The Complex Systems Approach to Behavioural Science

Fred Hasselman

2019

Contents

Course Guide

This is a companion book for a number of courses listed on this website: https://complexity-methods.github.io :

- Research Master Behavioural Science curriculum: **Dynamics of Complex Systems**
- Radboud University Summerschool: **Complexity Methods for Behavioural Science: A toolbox for studying change.**
- Shorter workshops, for example: 2.5 day course in Helsinki 2020

Image from Grip on Complexity

The Complex Systems Approach

Complexity research transcends the boundaries between the classical scientific disciplines and is a hot topic in physics, mathematics, biology, economy as well as psychology and the life sciences. This course will discuss techniques that allow for the study of human behaviour from the perspective of the Complexity Sciences, specifically, the study of complex physical systems that are alive and display complex adaptive behaviour such as learning, development and creativity. Contrary to what the term "complex" might suggest, complexity research is often about finding simple models/explanations that are able to describe a wide range of qualitatively different behavioural phenomena. "Complex" generally refers to the object of study: Complex systems are composed of many constituent parts that interact with one another across many different temporal and spatial scales to generate behaviour at the level of the system as a whole, in complex systems "everything is interacting with everything".

The idea behind many methods for studying the dynamics of complex systems is to exploit the fact that "everything is interacting" and quantify the degree of periodicity, nonlinearity, context sensitivity or resistance to perturbation (resilience) of system behaviour. Applications in the behavioural sciences are very diverse and concern analyses of continuous time or trial series data such as response times, heart rate variability or EEG to assess proficiency of skills, or health and well-being. Complexity methods can also be used for the analysis of categorical data, such as behaviour observation of dyadic interactions (client-therapist, child-caregiver), daily experience sampling, social and symptom networks. The complex systems approach to behavioural science often overlaps with the idiographical approach of "the science of the individual", that is, the goal is not to generalise properties or regularities to universal or statistical laws that hold at the level of infinitely large populations, but to apply general principles and universal laws that govern the adaptive behaviour of all complex systems to a specific case,

in a specific context, at a specific moment in time.

The main focus of the course will be hands-on data-analysis. Practical sessions will follow after a lecture session in which a specific technique will be introduced.

We will cover the following topics:

- Theoretical background of phase transitions (self-organised criticality), synchronisation (coupling dynamics) and resilience (resistance to perturbation) in complex dynamical systems and networks.

- Simple models of linear and nonlinear dynamical behaviour (Linear & logistic growth, Predator-Prey dynamics, Deterministic chaos),

- Analysis of (multi-) scale dependence in time and trial series (Entropy, Relative roughness, Standardized Dispersion Analysis, (multi-fractal) Detrended Fluctuation Analysis).

- Quantification of temporal patterns in time and trial series including dyadic interactions (Phase Space Reconstruction, [Cross-] Recurrence Quantification Analysis).

- Dynamical network analyses for univariate (recurrence networks) and multivariate time series (multiplex recurrence networks).

- Using the method of surrogate data analysis (constrained realisations of time series data) to test hypotheses about the nature of the data generating process.

Learning outcomes

After completing a course you will able to:

- Simulate linear, nonlinear and coupled dynamics using simple models.
- Conduct (multi-fractal) Detrended Fluctuation Analysis and related techniques to quantify global and local scaling relations.
- Conduct Recurrence Quantification Analysis and related techniques to quantify temporal patterns, synchronisation and coupling direction.
- Conduct analyses on (multiplex) Recurrence Networks to quantify structure and dynamics of (multivariate) time series.

Naturally the (depth of) topics discussed will be limited by the duration of the course.

Level of participant

- Master
- PhD
- Post-doc
- Professional

For whom are these courses designed?

The courses are designed for all researchers who are interested in acquiring hands-on experience with applying research methods and analytic techniques to study human behaviour from the perspective of Complexity Science. Prior knowledge is not required, some experience using R is recommended.

Admission requirements

During the course we will mostly be using the R statistical software environment. Basic experience with R is highly recommended (e.g. installing packages, calling functions that run analyses, handling and plotting data). We also offer a module for the Jamovi software with which the most basic analyses can be conducted. Using Jamovi does not require any prior knowledge of R, but you will not be able to use more advanced features of certain analyses.

Please bring your own laptop to the course. We will help you to install the necessary open source software, all of which can run on Windows, MacOS and most likely also on common varieties of Unix/Linux. The specifications for your computer are simply this: You need to be able to connect to a wireless network (wifi) and you should be able to install and run R (https://www.r-project.org). In addition, you might want to be able to use RStudio (https://www.rstudio.com) and Jamovi (https://www.jamovi.org).

If you do not have the resources to bring a laptop that meets the required specifications, please let us know in advance so we can try to find an alternative solution.

Literature

Pre-course literature:

It will be helpful to read the following articles before the first day of the course:

- Molenaar, P. C., & Campbell, C. G. (2009). The new person-specific paradigm in psychology. Current directions in psychological science, 18(2), 112-117.
- Kello, C. T., Brown, G. D., Ferrer-i-Cancho, R., Holden, J. G., Linkenkaer-Hansen, K., Rhodes, T., & Van Orden, G. C. (2010). Scaling laws in cognitive sciences. Trends in cognitive sciences, 14(5), 223-232.
- Thelen, E., & Ulrich, B. D. (1991). Hidden skills: A dynamic systems analysis of treadmill stepping during the first year. *Monographs of the Society for Research in Child Devevelopment, 56(1)*, 1-98; discussion 99-104. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/1922136
- Lewis, M. D. (2000). The promise of dynamic systems approaches for an integrated account of human development. Child development, 71(1), 36-43.

Selected chapters from these books will be made available so you can make a personal copy:

- Friedenberg, J. (2009). Dynamical psychology: Complexity, self-organization and mind. ISCE Publishing.
- Kaplan, D., & Glass, L. (2012). Understanding nonlinear dynamics. Springer Science & Business Media.
- Rose, T. (2016). The end of average: How we succeed in a world that values sameness. Penguin UK.

Links to online materials on specific topics will be provided (*Study Materials*) that may provide additional explanation and information about key concepts. These materials are not obligatory, but highly recommended to study at least once.

Notes about this book and the assignments

The texts in the chapters of this book are somewhat of a work in progress, and are intended as a rough introductory guide to accompany the lectures. Sometimes,

you will notice a paragraph or chapter rather resembles a set of lecture notes instead of a self-contained text. Do not hesitate to let us know if you think anything is unclear or too far out of context for you to understand.

An essential part of the course are the assignments that are available online and are linked to from the course pages, for example: https://complexity-methods.github.io/courses/helsinki-workshop-2020/day1_2/

The text inside these blocks provides important information about the course, the assignments, or the exam.

The text inside these blocks provides examples, or, information about a topic you should pay close attentiont to and try to understand.

The text inside these blocks provides a note, a comment, or observation.

The content in these blocks are often questions about a topic, or, suggestions about connections between different topics discussed in the book and the assignments. You should decide for yourself if you need to dig deeper to answer the questions or if you want to discuss the content. One way to find an answer or start a discussion is to open a thread in the discussion forum on Blackboard labelled *ThinkBox*.

The content in these blocks is provided as entertainment :)

Schedule

You can find detailed schedules on the course website: https://complexity-methods.github.io/courses/

We used R!

This text was transformed to `HTML`, `PDF` en `ePUB` using `bookdown`(**?**) in **RStudio**, the graphical user interface of the statistical language **R** (**?**). `bookdown` makes use of the `R` version of markdown called Rmarkdown (**?**), together with knitr (**?**) and pandoc.

We'll use some web applications made in Shiny (**?**)

Other `R` packages used are: `DT` (**?**), `htmlTable` (**?**), `plyr` (**?**), `dplyr` (**?**),`tidyr` (**?**), `png` (**?**), `rio` (**?**).

Part I

Introduction

Chapter 1

A Quick Guide to Scientific Rigour

> ``Meanwhile our eager-beaver researcher, undismayed by logic-of-science considerations and relying blissfully on the ``exactitude'' of modern statistical hypothesis-testing, has produced a long publication list and been promoted to a full professorship. In terms of his contribution to the enduring body of psychological knowledge, he has done hardly anything."
>
> ---Paul Meehl (**?**, p. 114)

Before we can begin our introduction to the wonderful world of Complex Adaptive Systems and Complex Networks, we briefly discuss the philosophy of science and perspective on the goal of scientific inquiry that is used throughout this book. This will allow us to highlight some differences between the **Complex Systems Approach (CSA)** we propose for the scientific study of human nature and the perspective (implicitly) used in most disciplines of the social and life sciences, we will call the **Machine Metaphor Approach (MMA)**, **Cognitivism**, or, **Computationalism**.

Use of the **scientific method** is what separates scientific, from non-scientific claims about the nature of reality. It consists of all philosophical, theoretical, and empirical tools that can be used to systematically evaluate the veracity of such explanatory claims. The repeated application of the scientific method, to study scientific questions, promises to generate **valid** (accurate) inferences and **reliable** (precise) facts about a certain explanatory domain. It does not guarantee that any kind of absolute 'truth' will be discovered.

The '**scientific method song**' discusses the most important phases of the *empirical cycle*. Be aware that there is also a *theoretical cycle* and a *diagnotsic cycle*.

One factor affecting the perceived veracity of scientific inferences, is the quality of the body of scientific knowledge from which the inferences were deduced, induced or abducted. For example, when a *crisis of confidence* about the trustworthiness of facts in the scientific record generated by some sub disciplines of psychological science was suggested (**?**), the immediate consequence was that the veracity of all claims by psychological science was called into question.

1.1 Rigorous Open Science

Less tangible, but not less important for the perceived veracity of scientific knowledge are concepts such as *intellectual honesty* and *scientific integrity* of the scientists laying explanatory claims on some domain in reality. Merely checking whether the scientific method has been applied does not fully grasp all the prerequisites for generating a solid body of knowledge. We will use the term **rigorous open science** to denote the ideal set of conditions that should be in place to allow us to distinguish scientific claims that are likely to be false, from claims that are likely to be true, given the perceived *verisimilitude* (truth-likeness) of the knowledge accumulated in the scientific record.

Figure 1.1 – Rigorous Science according to @casadevall2016a.

When a claim is based on **Scientific Rigour** (**?**), we mean it was posited based on the following set of principles:

1. **Experimental Redundancy** - The claim has been examined by all method-

ological and analytical tools that are available and are appropriate given the context. Rigorous Science does not rely on one type of experimental design or one type of statistical analysis.

2. **Recognition of Error** - Without failure there can be no progress, therefore we should carefully study failures and not just report success stories. Any sources of error should be carefully studied and reported to the scientific community.
3. **Sound Probability & Statistics** - Use of the most recent and appropriate statistical theories, models and analytical techniques. Statistical modelling techniques become more realistic over time and often the models that were taught in undergraduate statistics courses have long been replaced and should not be used any more.
4. **Efforts to Avoid Logical Traps** - When generating theories and defining constructs and laws, make sure logical inconsistencies are avoided. When making inferences, avoid the common logical traps such as *The Effect = Structure Fallacy* in null hypothesis significance testing (NHST).
5. **Intellectual Honesty** - Rigorous science is ethical, has integrity and thrives on critical reflection on scientific practice. The right mindset is *"Prove yourself wrong!"*, not *"Prove yourself right!"*

We add to the list that science must be open and transparent. This may seem like an obvious statement to a fresh student of human behaviour, but concepts that make up an essential part of the scientific debate in 2017, such as *open science*, *open data*, *reproducibility*, *Questionable Research Practices (QRPs)*, *Hypothesizing After the Results are Known (HARKing)* and *preregistration*, were practically unknown 5 years ago.

Comedian John Oliver discusses how and why media outlets so often report untrue or incomplete information as science:

Table **1.1**
Strong Inference according to @platt1964strong

	Strong inference consists of applying the following steps to every problem in science, formally and explicitly and regularly:
1.	Devising alternative hypotheses
2.	Devising a crucial experiment (or several of them), with alternative possible outcomes, each of which will, as nearly as possible, exclude one or more of the hypotheses
3.	Carrying out the experiment so as to get a clean result
1'	Recycling the procedure, making subhypotheses or sequential hypotheses to refine the possibilities that remain
...	and so on.

Strong Inference

> A difficulty of much psychological theorizing is vagueness in the terms employed. In this work, the above ideas have been studied in mathematical form throughout, the definitions and proofs being given corresponding precision.
>
> ---W. R. Ashby in `The Physical Origin of Adaptation by Trial adn Error' (**?**, p. 13)

The Effect = Structure Fallacy refers to the logical error that occurs a predicted effect is observed (i.e. a statistically significant test result leads to a rejection of the null hypothesis), it is not valid to infer the existence of the assumed cause was evidenced. NHST is based on *the falsification principle*, which means the perceived veracity of a scientific claim will increase only if it has resisted many rigorous attempts to prove it is wrong. If a scientific claim has a large track-record of resisting falsification attempts, we can call it plausible, or high in verisimilitude, but this could all change with one crucial experiment. Contrary to what some scholars suggest, falsifiability is not optional in a rigorous science (**?**).

An excellent recipe for a rigorous application of the scientific method was provided by **?**. Perhaps we should implement it and get us out of the curious situation in which so many different “theories” competing to explain the same pehnomena can be considered to be “true” at the same point in time.

1.2 Theoretical Tunnelvision

> ``It is the theory that decides what we may observe''
>
> ---Einstein (as quoted by Heisenberg)

Many of the initiatives proposed to improve the social and life sciences focus on improving methodology and statistics. This is understandable, it's where errors are easily made (and discovered) and it allows for relatively simple interventions, e.g. more stringent control on appropriate use of statistics by journals. However, the goal of generating empirical facts is ultimately because we want to find out which scientific claim about the structure of reality best explains why those empirical facts were observed.

The quote attributed to Einstein refers to an important, and grossly underestimated phenomenon one might call the *theoretical tunnelvision*. It is best explained by an example that is commonly encountered in the literature in psychological science and goes something like this:

1. A study tries to find independent causes (predictors) of a certain disease-entity, a pathological state or behavioural mode people can 'get stuck in'.
2. Typically, a statistical model fitted on a large, representative sample of individuals in which many different predictors were measured will yield associations between predictor and disease-entity that are significant but small (on average $r \approx 0.3$, or $\approx 9\%$ explained variance).
3. Often, if other known (non-clinical) covariates are included in a model, or, if the multivariate nature of the phenomenon is taken seriously by including repeated measurements and/or multiple dependent variables, these predictors will no longer explain any unique variance in the outcome measures.

Here's an example of a 'predictor' study (**?**) to find predictors of persistence of Major Depressive Disorder MDD 10 over the course of 10 years in a representative sample of 331 individuals who suffered MDD 10 years earlier:

> ``Clinical variables in this analysis were not strongly associated with persistence of MDD over the course of 10 years. Comorbid generalized anxiety disorder, baseline depression severity, and taking a prescription for nerves, anxiety, or depression were significantly associated with persistent depression in the unadjusted logistic regres-

> sion models, but the associations became non-significant when in the multivariate model. These findings are in contrast to the results from several other studies."

The study concludes by discussing three factors that play a statistically significant role in the persistence of MDD (text between brackets not in original):

- "*having two or more chronic medical conditions [in 1995-1996] contributes to experiencing depression ten years later.* [2.89 more likely] *However, only having one chronic medical condition did not increase the odds of being classified as having MDD in 2004–2006.*"
- "*days of activity limitation in 1995–1996 were significantly associated with a greater risk of depression ten years later,* [2.19 more likely] *independent of the number of chronic medical conditions a person had.*"
- "*Individuals who were in contact with family less than once a week* [in 1995-1996] *were more likely to have MDD in 2004–2006.* [2.07 more likely] *Likewise, people who were married were less likely to have persistent depression compared to those who have never married* [never married 2.42 more likely]"

So? What's wrong?

So what's wrong with these inferences? The study shows some previous assumptions about the relevance of clinical predictors should be reconsidered, and it adds to scientific record some facts about risk factors that might have eluded scientists, clinicians and health professionals. Let's look at the main conclusion of the study, in addition to a plea for more attention for people with two or more chronic medical conditions, **?** end the article with:

> Future research should continue to examine the complex nature of the relationship between chronic medical disorders and comorbid psychiatric conditions. Addressing these conditions and strengthening social support systems could be important strategies for reduce the burden of depression.

Here's what is odd from the perspective of *rigorous science*:

1. If clinical predictors play no role in explaining why some people remain depressed for such long periods of time, why isn't the main conclusion of the

study that we must re-appraise the scientific theories laying explanatory claim on the aetiology of MDD? It is from these theories that the diagnostic tools, the medical, and psychological interventions to which these patients have been exposed, were derived.

2. Even though the authors acknowledge –and indeed show– that the propagation of a pathological state like MDD over many years is a very complex multivariate phenomenon, their suggestion for future research is still based on an implicit assumption about causation that is extremely simple. The idea is that there is a chain of unique (efficient) causes, each contributing independently to the emergence, and persistence in time of the MDD state. The authors basically suggest some component causes have to be added to the aetiology. The metaphor is that of a *machine* of which the sum output of its constituent components is equal to the purpose or function of the machine as a whole. Should a component fail, then it can be repaired or replaced as long as it performs the same function as the defective part, thereby restoring the function of the machine as a whole. This is why the authors suggest that strengthening social support systems could be an intervention to reduce the burden of depression: The absence of a partner or visits by family members were predictors that explained some unique variance in the data on the persistence of MDD. Obviously, restoring this defective social support component should restore or at least facilitate the escape from the MDD state. Meanwhile, they seem to forget that they convincingly argued that MDD is a very complex phenomenon that cannot be dissected into neat, independent component causes.

3. Very much related to the previous point: The authors mention three important factors in the discussion and conclusion section, however, the results section contains another factor that was omitted, it is in fact the second most important predictor of the persistence of MDD:

> ``Women had 2.48 the odds of remaining depressed compared to men''

Why did they ignore this predictor in the discussion? This is speculation, but could it be that this factor is not mentioned because it would have to be considered a 'deficient' component and suggesting any kind of 'treatment' intended to 'repair' it is of course beyond the realm of sane things to suggest. Nevertheless, it does seem rather important to figure out why women are 2.5 times more

likely than men to still be depressed after 10 years. Perhaps *not* considering gender to be a unique causal component in a chain of *in*dependent predictors might help. Instead, gender could be considered a complex aggregate, or, contextual variable that is associated to the dependent variable through a vast network of *inter*dependent facts, events and states of affair. An obvious factor of importance is that effect-studies of medical interventions are mainly conducted on white, male, 20-30 year old, right-handed, subjects with above average SES. Also, it is likely that on average, the stability of mood over longer periods of time is more variable in women than in men due to fluctuations of hormone levels, but also due to antenatal and postnatal depression (**?**). It does not seem unreasonable to suggest this poses extra challenges for women who want to escape the MDD state.

No such thing as theory-free 'facts'

The analytical tools selected by the researchers (a generalized linear statistical model) restricts the kinds of associations we might observe in the data. In the the present case all associations will –after transformation– be linear compositions of independent components.[1] One never reads this valid equally valid conclusion: "*We conclude that the linear model is inadequate to describe the complexity of this phenomenon.*" The reason is that the implicit assumptions about causality underlying scientific claims never enter the empirical cycle and therefore escape falsification by the repeated application of the scientific method even though those causality assumptions are also based on a scientific theory about the structure of reality that is in principle falsifiable.

[1] Naturally, if one would use mixed models we can account for dependencies in the data, but they will still be limited to linear associations.

Study Materials

Phenomena, theories, facts and laws

``All science is either physics or stamp collecting.''

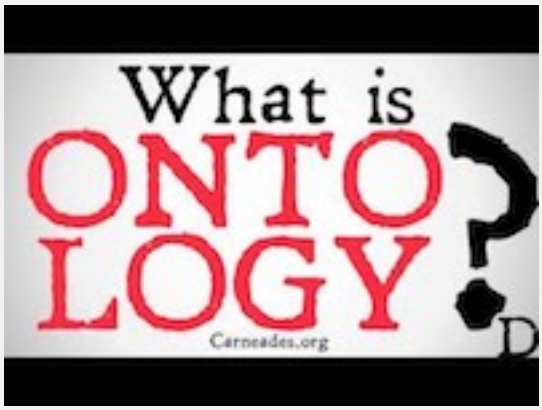
---Ernest Rutherford (Physics Nobel Laureate, 1872-1937)

It's important to distinguish between phenomena, hypothesis, theory and law. For example, we will be discussing, nonlinear *phenomena*, catastrophe *theory* and power *law* scaling.

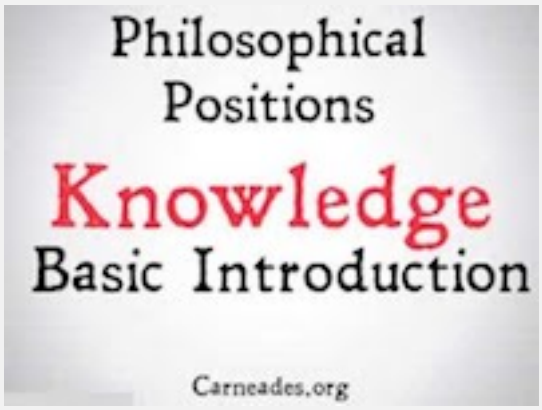
The video is provides a very clear explanation of the differences between these concepts.

You might also want to refresh your knowledge about some important aspects of scientific theorising about reality: *Ontology* and *Epistomology*

Ontology.



Epistemology



Intellectual Honesty and *Epistemic Responisiblity*:

Epistemic Responsibility


CC
14
conspiracy theories
& EPISTEMIC
RESPONSIBILITY

Chapter 2

Introduction to Complexity Methods

Psychological systems are biological systems which are physical systems that are alive. Therefore, any theory that lays explanatory claim to phenomena of the mind, ultimately must be a theory about how a physical system is able to accumulate non-random order into its internal structure that appears to codetermine its behaviour. Less formally stated, a science that studies the behaviour of physical systems that are alive, that appear to have a memory which makes their behaviour adaptive, future oriented and intelligent, should be grounded in physical and biological principles and laws.

For now, generating such a theory might be a bridge too far (however, see **?**), the least we may demand is that our current theories of human behaviour should *not* contradict highly corroborated theories of physics that describe (constituent components of) simple or complex dynamical systems. This is arguably not the case in current psychological theorising, theories assume internal, highly organised structures (such as mental representations) as causes for behaviour, without explaining where the order came from, or how it is maintained or increased. Well studied and formally defined constructs from other scientific disciplines are often imported at a metaphorical level, or are misinterpreted and essentially wrong. For example, *plasticity*, *holism*, *behavioural state/mode/change*, and especially, any concept related to the term *information* (computation, coding/decoding, information processing/storage/retrieval, entropy, etc.). Information is a formally defined quantity that resolves uncertainty about the states or properties of a theoretical object of measurement (e.g. a system, a signal) relative to its degrees of freedom, by assigning it (the uncertainty), a value. If a system represents 1 bit

of information (e.g. a coin-toss system), this means it means it can be in 1 of 2 states, or have one of 2 distinct values.

This is clearly not the same as “meaning” with which it is often conflated in theorising about cognition and behaviour. Shannon lucidly explained this in his seminal paper, which was to be the start of a new scientific discipline, information theory:

> ``The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or *conceptual entities*. These *semantic aspects of communication* are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.''
>
> ---Shannon (1948, p.379)

2.1 The Complex Systems Approach

The *Complex Systems Approach* to behavioural science departs from the assumption (which is probably not very constroversial) that human behaviour originates from a complex adaptive system.

A *system* is an entity that can be described as a composition of components, according to one or more organizing principles. The organizing principles can take many different forms, but essentially they decide three important features of systems that have to do with the relationship between parts and wholes and therefore whether we would call a system complex or not.

In order to find out what kind of system we are dealing with, we can ask three basic questions:

1. What are the relevant scales of observation of the system?
2. What are the relevant phenomena that may be observed at the different scales of observation, and are there any interactions across the relevant scales of observation that are needed to explain the relevant phenomena?
3. Can interactions with the internal and external environment of a system occur, and if so, do these interactions have any after-effects on the structure and/or behaviour of the system?

If the answer to the first question is “many” and to the second and third “yes” it is very likely we are dealing with a complex dynamical system.

So let’s look at some properties of this system that generates human behaviour, it’s a system:

- … which has many different constituent parts, and those parts are often also systems with many different constituent parts (the tRNA system, the prefrontal cortex, the respiratory system, the speech system, the endocrine system, the microbiome, etc.).
- … that is *open* and can exchange energy, matter and information with its internal and external environment, as a consequence, dissipating heat (disorder, entropy) back into the environment.
- … which has many different internal states that can have their own specific dynamics, sometimes appearing to be independent of, but oftentimes coupled to, the dynamics of other internal states (emotional states, motivational states, attentional states, physical fitness, general health, biological development, etc.).

- … in which there are many potential levels of organisation and, therefore, potentially many different levels of analysis (cognitive development, cognitive neuroscience, lifespan IQ, socio-cultural differences in IQ, etc.).
- … in which many processes operate on, and, interact across, many different spatial and temporal scales (Studying proficiency at playing chess: social/cultural/pedagogical/genetic contexts, development of social/emotional/motor/cognitive skills, availability/quality of education, motivation, personality, etc.).

So… that's probably a *yes* on complex dynamical system. To be more specific, we can state that most living organisms, including human beings, are **complex adaptive systems with internal state dynamics**.

