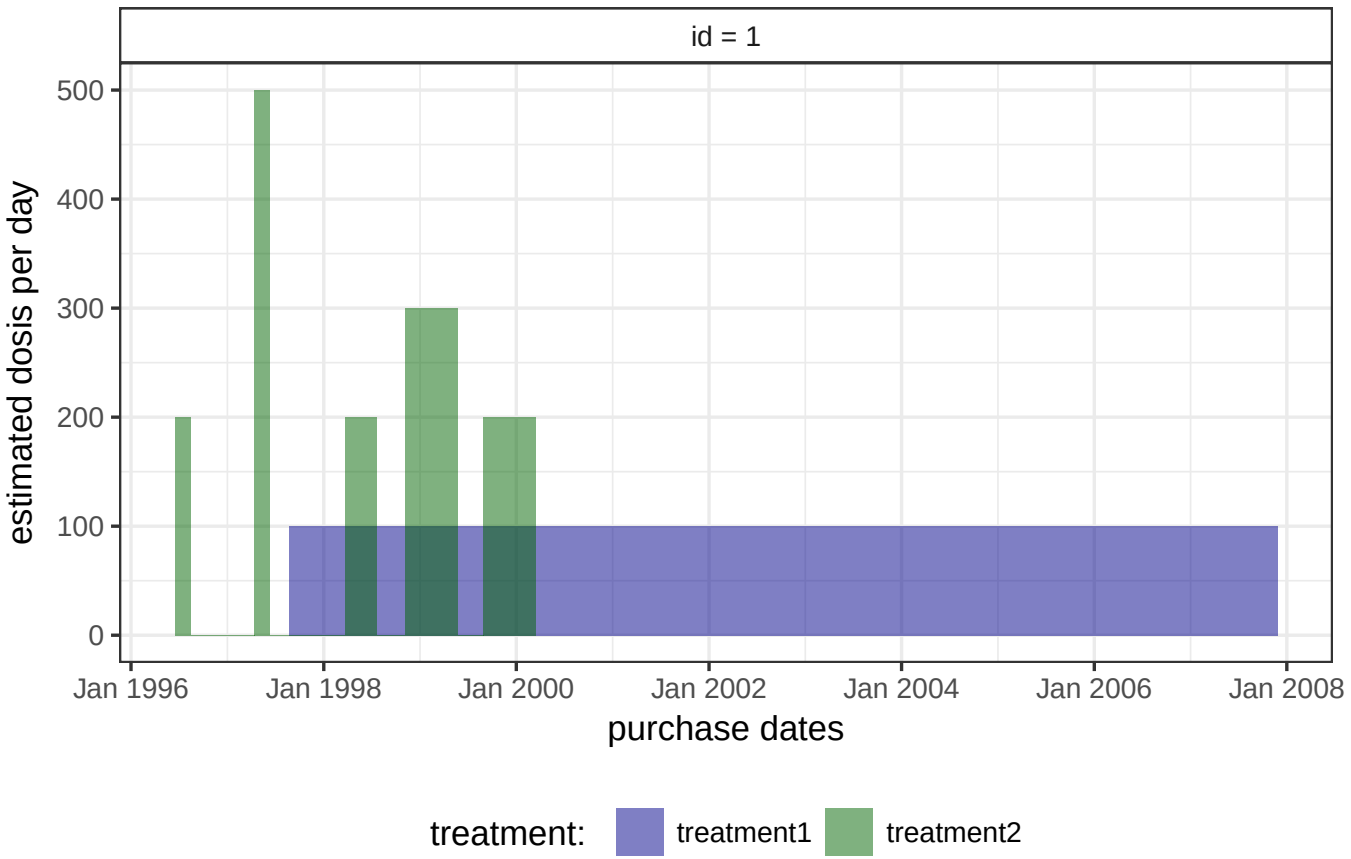
```
process(d, id = 9)
```

```
$treatment1
  id   X          B          E
1  9 100 1995-05-09 1999-06-18
2  9   0 1999-06-19 1999-11-18
3  9 100 1999-11-19 2001-06-03

$treatment2
  id   X          B          E
1  9 200 1996-02-22 1996-04-08
2  9 500 1996-04-09 1996-05-26
3  9   0 1996-05-27 1998-05-22
4  9 300 1998-05-23 1998-06-11
5  9   0 1998-06-12 1999-11-21
6  9 500 1999-11-22 2000-09-16
```

Built-in tools for output visulizations

A `plot()`-function to visualize the output:

```
out <- process(d)
plot(out, idmax = 4)
```



Technical details

. . . the mathematical part . . .

