Nonlinear Methods Workshop Lecture Notes

Dan Mønster

2024-04-16

Table of contents

Preface

This is a demonstration and test of online lecture notes for an R-based course or workshop. In this case the Nonlinear Methods Workshop held at Leuphana University in Lüneburg.

The lecture notes are written in Quarto markdown in Rstudio using the Quarto book document format.

For a “normal” page, see Chapter 1. For a more advanced example where the user can interact with a live instance of R, see Chapter 9.

Note that references are handled nicely, and there is even a PDF version of the lecture notes that can be downloaded.

The book is fully searchable, and there is an option for users to annotate both privately and as part of a group (but this is not enabled in the current demonstration).

To learn more about Quarto books visit https://quarto.org/docs/books.

1 Introduction to recurrence plots

This chapter introduces what a recurrence plot is and explains how it can be analyzed using the most common measures that quantify features of the recurrence plot. We will use some relatively simple examples, so that the basic concepts are not obscured by complications arising from more realistic data. We will deal with those issues in due time, but see Marwan et al. (2007) for a comprehensive introduction to recurrence plots with all the mathematical details.

Now, let us explore what a recurrence plot is.

1.1 A simple example

We start with the children's rhyme by Helen H. Moore, shown below, to explore what recurrence in a time series is — exemplified here by the text, where each letter is a datum and time is represented by the position of the letter in the text.

```
Pop, pop, popcorn
Popping in the pot!
Pop, pop, popcorn
Eat it while it's hot!
Pop, pop, popcorn
Butter on the top!
When I eat popcorn
I can't stop
```

The letters of the rhyme are encoded in the variable `popcorn` and we can the use the `crqa()` function from the package of the same name (Coco et al. 2021) to create the recurrence plot.

The first 30 elements of the `popcorn` vector are: P, O, P, ␣, P, O, P, ␣, P, O, P, C, O, R, N, ␣, P, O, P, P, I, N, G, ␣, I, N, ␣, T, H, E. Here the character "␣" represents a space between words. Line breaks and punctuation marks have been removed from the poem, and all letters are upper case, since we do not want to distinguish between lower case and upper case letters.

The R code shown below will compute and display the recurrence plot.

```r
library(crqa)
# We could create a package, NLM, instead
source("R/utils.R")

rp <- crqa(popcorn, popcorn,
           delay = 0,
           embed = 0,
           radius = 0.1,
           method = "rqa",
           datatype = "categorical")

plot_rp(rp$RP)
```

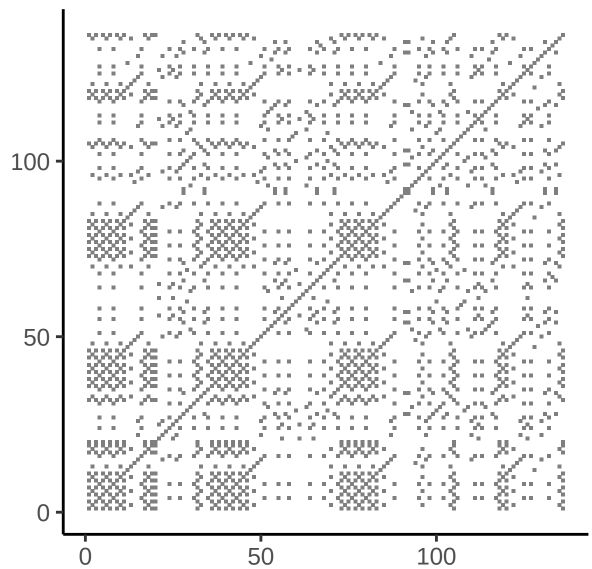


Figure 1.1: Recurrence plot of poem

We can also look at the first two verses of the poem, and put the letters on the axes, to make it a little easier to see how the underlying time series results in a recurrence plot.

