Exercises

Marcelo F C Gomes

SMIDD 2024

Epidemic thresholds

First, let's load some packages and datasets

```
require(R0)
data("Germany.1918")
require(mem)
data("flucyl")
require(ggplot2)
require(tidyverse)
source('theme.publication.R')
```

To use the data on MEM functions, it's best to have it as a data frame object. So let's convert the German data:

```
df.Germany.1918 <- as.data.frame(Germany.1918)
head(df.Germany.1918)
```

```
##            Germany.1918
## 1918-09-29           10
## 1918-09-30            4
## 1918-10-01            4
## 1918-10-02           19
## 1918-10-03            6
## 1918-10-04           13
```

```
head(flucyl)
```

```
##    2001/2002 2002/2003  2003/2004 2004/2005 2005/2006 2006/2007 2007/2008
## 40  4.381929  0.000000   4.231192  5.477651  0.000000  0.000000  0.000000
## 41  7.203573  7.500188  30.432136  9.396730  3.594536  3.690990 12.019712
## 42 10.962108  7.828401  47.626026  8.807857  6.643194 14.146773 29.476787
## 43 14.291329 11.134204 107.816712  0.000000  0.000000  3.804306  9.408223
## 44 14.449301  0.000000 177.264841  4.263665  3.668379  0.000000  4.238905
## 45 28.526601  6.774150 240.764427 12.937726  4.100041  3.658046  7.599939
##    2008/2009
## 40 39.996000
## 41 12.558607
## 42  3.807783
## 43 16.697975
## 44  4.043672
## 45 11.343442
```

Observe that flucyl has several seasons, separated by column. As in our previous class, let's start with just one of them.

```r
flucyl.2001 <- flucyl$`2001/2002`
head(flucyl.2001)
```

```
## [1]  4.381929  7.203573 10.962108 14.291329 14.449301 28.526601
```

```r
flucyl.2001.timming <- memtiming(flucyl.2001)
print(flucyl.2001.timming)
```

```
## Call:
## memtiming(i.data = flucyl.2001)
##
## Optimum:
## [1] 10
##
## Timing:
## [1] 14 23
```

```r
rownames(flucyl)[14]
```

```
## [1] "1"
```

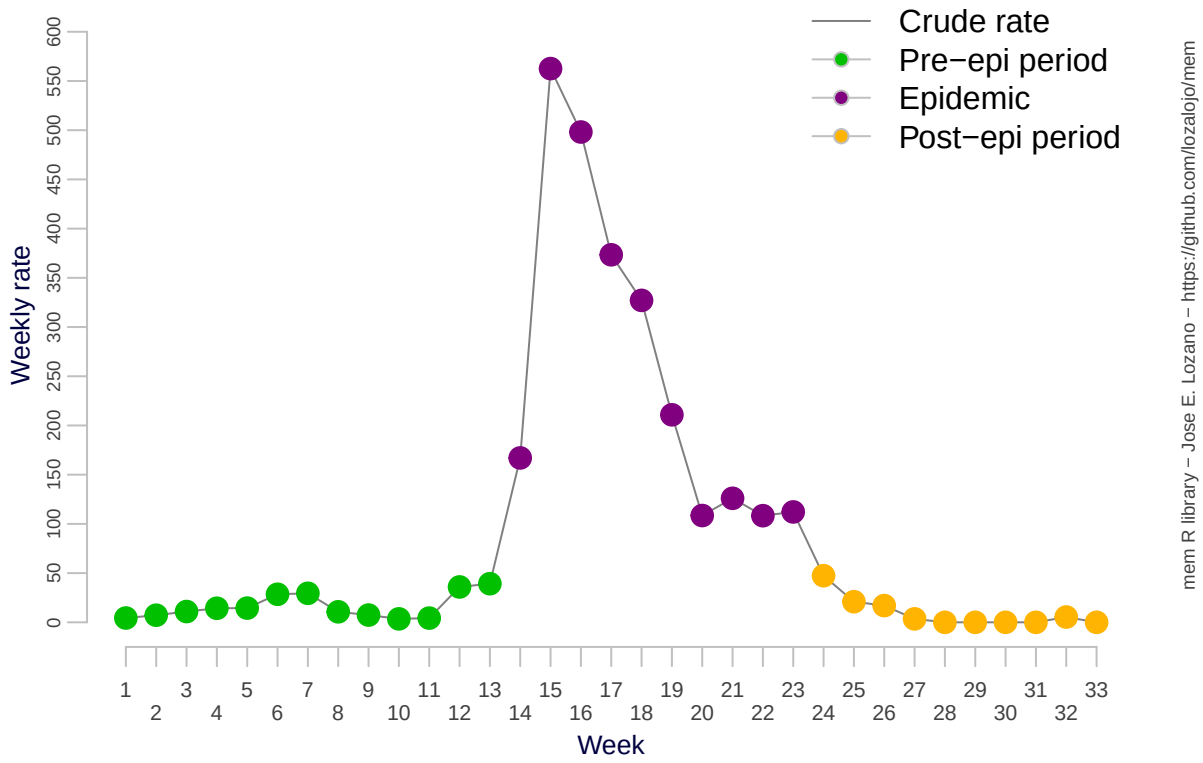
```r
rownames(flucyl)[23]
```

```
## [1] "10"
```

```r
summary(flucyl.2001.timming)
```

```
## Call:
## memtiming(i.data = flucyl.2001)
##
## Optimum:
## [1]   10.00000   89.49422 2594.10739
##
## Timing:
## [1] 14 23
##
## Pre-epidemic values:
## [1] 39.24867 35.93245 29.49091 28.52660 14.44930
##
## Post-epidemic values:
## [1] 47.364011 20.966998 16.993075  5.369704  3.500053
```

```r
plot(flucyl.2001.timming)
```