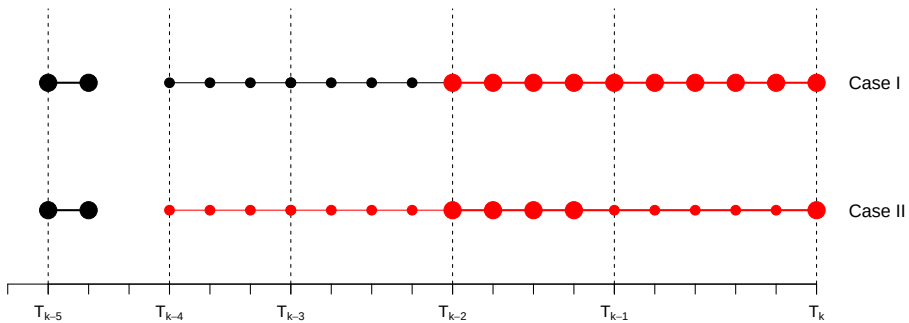
- The `R`-interface and the following formulas are all based on the implementation of `medicin macro` (`left_only`).
- The computations performed consists basically of an averaging over a set of prescriptions back in time (decided by the user)
- A number of things will for each prescription date help us determine how many dates back in time we should use for the calculations:
  - The number of days of supply of a certain drug is calculated based on the minimal possible doses for a drug
  - The actual number of dates between the prescription periods (where the number of days hospitalized is subtracted)
  - Whether or not the total amount of drug purchased at time k is approximately the same as purchased at earlier times
- Exposure periods are then calculated based on these average dose amounts

Final formula (a snippet of what we have worked on)

$$
\begin{aligned}
X_k &= (1-u_{k-1})\, s^*_{b(k)} && \text{(No overlap)}\\
&+ u_{k-1}\Bigg[&& \text{(Overlap)}\\
&\quad 1\Big\{S_{b(k-1)} = S_{b(k)}\Big\}\Bigg(1\left\{W_k > s^{\max}_{b(k)}\right\} s^{\max}_{b(k)}\\
&\quad + 1\left\{W_k < s^{\min}_{b(k)}\right\} s^{\min}_{b(k)} && \text{(I)}\\
&\quad + 1\left\{W_k \le s^{\max}_{b(k)}\right\} 1\left\{W_k \ge s^{\min}_{b(k)}\right\} W_k\Bigg)\Bigg].\\
&\quad + 1\Big\{S_{b(k-1)} \neq S_{b(k)}\Big\}\Bigg(1\left\{M^{(2)}_k > s^{\max}_{b(k)}\right\} s^{\max}_{b(k)}\\
&\quad + 1\left\{M^{(2)}_k < s^{\min}_{b(k)}\right\} s^{\min}_{b(k)} && \text{(II)}\\
&\quad + 1\left\{M^{(2)}_k \le s^{\max}_{b(k)}\right\} 1\left\{M^{(2)}_k \ge s^{\min}_{b(k)}\right\} s^*_{b(k)}\Bigg)\Bigg].
\end{aligned}
$$