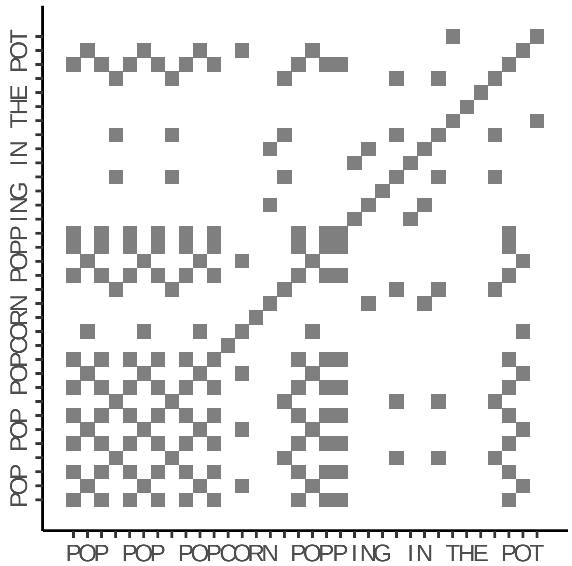


Figure 1.2: Recurrence plot of first two verses of the poem

1.2 Quantifying the recurrence plot

With some training you can learn how to spot various properties of a time series simply by looking at the corresponding recurrence plot. In the case of the poem, we can see, for instance, that certain motifs are repeated in the poem. This is evident from the blue diagonal lines in the figure below, which correspond to the recurring motif "POP POP POPCORN".

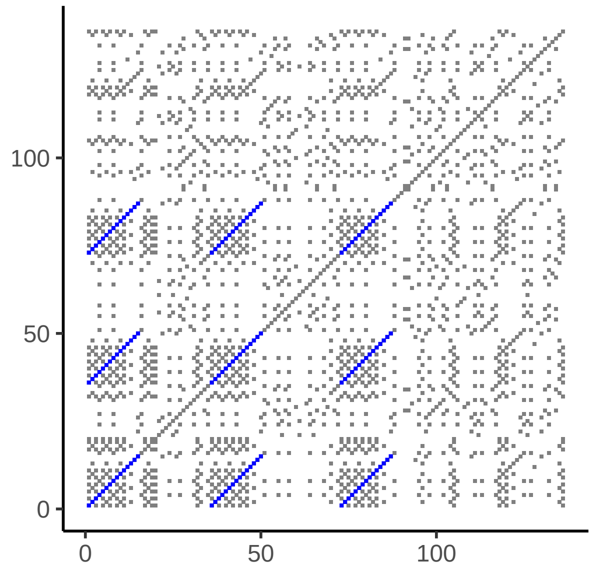


Figure 1.3: Recurring motifs in the poem

While we can gain some insights about the underlying time series from a visual inspection of the recurrence plot, we do not wish to rely on that. First, simply because we could overlook important features, and furthermore, it it not feasible for long time series or when comparing many time series. And, perhaps most importantly, we want *objective* measures, that do not depend on the skill of the person assessing the recurrence plot. With that said, it is still very useful to be able to visually inspect recurrence plots and infer something about the time series, so we definitely encourage you to work towards developing that skill as well.

But having objective quantitative measures that characterize features of a given recurrence plot is what has made recurrence plot analysis an effective tool, and that is why we will now look at *recurrence quantification measures*.

Some of the most common recurrence quantification measures are reproduced in table Table 1.1, reproduced from Coco et al. (2021).

Table 1.1: Common recurrence quantification measures

Measure	**Abbreviation**	**Definition**
Recurrence Rate	RR	$\frac{1}{N^2}\sum_{i,j=1}^{N} R_{ij}$
Determinism	DET	$\sum_{l=l_{\min}}^{N} lP(l) \bigg/ \sum_{l=1}^{N} lP(l)$

Measure	**Abbreviation**	**Definition**
Average Diagonal Line Length	L	$\sum_{l=l_{\min}}^{N} lP(l) \Big/ \sum_{l=l_{\min}}^{N} P(l)$
Maximum Diagonal Line Length	maxL	$\max(\{l_i\}_{i=1}^{N_l}), \quad N_l = \sum_{l \geq l_{\min}} P(l)$
Diagonal Line Entropy	ENTR	$-\sum_{l=l_{\min}}^{N} p(l) \log p(l)$
Laminarity	LAM	$\sum_{v=v_{\min}}^{N} vP(v) \Big/ \sum_{v=1}^{N} vP(v)$
Trapping Time	TT	$\sum_{v=v_{\min}}^{N} vP(v) \Big/ \sum_{v=v_{\min}}^{N} P(v)$
Categorical Area-based Entropy	catH	$-\sum_{a>1}^{N_a} p(a) \log p(a)$

2 Recurrence Quantification Analysis

Some text …

An inline equation can use LaTeX mathematics, e.g., $f(t) = t^2 e^{-2t}$.