2.2 Ergodicity and the Measurment Problem

It does not take an expert on *population statistics* to see there is probably a mismatch between the interesting behavioural phenomena and the analytical toolbox most frequently used to study human behaviour in the social and life sciences. Anyone who took an introductory class in inferential statistics will remember the assumptions of statistical models require observations to be independent of one another, variances to be homogeneous (e.g. Levene's test), and measurement error to be essentially random in nature and normally distributed, not correlated to any other factors that might cause the phenomenon under scrutiny.

Given the nature of the phenomena of interest and the properties of the system under scrutiny, there are two main concerns about the scientific study of human behaviour:

1. The assumption that *the ergodic theorems apply* to the theoretical objects of measurement and data generating processes (**??**): Ensemble averages of variables observed in samples of sufficiently many individuals are expected to be arbitrarily similar to the time averages of variables evolving over a sufficiently long interval of time, from any single initial condition.
2. The assumption that the interpretation of outcomes of psychological measurement is, or should be, equivalent to *classical physical measurement* (**?**): It is considered unproblematic to interpret a measurement outcome as a property of the theoretical object of measurement confounded by some random additive measurement noise or sampling error.

The validity of the assumptions related to ergodicity (i.e. stationarity and homogeneity of central moments) are obviously important for making valid statistical inferences and generalizations. However, even if some of the core assumptions for an ergodic data generating process are formally valid, one cannot rely on parameter estimates to converge on a characteristic expected value within the time scale of observation, or, scale of fluctuation, as is the case when the process samples from a stable distribution with one or more undefined central moments like the Cauchy distribution. This has led some scholars to suggest that "*the very notion of probability may not make sense*" (**?**) when studying complex systems with internal state dynamics.

Recent observations of discrepancies between inferred properties at the ensemble level (inter-individual) and the individual level (intra-individual), have been suggested as a cause of the so-called reproducibility crisis in the social and life

sciences (**???**). A study which observed a lack of 'group-to-individual generalizability' in the context of psychopathology described the phenomenon as a threat to human subjects research: "*In clinical research, diagnostic tests may be systematically biased and our classification systems may be at least partially invalid. In terms of theory development, we may have a misleading impression about the nature of psychological variables and their interactions.*" (**?**). A study of the neuroanatomical phenotypes of schizophrenia and bi-polar disorder (**?**) concluded: "*This study found that group-level differences disguised biological heterogeneity and interindividual differences among patients with the same diagnosis. This finding suggests that the idea of the average patient is a noninformative construct in psychiatry that falls apart when mapping abnormalities at the level of the individual patient.*"

The second concern is about the lack of a clear notion in psychology and the life sciences of how to incorporate the measurement context and the act of measurement into the description of a phenomenon (**?**). Psychological measurement is an interaction between a (prepared) theoretical object of measurement and the elements of the measurement procedure (experimental design, instruments, etc.). The very act of asking someone to project their current internal state of happiness onto an arbitrary ordinal scale will interfere with their "true" state of happiness (if such a thing even exists without the measurement context). There is no "happiness" equivalent for unobtrusive measurement of body temperature using an infra-red camera.

Resolutions to these and other problems with psychological measurement have been proposed, for example the various types of conjoint measurement (**??**), or suggestions to adopt concepts from quantum measurement (**??**). However, when measurement and analysis of the temporal evolution of internal states is concerned, problems arise due to the fact that living systems are subject to *ageing* (loss of identity over time) and appear to be able to coordinate their current behavior relative to some record of previously experienced events. In more general terms, the behavior of a complex adaptive system will display after-effects of interactions with its internal or external environment that extend far beyond any timescale that might be understood as a simple stochastic process with autoregressive components. Time series of observables of living systems will often lack the memoryless-ness property (**??**), suggesting anomalous, rather than normal diffusion processes should be considered as a model for the data generating process (**?**).

2.3 Component- vs. Interaction-dominant Dynamics

A helpful framework for discussing the differences between a Complex Systems Approach and a Machine Metaphor Approach to the scientific study of human behaviour is to describe the causal ontology used to explain behaviour. Familiar "degrees of causation", or entailment are possible in component dominant dynamics, such as uniquely explained variance, beta weights or effect sizes. In general, a linear arrangement of partial causes always neatly sum up to produce the behaviour of interest. An alternative causal ontology is interaction dominant dynamics in which not the components themselves, but their interactions as a whole are the source of the observed behaviour (Ihlen & Vereijken, 2010; Kello, Beltz, Holden, & Van Orden, 2007; Van Orden, Holden & Turvey, 2003; Van Orden, Holden & Turvey, 2005; Wijnants, Cox, et al., 2012). Here the contribution of components is not additive, but multiplicative and nonlinear (Holden et al., 2009; van Rooij, Nash, Rajaraman, & Holden, 2013). Such interaction dominant dynamics render individual component behaviour (which are still posited to exist), such as poor performance on ability X, impaired representation of that feature Y, as a less interesting object of theoretical and empirical inquiry.

As a consequence, theoretical and empirical inquiry is aimed at identifying and understanding the contexts in which impaired behaviour emerged. Adopting such a perspective entails that all observable behaviour can only be understood relative to the context in which it was observed, that is, the measurement context (cf. Holden, Choi, Amazeen, & Van Orden, 2010; Van Orden, Kello, & Holden, 2010). Figure presents the fundamental differences between the two ontologies in their assumptions about the causes of behaviour and their assumed place of measurement. Figure may reveal why the nature of cognitive components and processes remain elusive in their causal role. They are inferred, not postulated, based on data from different places of measurement. Their causal structure does not incorporate the nested nature of both measurements as well as posited entities. Applying the concept of the complex conditional reveals hierarchical dependencies of one condition on another and such a complex, if it were composed of the correct conditionals, should be considered as a whole. As a consequence, impaired behaviour should be understood as emerging from the whole of constituent components, not from an individual component. The notion of a cause is somewhat more radical than the complex conditionals and is known as impredicative, circular causation (Chemero & Turvey, 2010; Freeman, 1999; Turvey, 2007), or nested causation.

2.4 A Behavioural Science of the Individual?

The dictum "first analyze, then aggregate" (by Peter Molenaar as quoted in **?**), advocates an order reversal of the methods commonly employed by the behavioural sciences to advance scientific knowledge: Measurement outcomes of psychological variables observed in samples of individuals that share some identity are first aggregated, then analyzed, in order to decide whether the statistical properties of the variables realized in the sample can be considered general regularities or true characteristics of the identity under scrutiny. As such, nomothetic behavioural science produces knowledge about psychological and behavioural phenomena that are expected to be robust at the level of aggregated wholes. However, contrary to the applied branches of economic, sociological and political science, the scientist practitioner generally does not deal with the behaviour of aggregated wholes such as economies, societies and groups of voters. The domain in which they have to apply this knowledge concerns the explanation, intervention, or prediction of the behaviour of particular patients, children, or employees, that is, they deal with the individuals that constitute the aggregated wholes. As mentioned earlier, recent studies have shown that applying the statistical syllogism to nomothetic knowledge in psychology may be very problematic.

One could argue that the rules of sample-based statistical inference do not warrant generalizations to a particular case, so one should not engage in such inductive reasoning in the first place. This is of course an unsatisfactory solution and about 15 years ago Molenaar presented an alternate path for psychological science in a paper entitled: "A Manifesto on Psychology as Idiographic Science: Bringing the Person Back Into Scientific Psychology, This Time Forever" (**?**). It's still too early to call whether it will indeed be forever, but the past decade did see a surge of scientific studies employing a person-centered or idiographic approach, most prominently in the field of psychopathology. Many researchers have been exploring what can be described as a small data paradigm (cf. **?**), which departs from the same goals and principles as so-called precision, or personalized medicine (**?**), promising to yield new insights and better tools for scientist practitioners to apply to the particular cases they encounter in their daily practice (see e.g., **??**). In general, the approach entails studying a limited number of cases intensively, for example, by collecting densely sampled multivariate time series data that represent self-reports of emotional states or well-being, using the so-called Experience Sampling Method, or Ecological Momentary Assessment (**???**). Naturally, this change of focus in data collection from static sampling from homoge-

neous populations, to dynamic process monitoring in individuals is accompanied by the development of new theories, models and analytic techniques, of which the network approach to psychopathology has probably been the most popular and revolutionary approach (cf. **??**).

There appears to be a substantial gap between the conceptual changes in the theoretical ideas about the nature of the system in which psychological phenomena can be observed and the analytic techniques and rules of inference employed to study such systems. Consider the following description of the personalized approach:

> ``The personalized approach to psychopathology conceptualizes mental disorder as a complex system of contextualized dynamic processes that is non-trivially specific to each individual, and seeks to develop formal idiographic statistical models to represent these individual processes.'' (**?**)

Although many studies using the personalized approach indeed depart from the idea that human behaviour, whether pathological or not, should be considered to arise from a complex dynamical system (see e.g., **?**), few studies actually make use of the methods and analyses from Complexity Science that were developed to study such systems. It appears to be the case that in terms of methods and analytic techniques, the idiographic revolution is taking place right now, but the complexity revolution has yet to occur. Two recent reviews of the personalized approach (**??**) fail to discusses, or even mention, the use of complexity methods to study multivariate time series data in the context of psychopathology. Such studies do exist and have been a part of the scientific record for at least as long as studies using the Gaussian Graphical Model to model multivariate ESM data (see e.g., **????????**).

It is the purpose of this book to bridge the gap and introduce complexity methods that will be essential tools for the analytic toolbox of a behavioural science of the individual.

Study Materials and Resources

Graphical representation of complex system properties

This is a link to a 'cheat sheet'-style document of complex system concepts created by dr. Joanna Boehnert: https://www.cecan.ac.uk/news/the-visual-representation-of-complexity/

The Complexity Explorer Glossary

There is a glossary of terms in Appendix C, but the ultimate glossary can be found on the Complexity Explorer site maintained by the Santa Fé institute: https://www.complexityexplorer.org/explore/glossary/

TED talks

There are quite a lot of interesting TED talks on complexity and related topics. Some of them are highlighted here, but you are encouraged to look for them yourself as well.

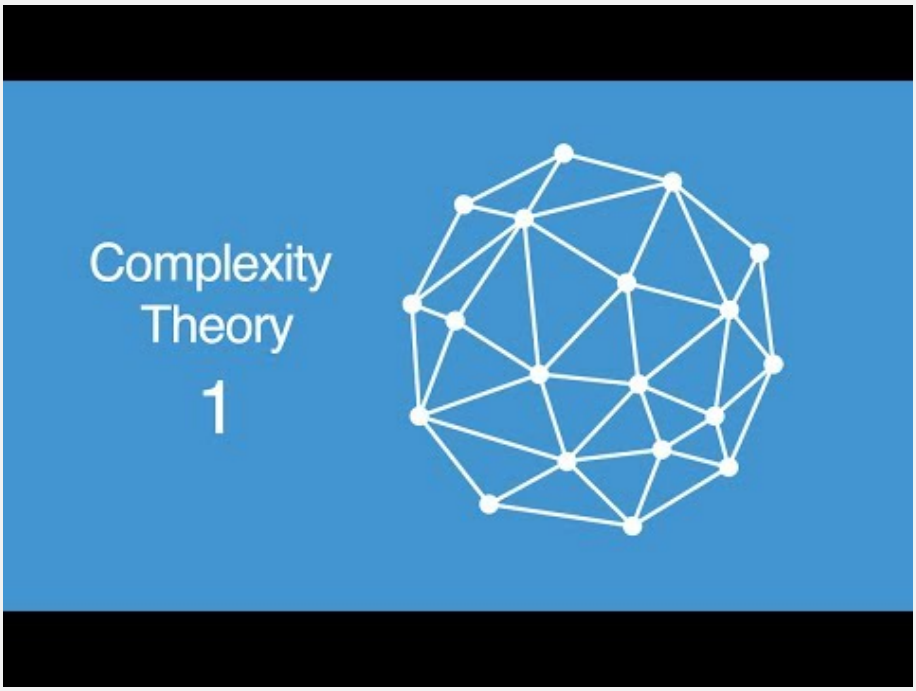
Simplicity in Complexity

Complex Adaptive Systems

Systems Innovation videos

The Systems Innovation platform has lots of resources on Complex Systems, Complex Networks and related topics. Their YouTube channel contains a wealth of informative videos.

A working definition of complex systems



Self-organization

The butterfly effect


Nonlinear
Systems

Part II

Mathematics of Change

Chapter 3

Introduction to the Mathematics of Change

The simplest non-trivial *iterative change process* can be described by the following *difference equation*:

$$Y_{i+1} = a * Y_i$$

The equation describes the way in which the value of Y changes between two adjacent, **discrete** moments in time (hence the term difference equation, or recurrence relation). There are two parameters resembling an intercept and a slope:

1. The starting value Y_0 at $i = 0$, also called the *starting value*, or the *initial conditions*.
2. A rule for incrementing time, here the change in Y takes place over a discrete time step of 1: $i + 1$.

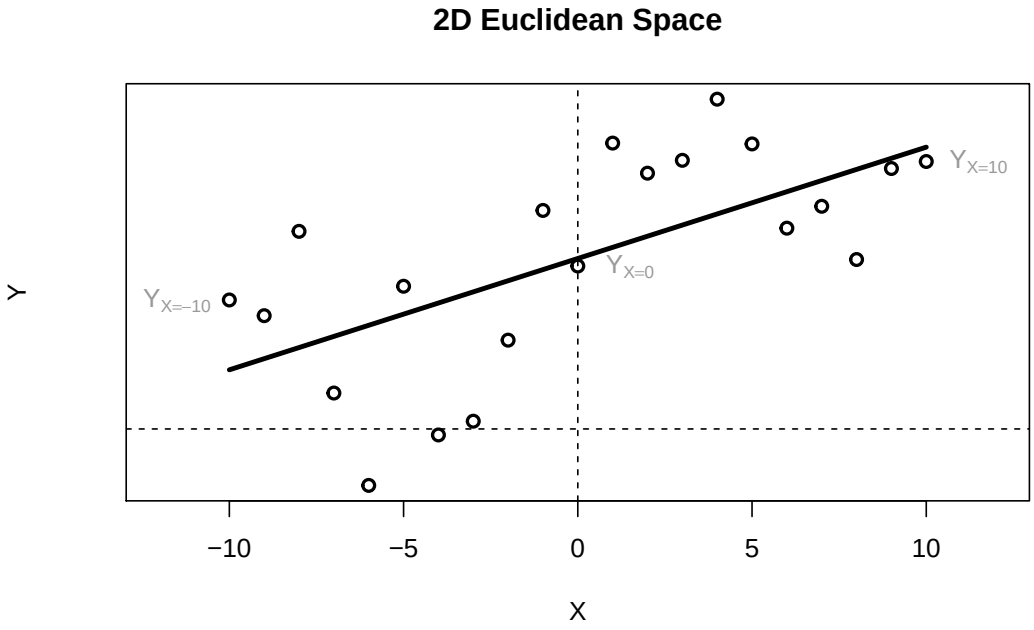
The values taken on by variable Y are considered to represent the states quantifiable observable alternative ways to describe the change of states :

- A dynamical rule describing the propagation of the states of a system observable measured by the values of variable Y through discrete time.
- A dynamic law describing the time-evolution of the states of a system observable measured by the variable Y.

These descriptions all refer to the change processes that govern system observables (properties of dynamical systems that can be observed through measurement).

3.1 It's a line! It's a plane!

The formula resembles the equation of a line. One could consider the value Y_0 a constant like the intercept of a line. The proportion of the value of Y at time i by which Y changes is given by parameter a. This is similar to the slope of a line. However, in a 2D (X, Y) plane there are two 'spatial' (metric) dimensions representing the values two variables X and Y can take on (see figure), in the case of a time series model, the X-axis represents the time dimension.



The best fitting straight line through the points in the (X, Y) plane is called a statistical model of the linear relationship between the observed values of X and Y. It can be obtained by fitting a General Linear Model (GLM) to the data. If X were to represent repeated measurements the multivariate GLM for repeated measures would have to be fitted to the data. This can be very problematic, because statistical models rely on the assumptions of Ergodic theory:

> ``... it is the study of the long term average behavior of systems evolving in time.''

The ergodic theorems require a process to be stationary and homogeneous (across time and/or different realisations of the process, i.e. between different individuals).

In other words: If you throw 1 die 100 times in a row, the average of the 100 numbers is the **time-average** of one of the observables of die-throwing systems. If this system is ergodic, then its **time-average** is expected to be similar to the average of the numbers that turn up if you throw 100 dice all at the same instance of time. The dice layed out on the table represent a spatial sample, a snapshot frozen in time, of the possible states the system can be in. Taking the average would be the **spatial average** this observable of die-throwing systems. This ergodic condiciotn is often implicitly assumed in Behavioural Science when studies claim to study change by taking different samples of individuals (snapshots of system states) and comparing if they are the same.

One also needs to assume the independence of measurements within *and* between individual realisations of the process. These assumptions can be translated to certain conditions that must hold for a (multivariate) statistical model to be valid. Some well known conditions are *Compound Symmetry* and *Sphericity*:

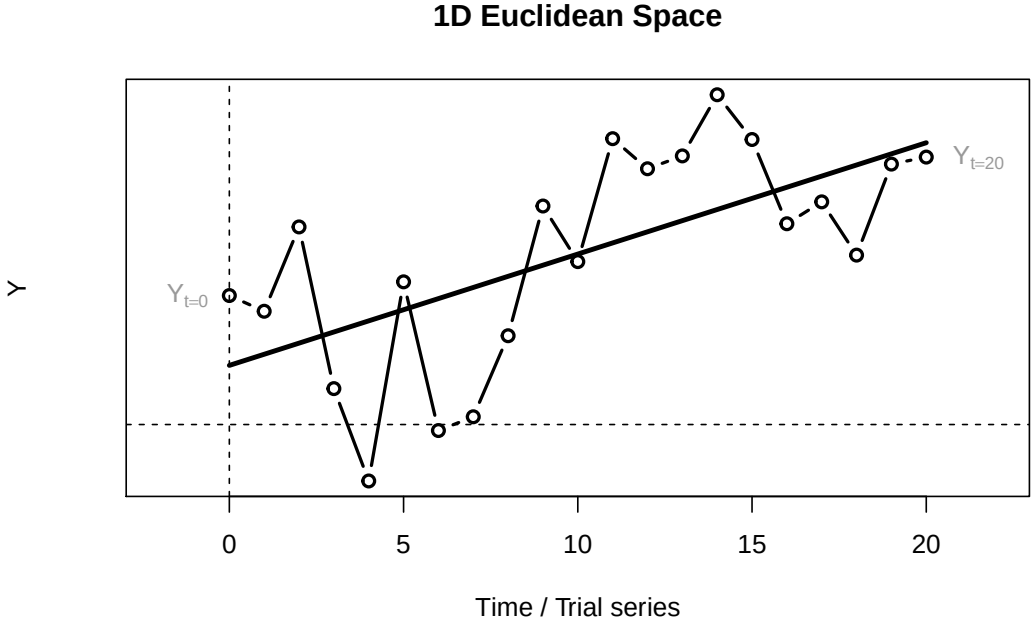
> The compound symmetry assumption requires that the variances (pooled within-group) and covariances (across subjects) of the different repeated measures are homogeneous (identical). This is a sufficient condition for the univariate F test for repeated measures to be valid (i.e., for the reported F values to actually follow the F distribution). However, it is not a necessary condition. The sphericity assumption is a necessary and sufficient condition for the F test to be valid; it states that the within-subject ``model'' consists of independent (orthogonal) components. The nature of these assumptions, and the effects of violations are usually not well-described in ANOVA textbooks; [^assumptions]

As you can read in the quoted text above, these conditions must hold in order to be able to identify unique independent components (i.e. the linear predictor X) as the sources of variation of Y over time within a subject.

If you choose to use GLM repeated measures to model change over time, you will only be able to infer independent components that are responsible for the time-evolution of Y. As is hinted in the last sentence of the quote, the validity of such inferences is not a common topic of discussion statistics textbooks.

3.2 No! … It's a time series!

The important difference between a regular 2-dimensional Euclidean plane and the space in which we model change processes is that the X-axis represents the physical dimension **time**. In the case of the Linear Map we have a 1D space with one 'spatial' dimension Y and a time dimension t (or i). This is called a *time series* if Y is sampled as a continuous process, or a *trial series* if the time between subsequent observations is not relevant, just the fact that there was a temporal order (for example, a series of response latencies to trials in a psychological experiment in the order in which they were presented to the subject).



Time behaves different from a spatial dimension in that it is directional (time cannot be reversed), it cannot take on negative values, and, unless one is dealing with a truly random process, there will be a temporal correlation across one or more values of Y separated by an amount of time. In the linear difference equation this occurs because each value one step in the future is calculated based on the current value. If the values of Y represent an observable of a dynamical system, the system can be said to have a history, or a memory.

Ergodic systems do *not* have a history or a memory that extends across more than a sufficiently small time scale (e.g. auto-correlations at lags of ±1-5 can be expected, but there should be no systematic relationships that span several

decades). Assuming such independence exists is very convenient, because one can calculate the expected value of a system observable (given infinite time), by making use of of the laws of probabilities of random events (or random fields). This means: The average of an observable of an Ergodic system measured across infinite time (its entire history, the **time-average**), will be the be the same value as the average of this observable measured at one instance in time, but in an infinite amount of systems of the same kind (the population, the **spatial average**) [^dice].

The simple linear growth difference equation will always have a form of *perfect memory* across the smallest time scale (i.e., the increment of 1, from t to $t+1$). This 'memory' just concerns a correlation of 1 between values at adjacent time points (a short range temporal correlation, SRC), because the change from Y_t to Y_{t+1} is exactly equal to $a * Y_t$ at each iteration step. This is the meaning of deterministic, not that each value of Y is the same, but that the value of Y now can be perfectly explained form the value of Y, one moment in the past.

Summarising, the most profound difference is not the fact that the equation of linear change is a deterministic model and the GLM is a probabilistic model with parameters fitted from data, this is something we can (and will) do for a as well. The profound difference between the models is the role given to the passage of time:

- The linear difference equation represents changes in Y as a function of the physical dimension *time* and Y itself.
- The GLM represents changes in Y as a function of a linear predictor composed of additive components that can be regarded as independent sources of variation that sum up to the observed values of Y.

Figure **??** shows the main differences between the GLM and Differential/Difference equation models.

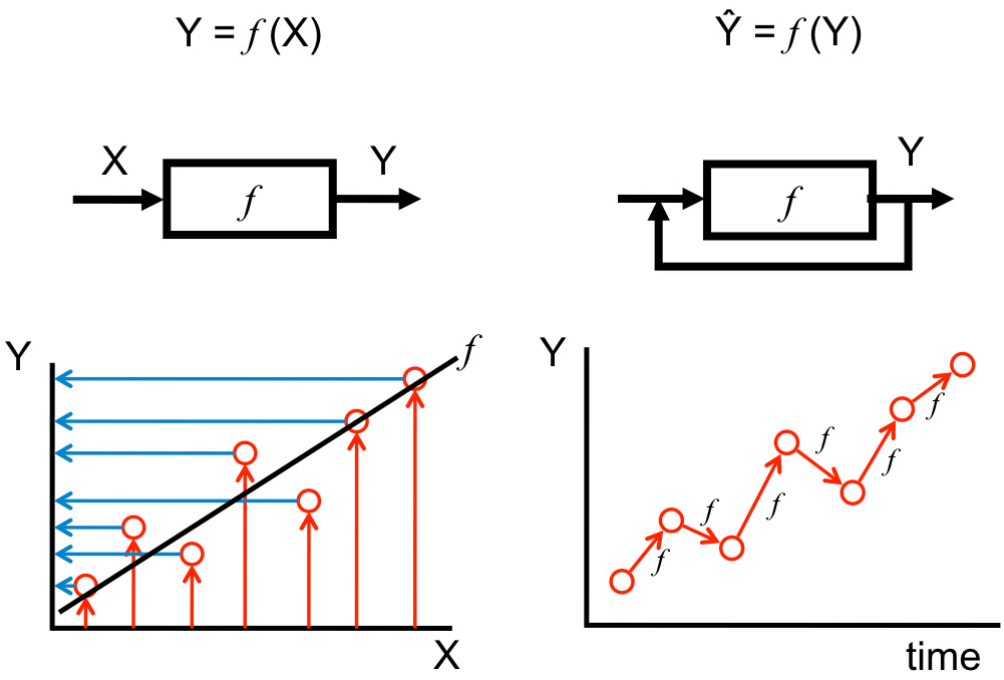


Figure 3.1 – General Linear Model versus Differential/Difference equations

3.3 Implementing iterative functions

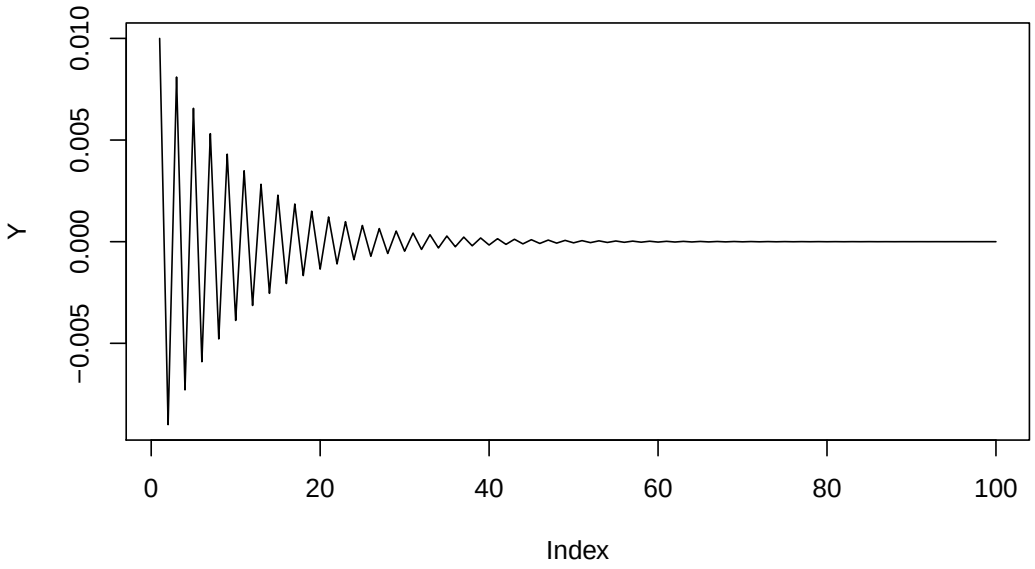
Coding change processes (difference equations) in `Matlab` and `R` is always easier than using a spreadsheet. One obvious way to do it is to use a counter variable representing the iterations of time in a `for ... next` loop (see tutorials). The iterations should run over a vector (which is the same concept as a row or a column in a spreadsheet: An indexed array of numbers or characters). The first entry should be the starting value, so the vector index 1 represents Y_0.