$M^{(1)}_k$, $M^{(2)}_k$ are average doses over the periods

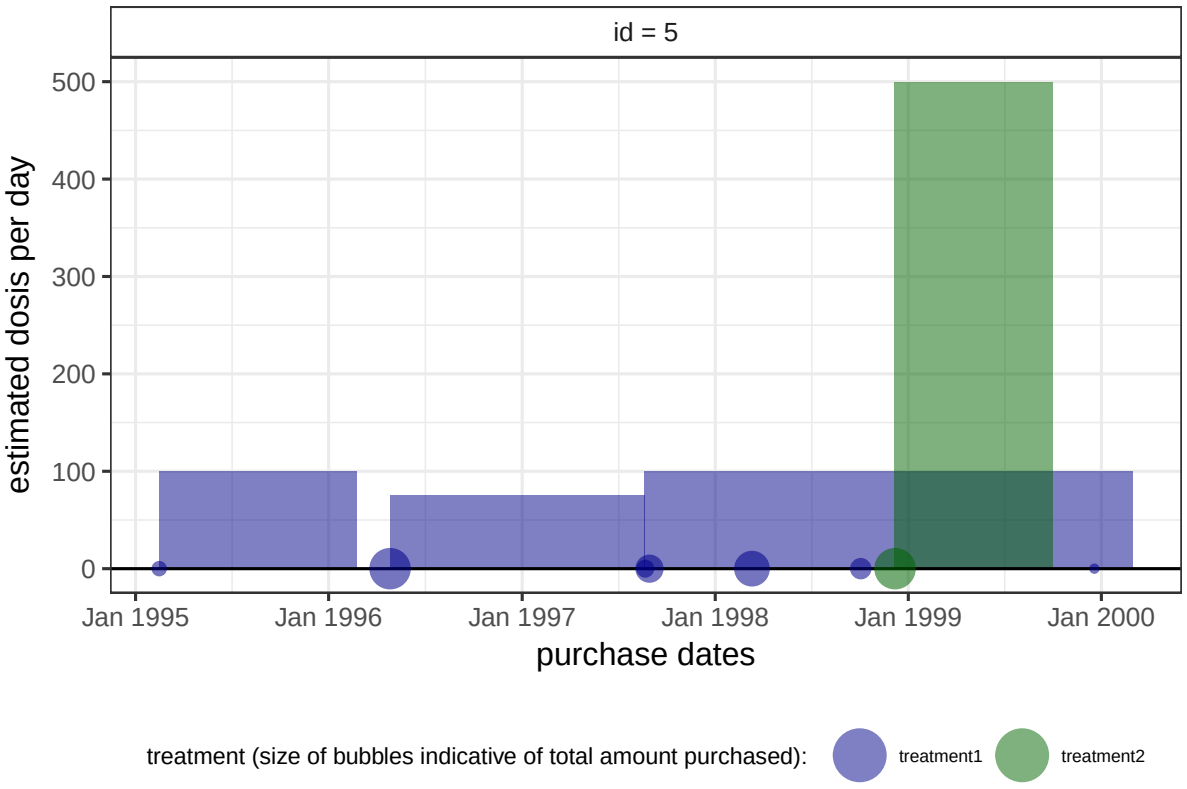
W_k is a rounding of $M^{(1)}_k$ to nearest multiple of relevant minimal dose



More output visulizations

We may also take a closer view on the underlying purchases behind the final exposures estimated:

```
out1 <- process(d, keep_data = TRUE)
plot(out1, id = 5, trace = TRUE)
```



Example: Omeprazol

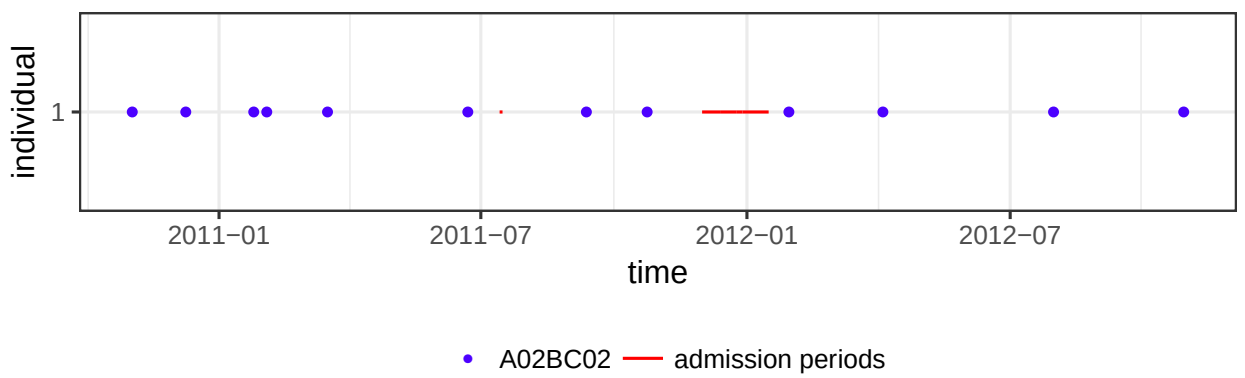
Drug purchases:

	pnr	eksd	packsize	strnum	apk	atc
1	1	25/01/2011	56	40	1	A02BC02
2	1	29/10/2012	100	40	1	A02BC02
3	1	31/07/2012	100	40	1	A02BC02
4	1	12/09/2011	28	40	1	A02BC02
5	1	24/10/2011	28	40	1	A02BC02
6	1	03/02/2011	56	40	1	A02BC02
7	1	09/12/2010	56	40	1	A02BC02
8	1	02/11/2010	56	40	1	A02BC02
9	1	04/04/2012	98	40	1	A02BC02
10	1	30/01/2012	98	40	1	A02BC02
11	1	22/06/2011	98	40	1	A02BC02
12	1	17/03/2011	98	40	1	A02BC02

Admission dates:

	inddto	uddto
2004-01-20	12437	12437
2004-01-22	12439	12440
2006-06-20	13319	13319
2006-06-23	13322	13322
2010-01-21	14630	14629
2010-01-14	14623	14635
2010-01-26	14635	14650
2010-07-05	14795	14795
2010-10-21	14903	14911
2011-07-14	15169	15171
2011-12-01	15309	15322
2011-12-14	15322	15333
2011-12-25	15333	15337
2011-12-29	15337	15355

Plotting the data:



Example: Omeprazol

Using medicin-macro:

```
%x_recepter(PPI, /* forventes at indeholde variable - skulle gerne passe med DST-standarder:
                    pnr - cpr/patientidentifikation
                    atc - ATC kode
                    eksd - udleveringsdato som sas-dato
                    strnum - numerisk styrke
                    apk - antal udleverede pakker
                    packsize - antal piller i hver pakke*/

            admData, /* Et produkt af medicin-hjælpe-macro eller andet program som ordner ALLE indlæggelser pr PNR på
                    EEN record med fortløbende inddto uddto */


            omeprazol, /* tabel over behandlingsperioder - navn på SAS datasæt valgt af brugeren*/
            A02BC02, /* atc kode - den behandling som der skal beregnes på*/
            5, /* antal recepter der indgår i beregninger - testet med 5, altså op til 2 før og 2 efter interesserecept */
            10, 20, 40, 40, /* Doser svarende til de følgende variable - det er pillestørrelser
                   - her og de følgende variable skal ALLE have en værdi. Hvis der findes færre skal der blot gentages*/
            10, 20, 40, 40,    /* Mindst accepterede dosis af lægemidler på hver pillestyrke*/
            20, 40, 60, 80,    /* Max accepterede dosis*/
            10, 20, 40, 40,    /* Typiske doser - en slags "default" dosis - og startdosis altid ved left_only */
            4000, /* Maximum sktørrelse af "restdosis" som kan overføres til følgende receptperioder. Denne giver mulighed
             at forhindre excessiv ophobning hvis små antagelser om maxdosis medfører til tiltagende stort depot
             Max_depot er piller*styrke - Hvis der højst må gemmes 100 piller a 10 mg, så er max_depot 1000
             */
            '01jan1997'd, /* første og sidste dato som har interesse kan angives som en "SAS-dato" eller med konventionen
               'ddmmyy'd     */
            '31dec2012'd,
            1, /* Hvis værdien er 1 så kommer der tracking udskrift i loggen - hvis nul, så ikke. Tilsvarende slettes en ræ
              temporære datasæt hvis værdien er 0 */
            1, /* Hvis værdien er 1 så kommer der grafer */
            test, /* præfix på generede variable som kan benyttes til at skelne fra lignende variable genereret i andre tri
            1 /* danner tabeller "l_" hvor doser og sluttider KUN regnes bagud*/
            );
```