It is also possible to have display equations that take up a whole line:

$$z_n = z_{n+1}^2 + c.$$

Consider the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.

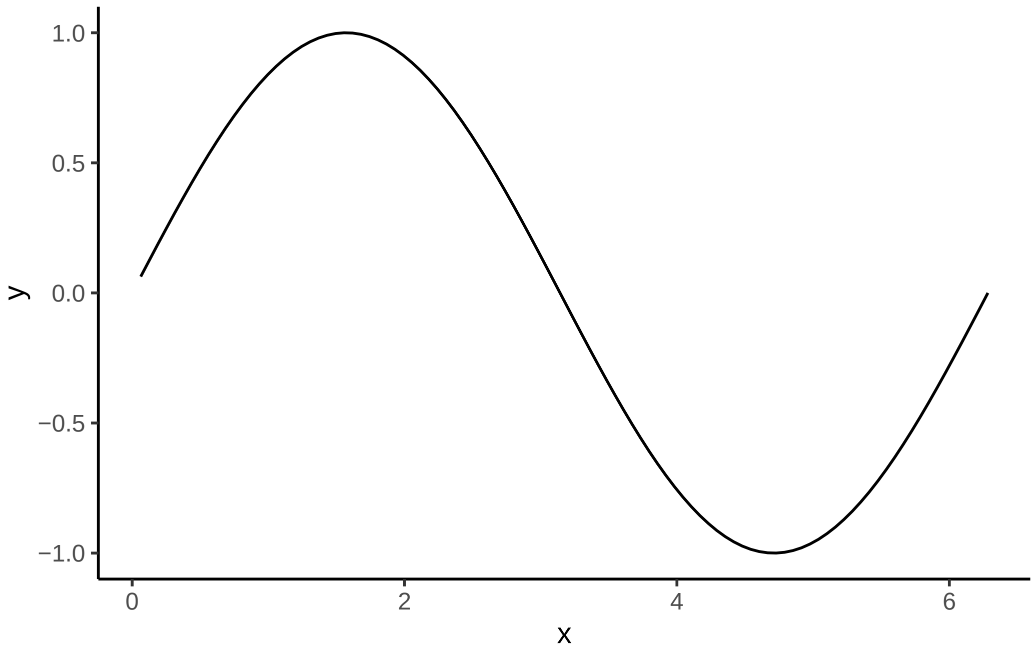
You can easily sum these (stored in the variable `numbers`) using R:

```r
summed_numbers <- sum(numbers)
```

If you are curious, the sum is 5050, something the young Carl Friedrich Gauss, allegedly, solved in class as a young schoolboy.

Plotting the sine of these numbers mapped to the interval $[0, 2\pi]$ is undoubtedly a much more daunting task to perform manually, but — again — we can ask R to do it for us:

```r
library(ggplot2)
ggplot(data.frame(x = numbers * 2 * pi / 100),
       aes(x = x)) +
  geom_function(fun = sin) +
  theme_classic()
```


1.0
0.5
0.0
−0.5
−1.0
y
0
2
4
6
x

3 Cross Recurrence Quantification Analysis

Some text …

An inline equation can use LaTeX mathematics, e.g., $f(t) = t^2 e^{-2t}$.

It is also possible to have display equations that take up a whole line:

$$z_n = z_{n+1}^2 + c.$$

Consider the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.