Germany/1918:

```
Germany.1918.timming <- memtiming(df.Germany.1918)
print(Germany.1918.timming)
```

```
## Call:
## memtiming(i.data = df.Germany.1918)
##
## Optimum:
## [1] 9
##
## Timing:
## [1] 27 35
```

```
rownames(df.Germany.1918)[27]
```

```
## [1] "1918-10-25"
```

```
rownames(df.Germany.1918)[35]
```

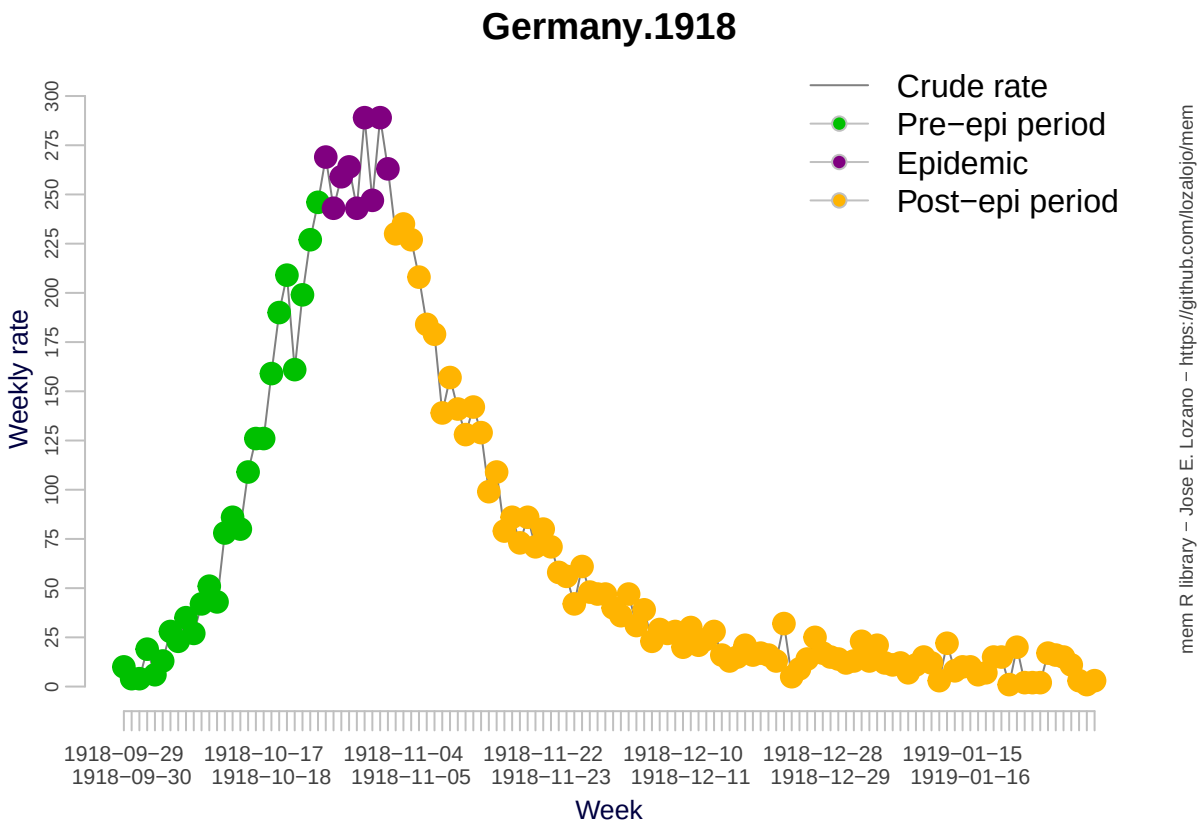
```
## [1] "1918-11-02"
```

```
summary(Germany.1918.timming)
```

```
## Call:
## memtiming(i.data = df.Germany.1918)
##
## Optimum:
## [1]    9.00000   26.55145 2366.00000
##
## Timing:
## [1] 27 35
##
```

```
## Pre-epidemic values:
## [1] 246 227 209 199 190
##
## Post-epidemic values:
## [1] 235 230 227 208 184
```

```
plot(Germany.1918.timming)
```



By default, memtimming uses "method 2", which is a threshold of 2.8 for the MAP curve derivative.

Let's play with those options

Flu Castillo y León, method 1:

```
flucyl.2001.timming <- memtiming(flucyl.2001, i.method=1)
print(flucyl.2001.timming)
```