The loop can be implemented a number of ways, for example as a function which can be called from a script or the command or console window. In `R` working with **functions** is easy, and very much recommended (see tutorials), because it will speed up calculations considerably, and it will reduce the amount of code you need to write. You need to gain some experience with coding in `R` before you'll get it right. In order to get it lean and clean (and possibly even mean as well) you'll need a lot of experience with coding in `R`,therefore, we will (eventually) provide you the functions you'll need to complete the assignments in the **Answers** section of the assignments. If you get stuck, look at the answers. If you need to do something that reminds you of an assignment, figure out how to modify the answers to suit your specific needs.

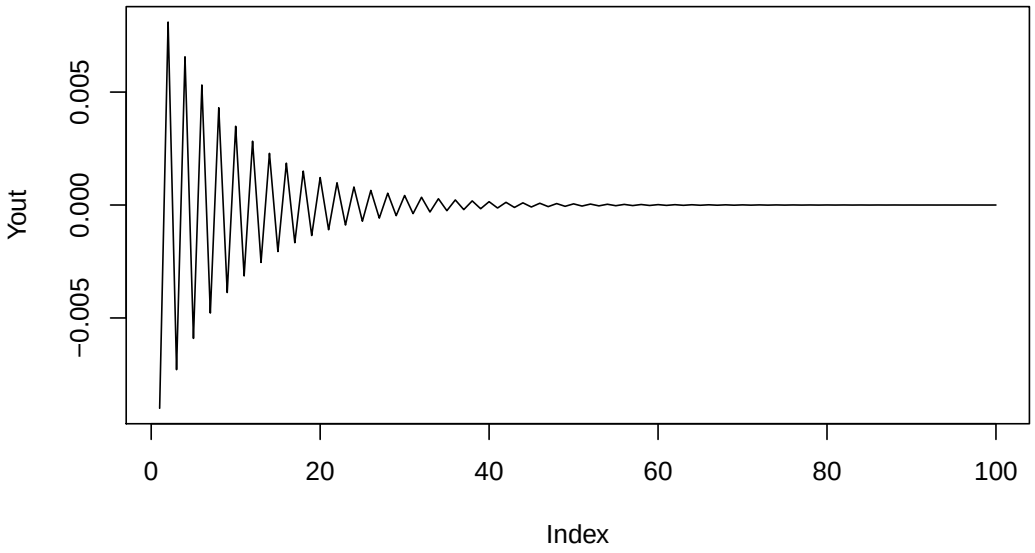
We'll use the linear map $Y_{i+1} = r * Y_i$ as an example and show three different ways to implement iterative processes:

1. The `for...` loop
2. The `-ply` family of functions
3. User defined `function()` with arguments

```
# for loop
N  <- 100
r  <- -.9
Y0 <- 0.01
Y  <- c(Y0,rep(NA,N-1))

for(i in 1:(N-1)){
  Y[i+1] <- r*Y[i]
}
plot(Y,type = "l")
```

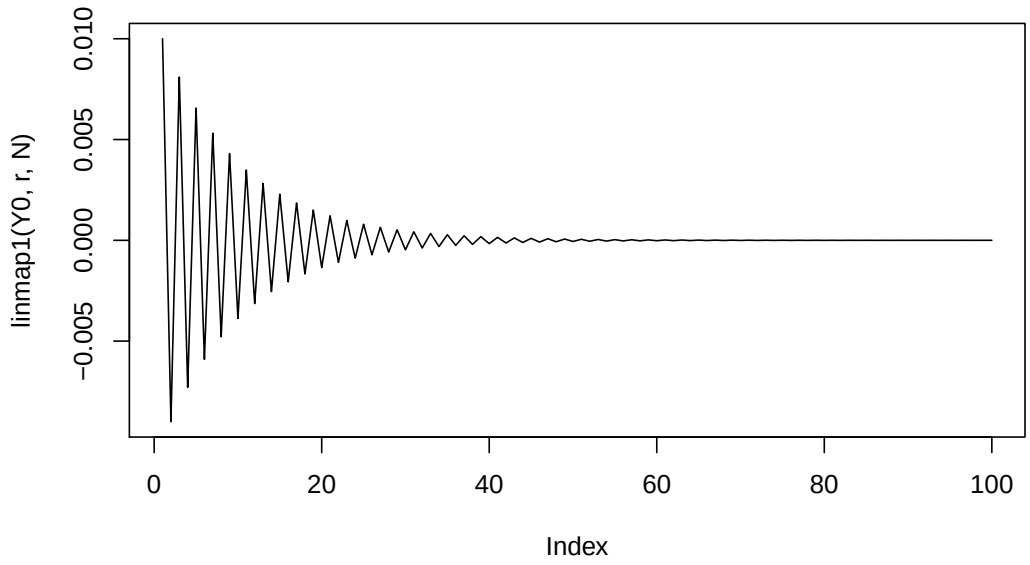



```r
# -ply family: sapply
Yout <- sapply(seq_along(Y),function(t) r*Y[t])
plot(Yout,type = "l")
```




```r
# function with for loop
linmap1 <- function(Y0,r,N){
  Y  <- c(Y0,rep(NA,N-1))
  for(i in 1:(N-1)){
```

```
    Y[i+1] <- r*Y[i]
  }
  return(Y)
}
plot(linmap1(Y0,r,N),type = "l")
```



3.4 Numerical integration to simulate continuous time

In order to 'solve' a differential equation for continuous time using a method of numerical integration, one could code it like in the spreadsheet assignment below. For R and Matlab there are so-called *solvers* available, functions that will do the integration for you. For R look at the Examples in package deSolve.

Euler's method and more...

The result of applying a method of numerical integration is called a **numerical solution** of the differential equation. The **analytical solution** is the equation which will give you a value of Y for any point in time, given an initial value Y_0. Systems which have an analytical solution can be used to test the accuracy of **numerical solutions**.

Analytical solution

Remember that the analytical solution for the logistic equation is:

$$Y(t) = \frac{K * Y_0}{Y_0 + (K - Y_0) * e^{-r*t}}$$

This can be 'simplified' to

$$Y(t) = \frac{K}{1 + \left(\frac{K}{Y_0 - 1}\right) * e^{-r*t}}$$

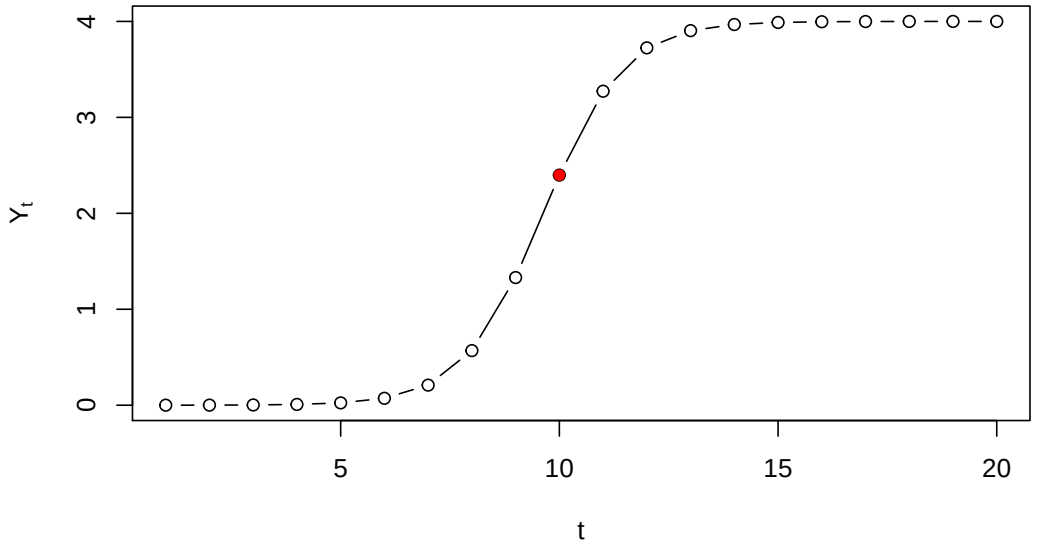
If we want to know the growth level Y_t at $t = 10$, with $Y_0 = .0001$, $r = 1.1$ and $K = 4$, we can just fill it in:

```
# Define a function for the solution
logSol <- function(Y0, r, K, t){K/(1+(K/Y0-1)*exp(-r*t))}

# Call the function
logSol(Y0=.0001, r=1.1, K=4, t=10)
```

```
> [1] 2.398008
```

We can pass a vector of time points to create the exact solution, the same we would get if we were to iterate the differential/difference equation.

```
# Plot from t=1 to t=100
plot(logSol(Y0=.0001, r=1.1, K=4, t=seq(1,20)), type = "b",
     ylab = expression(Y[t]), xlab = "t")
# Plot t=10 in red
points(10,logSol(Y0=.0001, r=1.1, K=4, t=10), col="red", pch=16)
```



Numerical solution (discrete)

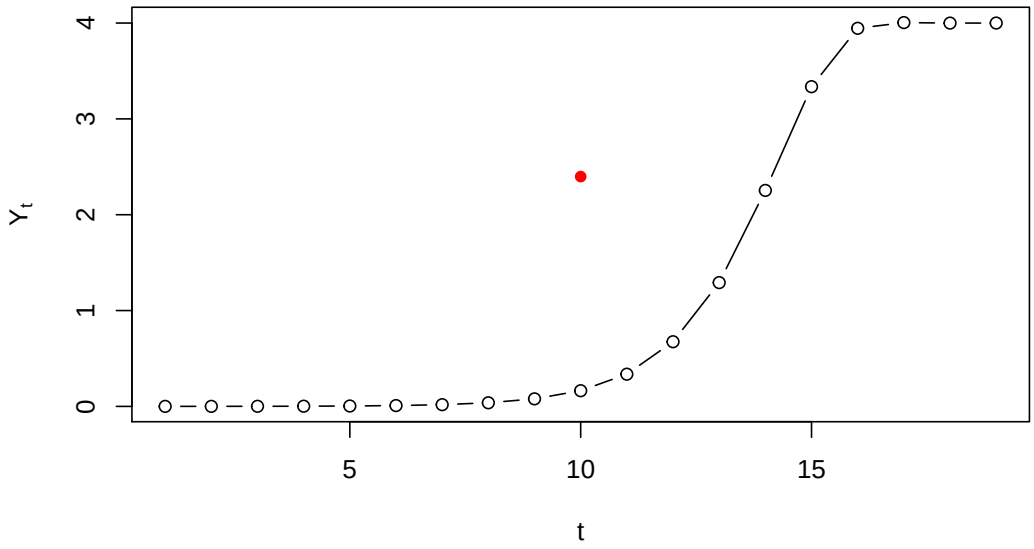
If we would iterate the differential equation ...

$$\frac{dY}{dt} = Y_t * (1 + r - r * \frac{Y_t}{K})$$

... as if it were a difference equation, we are *not* simulating continuous time, but a discrete time version of the model:

$$Y_{i+1} = Y_i * (1 + r - r * \frac{Y_i}{K})$$

```
logIter <-  function(Y0,r,K,t){
  N <- length(t)
  Y <- as.numeric(c(Y0, rep(NA,N-2)))
  sapply(seq_along(Y), function(t){ Y[[t+1]] <<- Y[t] * (1 + r - r * Y[t] / K)})
  }

# Plot from t=1 to t=100
plot(logIter(Y0=.0001, r=1.1, K=4, t=seq(1,20)), type = "b",
     ylab = expression(Y[t]), xlab = "t")
# Plot t=10 in red
points(10,logSol(Y0=.0001, r=1.1, K=4, t=10), col="red", pch=16)
```



3.4.1 Euler vs. Runge-Kutta

The method developped by Runge and Kutta takes a harmonic mean over a number of points, R-K4 takes 4 points, R-K6 takes 6, but there are many more variants.

Here's an exampkle with **Predator-Prey dynamics** comparing Euler's method to R-K4.

```
library(plyr)
library(tidyverse)
library(lattice)
```

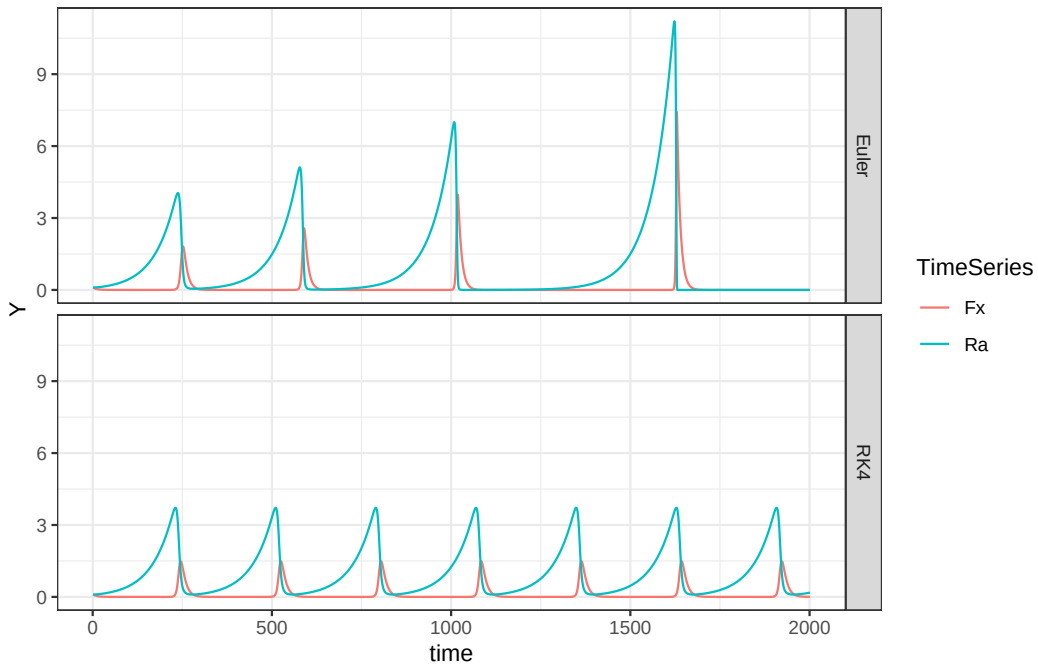
```r
# Lotka-Volterra Euler
lvEuler <- function(R0,F0,N,a,b,c,d,h){

  # Init vector
  Ra <- as.numeric(c(R0, rep(NA,N-1)))
  Fx <- as.numeric(c(F0, rep(NA,N-1)))

  for(t in 1:N){
  # Euler numerical solution of the predator-prey model
  Ra[t+1] <- Ra[t] + (a - b * Fx[t]) * Ra[t] * h
  Fx[t+1] <- Fx[t] + (c * Ra[t] - d) * Fx[t] * h
  }

  return(data.frame(time=1:NROW(Ra),Ra=Ra,Fx=Fx,method="Euler"))
}

# Lotka-Volterra Runge Kutta 4
lvRK4 <- function(R0,F0,N,a,b,c,d,h){

  # Init vector
  Ra <- as.numeric(c(R0, rep(NA,N-1)))
  Fx <- as.numeric(c(F0, rep(NA,N-1)))

  for(t in 1:N){
  # RK4 numerical solution of the predator-prey model
  k1_R=(a - b * Fx[t]) * Ra[t]
  k1_F=(c * Ra[t] - d) * Fx[t]

  k2_R=(a - b * (Fx[t]+h*k1_F/2)) * (Ra[t]+h*k1_R/2)
  k2_F=(c * (Ra[t]+h*k1_R/2) - d) * (Fx[t]+h*k1_F/2)

  k3_R=(a - b * (Fx[t]+h*k2_F/2)) * (Ra[t]+h*k2_R/2)
  k3_F=(c * (Ra[t]+h*k2_R/2) - d) * (Fx[t]+h*k2_F/2)

  k4_R=(a - b * (Fx[t]+h*k3_F)) * (Ra[t]+h*k3_R)
  k4_F=(c * (Ra[t]+h*k3_R) - d) * (Fx[t]+h*k3_F)
```

```
  # Iterative process
  Ra[t+1] <- Ra[t] + (1/6)*h*(k1_R+2*k2_R+2*k3_R+k4_R)
  Fx[t+1] <- Fx[t] + (1/6)*h*(k1_F+2*k2_F+2*k3_F+k4_F)
  }

  return(data.frame(time=1:NROW(Ra),Ra=Ra,Fx=Fx,method="RK4"))
}
```

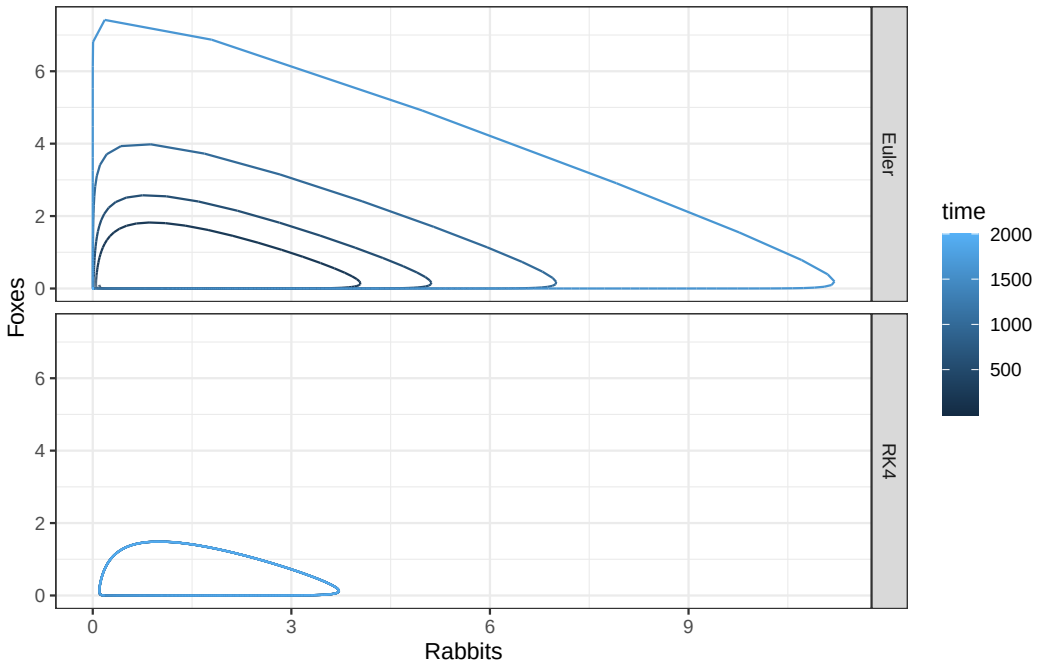
Now that we have the fuctions, we'll plot the numerical solutions for the same set of parameters. The continuous mathematics (= if you do some calculations to find the fixed points of the system) ensure us that the system should be in an equilibrium state in which the populations keep going around in the same cycle of growth and collapse. Let's see what happens…

```
# Parameters
N  <- 2000

# Equilibrium
a  <- 1/6
b  <- 4/3
c  <- d  <- 1
R0 <- F0 <- 0.1

# Time constant
h <- 0.1

# Get the results
pp1 <- lvEuler(R0,F0,N,a,b,c,d,h)
pp2 <- lvRK4(R0,F0,N,a,b,c,d,h)

# Make a long dataframe
pp <- rbind(pp1,pp2)

pp.long <- pp %>%
  gather(key = TimeSeries, value = Y, -c("time","method"))

# Time series plots
ggplot(pp.long, aes(x=time,y=Y,colour=TimeSeries)) +
```

```
  geom_line() +
  facet_grid(method~.) +
  theme_bw()
```




```
# Phase plane plots
ggplot(pp, aes(x=Ra,y=Fx,colour=time)) +
  geom_path() +
  facet_grid(method~.) +
  xlab("Rabbits") + ylab("Foxes") +
  theme_bw()
```



Using the Euler method predator and prey populations do not 'die out', but in phase space they seem to occupy different behavioural regimes. This looks like an unstable periodic orbit, or an unstable limit cycle, but it is in fact caused by the inaccuarcy of Euler's method. Here *RK4* clearly outperforms *Euler*.

3.5 Deterministic Chaos

``The study of things that look random -but are not''

---E. Lorenz

[This Google sheet]((https://docs.google.com/spreadsheets/d/1xaZJLYfZzkxg_PXCAa7QT7p4R_JTnUIxzs_Qs1-RAOQ/edit?usp=sharing) displays the extreme sensitivity to changes in initial conditions displayed by the Logistic Map for specific parameter settings. This specific phenomenon is more commonly referred to as **The Butterfly Effect**. It is a characteristic of a very interesting and rather mysterious behaviour displayed by deterministic dynamical equations known as **deterministic chaos**.

$$Y_{i+1} = rY_i\,(1-Y_i)$$

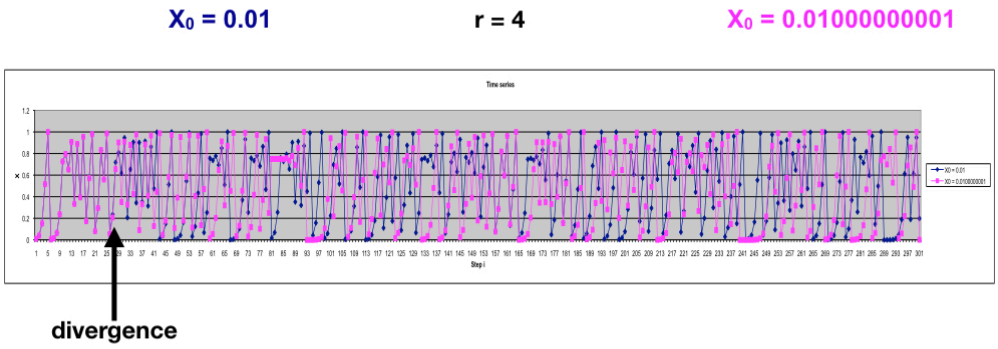


Figure 3.2 – Butterfly Effect

The Logistic Map is the simplest nontrivial model that can display deterministic chaos. For continuous time models, one needs at least 3 coupled differential

equations to generate chaotic behavior. There is no clear definition of deterministic chaos, but there are at least 4 ingredients: The dynamics are **a-periodic**, **bounded** and **sensitively depend on initial conditions**, moreover, the system generating these dynamics is **deterministic**.

The video below by YouTuber Veritasium does an amazing job at explaining what deterministic chaos is based on the famous Lorenz system.

The science behind the butterfly effect

3.6 Modeling interactions between processes and agents

3.6.1 The Competetive Lottka-Volterra Equations

The coupled predator-prey dynamics in the previous assignment are not a very realistic model of an actual ecological system. Both equations are exponential growth functions, but **R**abbits for example, also have to eat! One way to increase realism is to consider coupled logistic growth by introducing a carrying capacity.

- Follow the link to this Wiki page and try to model the system!

> This is what *interaction dynamics* refers to, modeling mutual dependencies using the `if ... then` conditional rules isn't really about interaction, or coupling between processes.

3.6.2 Predator-Prey (and other) dynamics as Agent Based Models

Agent-Based models are an expansion of the idea of "connected growers" that includes a spatial location of the things that is subject to change over time.

Have a look at some of the NETlogo demo's:

- Rabbits Weeds Grass
- Wolf Sheep Grass

3.6.3 The dynamic field model

Probably the most impressive modelling example in developmental psychology is the Dynamic Field Model for infant perservative reaching, also known as the *A-not-B error*:

- Thelen, E., Schöner, G., Scheier, C., & Smith, L. (2001). The dynamics of embodiment: A field theory of infant perseverative reaching. Behavioral and Brain Sciences, 24(1), 1-34. doi:10.1017/S0140525X01003910

The model makes some very interesting predictions that have been confirmed and it has been generalized to other phenomena and scientific disciplines as well

- Smith, L. B., & Thelen, E. (2003). Development as a dynamic system. Trends in cognitive sciences, 7(8), 343-348.
- Schöner, G., & Thelen, E. (2006). Using dynamic field theory to rethink infant habituation. Psychological review, 113(2), 273.
- TWOMEY, K. E., & HORST, J. S. (2014). TESTING A DYNAMIC NEURAL FIELD MODEL OF CHILDREN'S CATEGORY LABELLING. In Computational Models of Cognitive Processes: Proceedings of the 13th Neural Computation and Psychology Workshop (pp. 83-94).