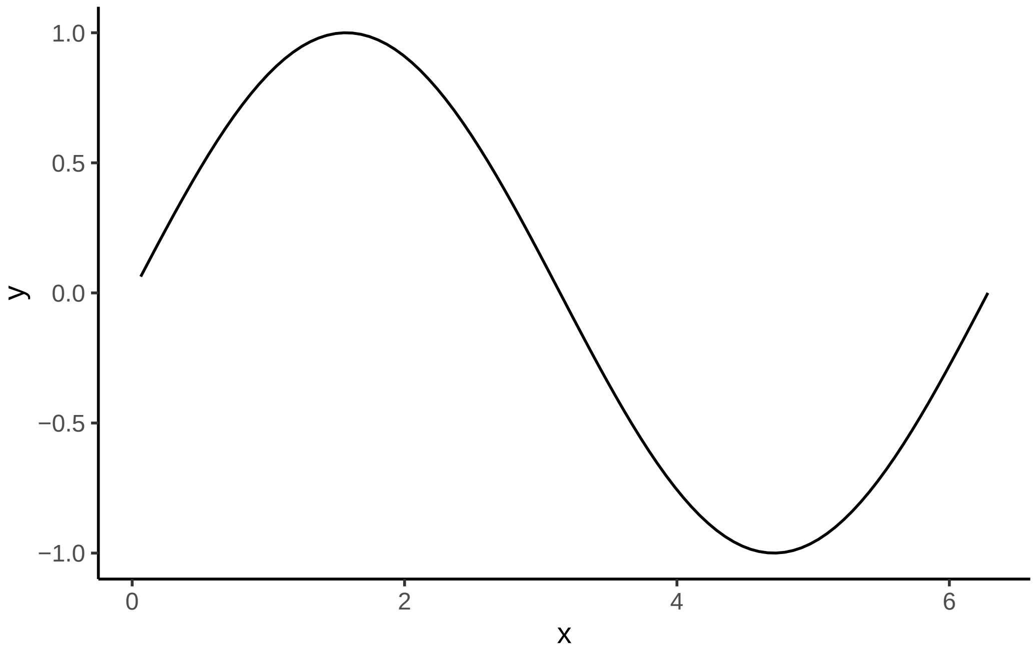
You can easily sum these (stored in the variable `numbers`) using R:

```r
summed_numbers <- sum(numbers)
```

If you are curious, the sum is 5050, something the young Carl Friedrich Gauss, allegedly, solved in class as a young schoolboy.

Plotting the sine of these numbers mapped to the interval $[0, 2\pi]$ is undoubtedly a much more daunting task to perform manually, but — again — we can ask R to do it for us:

```r
library(ggplot2)
ggplot(data.frame(x = numbers * 2 * pi / 100),
       aes(x = x)) +
  geom_function(fun = sin) +
  theme_classic()
```


1.0
0.5
0.0
−0.5
−1.0
y
0
2
4
6
x

4 Multivariate Recurrence Quantification Analysis

Some text …

An inline equation can use LaTeX mathematics, e.g., $f(t) = t^2 e^{-2t}$.

It is also possible to have display equations that take up a whole line:

$$z_n = z_{n+1}^2 + c.$$

Consider the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.

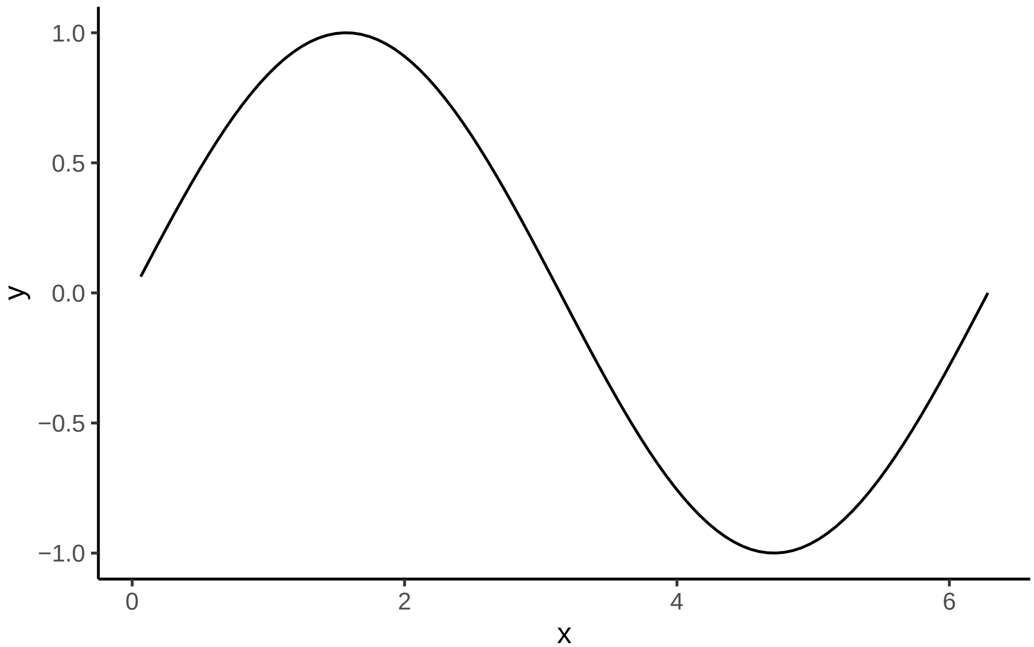
You can easily sum these (stored in the variable `numbers`) using R:

```
summed_numbers <- sum(numbers)
```