Example: Omeprazol

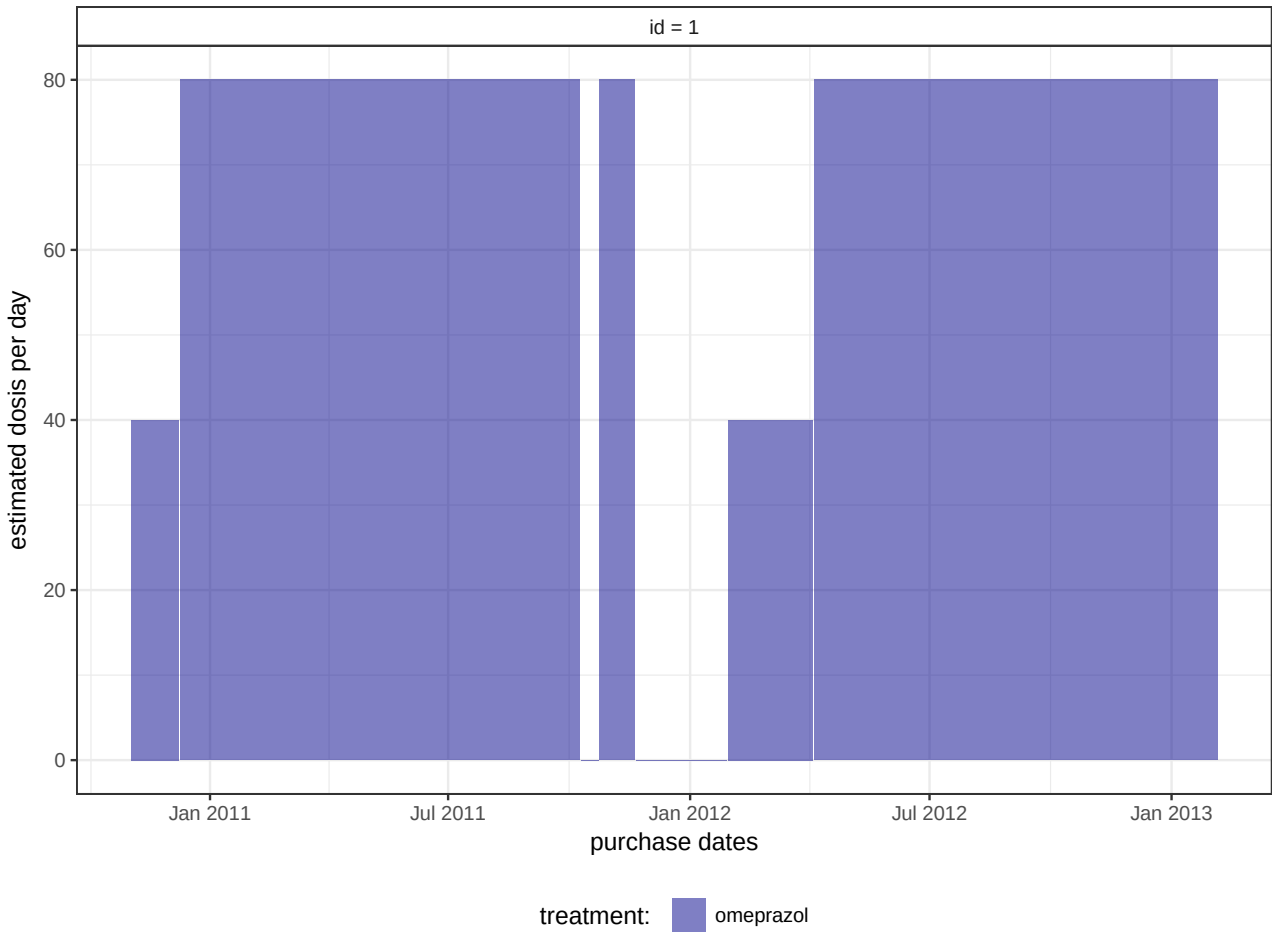

```
library(heaven)

d <- dpp()
drugdb(d) <- PPI
admdb(d) <- admData

drug(d, "omeprazol") <- atc("A02BC02")
drug(d, "omeprazol") <- pack(c(10, 20, 40, 40),
                  min = c(10, 20, 40, 40),
                  max = c(20, 40, 60, 80),
                  def = c(10, 20, 40, 40))

period(d) <- sapply(c("1997-01-01", "2012-12-31"), as.Date)
pwindow(d) <- 2
maxdepot(d) <- 4000

process(d)
```

```
$omeprazol
  id  X          B          E
1  1 40 2010-11-02 2010-12-08
2  1 80 2010-12-09 2011-10-09
3  1  0 2011-10-10 2011-10-23
4  1 80 2011-10-24 2011-11-20
5  1  0 2011-11-21 2012-01-29
6  1 40 2012-01-30 2012-04-03
7  1 80 2012-04-04 2013-02-05
```