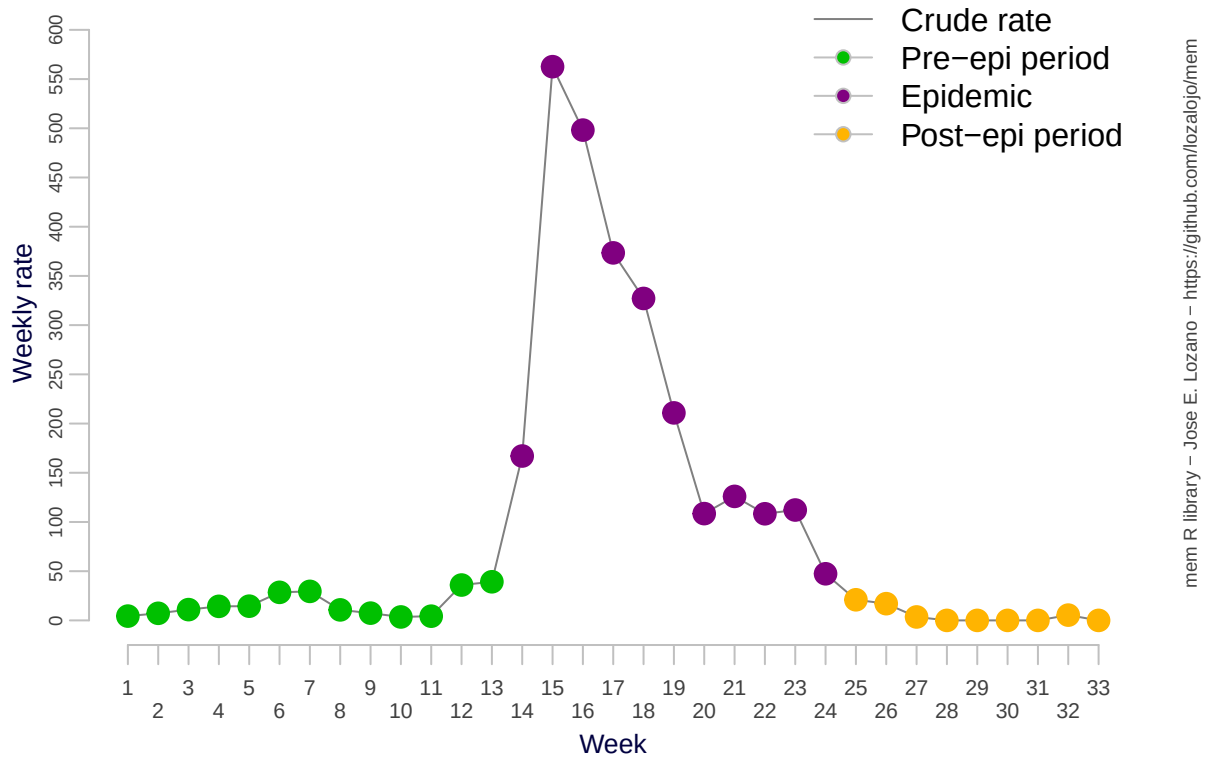
```
## Call:
## memtiming(i.data = flucyl.2001, i.method = 1)
##
## Optimum:
## [1] 11
##
## Timing:
## [1] 14 24
```

```
summary(flucyl.2001.timming)
```

```
## Call:
## memtiming(i.data = flucyl.2001, i.method = 1)
```

```
##
## Optimum:
## [1]   11.00000   91.12823 2641.47140
##
## Timing:
## [1] 14 24
##
## Pre-epidemic values:
## [1] 39.24867 35.93245 29.49091 28.52660 14.44930
##
## Post-epidemic values:
## [1] 20.966998 16.993075  5.369704  3.500053  0.000000
```

```
plot(flucyl.2001.timming)
```



Flu Castillo y León, method 2, $\delta = 2$:

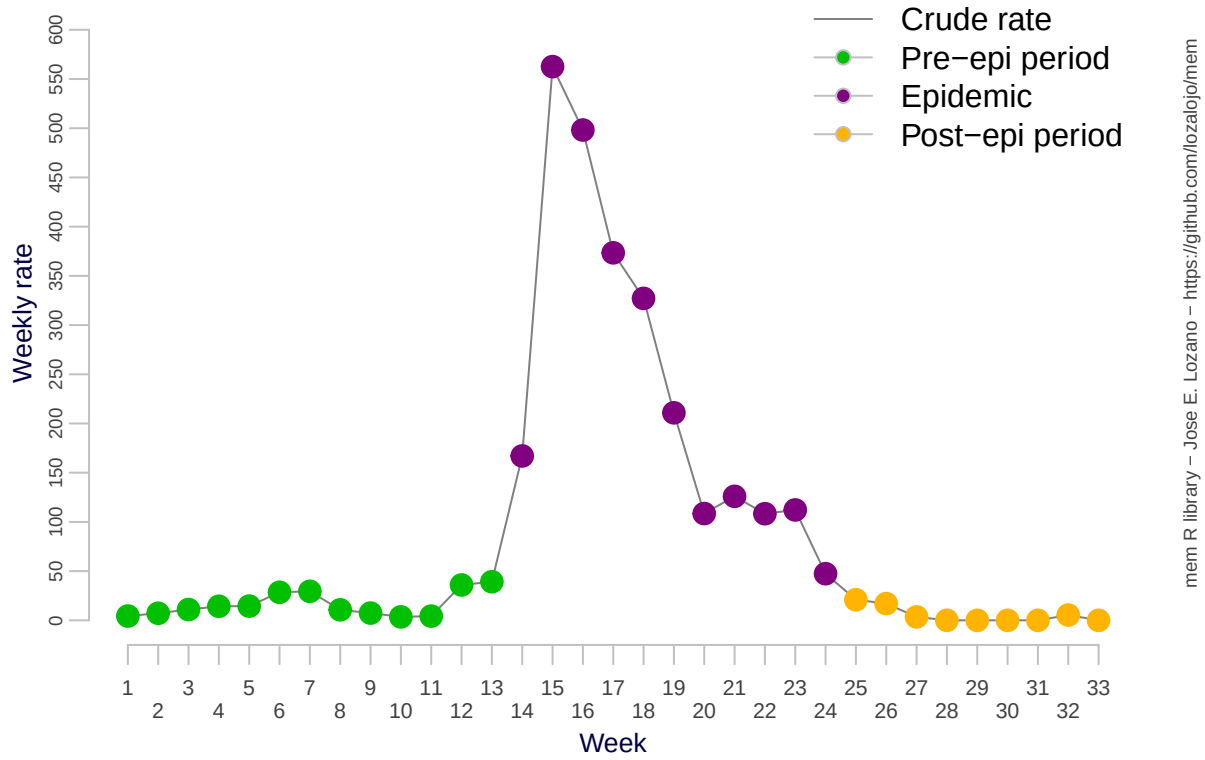
```
flucyl.2001.timming <- memtiming(flucyl.2001, i.method=2, i.param = 2)
print(flucyl.2001.timming)
```

```
## Call:
## memtiming(i.data = flucyl.2001, i.method = 2, i.param = 2)
##
## Optimum:
## [1] 11
##
## Timing:
## [1] 14 24
```

```r
summary(flucyl.2001.timming)
```

```
## Call:
## memtiming(i.data = flucyl.2001, i.method = 2, i.param = 2)
##
## Optimum:
## [1]   11.00000   91.12823 2641.47140
##
## Timing:
## [1] 14 24
##
## Pre-epidemic values:
## [1] 39.24867 35.93245 29.49091 28.52660 14.44930
##
## Post-epidemic values:
## [1] 20.966998 16.993075  5.369704  3.500053  0.000000
```

```r
plot(flucyl.2001.timming)
```