If you are curious, the sum is 5050, something the young Carl Friedrich Gauss, allegedly, solved in class as a young schoolboy.

Plotting the sine of these numbers mapped to the interval $[0, 2\pi]$ is undoubtedly a much more daunting task to perform manually, but — again — we can ask R to do it for us:

```
library(ggplot2)
ggplot(data.frame(x = numbers * 2 * pi / 100),
       aes(x = x)) +
  geom_function(fun = sin) +
  theme_classic()
```


1.0
0.5
0.0
−0.5
−1.0
y
0
2
4
6
x

5 Parameter Estimation for RQA

Some text …

An inline equation can use LaTeX mathematics, e.g., $f(t) = t^2 e^{-2t}$.

It is also possible to have display equations that take up a whole line:

$$z_n = z_{n+1}^2 + c.$$

Consider the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.

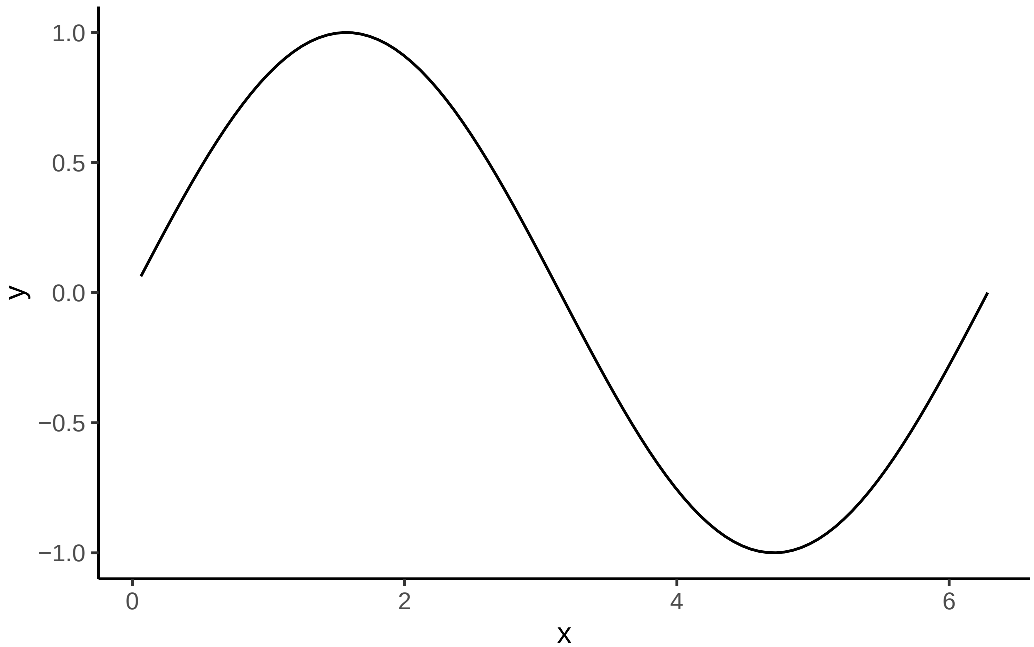
You can easily sum these (stored in the variable `numbers`) using R:

```
summed_numbers <- sum(numbers)
```

If you are curious, the sum is 5050, something the young Carl Friedrich Gauss, allegedly, solved in class as a young schoolboy.

