demonstration R code for some recurrence over text and other categorical series

```
#
# demonstration R code for some recurrence over text and other categorical series
#
# demo coded by: Rick Dale, cognaction.org/rick/
#
# shared for: Nonlinear ATI Summer, 2016 in Cincinnati, OH
#
# if you are fresh to R (and RStudio) see:
# https://www.datacamp.com/courses/free-introduction-to-r
#
# if you find this code useful we encourage you to cite the crqa library publication:
#
# Coco, M. I. & Dale, R. (2014). Cross-recurrence quantification analysis of
# categorical and continuous time series: an R package. Frontiers in
# Quantitative Psychology and Measurement, 5, 510.
# http://journal.frontiersin.org/article/10.3389/fpsyg.2014.00510/full
#

library(tm)
```

```
## Loading required package: NLP
```

```
library(stringi)
library(crqa)
```

```
## Warning: package 'crqa' was built under R version 3.1.3
```

```
## Loading required package: Matrix
## Loading required package: tseriesChaos
## Loading required package: deSolve
## Loading required package: fields
## Loading required package: spam
## Loading required package: grid
## Spam version 1.0-1 (2014-09-09) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
##
## Loading required package: maps
## Loading required package: plot3D
## Loading required package: pracma
##
## Attaching package: 'pracma'
##
## The following object is masked from 'package:deSolve':
##
##     rk4
##
## The following objects are masked from 'package:Matrix':
##
##     expm, lu, tril, triu
```

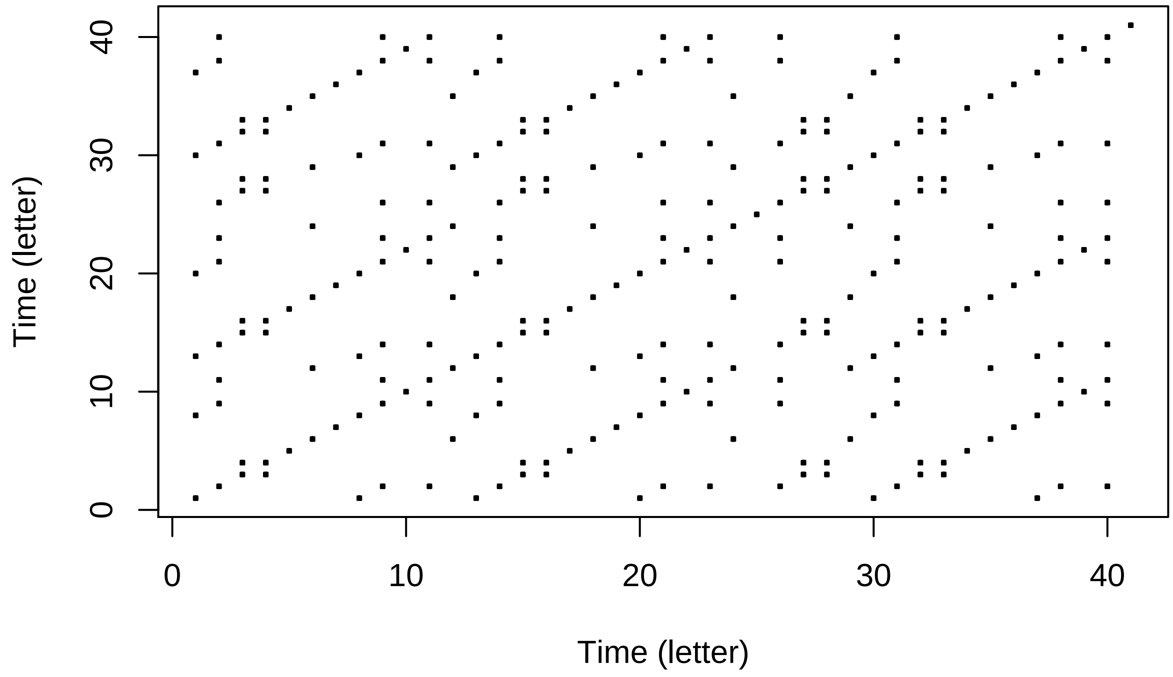
```
library(SnowballC)
source('functions.R') # functions using crqa; see that .R

setwd('~/Dropbox/crqa-quickstart') # move R to the folder with our data files

# these functions are in functions.R
words = makeWordSequence('data/testText.txt',stemWords=TRUE,trimPunctuation=TRUE,numbersToFile='data/ou
characters = makeCharacterSequence('data/testText.txt',numbersToFile='data/outputChars.txt')

# note we set radius to *near* 0
resChars = crqa(characters, characters, 1, 1, 1, 0.001, F, 2, 2, 0, F, F)

resChars$RR # you get your RQA measures!l
```

```
## [1] 15.76443
```