Flu Castillo y León, method 2, $\delta = 4$:

```r
flucyl.2001.timming <- memtiming(flucyl.2001, i.method=2, i.param = 4)
print(flucyl.2001.timming)
```

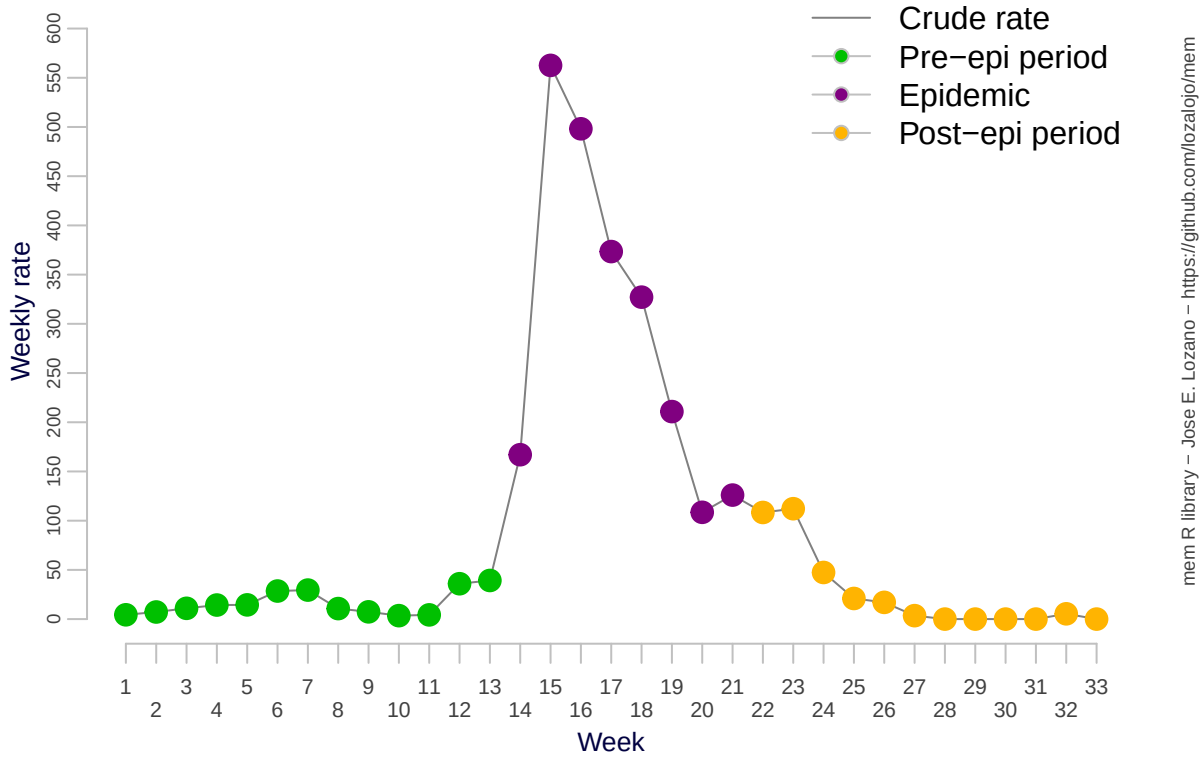
```
## Call:
## memtiming(i.data = flucyl.2001, i.method = 2, i.param = 4)
##
## Optimum:
## [1] 8
##
```

```
## Timing:
## [1] 14 21
```

```
summary(flucyl.2001.timming)
```

```
## Call:
## memtiming(i.data = flucyl.2001, i.method = 2, i.param = 4)
##
## Optimum:
## [1]    8.00000   81.88731 2373.61116
##
## Timing:
## [1] 14 21
##
## Pre-epidemic values:
## [1] 39.24867 35.93245 29.49091 28.52660 14.44930
##
## Post-epidemic values:
## [1] 112.15792 108.33831  47.36401  20.96700  16.99308
```

```
plot(flucyl.2001.timming)
```