Plotting the sine of these numbers mapped to the interval $[0, 2\pi]$ is undoubtedly a much more daunting task to perform manually, but — again — we can ask R to do it for us:

```
library(ggplot2)
ggplot(data.frame(x = numbers * 2 * pi / 100),
       aes(x = x)) +
  geom_function(fun = sin) +
  theme_classic()
```


1.0
0.5
0.0
−0.5
−1.0
y
0
2
4
6
x

6 Parameter Sensitivity Analysis for RQA

Some text …

An inline equation can use LaTeX mathematics, e.g., $f(t) = t^2 e^{-2t}$.

It is also possible to have display equations that take up a whole line:

$$z_n = z_{n+1}^2 + c.$$

Consider the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.

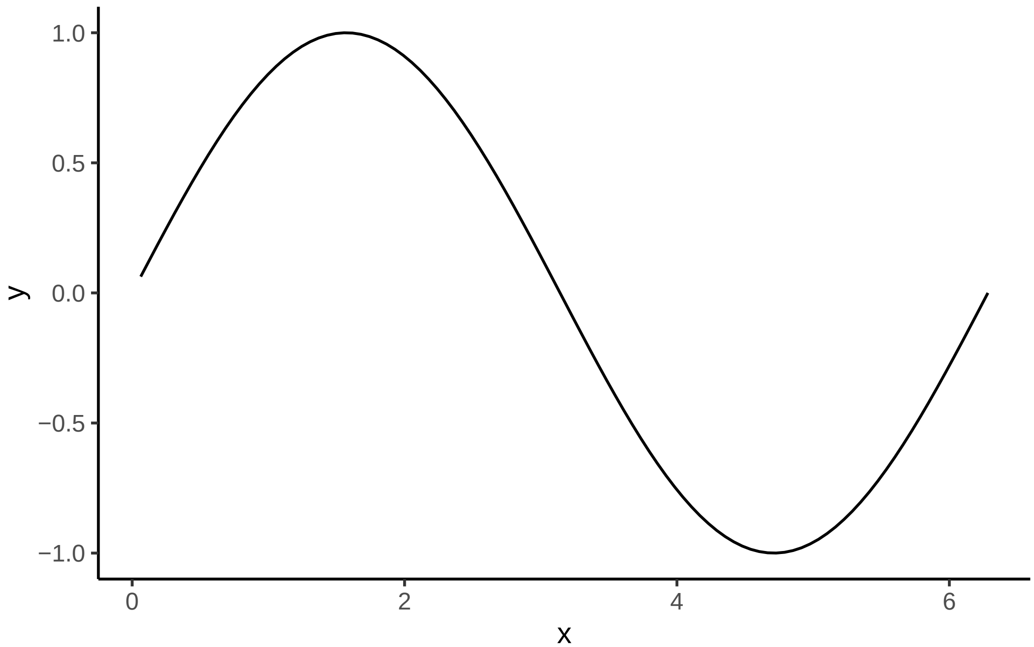
You can easily sum these (stored in the variable `numbers`) using R:

```
summed_numbers <- sum(numbers)
```

If you are curious, the sum is 5050, something the young Carl Friedrich Gauss, allegedly, solved in class as a young schoolboy.

Plotting the sine of these numbers mapped to the interval $[0, 2\pi]$ is undoubtedly a much more daunting task to perform manually, but — again — we can ask R to do it for us:

```
library(ggplot2)
ggplot(data.frame(x = numbers * 2 * pi / 100),
       aes(x = x)) +
  geom_function(fun = sin) +
  theme_classic()
```


1.0
0.5
0.0
−0.5
−1.0
y
0
2
4
6
x

7 Fractal Analysis

Some text …

An inline equation can use LaTeX mathematics, e.g., $f(t) = t^2 e^{-2t}$.

It is also possible to have display equations that take up a whole line:

$$z_n = z_{n+1}^2 + c.$$

Consider the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.