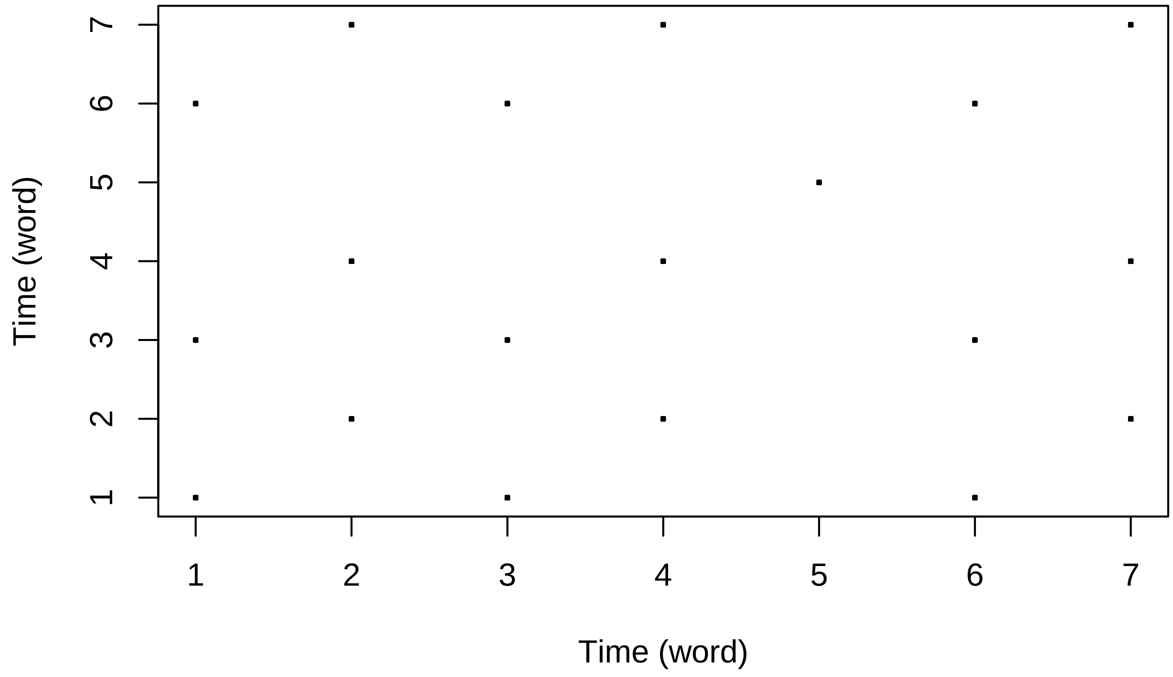
```
resChars$DET
```

```
## [1] 62.26415
```

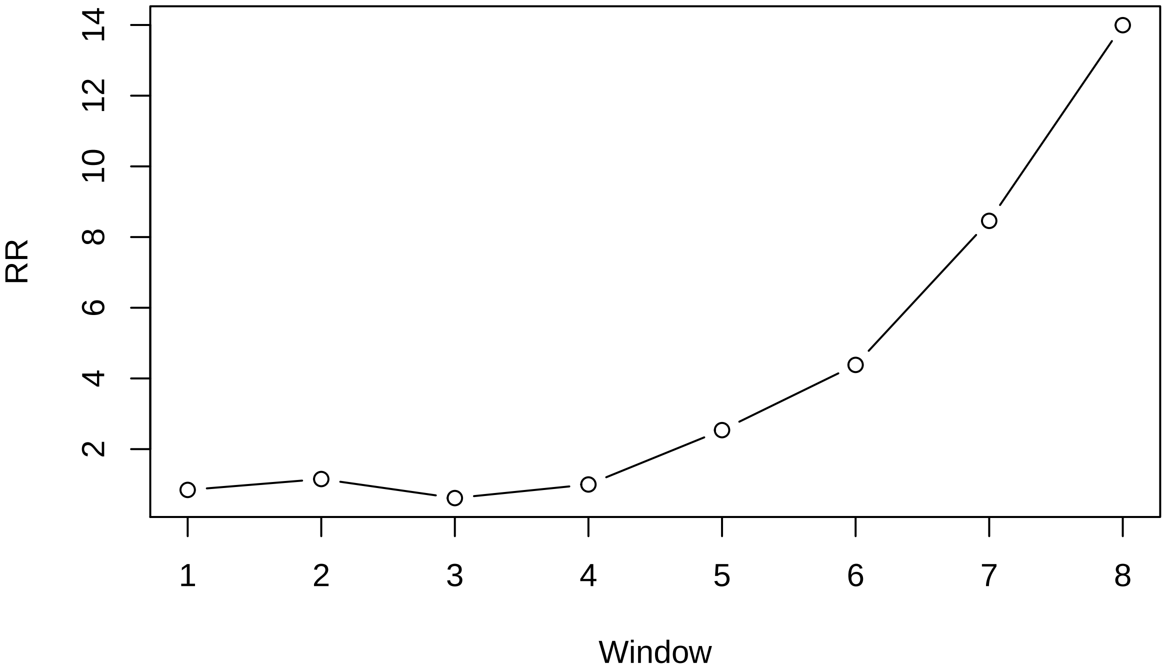
```
# remaining parameters see help(crqa) == uses same parameters as ATI software!
resWords = crqa(words, words, 1, 1, 1, 0.001, F, 2, 2, 0, F, F)

plotRP(resChars$RP,'Time (letter)','Time (letter)')
```



```
plotRP(resWords$RP,'Time (word)','Time (word)')
```



```
# windowed recurrence example, with fuller parameter specification
# using the cattail down lyrics example where the song ends on intense, extended
# mind repetition with the deer telling you you don't know who you are
words = makeWordSequence('data/cattaildown_end.txt',stemWords=TRUE,trimPunctuation=TRUE,numbersToFile='
res = wincrqa(words, words, windowstep=10,windowsize=50,
              delay=1, embed=1, rescale=1, radius=.001,
              normalize=0, mindiagline=2, minvertline=2, tw=1, whiteline=F, trend=F)
plot(res$crqwin[,1],type='b',xlab='Window',ylab='RR') # columns in crqwin represent our measures as lis
```


RR
2
4
6
8
10
12
14
1
2
3
4
5
6
7
8
Window