You can learn about it on the Dynamic Field Theory website centered around the book:

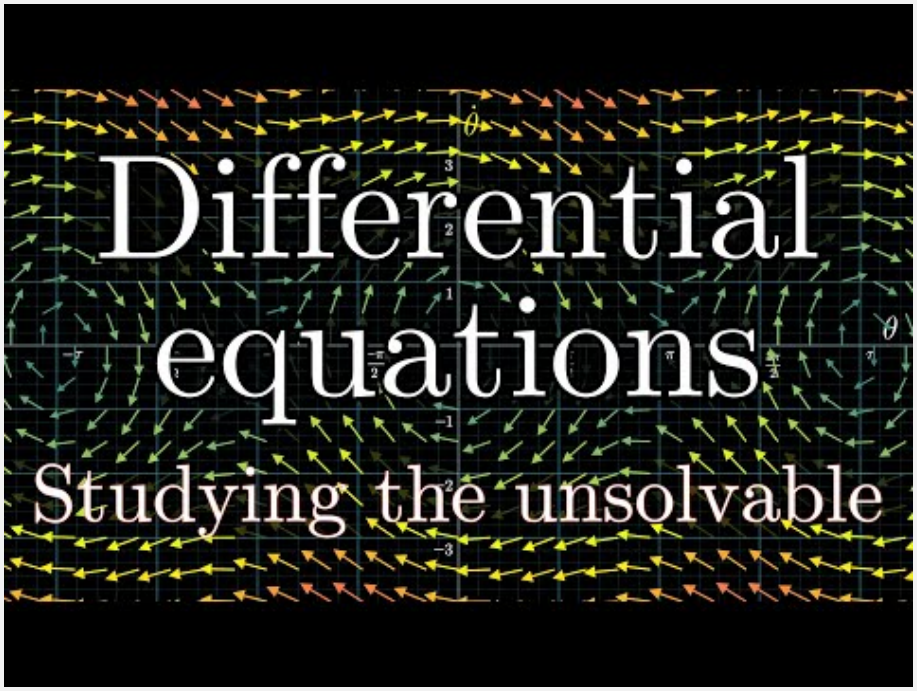
- Schöner, G., & Spencer, J. (2015). Dynamic thinking: A primer on dynamic field theory. Oxford University Press.

Study Materials and Resources

3.6.4 Differential equations

This is a very informative 5 part series of videos by 3Blue1Brown about differential equations.

Differential equations



Part III

Basic (Nonlinear) Time Series Analysis

Chapter 4

Basic Time Series Analysis

The main purpose of all time series analysis is to quantify patterns in time ordered data. Sometimes these patterns can be derived from by fitting statistical models of change processes to the data, but such analyses will not be the topic of this chapter We will focus on time series analyses that can provide hints about the nature of the data generating process, and not test whether some known model will fit the data.

Most of the techniques we'll discuss can be considered nonlinear time series analyses, we'll start though by looking at the most important linear tools for studying temporal patterns (correlation functions).

First, here is some very important advice you should consider before you get started with time series analysis…

4.1 Always plot your data!

Meet *the data saurus dozen*, a stark reminder to always plot your data and never rely on summary statistics only!

https://d2f99xq7vri1nk.cloudfront.net/DinoSequentialSmaller.gif

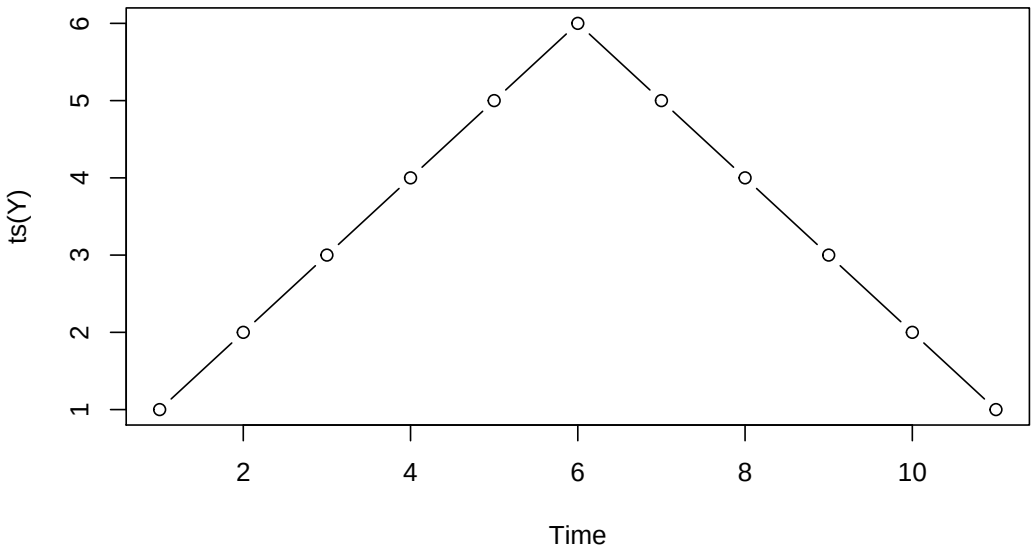
> Justin Matejka, George Fitzmaurice (2017) Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing CHI 2017 Conference proceedings: ACM SIGCHI Conference on Human Factors in Computing Systems

4.2 Correlation Functions

Correlation functions are intuitive tools for quantifying the temporal structure in a time series. As you know, the correlation measure can only quantify linear regularities between variables, which is why we discuss them here as `basic` tools for time series analysis. So what are the variables? In the simplest case, the variables between which we calculate a correlation are between a data point at time *t* and a data point that is separated in time by some *lag*, for example, if you would calculate the correlation in a lag-1 return plot, you would have calculated the 1st value of the correlation function (actually, it is 2nd value, the 1st value is the correlation of time series with itself, the lag-0 correlation, which is of course $r = 1$).

Suppose we have a time series $Y_i = 1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1,$

```
Y <- c(1,2,3,4,5,6,5,4,3,2,1)
plot(ts(Y),type="b")
```



We can create the pairs of lagged values, here we'll study lags from `0` to `4`:

Y

lag0

lag1

lag2

lag3

lag4

1

1

2

3

4

5

2

2

3

4

5

6

3

3

4

5

6

5

4

4

5

6

5

4

5

5

6

5

4

3

6

6

5

4

3

2

5

5

4

3

2

1

4

4

3

2

1

NA

3

3

2

1

NA

NA

2

2

1

NA

NA

NA

1

1

NA

NA

NA

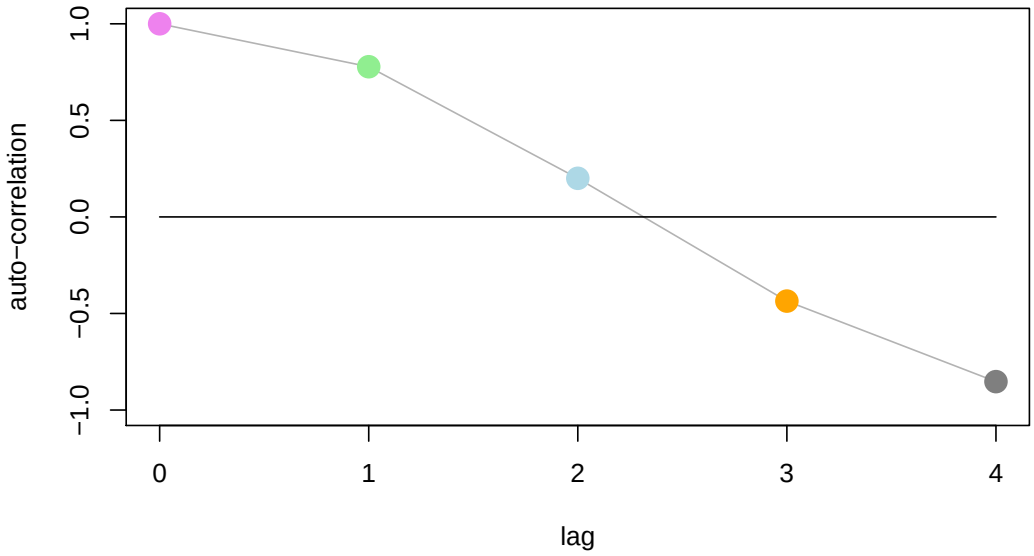
NA

Now we can simply calculate the correlation for each pair of Y with a lagged version of Y. This is the auto-correlation, because we are basically comparing Y with itself, just after some lag of time has passed.

```
(rlag0 <- cor(Y,Y))
> [1] 1
(rlag1 <- cor(Y[1:10],Y[2:11]))
> [1] 0.7777778
(rlag2 <- cor(Y[1:9],Y[3:11]))
> [1] 0.2
```

```
(rlag3 <- cor(Y[1:8],Y[4:11]))
> [1] -0.4358974
(rlag4 <- cor(Y[1:7],Y[5:11]))
> [1] -0.8529412
```

We can plot these correlations to create the so-called *autocorrelation function* or *ACF*.



The ACF shows a pattern indicating values separated by a step of 1 are positively correlated (of course, at a lag of 0 the correlation is 1). At lag 4 the correlation is negative, if you look at the plot of the time series you can see that for many time steps the values will be on opposite sides of the peak.

We can also decide whether the correlations deviate significantly from 0. If they do, this can be an indication of 'memory', or interdependence: There could be patterns in the data that are recurring with a particular frequency.

In Figure **??** the ACF and the *partial* ACF of a sine wave are shown (using function `plotRED_acf()`). The partial auto correlation function, 'partials out' the correlation that is known from the previous lag and displays the unique correlation that exists between data points seperated by the lags.

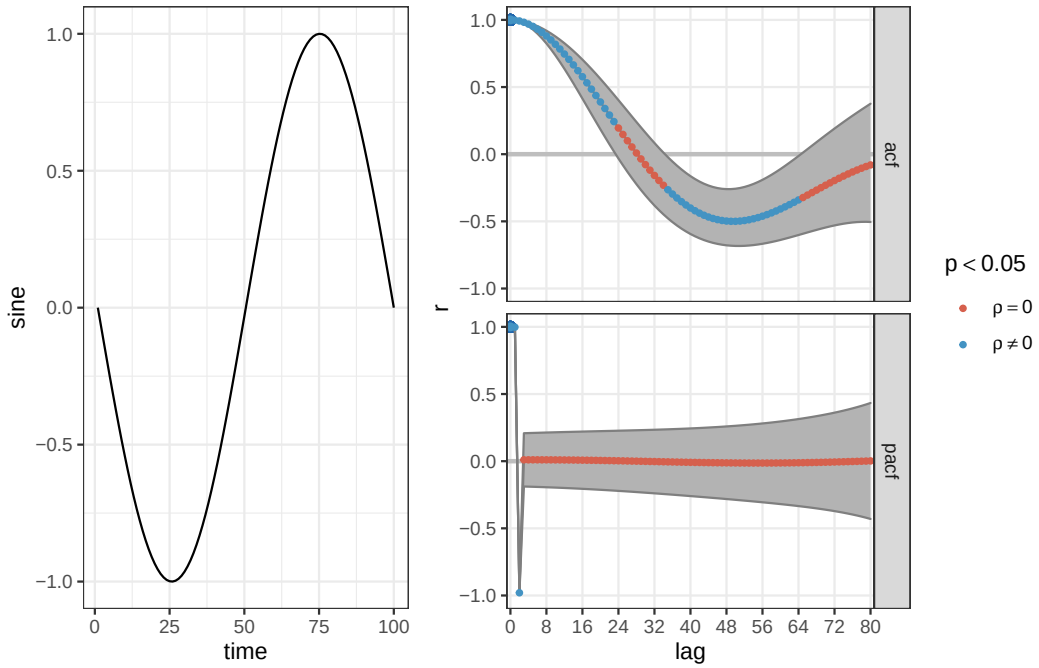


Figure 4.1 – (P)ACF of a sine wave

4.3 Autoregressive models

In this book we will not discuss the type of linear time series models known as Autoregressive Models (e.g. AR, ARMA, ARiMA, ARfiMA) summarised on this Wikipedia page on timeseries. There are many extensions to these linear models, check the `CRAN Task View` on `Time Series Analysis` to learn more (e.g. about package `zoo` and `forecast`).

We will in fact be discussing a lot of methods in a book the Wiki page refers to for *'Further references on nonlinear time series analysis'*: **Nonlinear Time Series Analysis** by Kantz & Schreiber. You do not need to buy this book, to understand what follows, but it can be a helpful reference if you want to go beyond the formal level (= mathematics) used in the CSA-book Some of the packages we use are based on the accompanying software to the book **TiSEAN** which is written in `C` and `Fortran` and can be called from the command line (Windows / Linux).

Chapter 5

Basic Nonlinear Time Series Analysis

Many nonlinear analyses can be considered "descriptive" techniques, that is, the aim is not to fit the parameters of a model, but to describe, quantitatively, some aspects of how one value changes into another value over time.

5.0.1 Intuitive notion of Fractal Dimension

You are probably familiar with the notion of *dimension*, that is, the notion displayed in Figure **??** a point has dimension 0, a line dimension 1, a plane dimension 2 and a cube dimension 3.

https://upload.wikimedia.org/wikipedia/commons/4/45/Dimension_levels.svg

The fractal dimension of an object indicates how much it 'spills over' into the next integer dimension. Mathematician Benoît B. Mandelbrot used it to (formally) study the `roughness` of the world, which was quite revolutionary in mathematics where everything is neat and smooth.

A qualitative description of the fractal dimension of a time series (or 1D curve) can be given by deciding whether the curve looks/behaves like a line, or, like a plane.

As can be seen in the Figure **??**, if slow processes (low frequencies) dominate the signal, they are more *line-like* (bottom of the figure) and will have a fractal dimension closer to `1`. If fast processes (high frequencies) dominate the signal, they are more *plane-like* and will have a fractal dimension closer to `2`.

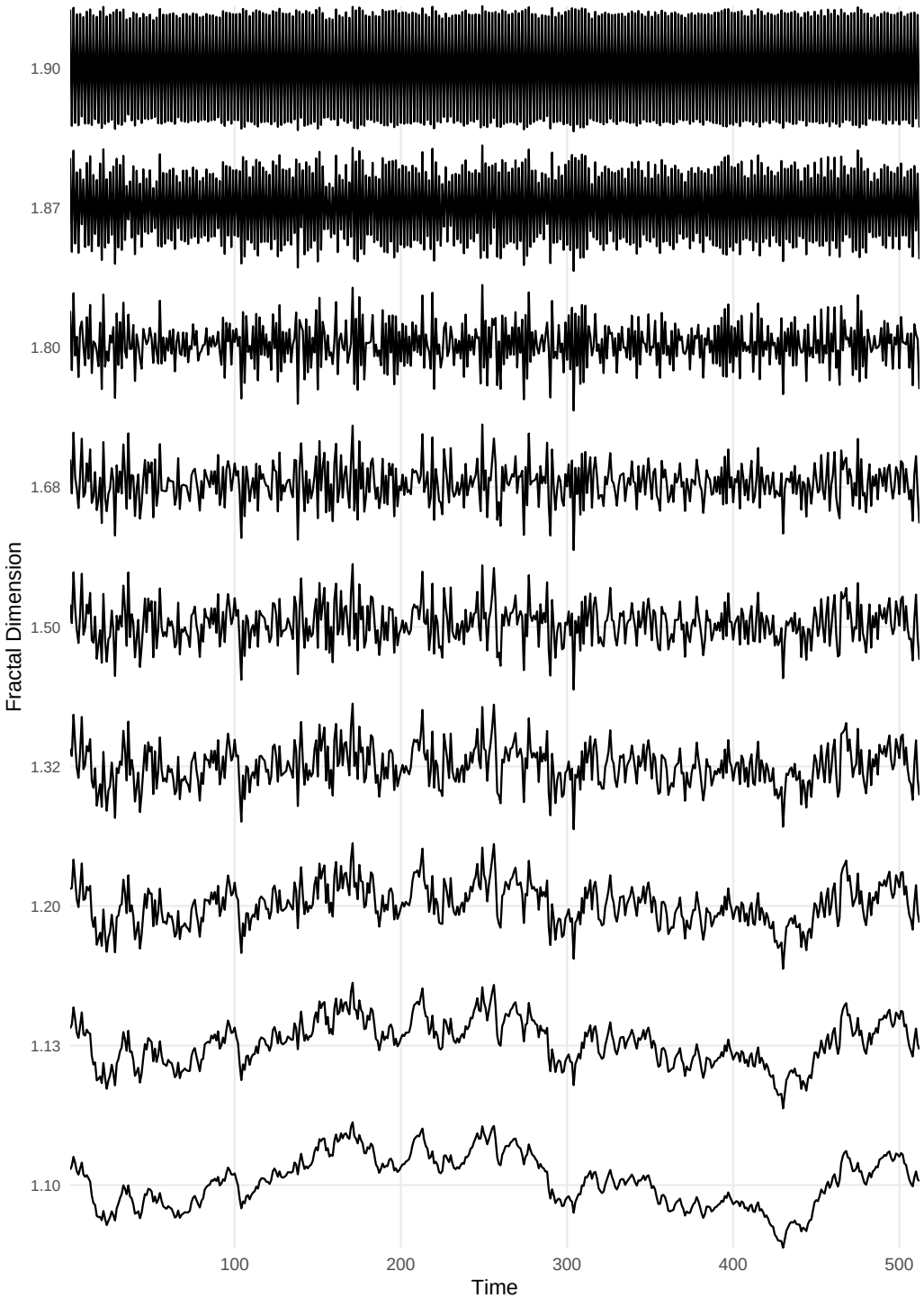
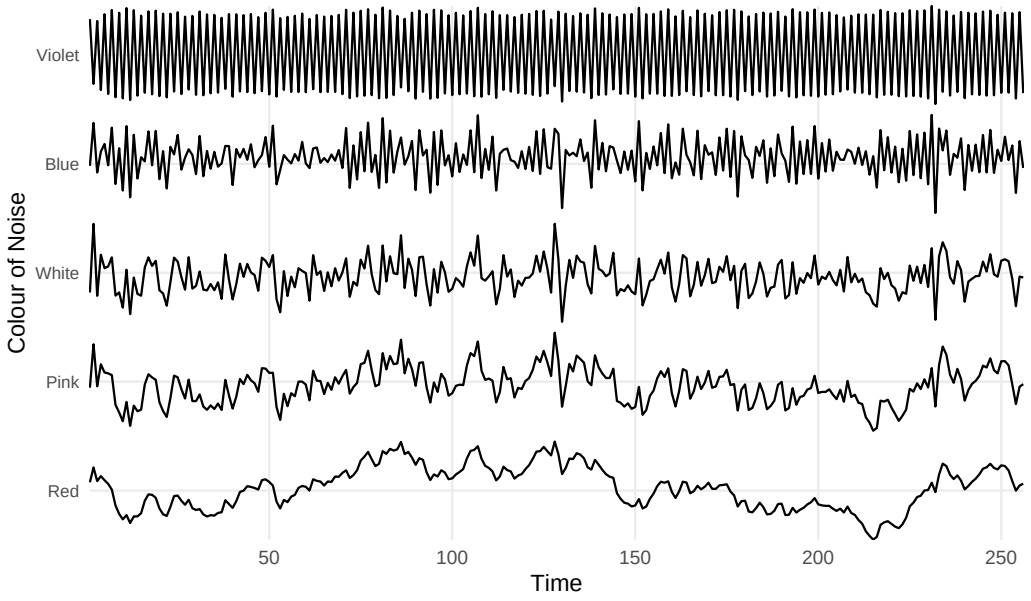


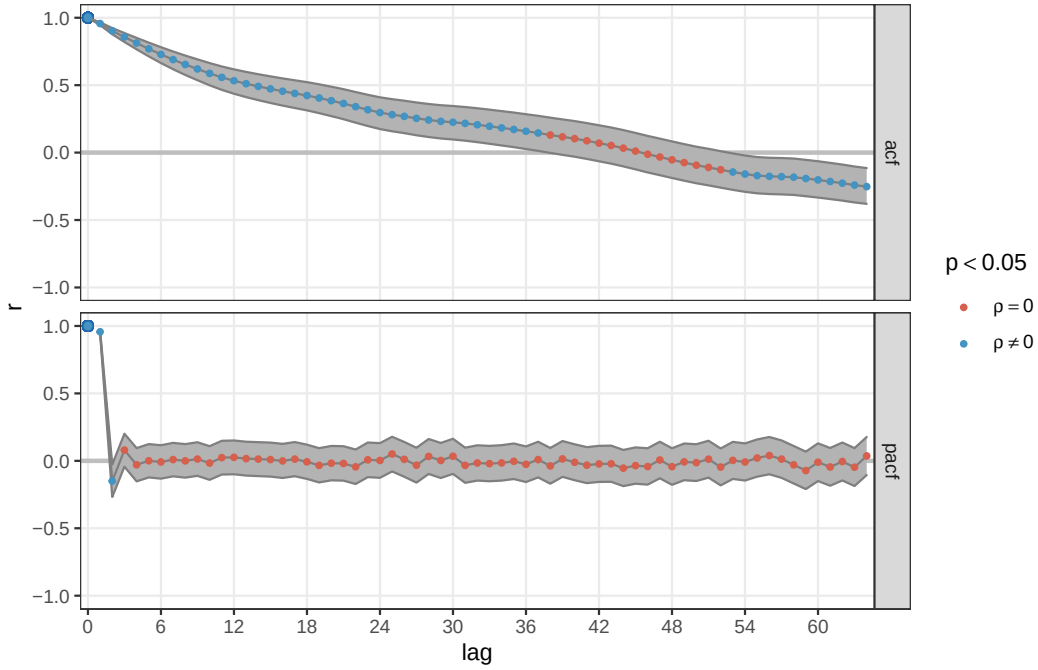
Figure 5.1 – The fractal dimension of different types of change processes

5.0.2 Coloured noise

You can also think of these different types of time-series as being generated by different kinds of processes, they have a different type of correlational structure, what varies is at which time scales the largest correlations can be found. So let's study the correlation functions of these noises, which are often denoted by a colour based on the [colours of the visible light spectrum](https://en.wikipedia.org/wiki/Colors_of_noise.

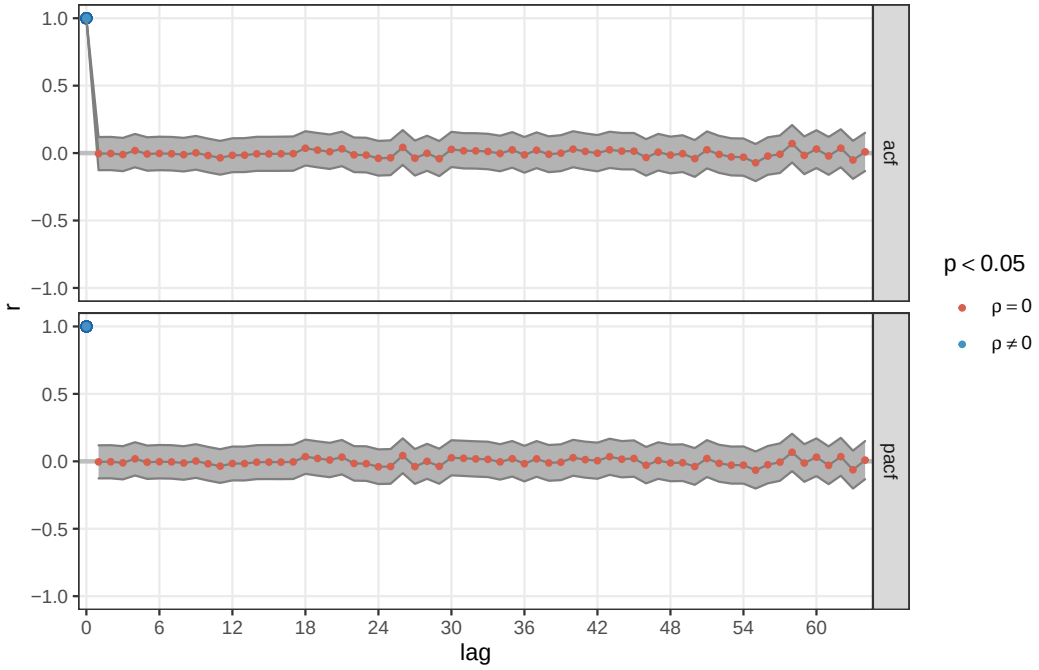


5.0.2.1 Brownian (red) noise



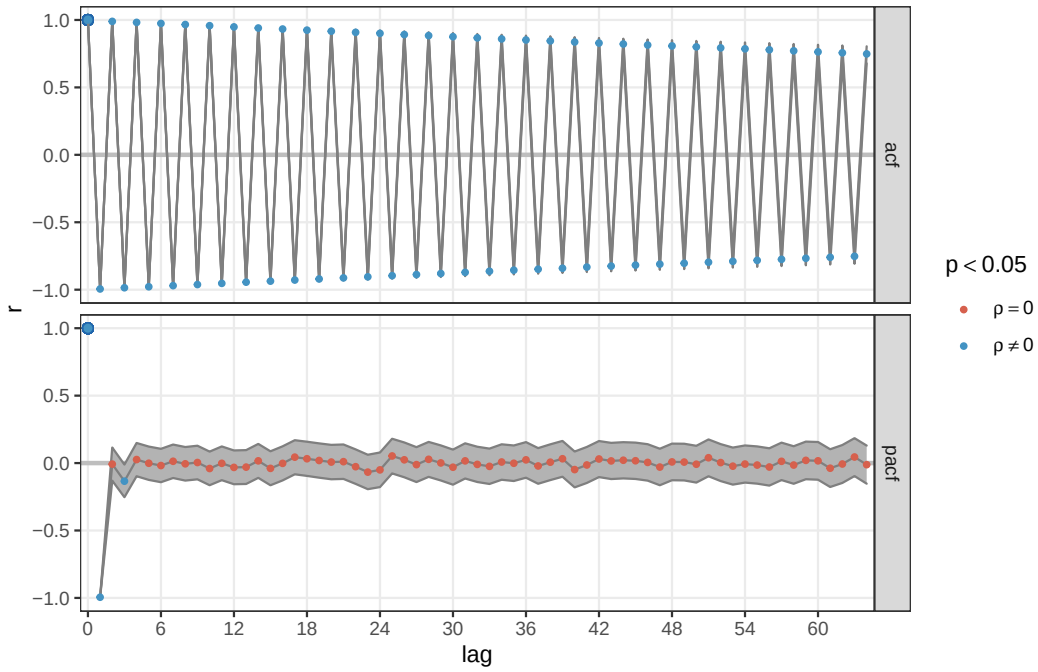
Persistent pattern, correlations exist at all lags, but in the partial acf at lag 1 $r \approx 1$.
Brownian noise is the cumulative sum of white noise, a random walk.