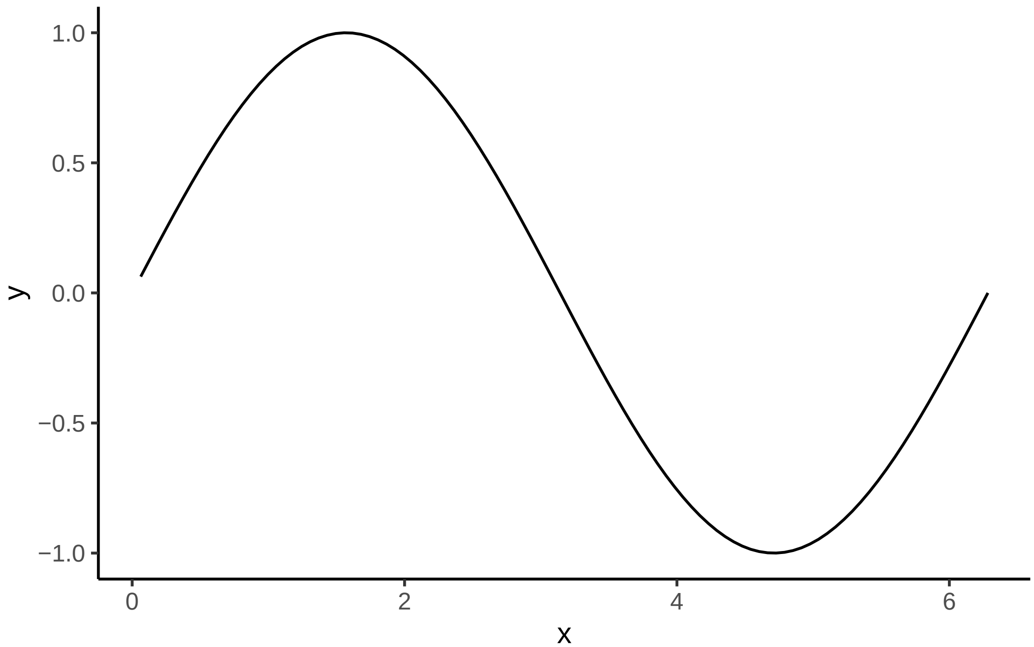
You can easily sum these (stored in the variable `numbers`) using R:

```
summed_numbers <- sum(numbers)
```

If you are curious, the sum is 5050, something the young Carl Friedrich Gauss, allegedly, solved in class as a young schoolboy.

Plotting the sine of these numbers mapped to the interval $[0, 2\pi]$ is undoubtedly a much more daunting task to perform manually, but — again — we can ask R to do it for us:

```
library(ggplot2)
ggplot(data.frame(x = numbers * 2 * pi / 100),
       aes(x = x)) +
  geom_function(fun = sin) +
  theme_classic()
```


1.0
0.5
0.0
−0.5
−1.0
y
0
2
4
6
x

8 Convergent Cross Mapping

Some text …

An inline equation can use LaTeX mathematics, e.g., $f(t) = t^2 e^{-2t}$.

It is also possible to have display equations that take up a whole line:

$$z_n = z_{n+1}^2 + c.$$

Consider the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.

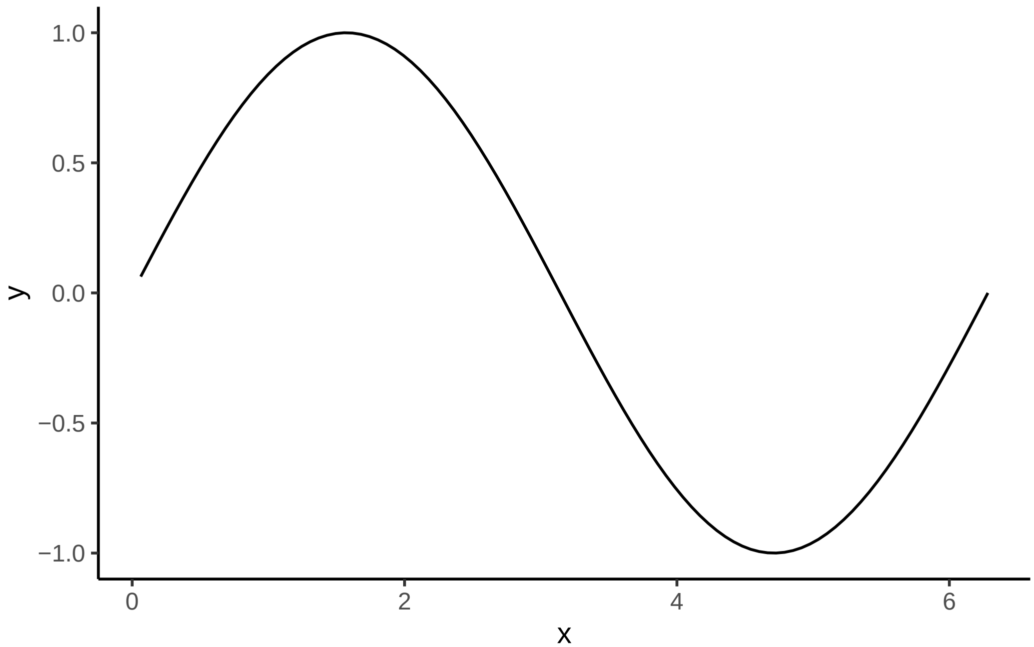
You can easily sum these (stored in the variable `numbers`) using R:

```
summed_numbers <- sum(numbers)
```