Germany, method 1:

```
df.Germany.1918.timming <- memtiming(df.Germany.1918, i.method=1)
print(df.Germany.1918.timming)
```

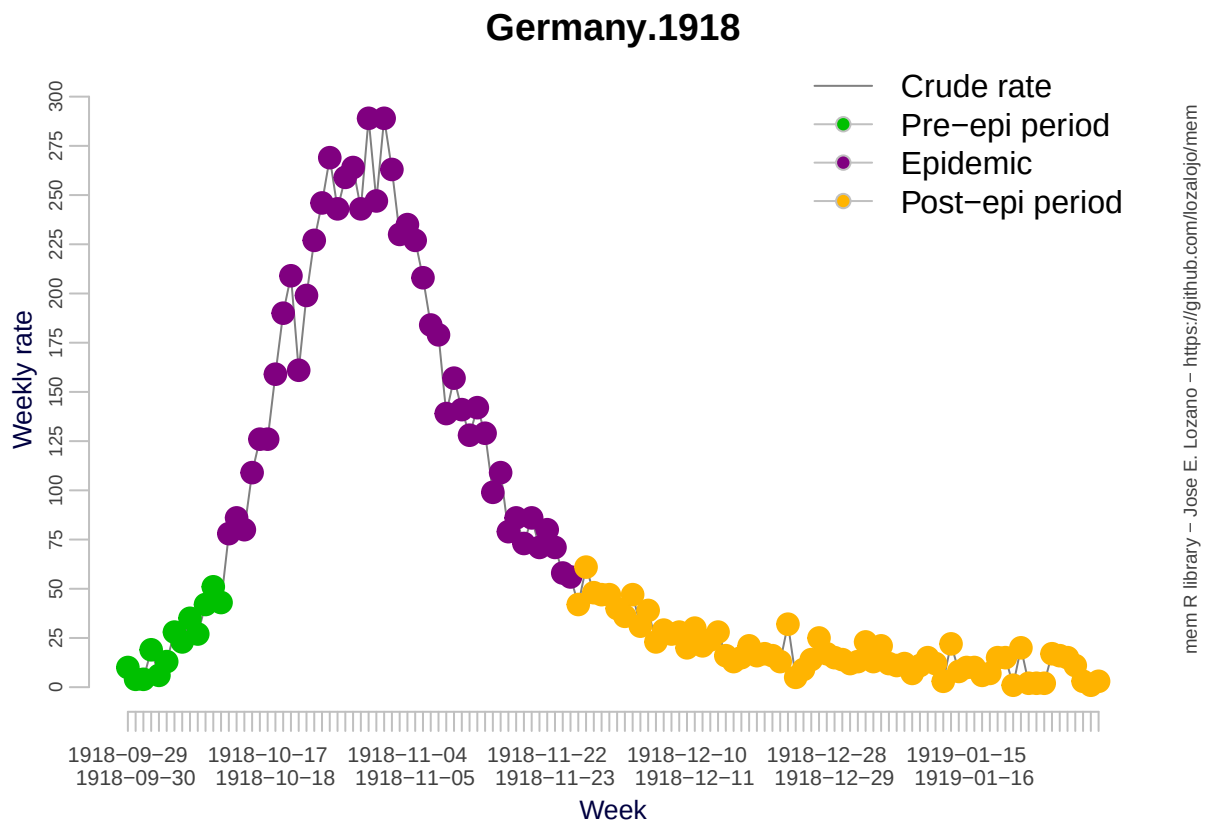
```
## Call:
## memtiming(i.data = df.Germany.1918, i.method = 1)
##
## Optimum:
```

```
## [1] 45
##
## Timing:
## [1] 14 58
```

```
summary(df.Germany.1918.timming)
```

```
## Call:
## memtiming(i.data = df.Germany.1918, i.method = 1)
##
## Optimum:
## [1]   45.00000   82.24666 7329.00000
##
## Timing:
## [1] 14 58
##
## Pre-epidemic values:
## [1] 51 43 42 35 28
##
## Post-epidemic values:
## [1] 61 48 47 47 47
```

```
plot(df.Germany.1918.timming)
```



Germany, method 2, $\delta = 2$:

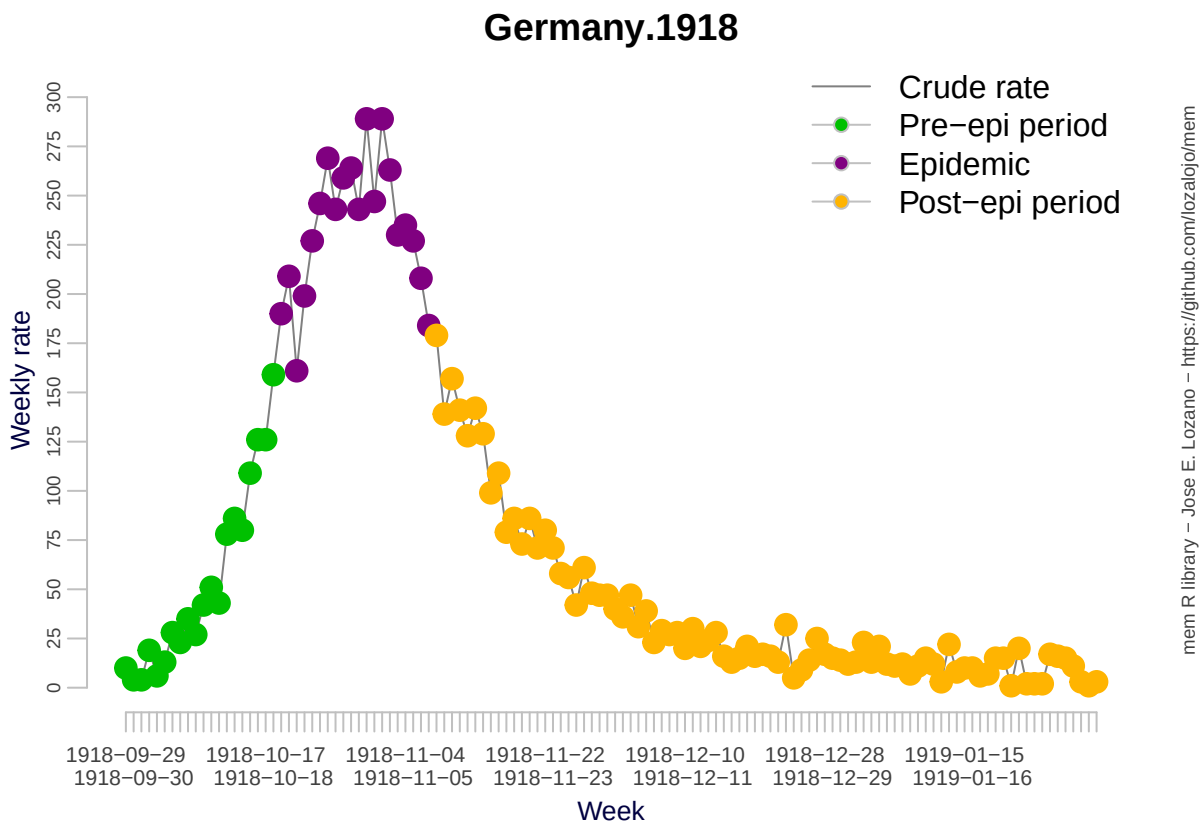
```
df.Germany.1918.timming <- memtiming(df.Germany.1918, i.method=2, i.param = 2)
print(df.Germany.1918.timming)
```

```
## Call:
## memtiming(i.data = df.Germany.1918, i.method = 2, i.param = 2)
##
## Optimum:
## [1] 20
##
## Timing:
## [1] 21 40
```

```
summary(df.Germany.1918.timming)
```

```
## Call:
## memtiming(i.data = df.Germany.1918, i.method = 2, i.param = 2)
##
## Optimum:
## [1]   20.0000   52.5418 4682.0000
##
## Timing:
## [1] 21 40
##
## Pre-epidemic values:
## [1] 159 126 126 109  86
##
## Post-epidemic values:
## [1] 179 157 142 141 139
```

```
plot(df.Germany.1918.timming)
```



Germany, method 2, $\delta = 4$:

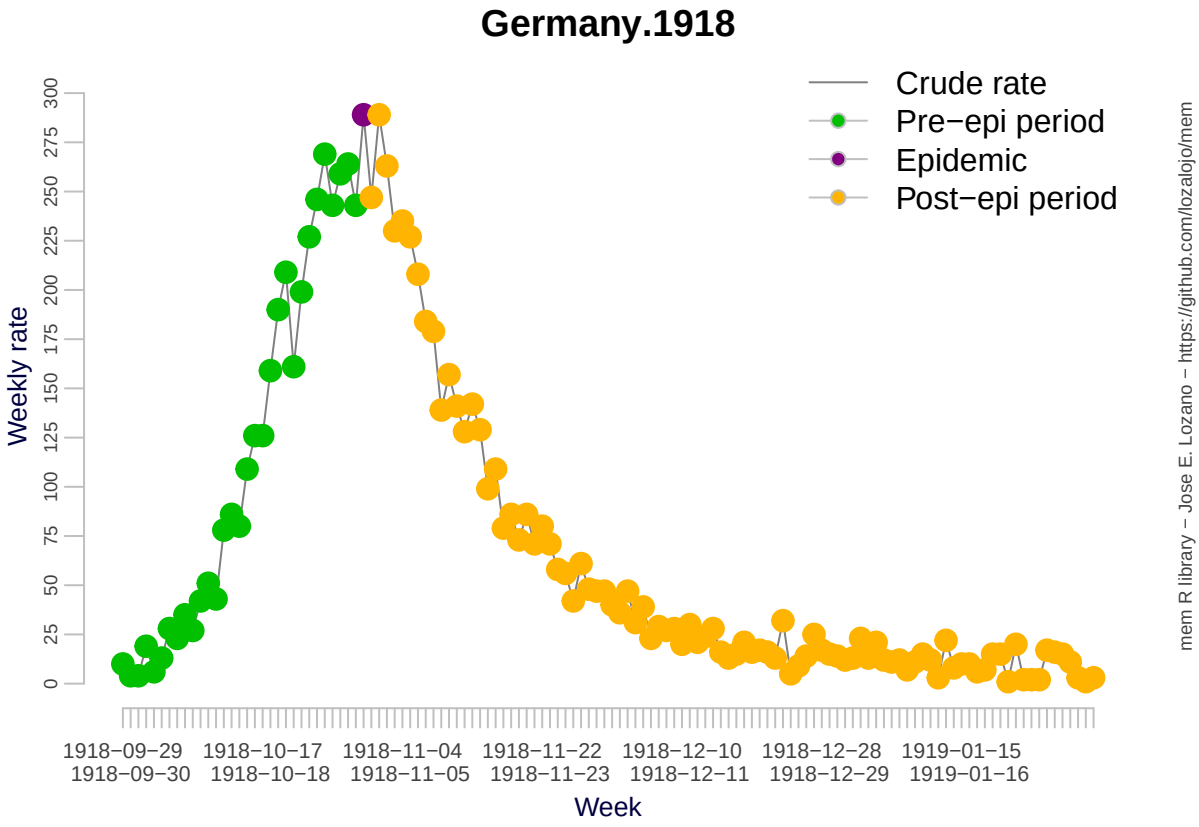
```
df.Germany.1918.timming <- memtiming(df.Germany.1918, i.method=2, i.param = 4)
print(df.Germany.1918.timming)
```

```
## Call:
## memtiming(i.data = df.Germany.1918, i.method = 2, i.param = 4)
##
## Optimum:
## [1] 1
##
## Timing:
## [1] 32 32
```

```
summary(df.Germany.1918.timming)
```

```
## Call:
## memtiming(i.data = df.Germany.1918, i.method = 2, i.param = 4)
##
## Optimum:
## [1]   1.000000   3.243183 289.000000
##
## Timing:
## [1] 32 32
##
## Pre-epidemic values:
## [1] 269 264 259 246 243
##
## Post-epidemic values:
## [1] 289 263 247 235 230
```

```
plot(df.Germany.1918.timming)
```