White noise



No correlations at any lag (except lag 0, where $r = 1$ of course).

Violet noise

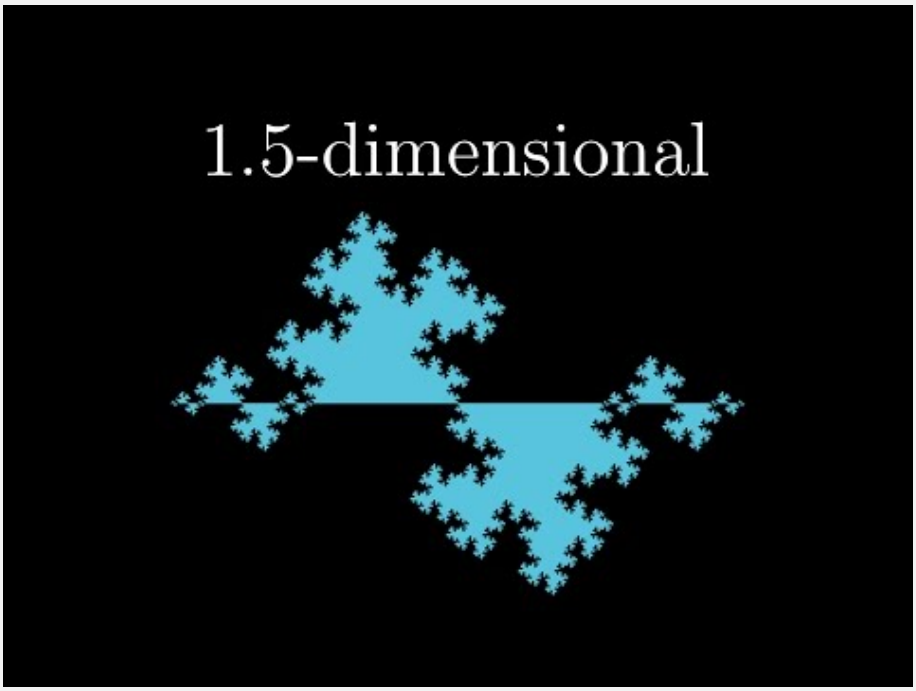


Anti-persistent pattern, correlations exist at all lags, but they change sign. In the partial acf at lag 1 $r \approx -1$.

Fractals

You have probably heard about *fractals* as being objects that are made out of little copies of the whole, self-similar objects. The video below provides a good introduction to what fractals are and what they are not, what the *fractal dimension* actually means and it introduces the concept of **roughness*, scaling* and *box-counting dimension*. We will eventually discuss all these topics, for now it is ok to stop after about 10 minutes.

Most fractals are not self-similar


1.5-dimensional

5.1 Relative Roughness

Relative Roughness evaluates the local variability relative to the global variability in a time series. It is calculated using the following formula:

$$RR = 2 * \left[1 - \frac{\gamma_1(x_i)}{Var(x_i)}\right] \tag{5.1}$$

The numerator in the formula stands for the `lag 1` auto-covariance of the time series x_i, this is the unstandardised lag1 autocorrelation (auto-covariance). The denominator stands for the (global) variance of x_i which all statistics packages can calculate. Another way to describe the global variance is: `lag 0` auto-covariance.

You can use Figure **??** to lookup which value of RR corresponds to which fractal dimension. The line-like time series will have low local variance and high global variance and therefore the Relative roughness will be small. The plane-like time series will have high local variance and low global variance, resulting in a high value for Relative Roughness.

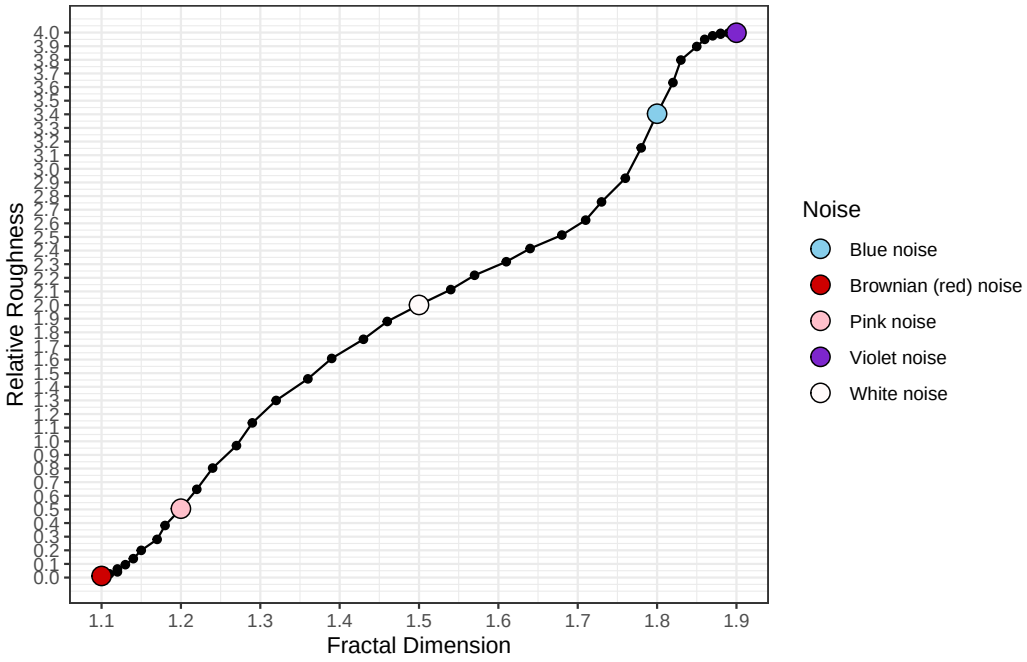


Figure 5.2 – Coloured Noise versus Relative Roughness

5.2 Entropy

Entropy is an important topic, but also often a misunderstood concept. See the *Study Materials and Resources* of this chapter for some excellent explanations of this quantity. Here we provide an explanation of Entropy based on its meaning in information theory.

Physical Information and Entropy

Classical, algorithmic and quantum information theory explicitly exclude meaningful, or, semantic information as part of their explanatory domain (cf. Desurvire, 2009, p. 38), or, as Shannon lucidly explained:

> ``The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is, they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.'' (Shannon, 1948, p. 379, emphasis in original)

The semantic aspects of a message are irrelevant for reproducing it, if they were relevant for message reproduction, a universal theory of information and communication would not be possible. In order to successfully communicate a message through a channel if semantics were involved, one would have to know particular facts about these correlations with "*some system with certain physical or conceptual entities*". What then is information?

In the most general terms, information can be defined as a quantity that resolves uncertainty about the state of an information source, a physical entity that can represent an amount of information. The information-theoretic quantity known as self-information, or information content (I) quantifies the reduction of uncertainty due to the observation of a particular state of the information source. For example, an information source that can be in 2 states, a fair coin, can represent 2 bits of information (2 things to be uncertain about). Observing 1 of the 2 possible

states ('heads') resolves uncertainty about the state of the system by an amount of 1 bit of information.

Another important information-theoretic quantity is entropy (H), which can be interpreted as the expected value of I over repeated observations of states of the information source. For a fair coin, the 2 possible states are equiprobable, so the expected amount of information represented by the observation either 'heads' or 'tails' will be 1. Figure **??** displays the relation between the information and entropy represented by each state for different configurations of the coin: Biased towards 'Tails' ($Pr(X = Heads) < .5$); fair ($Pr(X = Heads) = .5$); biased towards 'Heads' ($Pr(X = Heads) > .5$). The dotted lines show the Information represented by the observation of $X = Heads$ or $X = Tails$. The entropy is at its maximum $H = 1\ bit/symbol$ when the system is configured to be fair, In that case the observation of each state would represent the same amount of information ($I = 1$). in which case the observation of a particular state does not provide any information about the state that will be observed next, our best guess, is quite literally, to guess.

The maximum entropy state represents maximum unpredictability, in the case of a physical system one would refer to the maximum entropy state as a state of maximum disorder. For now, it is important to note that information is dimensionless in the sense that it does not represent a characteristic scale of observation, but has to be interpreted relative to the scale at which the microscopic states of the system are observed. If a computer scientist and a particle physicist have to estimate the amount of information that is represented by a 1GB memory chip, they will give two different, but correct answers (2^{33} and $\approx\ 2^{76}$ bits, respectively). This is due to the scale at which the degrees of freedom of the information source are identified. The computer scientist cares about the on and off states of the transistors on the chip, whereas the particle physicist is more likely interested in the states of billions of individual particles (example taken from Bekenstein, 2003, p. 59).

A user of a 1GB flash drive commonly wants to know whether there is enough free space to copy one or more files onto it. The amount of free space, expressed as a quantity of information, for example, 100MB, refers to the degrees of freedom at the level of the on and off states of the transistors on the chip that are still available to become "*correlated according to some system with certain physical or conceptual entities*". We will refer to such degrees of freedom as *non-information bearing d.o.f.*, or, *non-informative structure*, as opposed to *information-bearing*

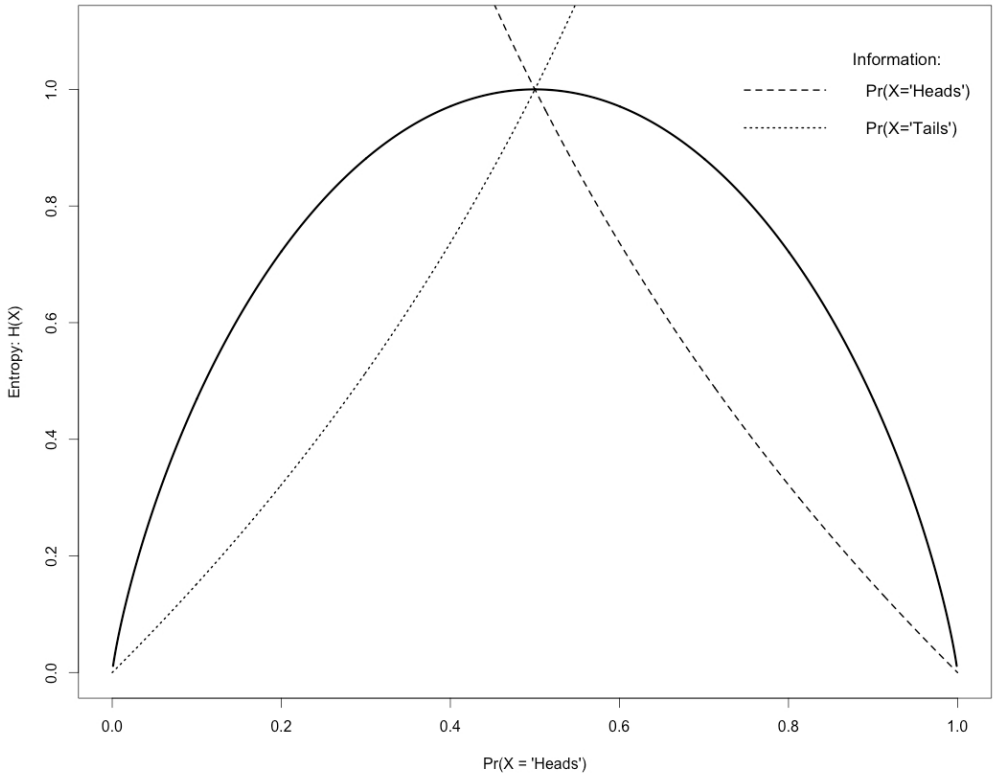


Figure 5.3 – Information and Entropy of a coin system.

d.o.f., or, *informative structure*. The latter would concern those transistors whose state is temporarily fixed, because they are part of a specific configuration of transistor states that is associated with one or more systems of physical or conceptual entities (i.e. the state encodes for whatever digital content was stored on the drive). The non-informative structure concerns transistors whose state is not fixed, and would be available to become associated to some other system of entities. Another way to describe this situation, is that there has been a (relative) reduction of the total degrees of freedom available to the system, from $8,589,934,592$ (1GB) to $838,860,800$ (100MB) on/off states. The informative structure represents systematicity, certainty, order: "order is essentially the arrival of redundancy in a system, a reduction of possibilities" (von Foerster, 2003). The non-informative structure represents possibility, uncertainty, disorder.

These quantities of information can be used to distinguish one system state relative to another (e.g. less or more redundancy/possibility, order/disorder, unavailable/available storage space), and this can lead to the emergence of identity (cf. Kak, 1996). However, this is not the same as the emergence of meaning, that is, it the quantity does not account for the emergence of informative structure, which would require knowledge about the system with which the configuration of degrees of freedom became associated. The amount of information represented by an information source does not specify what it codes for, it is meaningless (cf. Küppers, 2013). To 'make use' of the meaning encoded in an information source one would need to be able learn the systemic regularities it codes for by observing how the redundancies came about, or, figure out how to translate the configuration into another, known code (an equivalent to the Rosetta stone). Compared to the concept of internal or mental representation, the physical representation of information is much more an indication of a capacity for registering an amount of information (Lloyd, 2006). It refers to a potential for establishing a new order in the configuration of the system by recruiting available degrees of freedom, whereas the mental representations the post-cognitivists seek to dispense with, refer to previously realized order that was somehow trapped, stored, or imprinted into the structure of the system.

Entropy in time series

To estimate the entropy of an observed time series, we need some method for deciding how predictable the series is, what are the redundancies in the series we can exploit in order to know what is going to happen next? We'll use a mea-

sure called Sample Entropy or *SampEn*, which can be interpreted as measuring the average amount of information that is needed in order to represent all the (patterns) of values observed in the time series. If we need lots of information, this means the time series is unpredictable, random and the SampEn will be high. If we need just a few bits of information, this means the time series is very predictable, deterministic and the SampEn will be low.

More formally: Sample entropy is the negative natural logarithm of the conditional probability that a dataset of length N, having repeated itself within a tolerance r for m points, will also repeat itself for $m + 1$ points.

$P = \frac{distance\ between\ values\ in\ data\ segment\ of\ length\ m+1 < r}{distance\ between\ values\ in\ data\ segment\ of\ length\ m < r}$

$SampEn = -\log P$

Figure **??** can be used to lookup which value of `SampEn` corresponds to which type of dynamics. As expected, white noise has the highest SampEn, it is the most unpredictable type of noise. The other noises, towards (infra) red and (ultra) violet noise are characterised by increasingly dominant high and low frequency oscillations, which are redundancies, and they make the behaviour of the series predictable.

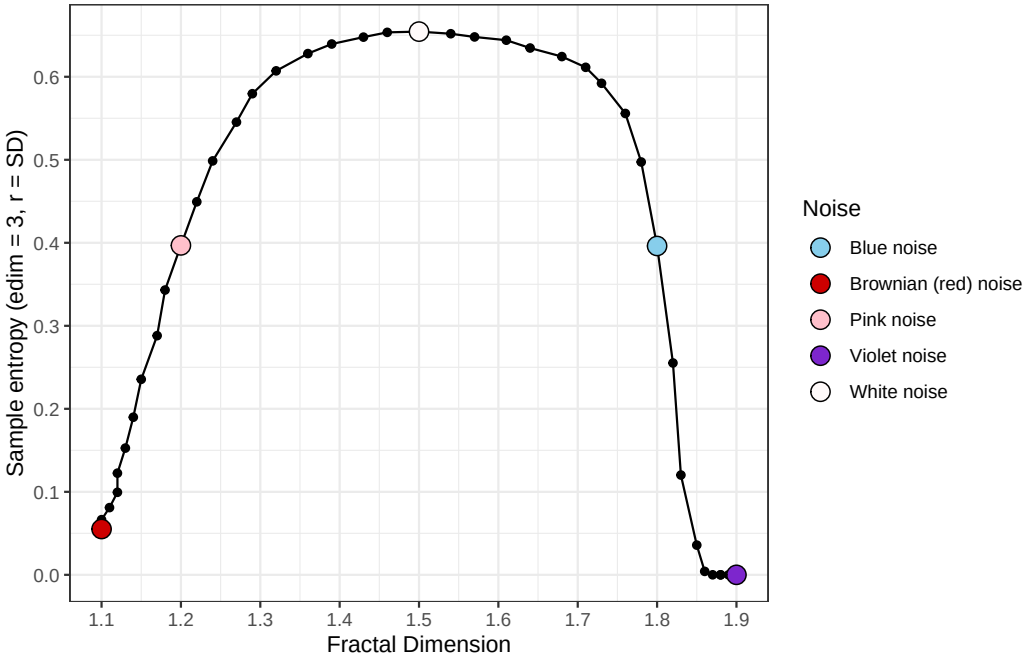


Figure 5.4 – Coloured Noise versus Sample Entropy

5.3 Other measures in *casnet*

Check the functions

fd_boxcount2D

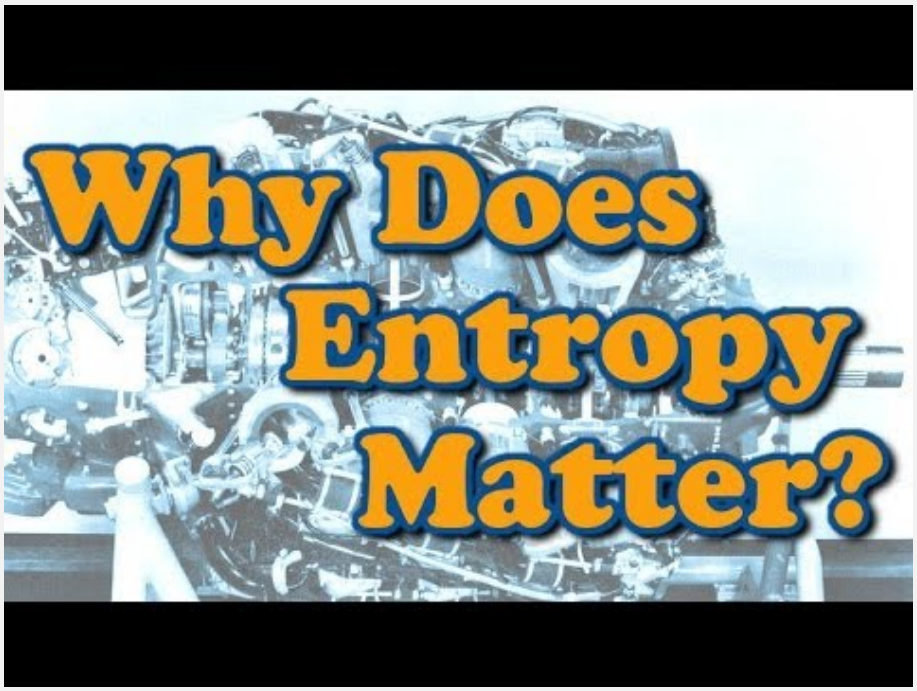
fd_allan

fd_sev

Study Materials and Resources

What is entropy?

Why does entropy even matter?



A better explanation of entropy

Part IV

Scaling Phenomena - Fluctuation Analyses

Chapter 6

Fluctuation Analyses: Global Scaling

> ``If you have not found the 1/f spectrum, it is because you have not waited long enough. You have not looked at low enough frequencies."
> - Machlup (1981)

As **?** noted, in order to find long range temporal correlations, you need to be able to observe them, that is, observe the process for a sufficiently long time. In general, variables measured from Complex Adaptive Systems will display long-range correlations (for examples in behavioural science see e.g. **?**, **?**, **?**, **?**, **?**, **?**). What the presence of such correlations, and specifically the pattern known as 1/f noise (pink noise), signifies is a matter of debate. This chapter lists some of the methods available in package `casnet` to quantify the presence of temporal patterns that can be associated to the different colours of noise discussed in the previous chapter. Most analysis outcomes can be converted to an estimate of the Fractal Dimension of the time series (see **?**, for rationale and conversion formula's).

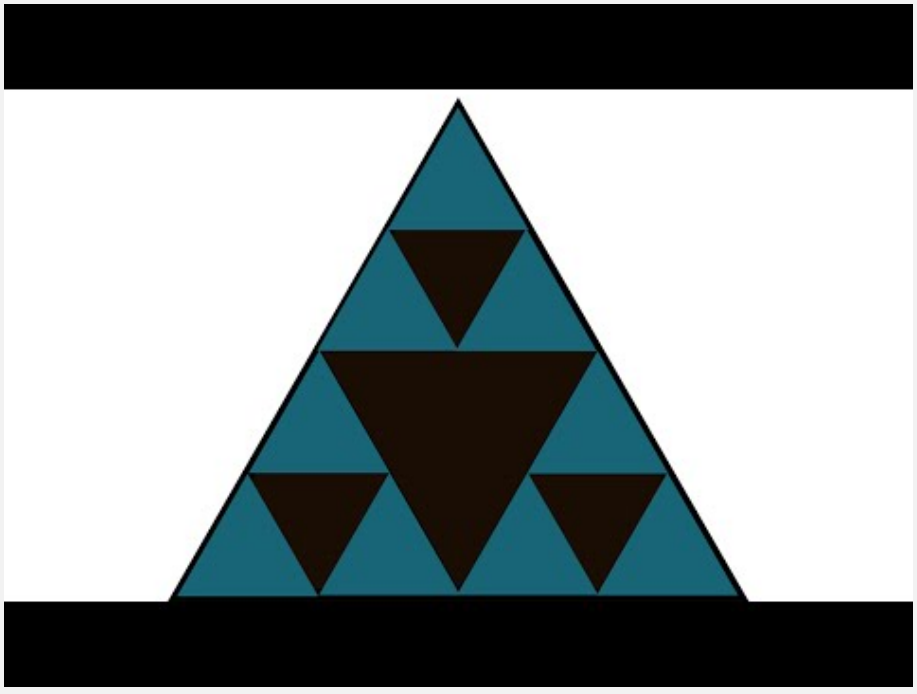
If you haven't done so already, it is recommended you study the materials listed in the paragraph **Fractals** of the previous chapter.

The presence of power-law scaling means that the observed time series (and potentially the data generating process) cannot be described by referring to a characteristic scale. This can be a scale of fluctuation (e.g. a population variance) or a central tendency (e.g. a population mean).

However, we do not know anything about the mechanism responsible for the emergence of the powerlaw. This is explained in the video linked below. Other useful resources are:

- Scaling laws in cognitive sciences. (**?**)
- A Review of Theoretical Perspectives in Cognitive Science on the Presence of 1/f Scaling in Coordinated Physiological and Cognitive Processes. (**?**)
- Fractal dynamics in physiology: Alterations with disease and aging. (**?**)
- Living in the Pink: Intentionality, Wellbeing, and Complexity. (**?**)

Fractals and Scaling: What do power laws mean?

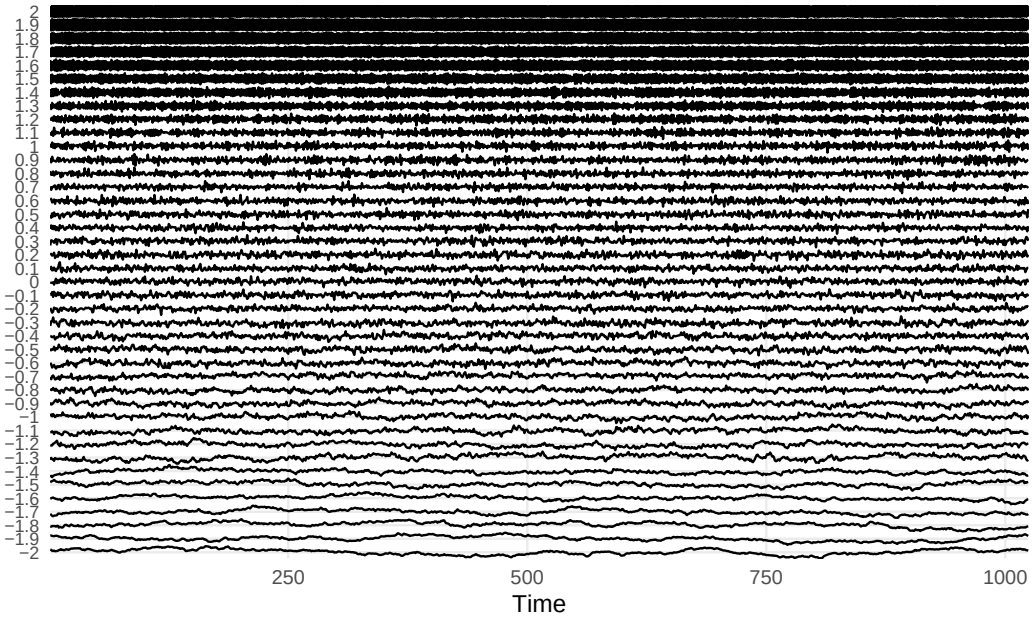


6.1 Power Spectral Density (PSD) slope

Power Spectral Density slope analysis first transforms the time series into the frequency domain by performing the Fourier transform. This breaks down the signal into sine and cosine waves of a particular amplitude that together "add-up" to represent the original signal. If there is a systematic relationship between the frequencies in the signal and the power of those frequencies (power = amplitude^2), this will reveal itself in log-log coordinates as a linear relationship. The slope of the best fitting line is taken as an estimate of the scaling exponent and can be converted to an estimate of the fractal dimension.