If you are curious, the sum is 5050, something the young Carl Friedrich Gauss, allegedly, solved in class as a young schoolboy.

Plotting the sine of these numbers mapped to the interval $[0, 2\pi]$ is undoubtedly a much more daunting task to perform manually, but — again — we can ask R to do it for us:

```
library(ggplot2)
ggplot(data.frame(x = numbers * 2 * pi / 100),
       aes(x = x)) +
  geom_function(fun = sin) +
  theme_classic()
```


1.0
0.5
0.0
−0.5
−1.0
y
0
2
4
6
x

9 Example of interactivity

This is an interactive web page that needs to load a few things before it is ready, so please be patient. It runs R in your browser, so all computations happen on your own computer, not on a server. You do not need to have R installed. Once you see WebR Status above as “Ready!” you can begin to run code and see the output (if any). You will need to run the code cells in order, since later cells depend on variables and computations in earlier ones. You can also edit the cells and change the code — something you will have to do in order to finish some incomplete code.

Below, is a categorical time series assigned to the vector `cat1`. In this example you will plot the time series and perform a categorical recurrence quantification analysis.

```
cat1 <- c(
  4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 1,
  1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4, 4, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 1, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3,
  3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4,
  4, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 4, 4, 4, 4, 1, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4,
  4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 1,
```

```
  1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4
)

message("Finished! You can now proceed.")
```

First, make a plot of the time series.

```
plot(cat1, type = 'b', axes = F, ylab = "Categorical TS1")
axis(2, at = c(1, 2, 3, 4), labels = c("1", "2", "3", "4"), las = 1)
axis(1, at = c(0, 100, 200, 300, 400, 500, 600),
     labels = c("0", "100", "200", "300", "400", "500", "600"), las = 1)
```

Now we load a special version of the `crqa` package that can run in the browser. There may be a warning that it was compiled under a different version of R, which you can ignore.

```
webr::install("crqa", repos = "https://tildeweb.au.dk/~au78495/repo")
library(crqa)
```

Now, perform a categorical recurrence analysis. Here, you will have to change some parameters in order to make it run correctly. These have been marked by comments in the code below.

```
RQA_cat1 <- crqa(cat1, cat1,
                 delay = 2,  # CHOOSE AN APPROPRIATE PARAMETER
                 embed = 3,  # CHOOSE AN APPROPRIATE PARAMETER
                 radius = 6, # CHOOSE AN APPROPRIATE PARAMETER
                 method = "rqa",
                 datatype = "categorical")
```

When you have made the changes, you can execute the code below to show the recurrence plot.

```
plot_rp(RQA_cat1$RP)
```

10 Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```

References

Coco, Moreno I., Dan Mønster, Giuseppe Leonardi, Rick Dale, and Sebastian Wallot. 2021. "Unidimensional and Multidimensional Methods for Recurrence Quantification Analysis with Crqa." *The R Journal* 13 (1): 145–63. https://doi.org/10.32614/RJ-2021-062.

Marwan, Norbert, M. Carmen Romano, Marco Thiel, and Jürgen Kurths. 2007. "Recurrence Plots for the Analysis of Complex Systems." *Physics Reports* 438 (5–6): 237–329. https://doi.org/10.1016/j.physrep.2006.11.001.