Example: Omeprazol

Plotting output:

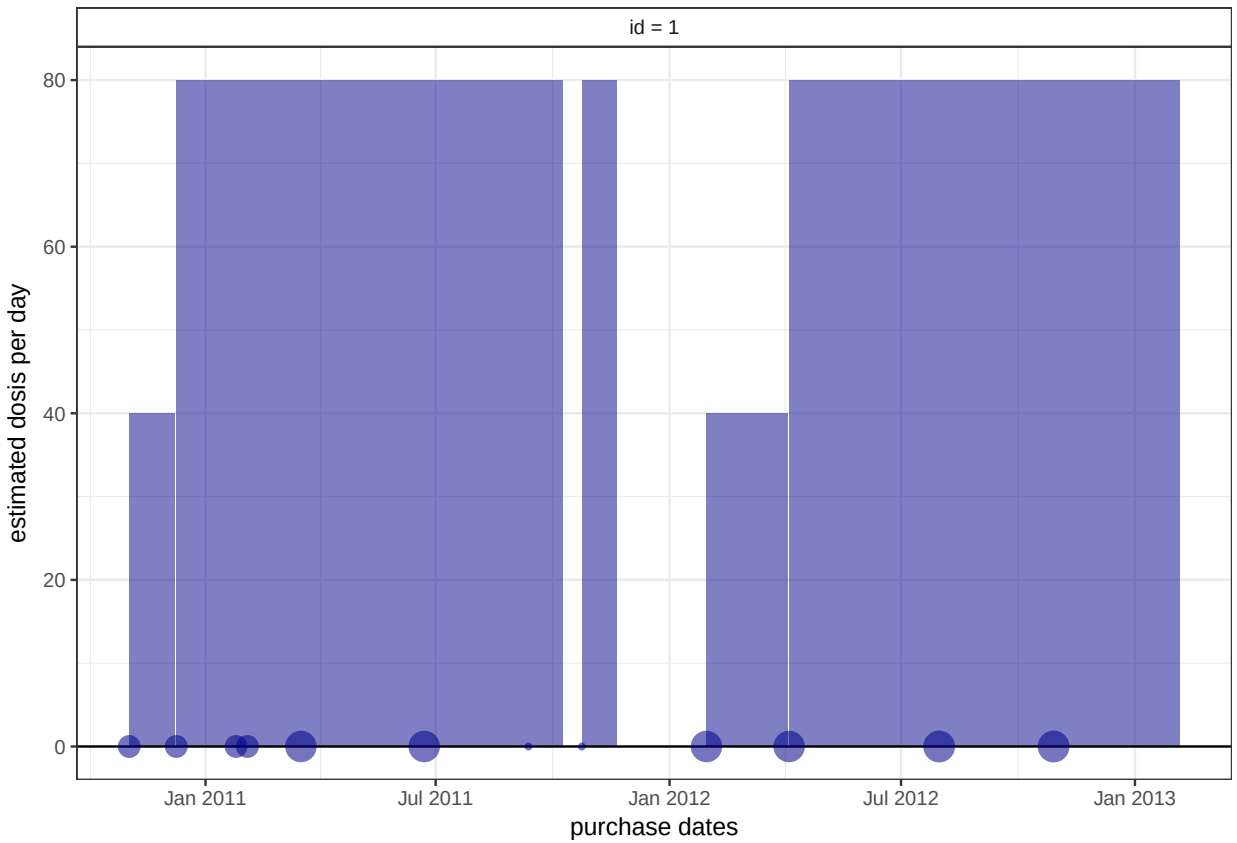
```
out <- process(d)
plot(out)
```



Example: Omeprazol

Plotting output with input:

```
out1 <- process(d, keep_data = TRUE)
plot(out1, trace = TRUE)
```



Thank you