Package `casnet` contains a dataset called `ColouredNoise`, the column names represent the scaling exponent of the time series simulated with `noise_powerlaw()`.

```
plotTS_multi(ColouredNoise)
```



Data preparation usually involves the steps required for conducting the Fourier analysis. This includes centring on the mean (or standardising) and detrending the time series (the Fourier transform is a linear technique which assumes stationarity).

Calling `fd_psd()` will produce a summary output in the console. For all functions of the `fd_` family, two scaling exponents will be estimated, one based on the entire

range of data points that represent a potential powerlaw, and one based on a restricted range. In the case of the spectral slope, it is custom to fit only over the scaling region in the lower frequencies (for negative scaling exponents) or higher frequencies (for positive scaling exponents). See the `fitMethod` argument in the manual for an explanation of the options for selecting a range of frequencies.

```
psdN1 <- fd_psd(ColouredNoise$`-1`, returnPlot = TRUE, tsName = "Pink noise", noTitle = TRUE, do
```

```
>
>
> fd_psd:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Power Spectral Density Slope
>
>  All frequencies (n = 512)
> Slope = -1.06 | FD = 1.19
>
>  Hurvich-Deo (n = 81)
> Slope = -1.04 | FD = 1.19
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
psd0 <- fd_psd(ColouredNoise$`0`, returnPlot = TRUE, tsName = "White noise", noTitle = TRUE, do
```

```
>
>
> fd_psd:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Power Spectral Density Slope
>
>  All frequencies (n = 512)
> Slope = -0.12 | FD = 1.45
>
```

```
>  Hurvich-Deo (n = 34)
> Slope = -0.35 | FD = 1.37
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
psdP1 <- fd_psd(ColouredNoise$`1`, returnPlot = TRUE, tsName = "Blue noise", noTitle = TRUE, doPlot = FA
```

```
>
>
> fd_psd:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Power Spectral Density Slope
>
>  All frequencies (n = 512)
> Slope = 1.02 | FD = 1.8
>
>  Hurvich-Deo (n = 85)
> Slope = 0.82 | FD = 1.76
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
cowplot::plot_grid(psdN1$plot,psd0$plot,psdP1$plot,ncol = 1)
```

The sample time series were generated using the inverse Fourier transform. Therefore, the `fd_psd()` estimates are expected to yield most accurate estimates the scaling exponent. Let's see how other approaches do.

6.2 Standardised Dispersion Analysis (SDA)

In Standardised Dispersion Analysis, the time series is converted to z0scores (standardised) and the way the average standard deviation (SD) calculated in bins of a particular size scales with the bin size should be an indication of the presence of power-laws. That is, if the bins get larger and the variability decreases, there probably is no scaling relation. If the SD systematically increases either with larger bin sizes, or, in reverse, this means the fluctuations depend on the size of the bins, the size of the measurement stick.

```
sdaN1 <- fd_sda(ColouredNoise$`-1`, doPlot = FALSE, returnPlot = TRUE, tsName = "Pink noise", noTitle = T
```

```
>
>
> fd_sda:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Standardised Dispersion Analysis
>
>  Full range (n = 10)
> Slope = -0.16 | FD = 1.16
>
>  Fit range (n = 9)
> Slope = -0.14 | FD = 1.14
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
sda0 <- fd_sda(ColouredNoise$`0`, doPlot = FALSE, returnPlot = TRUE, tsName = "White noise", noTitle = TR
```

```
>
>
> fd_sda:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Standardised Dispersion Analysis
```

```
>
>  Full range (n = 10)
> Slope = -0.53 | FD = 1.53
>
>  Fit range (n = 9)
> Slope = -0.56 | FD = 1.56
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
sdaP1 <- fd_sda(ColouredNoise$`1`, doPlot = FALSE, returnPlot = TRUE, tsName = "Blue noise", no
```

```
>
>
> fd_sda:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Standardised Dispersion Analysis
>
>  Full range (n = 10)
> Slope = -1.15 | FD = 2.15
>
>  Fit range (n = 9)
> Slope = -1.08 | FD = 2.08
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
cowplot::plot_grid(sdaN1$plot,sda0$plot,sdaP1$plot,ncol = 1)
```

6.3 Detrended Fluctuation Analysis (DFA)

The procedure for Detrended Fluctuation Analysis is similar to SDA, except that within each bin, the signal is first detrended, what remains is then considered the residual variance. The logic is the same, the way the average residual variance scales with the bin size should be an indication of the presence of power-laws. There are many different versions of DFA, one can choose to detrend polynomials of a higher order, or even detrend using the best fitting model, which is decided for each bin individually. See the manual pages of `fd_dfa()` for details.

```
dfaN1 <- fd_dfa(ColouredNoise$`-1`, doPlot = FALSE, returnPlot = TRUE, tsName = "Pink noise", n
```

```
>
>
> fd_dfa:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Detrended FLuctuation Analysis
>
>  Full range (n = 8)
> Slope = 0.95 | FD = 1.22
>
>  Exclude large bin sizes (n = 7)
> Slope = 0.94 | FD = 1.22
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
dfa0  <- fd_dfa(ColouredNoise$`0`, doPlot = FALSE, returnPlot = TRUE, tsName = "White noise", n
```

```
>
>
> fd_dfa:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Detrended FLuctuation Analysis
```

```
>
>  Full range (n = 8)
> Slope = 0.49 | FD = 1.51
>
>  Exclude large bin sizes (n = 7)
> Slope = 0.52 | FD = 1.48
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
dfaP1 <- fd_dfa(ColouredNoise$`1`, doPlot = FALSE, returnPlot = TRUE, tsName = "Blue noise", noTitle = T
```

```
>
>
> fd_dfa:   Sample rate was set to 1.
>
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Detrended FLuctuation Analysis
>
>  Full range (n = 8)
> Slope = 0.13 | FD = 1.86
>
>  Exclude large bin sizes (n = 7)
> Slope = 0.13 | FD = 1.85
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
cowplot::plot_grid(dfaN1$plot,dfa0$plot,dfaP1$plot,ncol = 1)
```

6.4 Other varieties of fluctuation analysis

Check the `casnet` manual for other functions in the `fd_` family!

Study Materials and Resources

Systems Innovation

The Systems Innovation platform has lots of resources on Complex Systems, Complex Networks and related topics. Their YouTube channel contains a wealth of informative videos.


Self-Organization

Chapter 7

Fluctuation Analyses: Local Scaling

7.1 Multi-fractal geometry in time series

7.2 Multi-fractal DFA

7.3 The Wavelet Transform Modulus Maxima (WTMM)

7.4 Multi-fractal Spectrum Measures

Part V

Recurrence Quantification Analysis

```r
# Seed for random number generation
set.seed(42)
knitr::opts_chunk$set(cache.extra = knitr::rand_seed)
#knitr::opts_chunk$set(dev = "tiff", dpi = 600)
```

Chapter 8

Categorical Data: Auto-RQA

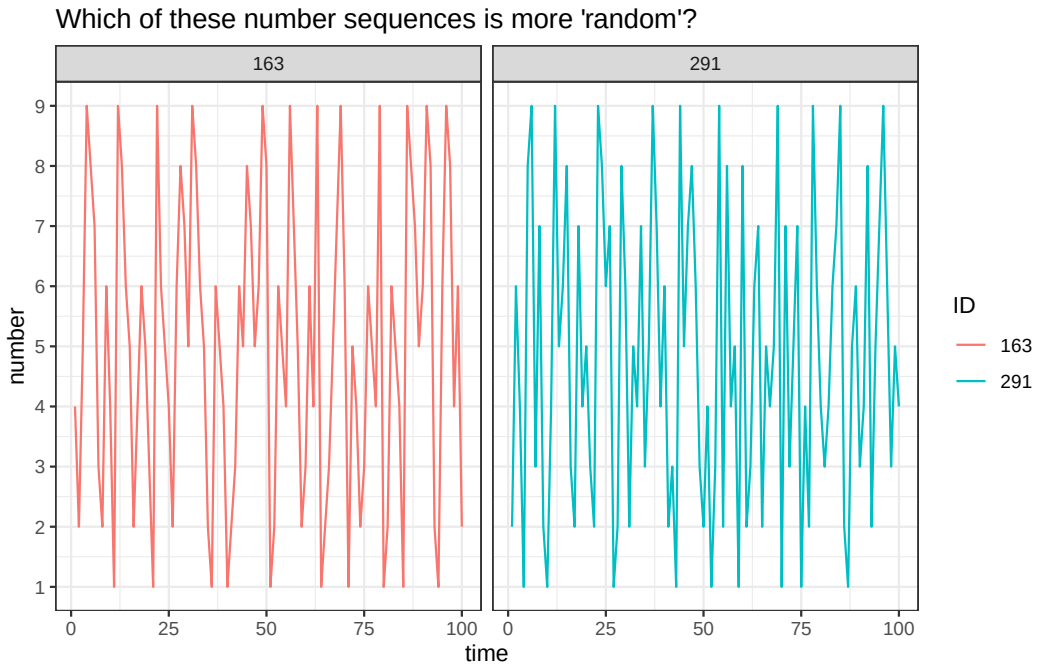
Recurrence Quantification Analysis on a single time series of categorical data, is called categorical Auto-RQA. The analysis involves determining, for each point in time, whether the observed value at that time will recur at some other point in the time series.

The main tool used to quantify the patterns of recurrent values is the Recurrence Matrix. A binary matrix of size N x N (where N is the length of the time series) in which 0s indicate no recurring value and 1s indicate a value that was observed at some point in time recurred at another point.

8.1 Human Random Number Generators

We'll use data from **?** in which 242 students were asked to generate random sequences of 100 numbers between 1 and 9 (for details see the article).

```
library(casnet)
library(ggplot2)

# Load the random number sequence data from Oomens et al. (2015)
data(RNG)

# Select a subject
IDs <- RNG$ID%in%c(163,291)

# Look at the sequence
ggplot(RNG[IDs,],aes(x=time,y=number,group=ID)) +
  geom_line(aes(colour=ID))+
  facet_grid(~ID) +
  scale_y_continuous(breaks = 1:9) +
  ggtitle("Which of these number sequences is more 'random'?") +
  theme_bw()
```



In order to answer the question in the Figure title, we'll run a Recurrence Quantification Analysis.

The data are unordered categorical, that is, the differences between the integers are meaningless in the context of generating random number sequences. This means the RQA parameters can be set to quantify recurrences of the same value:

- Embedding lag = 1
- Embedding dimension = 1
- Radius = 0 (any number $\leq$ 1 will do)

In the code block below the functions `rp()`, `rp_measures()` and `rp_plot()` are used to perform RQA on 2 participants in the dataset.

```
# Run the RQA analysis
y_1  <- RNG$number[RNG$ID==163]
y_2 <-  RNG$number[RNG$ID==291]

## Plot the recurrence matrix
# Get the recurrence matrix
rp_1 <- rp(y1=y_1, emDim = 1, emLag = 1, emRad = 1)
rp_2 <- rp(y1=y_2, emDim = 1, emLag = 1, emRad = 1)
```

```
# Get the plots
g_1 <- rp_plot(rp_1, plotDimensions = TRUE, returnOnlyObject = TRUE, title = "ID 163")
g_2 <- rp_plot(rp_2, plotDimensions = TRUE, returnOnlyObject = TRUE, title = "ID 291")

# Get the RQA measures, using silent = FALSE will produce output in the console.
crqa_1 <- rp_measures(rp_1, silent = FALSE)
```

```
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Global Measures
>          Global Max.rec.points N.rec.points Recurrence.Rate Singular.points Divergence Repetitiv
> 1 Recurrence Matrix        9900       3104     0.3135354         1324  0.1111111     1.174157
>
>
>  Line-based Measures
>  Line.based N.lines N.points.on.lines     Measure     Rate    Mean Max Entropy.of.lengths Relative
> 1  Diagonal    668          1780  Determinism 0.5734536 2.664671   9       1.1141896      0.242472
> 2  Vertical    959          2090 V Laminarity 0.6733247 2.179353   3       0.4704117      0.102372
> 3 Horizontal    959          2090 H Laminarity 0.6733247 2.179353   3       0.4704117      0.102372
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
crqa_2 <- rp_measures(rp_2, silent = FALSE)
```

```
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Global Measures
>          Global Max.rec.points N.rec.points Recurrence.Rate Singular.points Divergence Repetitiv
> 1 Recurrence Matrix        9900       3094     0.3125253         1456  0.1428571    0.9010989
>
>
>  Line-based Measures
>  Line.based N.lines N.points.on.lines     Measure     Rate    Mean Max Entropy.of.lengths Relative
> 1  Diagonal    668          1638  Determinism 0.5294118 2.452096   7       0.8901225      0.193710
> 2  Vertical    682          1476 V Laminarity 0.4770524 2.164223   3       0.4466065      0.097191
> 3 Horizontal    682          1476 H Laminarity 0.4770524 2.164223   3       0.4466065      0.097191
```

```
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
# Using rp_cl() would look very similar:
# rp_1 <- rp_cl(y1 = y_1, emDim  = 1, emLag = 1, emRad= 1)
# rp_2 <- rp_cl(y1 = y_2, emDim  = 1, emLag = 1, emRad= 1)
```

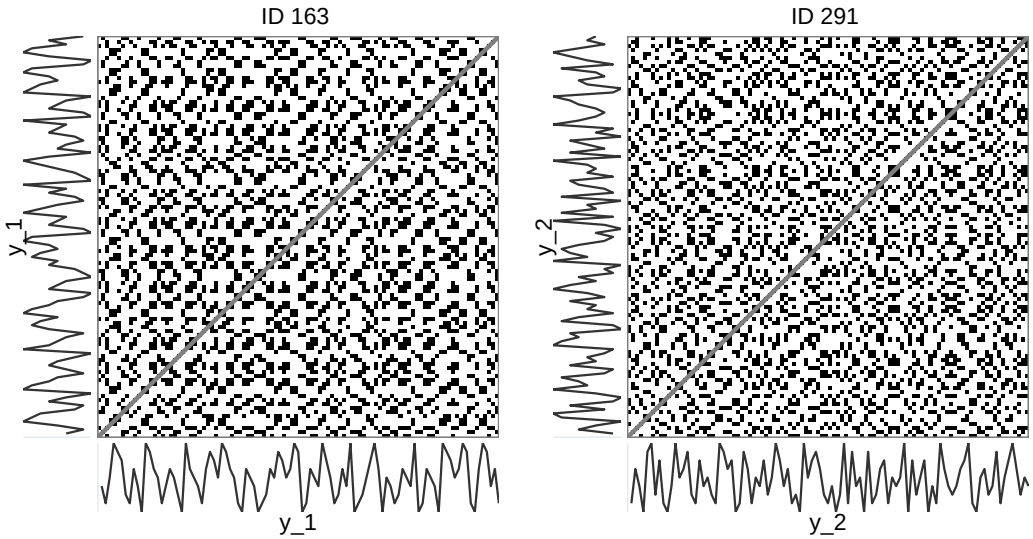
The output from `rp_measures()` is structured into *Global* and *Line-based* measures. In addition to information about the matrix size and number of points global measures are provided such as the *Recurrence Rate* (RR), the number of points that do **not** form a line (Singular Points) the *Divergence* (1 / the longest line structure) the average *Repetitiveness* (proportion of horizontal and vertical lines on the number of diagonal lines). The idea is to quantify how much of the deterministic structure (line structures) is due to repeating the same values (i.e., the horizontal and vertical lines, Laminarity). Finally the *Anisotropy* quantifies the symmetry of line structures in the plot by taking the ratio of he number of vertical lines over the numbers of horizontal lines in the plot. This should be 1 for Auto-RQA.

The *Line-based* output is a table listing the statistics for diagonal, vertical and horizontal lines (mean length, max length, rate, entropy of the distribution of line lengths and the same entropy but relative to the number of possible recurrent points and the coefficient of variation of line lengths).

The actual output object is a dataframe (which has more output fields, see the manual pages), the table output to the console is added as attribute `measuresTable`,

Below the data and plots are rearranged for ease of comparison.

```
library(cowplot)
# The recurrence plots
cowplot::plot_grid(g_1, g_2)
```




```
# The RQA measures
cbind.data.frame(subj163=t(crqa_1), subj291=t(crqa_1))
```

```
>                      1            1
> emRad         1.0000000    1.0000000
> RP_N       3104.0000000 3104.0000000
> RR            0.3135354    0.3135354
> SING_N     1324.0000000 1324.0000000
> SING_rate     0.4265464    0.4265464
> DIV_dl        0.1111111    0.1111111
> REP_av        1.1741573    1.1741573
> ANI           1.0000000    1.0000000
> N_dl        668.0000000  668.0000000
> N_dlp      1780.0000000 1780.0000000
> DET           0.5734536    0.5734536
> MEAN_dl       2.6646707    2.6646707
> MAX_dl        9.0000000    9.0000000
> ENT_dl        1.1141896    1.1141896
> ENTrel_dl     0.2424724    0.2424724
> CoV_dl        0.3923046    0.3923046
> N_vl        959.0000000  959.0000000
> N_vlp      2090.0000000 2090.0000000
> LAM_vl        0.6733247    0.6733247
> TT_vl         2.1793535    2.1793535
```

```
> MAX_vl       3.0000000    3.0000000
> ENT_vl       0.4704117    0.4704117
> ENTrel_vl    0.1023720    0.1023720
> CoV_vl       0.1761294    0.1761294
> REP_vl       1.1741573    1.1741573
> N_hlp     2090.0000000 2090.0000000
> N_hl       959.0000000  959.0000000
> LAM_hl       0.6733247    0.6733247
> TT_hl        2.1793535    2.1793535
> MAX_hl       3.0000000    3.0000000
> ENT_hl       0.4704117    0.4704117
> ENTrel_hl    0.1023720    0.1023720
> CoV_hl       0.1761294    0.1761294
> REP_hl       1.1741573    1.1741573
```

```
# The tables are stored in an attribute
attr(crqa_1,"measuresTable")
```

```
> $`Global Measures`
>          Global Max.rec.points N.rec.points Recurrence.Rate Singular.points Divergence Repetitiveness Ani
> 1 Recurrence Matrix        9900       3104     0.3135354         1324  0.1111111     1.174157       1
>
> $`Line-based Measures`
>   Line.based N.lines N.points.on.lines     Measure     Rate    Mean Max Entropy.of.lengths Relative.entropy
> 1   Diagonal    668          1780  Determinism 0.5734536 2.664671   9       1.1141896      0.2424724     0.39
> 2   Vertical    959          2090 V Laminarity 0.6733247 2.179353   3       0.4704117      0.1023720     0.17
> 3 Horizontal    959          2090 H Laminarity 0.6733247 2.179353   3       0.4704117      0.1023720     0.17
```

```
attr(crqa_2,"measuresTable")
```

```
> $`Global Measures`
>          Global Max.rec.points N.rec.points Recurrence.Rate Singular.points Divergence Repetitiveness Ani
> 1 Recurrence Matrix        9900       3094     0.3125253         1456  0.1428571    0.9010989       1
>
> $`Line-based Measures`
>   Line.based N.lines N.points.on.lines     Measure     Rate    Mean Max Entropy.of.lengths Relative.entropy
> 1   Diagonal    668          1638  Determinism 0.5294118 2.452096   7       0.8901225     0.19371040     0.31
> 2   Vertical    682          1476 V Laminarity 0.4770524 2.164223   3       0.4466065     0.09719148     0.17
> 3 Horizontal    682          1476 H Laminarity 0.4770524 2.164223   3       0.4466065     0.09719148     0.1
```

The sequence generated by participant 163 has a higher **DET**erminism (`DET = .40`) than the sequence by particpant 291 (`DET = .19`). The ratio of points on a diagonal line to the total number of recurrent point also quantifies this difference (`DET_RR`). Also interesting to note, both participants have a **LAM**inarity score of `0`. This implies they avoided to produce patterns in which the exact same numbers were repeated in succession. This is a tell-tale sign of the *non-random* origins of these sequences.

8.2 Hypothesis testing using constrained data realisations

A simple strategy to get some more certainty about the differences between the two sequences is to randomise the observed series, thus removing any temporal correlations that might give rise to recurring patterns in the sequences and re-run the RQA. If the repeated patterns generated by participant 163 are non-random one would expect the **DET**erminism to drop. If they do not drop this could indicate some random autoregressive process is causing apparent deterministic temporal patterns.

```
# Reproduce the same randomisation
set.seed(123456789)

# Randomise the number sequences
y_1rnd <- y_1[sample(1:NROW(y_1),size = NROW(y_1))]
y_2rnd <- y_2[sample(1:NROW(y_2),size = NROW(y_2))]

# Create the recurrence matrix
rp_1rnd <- rp(y1=y_1rnd, emDim = 1, emLag = 1,emRad = 1)
rp_2rnd <- rp(y1=y_2rnd, emDim = 1, emLag = 1,emRad = 1)

# Get the RPs
g_1rnd <- rp_plot(rp_1rnd, plotDimensions = TRUE, returnOnlyObject = TRUE, title = "ID 163 shuffled")
g_2rnd <- rp_plot(rp_2rnd, plotDimensions = TRUE, returnOnlyObject = TRUE, title = "ID 291 shuffled")

# Get CRQA measures
crqa_1rnd <- rp_measures(rp_1rnd, silent = FALSE)
```