Pre and post epidemic threshods

For Germany we can't use the memmodel function, since it needs at least 2 seasons.

```
flucyl
```

```
##     2001/2002  2002/2003  2003/2004  2004/2005  2005/2006  2006/2007  2007/2008
## 40   4.381929   0.000000   4.231192   5.477651   0.000000   0.000000   0.000000
## 41   7.203573   7.500188  30.432136   9.396730   3.594536   3.690990  12.019712
## 42  10.962108   7.828401  47.626026   8.807857   6.643194  14.146773  29.476787
## 43  14.291329  11.134204 107.816712   0.000000   0.000000   3.804306   9.408223
## 44  14.449301   0.000000 177.264841   4.263665   3.668379   0.000000   4.238905
## 45  28.526601   6.774150 240.764427  12.937726   4.100041   3.658046   7.599939
## 46  29.490913  11.356753 246.324378   4.199387   9.917683  10.640939  19.830253
## 47  10.676157  14.795088 260.279140  12.124151   3.276111   7.006481  28.148200
## 48   7.370283  25.347625 210.645710  34.480777   3.711952   7.693491  44.971382
## 49   3.500053  59.345109 113.925160  17.751479   0.000000   9.725734  31.251221
## 50   4.296640  43.713241  82.182774  62.129612  18.794166  12.966245  35.227806
## 51  35.932447  71.111855  61.006609 153.150017   4.090816  18.672393  74.994643
## 52  39.248668  68.511921  41.377884 283.655755  22.650057  31.595577 103.391232
## 1  167.025625  32.639739  42.973786 567.260940  24.233612   9.925558 257.408338
## 2  562.593145  39.454806  42.042501 559.750554  19.643793  90.970217 257.490637
## 3  498.154320 110.192837  21.262119 468.123934  18.349971 166.805671 225.660864
## 4  373.407527 168.846099  14.919806 266.983091  27.833002 248.377534 182.322054
## 5  327.092263 185.294990   4.261485 107.215185  28.117092 414.167381 146.576394
## 6  210.834763 220.526463   3.667033  77.389398  93.723640 395.033860 123.175017
## 7  108.501628 172.740821  11.400775  81.935400 145.822029 314.996063  99.619312
## 8  126.001890 158.577961  14.965579  67.903068 141.491636 117.861760  88.784715
```

```
## 9  108.338314  86.963097  19.073777  47.805302 157.601771  89.085272  63.449540
## 10 112.157918  59.569697  11.952191  57.054365 193.675493  41.760919  30.373595
## 11  47.364011  47.365736   4.150066  35.396664 208.938657  24.360536  17.707734
## 12  20.966998  41.485169   0.000000   8.826125 214.379420  34.127363  18.614348
## 13  16.993075  80.852628   0.000000   4.303111 145.337719  17.664724   3.560873
## 14   3.500053  31.539108   0.000000   8.105370 117.818283   0.000000  26.055237
## 15   0.000000  11.242692   0.000000   4.309602  41.088855  11.463508   7.217090
## 16   0.000000   0.000000   0.000000   8.550297  48.096489   7.754042   3.865033
## 17   0.000000  17.904462   0.000000   0.000000   3.690582   3.871018   0.000000
## 18   0.000000   8.000000   0.000000   0.000000   3.444238   0.000000   8.254230
## 19   5.369704   8.351428   0.000000   0.000000   6.902740   0.000000   0.000000
## 20   0.000000   0.000000   0.000000   0.000000   3.572322   0.000000   0.000000
##     2008/2009
## 40  39.996000
## 41  12.558607
## 42   3.807783
## 43  16.697975
## 44   4.043672
## 45  11.343442
## 46  31.229262
## 47  36.516341
## 48  17.719814
## 49  61.479347
## 50  77.432677
## 51 277.897561
## 52 250.777876
## 1  205.603902
## 2  284.820141
## 3  183.801005
## 4  109.911678
## 5   80.599944
## 6   57.228700
## 7   61.851919
## 8   38.898397
## 9   29.439906
## 10  19.074505
## 11  24.102193
## 12   3.673095
## 13   8.179959
## 14   8.694896
## 15   4.071164
## 16   0.000000
## 17   4.632847
## 18   0.000000
## 19   0.000000
## 20   0.000000
```

```
flucyl.memmodel <- memmodel(flucyl)
print(flucyl.memmodel)
```