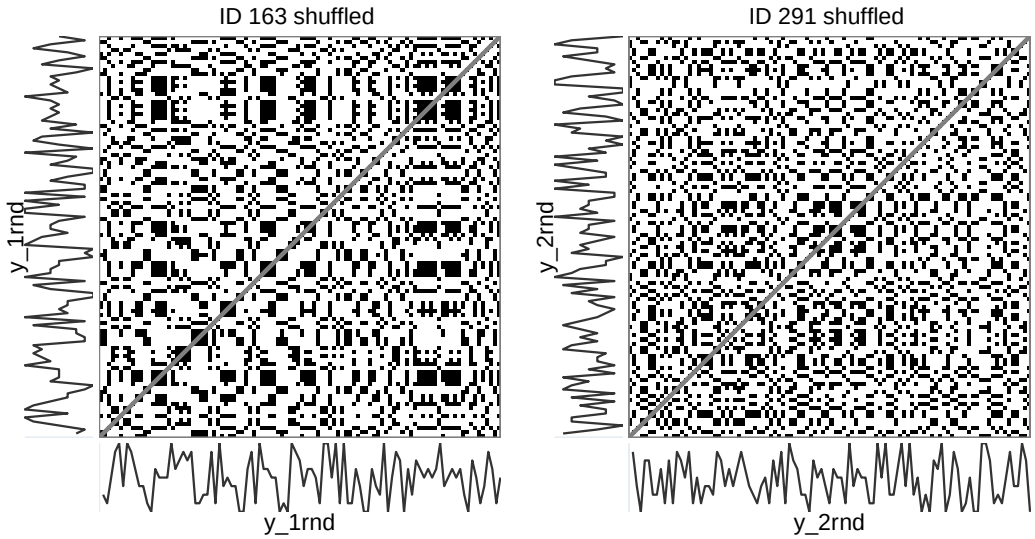
```
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Global Measures
>          Global Max.rec.points N.rec.points Recurrence.Rate Singular.points Divergence Repetitiveness Ani
> 1 Recurrence Matrix        9900       3104     0.3135354         1474  0.1666667     1.142945       1
>
>
>  Line-based Measures
```

```
>  Line.based N.lines N.points.on.lines     Measure     Rate    Mean Max Entropy.of.lengths Relative
> 1  Diagonal    652          1630  Determinism 0.5251289 2.500000   6       0.9496474      0.206664
> 2  Vertical    673          1863 V Laminarity 0.6001933 2.768202   6       1.1353416      0.247075
> 3 Horizontal    673          1863 H Laminarity 0.6001933 2.768202   6       1.1353416      0.247075
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
crqa_2rnd <- rp_measures(rp_2rnd, silent = FALSE)
```

```
>
> ~~~o~~o~~casnet~~o~~o~~~
>
>  Global Measures
>          Global Max.rec.points N.rec.points Recurrence.Rate Singular.points Divergence Repetitiv
> 1 Recurrence Matrix        9900       3094     0.3125253         1524  0.1666667     1.007643
>
>
>  Line-based Measures
>  Line.based N.lines N.points.on.lines     Measure     Rate    Mean Max Entropy.of.lengths Relative
> 1  Diagonal    652          1570  Determinism 0.5074337 2.407975   6       0.8384475      0.182464
> 2  Vertical    680          1582 V Laminarity 0.5113122 2.326471   4       0.7212808      0.156966
> 3 Horizontal    680          1582 H Laminarity 0.5113122 2.326471   4       0.7212808      0.156966
>
> ~~~o~~o~~casnet~~o~~o~~~
```

```
# Display recurrence plots
cowplot::plot_grid(g_1rnd, g_2rnd, align = "h")
```




```
# Display the RQA measures for ID 163
cbind.data.frame(subj163=t(crqa_1), subj163rnd=t(crqa_1rnd))
```

```
>                      1            1
> emRad        1.0000000    1.0000000
> RP_N      3104.0000000 3104.0000000
> RR           0.3135354    0.3135354
> SING_N    1324.0000000 1474.0000000
> SING_rate    0.4265464    0.4748711
> DIV_dl       0.1111111    0.1666667
> REP_av       1.1741573    1.1429448
> ANI          1.0000000    1.0000000
> N_dl       668.0000000  652.0000000
> N_dlp     1780.0000000 1630.0000000
> DET          0.5734536    0.5251289
> MEAN_dl      2.6646707    2.5000000
> MAX_dl       9.0000000    6.0000000
> ENT_dl       1.1141896    0.9496474
> ENTrel_dl    0.2424724    0.2066643
> CoV_dl       0.3923046    0.3299676
> N_vl       959.0000000  673.0000000
> N_vlp     2090.0000000 1863.0000000
> LAM_vl       0.6733247    0.6001933
> TT_vl        2.1793535    2.7682021
```

```
> MAX_vl       3.0000000    6.0000000
> ENT_vl       0.4704117    1.1353416
> ENTrel_vl    0.1023720    0.2470755
> CoV_vl       0.1761294    0.3826106
> REP_vl       1.1741573    1.1429448
> N_hlp     2090.0000000 1863.0000000
> N_hl       959.0000000  673.0000000
> LAM_hl       0.6733247    0.6001933
> TT_hl        2.1793535    2.7682021
> MAX_hl       3.0000000    6.0000000
> ENT_hl       0.4704117    1.1353416
> ENTrel_hl    0.1023720    0.2470755
> CoV_hl       0.1761294    0.3826106
> REP_hl       1.1741573    1.1429448
```

```
# Display the RQA measures for ID 291
cbind.data.frame(subj291=t(crqa_2),  subj291rnd=t(crqa_2rnd))
```

```
>                      1            1
> emRad     1.000000e+00    1.0000000
> RP_N      3.094000e+03 3094.0000000
> RR        3.125253e-01    0.3125253
> SING_N    1.456000e+03 1524.0000000
> SING_rate 4.705882e-01    0.4925663
> DIV_dl    1.428571e-01    0.1666667
> REP_av    9.010989e-01    1.0076433
> ANI       1.000000e+00    1.0000000
> N_dl      6.680000e+02  652.0000000
> N_dlp     1.638000e+03 1570.0000000
> DET       5.294118e-01    0.5074337
> MEAN_dl   2.452096e+00    2.4079755
> MAX_dl    7.000000e+00    6.0000000
> ENT_dl    8.901225e-01    0.8384475
> ENTrel_dl 1.937104e-01    0.1824648
> CoV_dl    3.195282e-01    0.3094559
> N_vl      6.820000e+02  680.0000000
> N_vlp     1.476000e+03 1582.0000000
> LAM_vl    4.770524e-01    0.5113122
```

```
> TT_vl     2.164223e+00    2.3264706
> MAX_vl    3.000000e+00    4.0000000
> ENT_vl    4.466065e-01    0.7212808
> ENTrel_vl 9.719148e-02    0.1569667
> CoV_vl    1.713084e-01    0.2542265
> REP_vl    9.010989e-01    1.0076433
> N_hlp     1.476000e+03 1582.0000000
> N_hl      6.820000e+02  680.0000000
> LAM_hl    4.770524e-01    0.5113122
> TT_hl     2.164223e+00    2.3264706
> MAX_hl    3.000000e+00    4.0000000
> ENT_hl    4.466065e-01    0.7212808
> ENTrel_hl 9.719148e-02    0.1569667
> CoV_hl    1.713084e-01    0.2542265
> REP_hl    9.010989e-01    1.0076433
```

Note that the number of recurrent points (RR) does not change whe we shuffle the data. What changes is the number of recurrent points that form line structures in the recurrence plot. Randomising the number sequences causes vertical line structures to appear in the recurrence plot (LAM, V_max, V_entr, TT), this is what we would expect if the data generating process were indeed a random process. Having no such structures means there were hardly any sequences consisting of repetitions of the same number. Participants may have adopted a strategy to avoid such sequences because they erroneously believed this to be a feature of non-random sequences.

8.2.1 A permutation test with surrogate time series

In order to get an idea about the meaningfulness of these differences, we can construct a surrogate data test for each participant. If we want a one-sided test with $\alpha = .05$, the formula for the number of constrained realisations M we minimally need is:

$$M = \frac{1}{\alpha} - 1 = 19$$

. Add the observed value and we have a sample size of $N = 20$. For a two sided test we would use

$$M = \frac{2}{\alpha} - 1 = 39$$

.

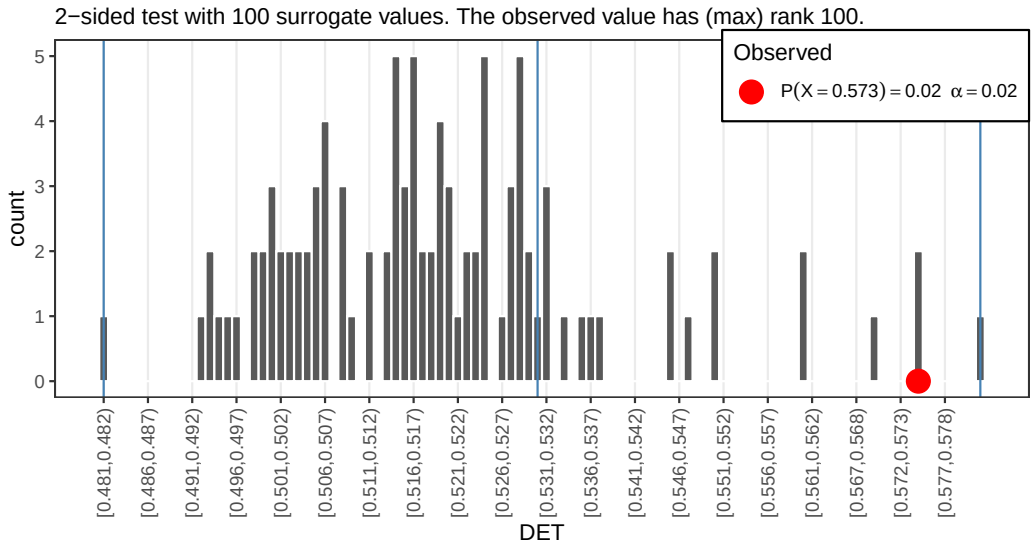
Of course, if there are no computational constraints on generating surrogate time series, we can go much higher, If we want $N = 100$, the test will be an evaluation of H_0 at $\alpha = .01$.

1. Create `99` realisations that reflect a test of the hypothesis $H_0 : X_i \sim \mathcal{U}(1, 9)$ at $\alpha = .01$.
2. Calculate the measure of interest, e.g. `DET`
3. If the observed `DET` value is at the extremes of the distribution of values representing H_0, the observed value was probably not generated by drawing from a discrete uniform distribution with finite elements `1` through `9`.

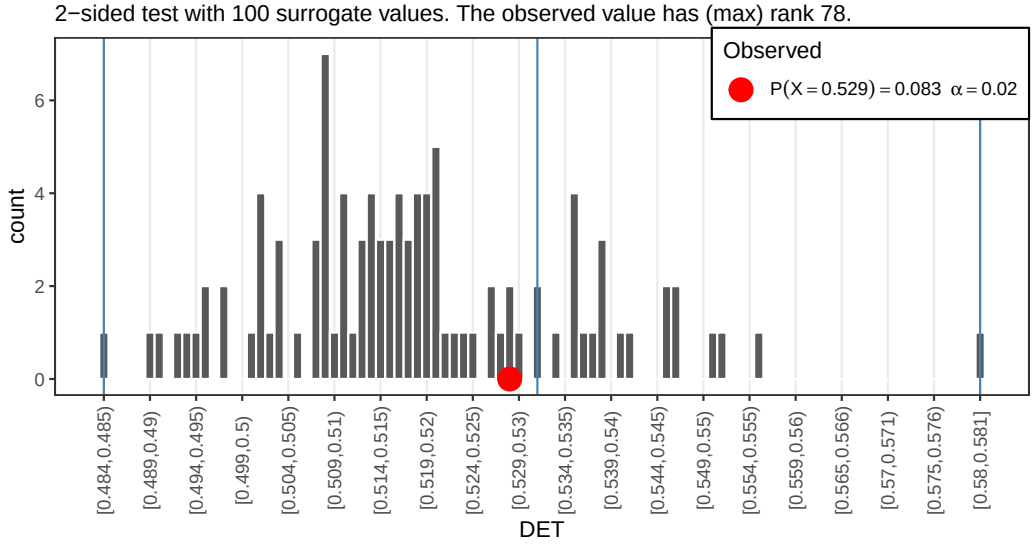
Use function `plorSUR_hist()` to get a p-value and plot the distributions. The red dots indicate the observed values.

```
# Get point estimates for p-values based on rank of observation (discrete distribution)
#   99 = (1 / alpha) - 1
# 99+1 = (1 / alpha)
alpha = 1/100

p_1 <- plotSUR_hist(surrogateValues = crqa_1rnd_sur$DET, observedValue = crqa_1$DET, measureNam
p_2 <- plotSUR_hist(surrogateValues = crqa_2rnd_sur$DET, observedValue = crqa_2$DET, measureNam

cowplot::plot_grid(p_1$surrogates_plot, p_2$surrogates_plot, labels = c("ID 163","ID 291"), nco
```

ID 163



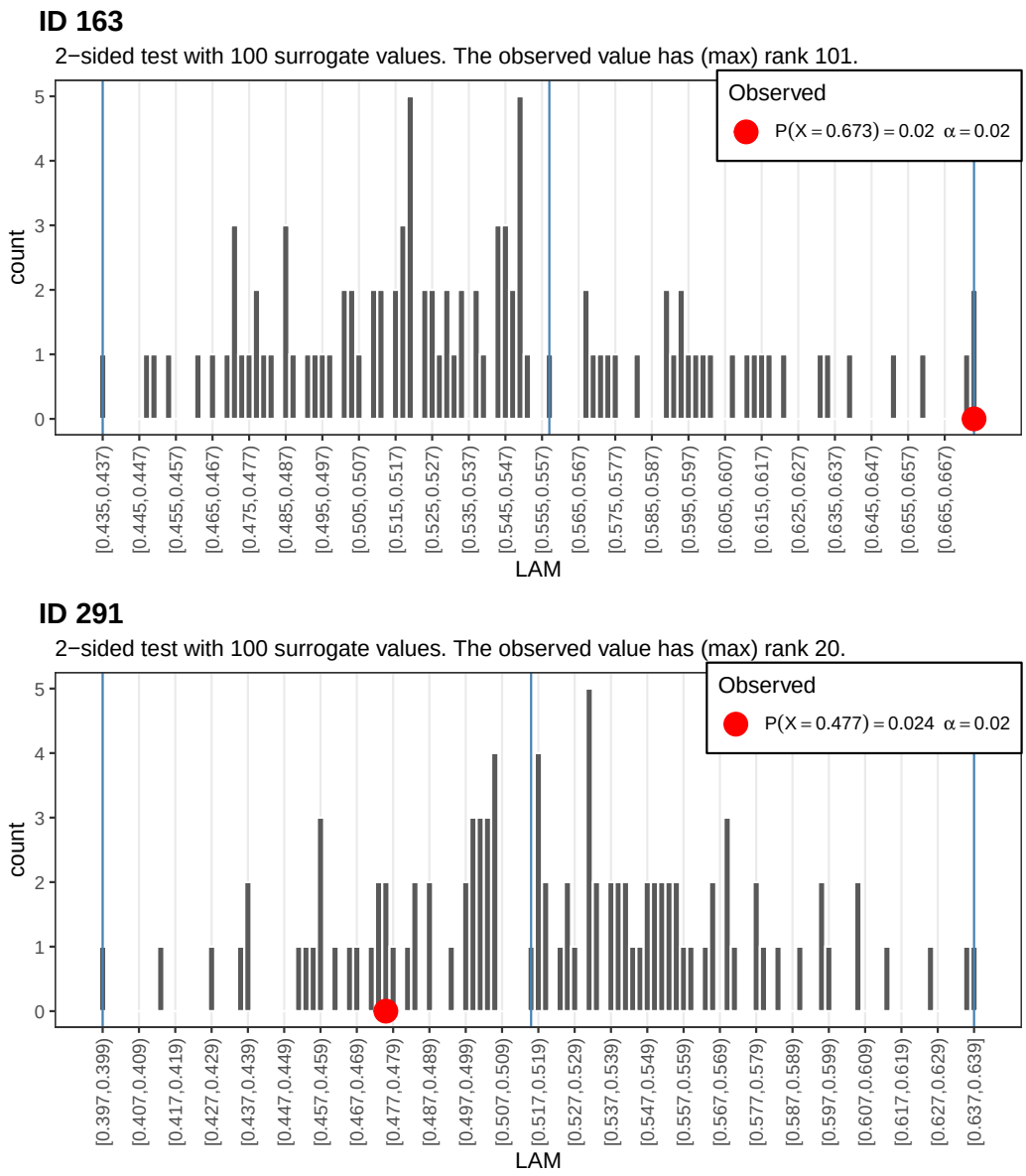
ID 291



To get the full picture, let's look at those missing repetitions of the same numbers.

```
# Get point estimates for p-values based on rank of observation (discrete distribution)
#   99 = (1 / alpha) - 1
# 99+1 = (1 / alpha)
alpha = 1/100
```

```
p_1 <- plotSUR_hist(surrogateValues = crqa_1rnd_sur$LAM_vl, observedValue = crqa_1$LAM_vl, measu
p_2 <- plotSUR_hist(surrogateValues = crqa_2rnd_sur$LAM_vl, observedValue = crqa_2$LAM_vl, measu

cowplot::plot_grid(p_1$surrogates_plot, p_2$surrogates_plot, labels = c("ID 163","ID 291"), nco
```



If we were naive to the origin of these number sequences, the results for

LAMinarity should make us doubt that they represent independent draws from a discrete uniform distribution of the type $X \sim \mathcal{U}(1, 9)$. If we had to decide which sequence was more, or, less random, then based on the **DET**erminism result, we would conclude that participant 163 produced a sequence that is less random than participant 291, the observed value of the former is at the right extreme of a distribution of DET values calculated from 99 realisations of the data constrained by H_0.

Chapter 9

Continuous Data: Recurrences in Phase Space

An important concept in recurrence-based time series analyses of continuous data is the Phase (or State) Space and its reconstruction. In order to explain the steps involved in reconstructing the phase space from a (multivariate) time series, consider the following system of $N = 4$ coupled competitive Lotka-Volterra equations previously studied by Vano et al. (**?**):

$$\frac{dY_i}{dt} = r_i Y_i \left(1 - \frac{\sum_{j=1}^{N} \alpha_{ij} Y_j}{K_i}\right), \quad i = 1, \dots, N \tag{9.1}$$

with K_i the carrying capacity set to a value of 10 for all Y_i, and r_i a vector of growth rates and α_{ij} a matrix of coupling parameters:

$$Y_i(t=0) = \begin{bmatrix} 8 \\ 2 \\ 2 \\ 2 \end{bmatrix}, \; K = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix}, \; r = \begin{bmatrix} 1 \\ 0.72 \\ 1.53 \\ 1.27 \end{bmatrix}, \text{ and } \alpha = \begin{bmatrix} 1 & 1.09 & 1.52 & 0 \\ 0 & 1 & 0.44 & 1.36 \\ 2.33 & 0 & 1 & 0.47 \\ 1.21 & 0.51 & 0.35 & 1 \end{bmatrix}. \tag{9.2}$$

The parameters in r and α are taken from Vano et al. (**?**), who describe the dynamics of the resulting attractor as chaotic (bounded, quasi-periodic, sensitive

dependence on initial conditions) and found several quantities to display power-law scaling, which is associated with self-organized criticality of multi-stable complex systems (**??**). Figure **??** shows $1,000$ iterations of the system (using the 4th order Runge-Kutta method). At $t = 700$ the coupling strengths in A are gradually increased by 0.01 at each iteration untill $t = 800$. This means the dynamics of all dimensions become more strongly coupled. This is one of the assumptions used by **?** to model symptom networks that are vulnerable for developing Major Depression. For the purpose of the present chapter, these time series could represent a single participant taking part in a study in which bi-daily ratings on a visual analog scale were collected. The dynamics are characterized by initial transient behaviour which settles into a stable, fixed point state, after the coupling strength is changed, a more complex state emerges as a (quasi-)periodic orbit (a trajectory that revisits the same region of phase space, but is not exactly the same as previous visits).

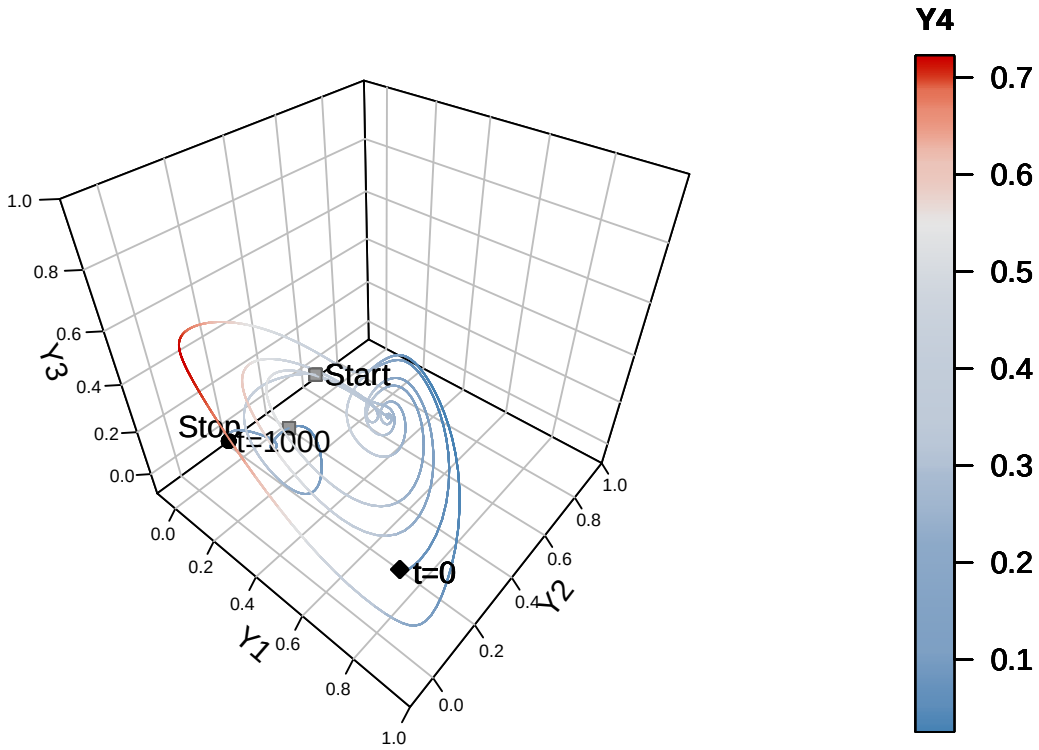
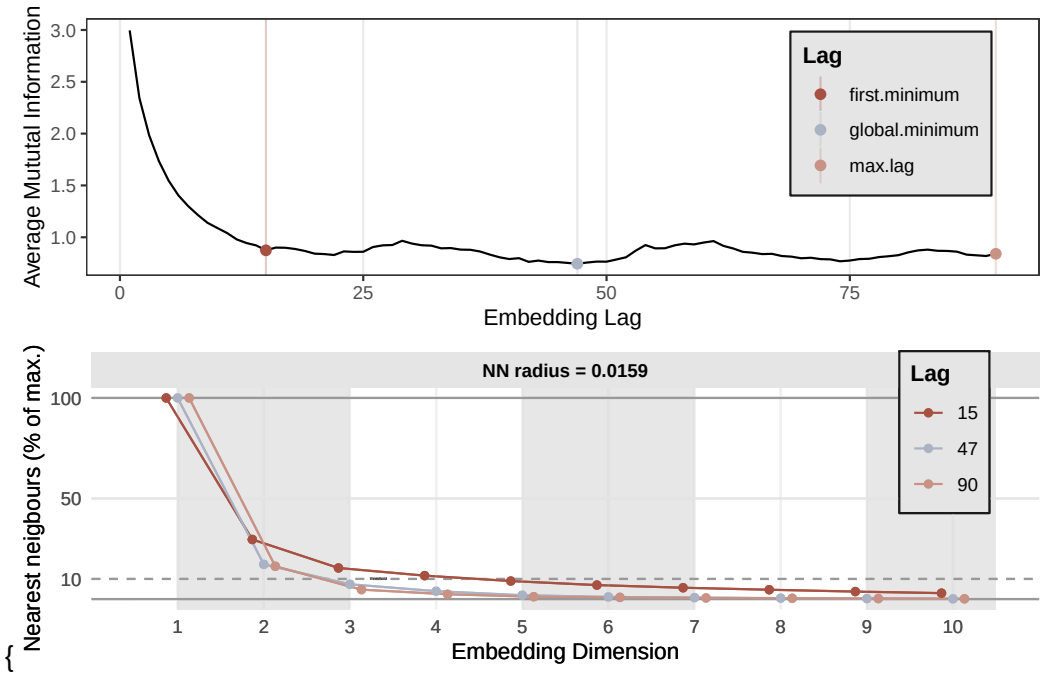


Figure 9.1 – The 4D state space of the simulated system. The grey squares mark t=700 and t=800 at which the coupling strengths between the 4 dimensions are gradually increased by 0.01 each iteration.

Recurrence analyses are based on a recurrence matrix which represents the states of the system that are re-visited at least once during the time it was observed. If a sequence of states is recurring, this is referred to as a trajectory in phase space, a relatively stable state, or, orbit of the system. There are several different ways to construct a recurrence matrix from one or more time series, in this tutorial the so-called multidimensional recurrence matrix (**?**), and the recurrence matrix based on phase space reconstruction will be discussed (cf. **?**). A (multivariate) time series can be interpreted as a finite representation of the trajectory or state-evolution of a stochastic or deterministic dynamic system: $y_{i}{}_{i=1}^{N}$, with $y_i = y(t_i)$ (cf. **?**). In the present example all relevant dimensions of the system have been observed and in such a case a multidimensional recurrence matrix can be constructed by considering each time series a state vector $\vec{y}_i$ of the m-dimensional phase space of the system. Figure **??** is a 3D representation of the 4D phase space of the system, with the values of dimension Y_4 represented by a colour gradient.

\begin{figure}



{

}

\caption{Embedding parameters for attractor reconstruction: The upper panel shows the time-delayed mutual information function with 3 embedding delays,

2 minima and the maximum possible lag for 10 surrogate dimensions given the length of the time series. The bottom panel shows the results of the false nearest neighbor analysis for each of the 3 delays. This plot is the ouput of function `est_parameters()` from package *casnet*.} \end{figure}

The recurrence matrix $\mathbf{R}_{i,j}$ is defined as:

$$\mathbf{R}_{i,j}(\varepsilon) = \Theta\left(\varepsilon - \|\vec{y}_i - \vec{y}_j\|\right), \quad i,j = 1,\dots,N \tag{9.3}$$

where $\|\cdot\|$ is a distance norm (e.g. Euclidean, Chebyshev, Manhattan), ε is a threshold value which determines at which distance value a state should be considered to recur, and Θ is the Heaviside function, which returns 1 if a distance value falls below ε and 0 otherwise. Figure **??** displays the unthresholded distance matrix based on the Maximum norm (Chebyshev distance). The distances of each state coordinate (a 4-tuple represented by the values of y_i at each time point) to every other state coordinate are represented by different a colours. The matrix is symmetric around the diagonal, the line of incidence, which is excluded from calculation of recurrence measures. The colour-bar displays several distance values on the right side, which, should they be chosen as the threshold value ε, would result in the Recurrence Rate (proportion of recurrent points in $\mathbf{R}_{i,j}$) displayed on the left.

```
> TableGrob (4 x 5) "di_rp_dim": 6 grobs
>   z     cells      name                 grob
> 1 1 (2-2,2-2) di_rp_dim        gtable[layout]
> 2 2 (3-3,2-2) di_rp_dim null[GRID.null.10393]
> 3 3 (2-2,3-3) di_rp_dim        gtable[layout]
> 4 4 (3-3,3-3) di_rp_dim        gtable[layout]
> 5 5 (2-2,4-4) di_rp_dim        gtable[layout]
> 6 6 (3-3,4-4) di_rp_dim null[GRID.null.10394]
```

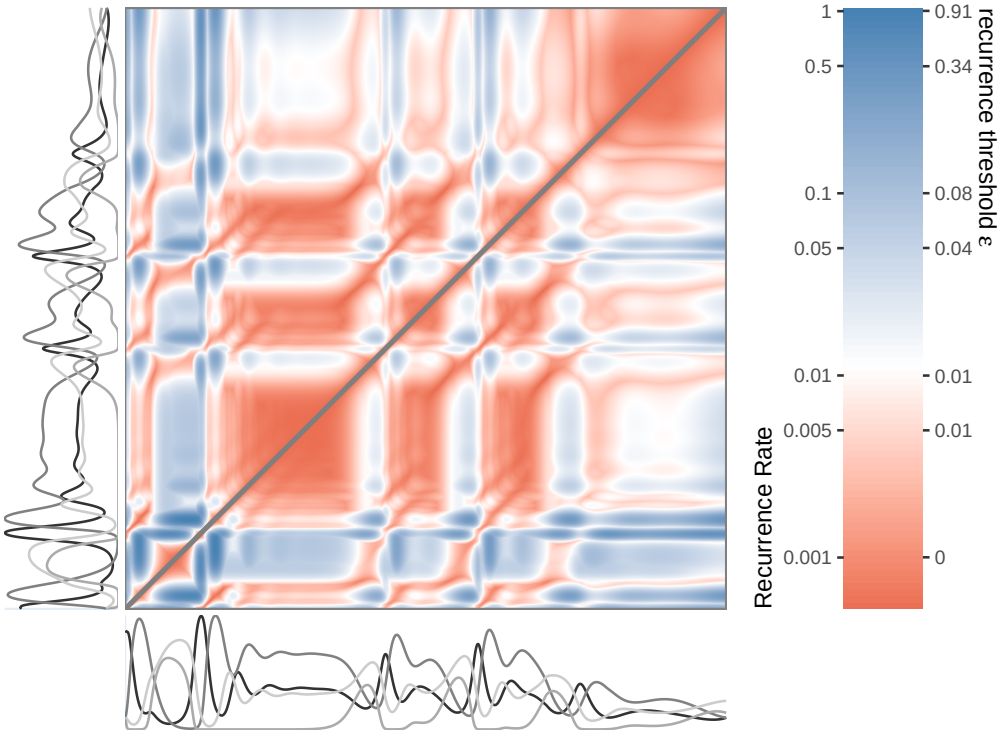


Figure 9.2 – Distance matrix of the simulated 4D phase space.