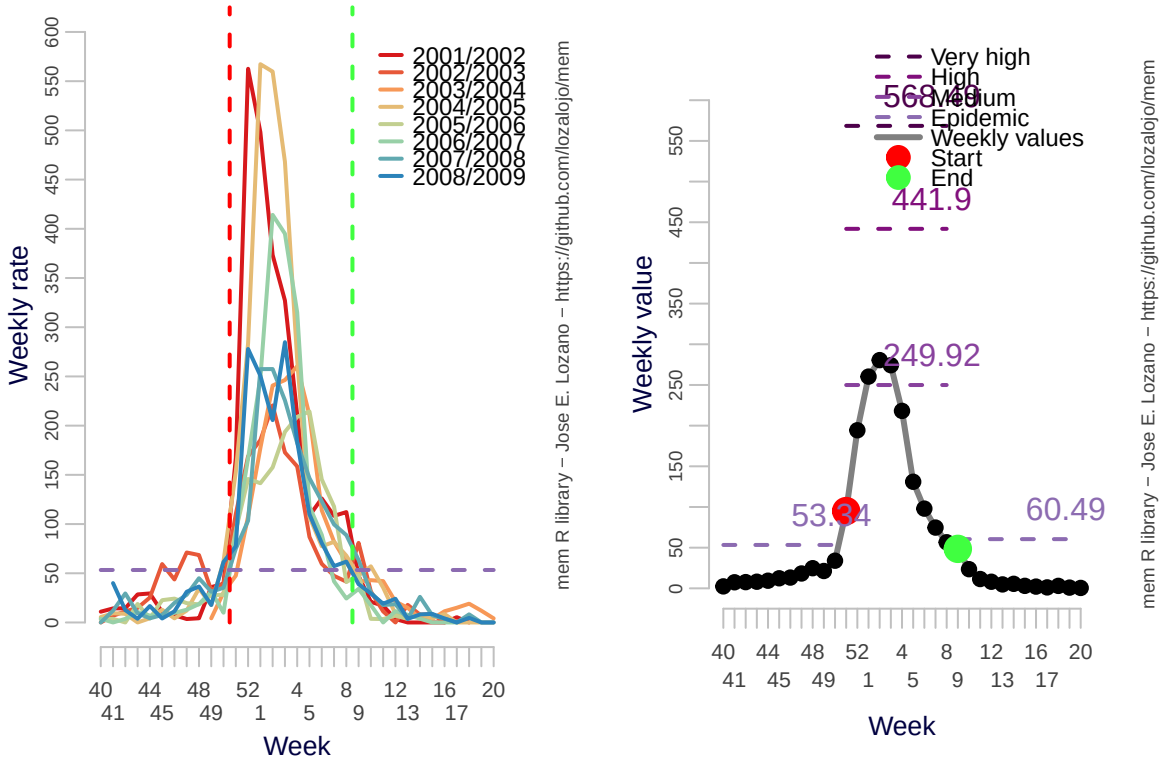
```
## Call:
## memmodel(i.data = flucyl)
##
## Epidemic threshold:
##             Pre  Post
```

```
## Threshold 53.34 60.49
##
## Intensity thresholds:
##                   Threshold
## Medium (40%)         249.92
## High (90%)           441.90
## Very high (97.5%)    568.49
```

```
summary(flucyl.memmodel)
```

```
## Call:
## memmodel(i.data = flucyl)
##
## Parameters:
##  - General:
##      + Number of seasons restriction:  Restricted to 10
##      + Number of seasons used:  8
##      + Seasons used:  2001/2002,2002/2003,2003/2004,2004/2005,2005/2006,2006/2007,2007/2008,2008/200
##      + Number of weeks:  33
##  - Confidence intervals:
##      + Epidemic threshold:  Arithmetic mean and its one sided 95% CI using 2*SD
##      + Intensity:  Geometric mean and its one sided 40,90,97.5% CI using (log) 2*SD
##      + Curve:  Geometric mean and its two sided 95% CI using the (log) normal approximation
##      + Others:  Median and its two sided 95% CI using the KC Method
##  - Epidemic timing calculation:
##      + Method:  2
##      + Parameter:  2.8
##  - Epidemic threshold calculation:
##      + Pre-epidemic values:  Optimized: 4
##      + Tails of CI:  1
##  - Intensity thresholds calculation:
##      + Number of values:  Optimized: 4
##      + Tails of CI:  1
##      + Levels of CI:  Medium: 40% High: 90% Very high: 97.5%
##  - Bootstrap (if used):
##      + Technique:  norm
##      + Bootstrap samples:  10000
##
## Epidemic description:
##  - Typical influenza season lasts  10  weeks.  95 %CI    [ 8 , 11 ]
##  - This optimal  10  weeks influenza season includes the 84.99 % of the total sum of rates
##
##
## Epidemic threshold:
##             Pre  Post
## Threshold 53.34 60.49
##
##
## Intensity thresholds:
##                   Threshold
## Medium (40%)         249.92
## High (90%)           441.90
## Very high (97.5%)    568.49
```

```
plot(flucyl.memmodel)
```



The expected "epidemic period" can be obtained from the object by means of it's entry ci.start

```
flucyl.memmodel$ci.start
```

```
##
##            3 12 15
##           42 51  2
## ic.inicio  3 12 15
```