9.1 Phase Space Reconstruction

A record of all the dimensions that span the phase space of a system will not be available in most cases, so a reconstruction of the attractor and dynamics in phase space will often be conducted. This a common step in many nonlinear time series analyses and can be achieved using the method of delay-embedding (**??**). [1] Due to Takens' theorem (**?**) a phase space can be reconstructed that is *topologically equivalent* to the original space, by creating an m-dimensional time-delay embedding of a single observed dimension of a system: $\vec{y}_i = (y_i, y_{i+\tau}, \dots, y_{i+(m-1)\tau})$, where τ is the embedding delay. The idea is to exploit the interaction-dominant dynamics that govern the behavior of complex systems. Due to the coupled dynamics of the 4 dimensions of the simulated system, there is information about the dynamics of Y_2, Y_3 and Y_4 encoded in the dynamics of Y_1. Therefore, any of the dimensions could be used to reconstruct the dynamics of the original phase space. The exact trajectories through phase space will be different, but the dynamical invariants of the original attractor will be retained by the reconstruction and recurrence analyses can be used to quantify (properties of) dynamical invariants. The attractor reconstruction procedure involves two steps, the first is choosing an embedding delay τ, the second concerns choosing an appropriate number of surrogate dimensions to embed the time series.

9.1.1 Choosing an embedding lag

The choice for the embedding delay is an optimization step and less crucial than choosing a sufficiently large embedding dimension. Takens' theorem guarantees topological equivalence if m is twice as large as the fractal dimension of the support (the nonzero elements) of the invariants generated by the dynamics in the original phase space (**?**). In other words, there should be enough nonzero data to make an embedding with enough surrogate dimensions that fully captures the original dynamics. Embedding a time series in a higher dimension than strictly necessary is unproblematic. Intuitively, τ should represent a lag of time at which the information about future states of the system as predicted by its history, are at a minimum, for example, the lag at which the autocorrelation

[1] Note that phase space reconstruction is not strictly necessary for recurrence quantification analysis and recurrence network analysis. The analyses presented in this paper can also be conducted on the non-embedded 'raw' time series, however, here we assume the embedded series to be more informative.

function goes to 0. However, because time series representing observables of complex systems more often than not violate the stationarity and homogeneity assumptions, a time-delayed mutual information function is commonly used (cf. **?**). As can be seen in the top panel of Figure **??** the mutual information function of dimension Y_1 has local and global minima that could be used as the embedding delay. A delay of 17.5 was used (the median of the first local minima found for the four series) for all reconstructions. By copying the time series starting at τ and treating it as the first point of a surrogate dimension, a state vector is created based on a single observed time series. This process is repeated up to m-dimensions, by copying the series at $2\tau, 3\tau, ... (m-1)\tau$. The question is, how to decide on the number of embedding dimensions required to represent the dynamics of the original attractor?

9.1.2 False Nearest Neighbor analysis

One way to estimate m is by False Nearest Neighbor (FNN) analysis (**?**), which will consider whether points that are close together in an m-dimensional reconstructed phase space, will still be close together in an $m+1$ embedding. If this is not the case, these points were 'false neighbors' of the points in m dimensions. The lower panel of Figure **??** shows the results of the FNN analysis for the simulated system. We know the correct number of dimensions should be $m = 4$ and this is indeed the case for Y_1 and the local minimum. With real-world data, one would look at the dotted line, where the percentage of false neighbors drops below 10%. The value obtained by FNN provides an estimate of the minimum number of coupled ODE's required to model the dynamics of the original phase space trajectory. Based on Y_2 and Y_4 a similar result is found, but Y_3 is an exception, here FNN indicates a much higher embedding dimension. The reason is the lack of support that is required by Takens' theorem, as can be seen in Figure **??**, series $Y3$ contains a lot of zeroes. If this were an observed data set and the aim was to compare analyses based on reconstructions of each of the four series, the largest embedding dimension would be used for all series, for demonstration purposes, we'll use $m = 4$ for all reconstructions. Note that the lack of support in Y_3 cannot be resolved by simply adding a constant and proceeding with attractor reconstruction. The lack of support is produced by the coupled dynamics, so there are basically two ways in which Y_3 could become a candidate for a succesful phase space reconstruction: i) Observe the dynamics for a longer period of time, potentially decreasing the relative frequency of zeros; ii) Change the parameters to change the dynamics of Y_3.

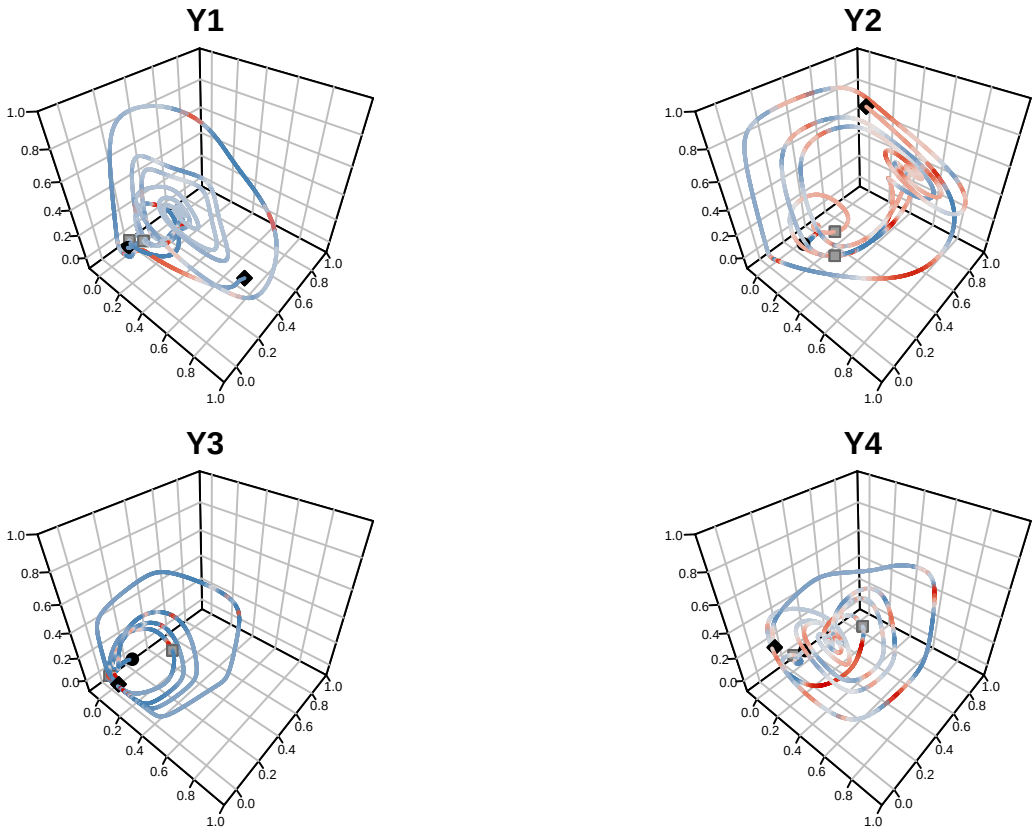


Figure 9.3 – The reconstructed phase space based on variable Y1, Y2, Y3 and Y4 respectively. The black diamond indicates t=0; the grey squares at t=700 and t=800 indicate where the coupling strength is increased; the black circle is at t=949.

Figure **??** displays reconstructed attractors based on each of the 4 dimensions. The topological equivalence may not be immediately apparent, but considering that for each reconstruction just 1 of the original dimensions was used it is quite remarkable we end up with a trajectory that appears to orbit around a specific region. According to the rules of topology, there exists a continuous deformation for each of these trajectories that will recover the exact same shape as the original attractor. The topological equivalence is visible in the unthresholded distance matrices displayed in Figure **??**. Here, the transition towards the stable state that persists for some time is clearly visible, as well as several other transitions to states that recur for a shorter duration of time.

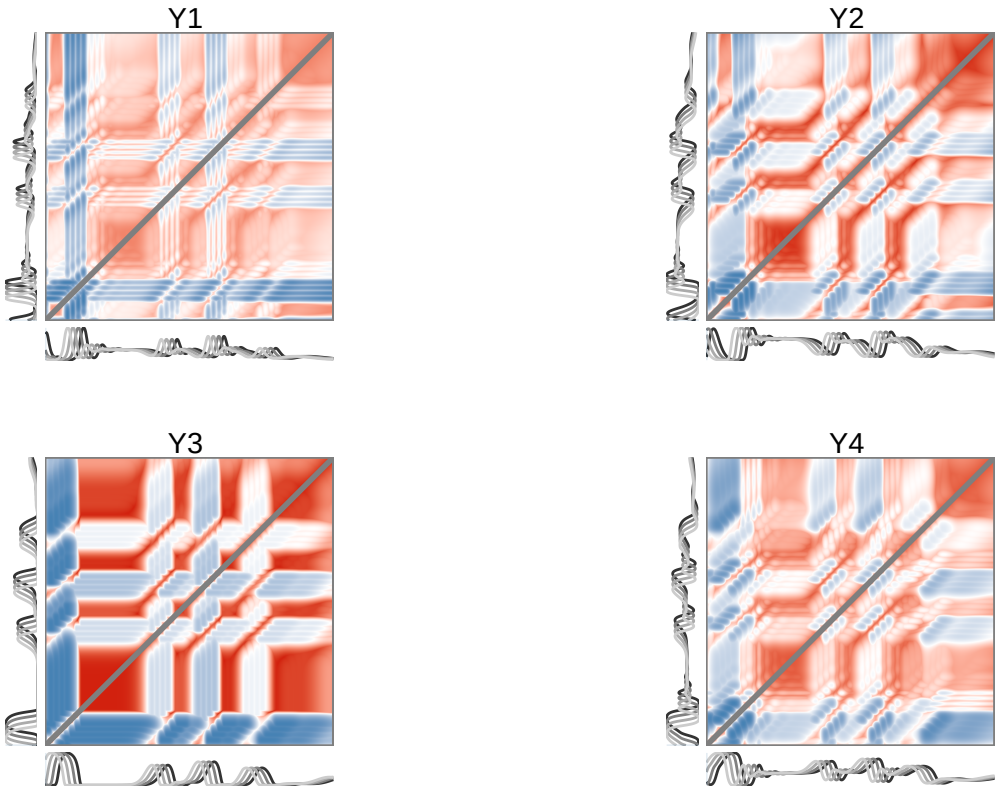


Figure 9.4 – Distance matrix of the reconstructed phase spaces based on Y1, Y2, Y3, and Y4, respectively.

9.2 From Distance Matrix to Recurrence Plot

The next step is to either turn the thresholded recurrence matrix into a Recurrence Plot (RP, **???**), or into a Recurrence Network (RN, **?**). In the present paper, the focus is on constructing and analyzing Recurrence Networks, not the analysis of Recurrence Plots (Recurrence Quantification Analysis, RQA). It is important to briefly discus RQA in order to point out some of the differences between the RN and RP analysis.

Figure **??** represents the RP based on the multidimensional recurrence matrix, the threshold ε was chosen to yield 5% recurrent points. The probability of a cell in the matrix being a recurrent point is expressed by the Recurrence Rate, the measure RR:

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} \mathbf{R}(i,j) \tag{9.4}$$

The recurrence rate of an RP is equivalent to the total degree of a network constructed from a recurrence matrix, because the recurrent points indicate time points are connected by an edge. Other RQA measures do not have clear network analogues, because they are based on the line structures in the RP. Time-adjacent recurrent points form diagonal lines ℓ, which are considered a recurrent trajectory through reconstructed phase space. If the distribution of diagonal line lengths is $P(\ell)$ then the deterministic structure of the reconstructed attractor (DET) is quantified by the proportion of recurrent points that form a diagonal line:

$$DET = \frac{\sum_{\ell \geq \ell_{\text{min}}}^{\ell_{\text{max}}} \ell \, P(\ell)}{\sum_{i,j=1} \mathbf{R}(i,j)} \tag{9.5}$$

Based on the distribution of line lengths, measures can be calculated such as the maximum and mean diagonal line length (DL_{max}, DL_{mean}) and the entropy of the line length distribution, $H(\ell)$, based on the observed frequencies, see equation (**??**):

$$p(\ell) = \frac{P(\ell)}{\sum_{\ell \geq \ell_{\min}}^{\ell_{\max}} P(\ell)} H(\ell) = -\sum_{\ell \geq \ell_{\min}}^{\ell_{\max}} p(\ell) \ln p(\ell) \tag{9.6}$$

DET together with $H(\ell)$ quantify different aspects of the coupled dynamics (**??**), a high determinism (DET) and a high entropy of the distribution $P(\ell)$ would represent a meta-, or, multi-stable regime (**?**), whereas a high determinism coinciding with low entropy of $P(\ell)$ would indicate a relatively stable dynamical regime. Similar measures can be calculated based on the vertical (or horizontal) line structures $P(\nu)$: Laminarity (LAM) is the proportion of recurrent trajectories that reflect recurrence of the same state, a laminar phase; Trapping time (TT, or VL_{mean}) is the average duration of a laminar phase; VL_{max} the maximal vertical line length; and $H(\nu)$, the entropy of the distribution of vertical lines.

The absolute values of these measures depend on the choice of the recurrence threshold ε, but also on the length of the time series. For the purpose of comparing measures calculated from different recurrence plots it can make sense to fix the recurrence rate by choosing a different threshold value for each analysis. It is also possible to pick a fixed threshold and study how the recurrence rate varies between different plots. Several methods exist for the optimization of choosing ε, some of which will yield different thresholds for different recurrence measures (e.g. the signal detection method, **?**). For these and other considerations when determining the recurrence analysis parameters see **?** and the references therein. It is important to note that ε is a crucial analysis parameter for recurrence-based methods and its value and selection should always be reported and justified. In general, it is recommended to study the effects of choosing different sets of parameters when comparing the recurrence measures obtained from different (samples of) time series.

In Table **??**, several common recurrence measures are provided for each of the phase spaces considered so far. The values for Determinism and laminarity are not surprising, after all, this is a completely deterministic system! The point here is to show that the dynamically invariant properties of the original attractor are retained for reconstructions based on Y_1, Y_2, Y_4. The reconstruction based on Y_3 was expected to be less accurate due to the lack of support (stationary level at zero for a large period of time). The recurrence threshold ε has to be

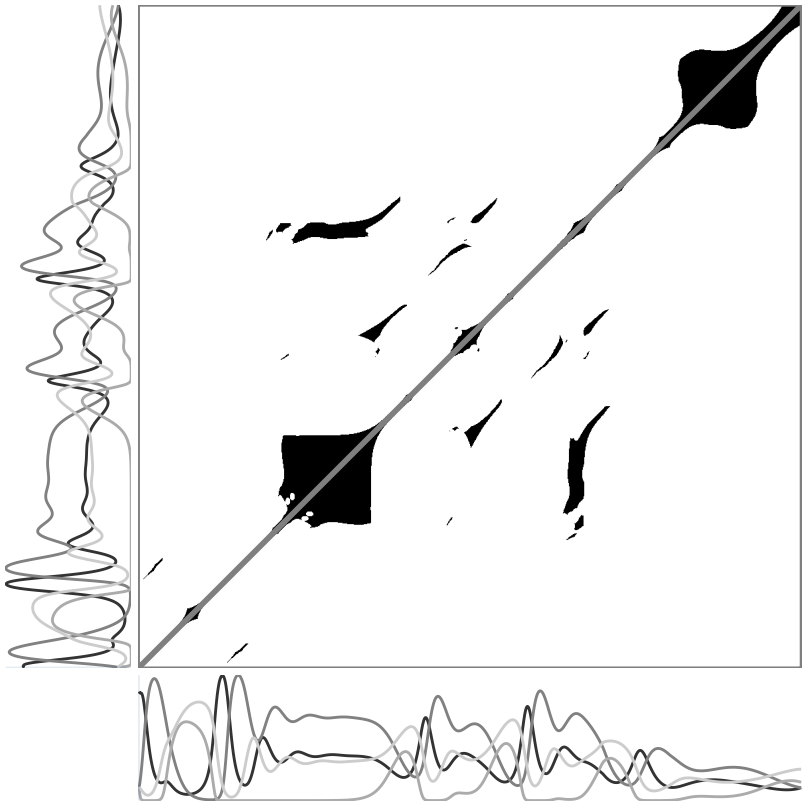


Figure 9.5 – A recurrence plot based on the multidimensional recurrence matrix of the coupled 4D system.

extremely small compared to the other values in order to get $RR = 0.045$. There are techniques that can be used to estimate how well a reconstructed phase space from, say, the first 75% of the time series predicts the remaining 25% (nonlinear prediction skill), or how well it predicts other observed dimensions (convergent cross-mapping) (**?**). Convergent cross-mapping can also be used to detect coupling direction and causality in dynamic or delayed interactions (see e.g., **???**).

Chapter 10

Notes on running Recurrence Quantification Analysis in R

This book discusses only R packages and functions for running RQA. There are howevr many othe options like Matlab, or dedicated software for specific platforms. Acomplehensive list is available here:
http://www.recurrence-plot.tk/programmes.php

10.1 casnet

There are 2 main ways to run Recurrence Quantification Analyses in `casnet`:

- Using functions `rp`, `rp_measures` and `rp_plot`
- Using function `rp_cl` which will run Norbert Marwan's commandline Recurrence Plots

The following example will use the native `casnet` functions, see the paragraph **An R interface to Marwan's commandline recurrence plots** to learn about using `rp_cl()`.

An R interface to Marwan's commandline recurrence plots

IMPORTANT: Currently `rp_cl` can only run on an operating system that allows execution of 32-bit applications!

The `rp_cl()` function is a wrapper for the commandline Recurrence Plots executable provided by Norbert Marwan.

The `rp` executable is installed on your machine when the function `rp_cl()` is called for the first time:

- It is renamed to `rp` from a platform specific file downloaded from the commandline Recurrence Plots site.
- The file is copied to the directory: [path to casnet]/exec/
  - Make sure that you have rights to execute programs in this directory!
- The latter location is stored as an option and can be read by calling `getOption("casnet.path_to_rp")`

If you cannot change the permissions on the folder where `rp` was downloaded, consider downloading the appropriate executable from the commandline Recurrence Plots site to a directory in which you have such permissions. Then change the `path_to_rp` option using `options(casnet.path_to_rp="YOUR_PATH_TO_RP")`. See the manual entry for `rp_cl()` for more details.

The platform specific `rp` command line executable files were created by Norbert Marwan and obtained under a Creative Commons License from the website of

the Potsdam Institute for Climate Impact Research at:
http://tocsy.pik-potsdam.de/

The full copyright statement on the website is as follows:

More information about recurrence quantification analysis can be found on the Recurrence Plot website.

10.2 Other options

The most comprehensive alternative to `casnet` is probably `crqa`. It has a great tutorial paper by Coco & Dale (2014).

Several other packages have dedicated functions, e.g. package `nonlinearTseries` has a function `RQA`.

10.3 Computational load: The Python solution [PyRQA]

When the time series you analyze are very long, the recurrence matrix will become very large and R will become very slow. One solution is to use R to run the Python program `PyRQA` or perhaps `pyunicorn`. The options for `PyRQA` are limited compared to the casnet functions, but the gain in processing speed is remarkable!

What follows is an example of how you could make `PyRQA` run in R using the package `reticulate`.

Setup the environment

Suppose you are on a machine that has both `R` and `Python` installed then the steps are:

- Make sure `Python` and `pip` are up to date
- Create a virtual (or a coda) environment.
- Install `PyRQA` into the virtual environment.

You should only have to create and setup the environment once.

```
library(reticulate)

# OS requirements
# Python3.X is installed and updated.
# On MacOS you'll probably need to run these commands in a Terminal window:
python3 pip install --update pip # Updates the Python module installer
python3 pip intall pyrqa # Installs the pyrqa module on your machine

# First make sure you use the latest Python version
# You can check your machine by calling: reticulate::py_discover_config()
reticulate::use_python("/usr/local/bin/python3")

# Create a new environment "r-reticulate", the path is stored in vEnv
# On Windows use coda_create() see the reticulate manual.
vEnv <- reticulate::virtualenv_create("r-reticulate")
```

```
# Install pyrqa into the virtual environment
reticulate::virtualenv_install("r-reticulate","pyrqa")

# If you wish to remove the environment use: reticulate::virtualenv_remove("r-reticulate")
```

After the environment is set up:

- Restart your R session and instruct the system to use Python in the virtual environment.
- Import `PyRQA` into your `R` session.
- Use the `PyRQA` functions that are now available as fields (`$`) of the imported object!

An important thing to note in the example below is the use of `as.integer()` to pass integer variables to Python.

```
# Make sure you associate reticulate with your virtual environment.
reticulate::use_virtualenv("r-reticulate", required = TRUE)

# Import pyrqa into your R session
pyrqa <- reticulate::import("pyrqa")

# Alternatively, you can import from a path in the virtual environment.
# On MacOS this will be a hidden folder in your home directory:
# '.virtualenvs/r-reticulate/lib/Python3.X/site-packages'
# pyrqa <- import_from_path(file.path(vEnv,"/lib/python3.9/site-packages"))

# Now perform RQA on your N = 10,000 time series!
Y <- cumsum(rnorm(10000))

# Automated parameter search will still take some time using casnet
system.time({
  emPar <- casnet::est_parameters(Y, doPlot = FALSE)
  emRad <- casnet::est_radius(y1 = Y, emLag = emPar$optimLag, emDim = emPar$optimDim)
  })
# user    system  elapsed
# 299.732 89.094  393.620

# About 5 minutes to find a delay, embedding dimension and radius yielding 5% recurrent points.
```

```
# Now do an RQA on the 10,000 x 10,000 matrix using Python
system.time({
time_series <- pyrqa$time_series$TimeSeries(Y,
                                            embedding_dimension= as.integer(emPar$optimDim),
                                            time_delay= as.integer(emPar$optimLag))
settings    <- pyrqa$settings$Settings(time_series,
                                       analysis_type = pyrqa$analysis_type$Classic,
                                       neighbourhood = pyrqa$neighbourhood$FixedRadius(emRad$Radius),
                                       similarity_measure = pyrqa$metric$EuclideanMetric,
                                       theiler_corrector = 0)
computation <- pyrqa$computation$RQAComputation$create(settings)
result      <- computation$run()
})
# user   system  elapsed
# 2.996  0.069   0.365

# About 3 seconds for the analysis...
# That's really fast!

print(result)
```

```
RQA Result:
===========

Minimum diagonal line length (L_min): 2
Minimum vertical line length (V_min): 2
Minimum white vertical line length (W_min): 2

Recurrence rate (RR): 0.050090
Determinism (DET): 0.955821
Average diagonal line length (L): 10.634044
Longest diagonal line length (L_max): 9866
Divergence (DIV): 0.000101
Entropy diagonal lines (L_entr): 3.064460
Laminarity (LAM): 0.969709
Trapping time (TT): 14.930102
Longest vertical line length (V_max): 345
```

```
Entropy vertical lines (V_entr): 3.386939
Average white vertical line length (W): 265.518914
Longest white vertical line length (W_max): 9161
Longest white vertical line length inverse (W_div): 0.000109
Entropy white vertical lines (W_entr): 4.726210

Ratio determinism / recurrence rate (DET/RR): 19.081989
Ratio laminarity / determinism (LAM/DET): 1.014530
```

You can also save the Recurrence Plot.

```
RPcomputation <- pyrqa$computation$RPComputation$create(settings)
RPresult <- RPcomputation$run()
pyrqa$image_generator$ImageGenerator$save_recurrence_plot(RPresult$recurrence_matrix_reverse,'r
```

```
knitr::include_graphics("images/recurrence_plot_python.png")
```

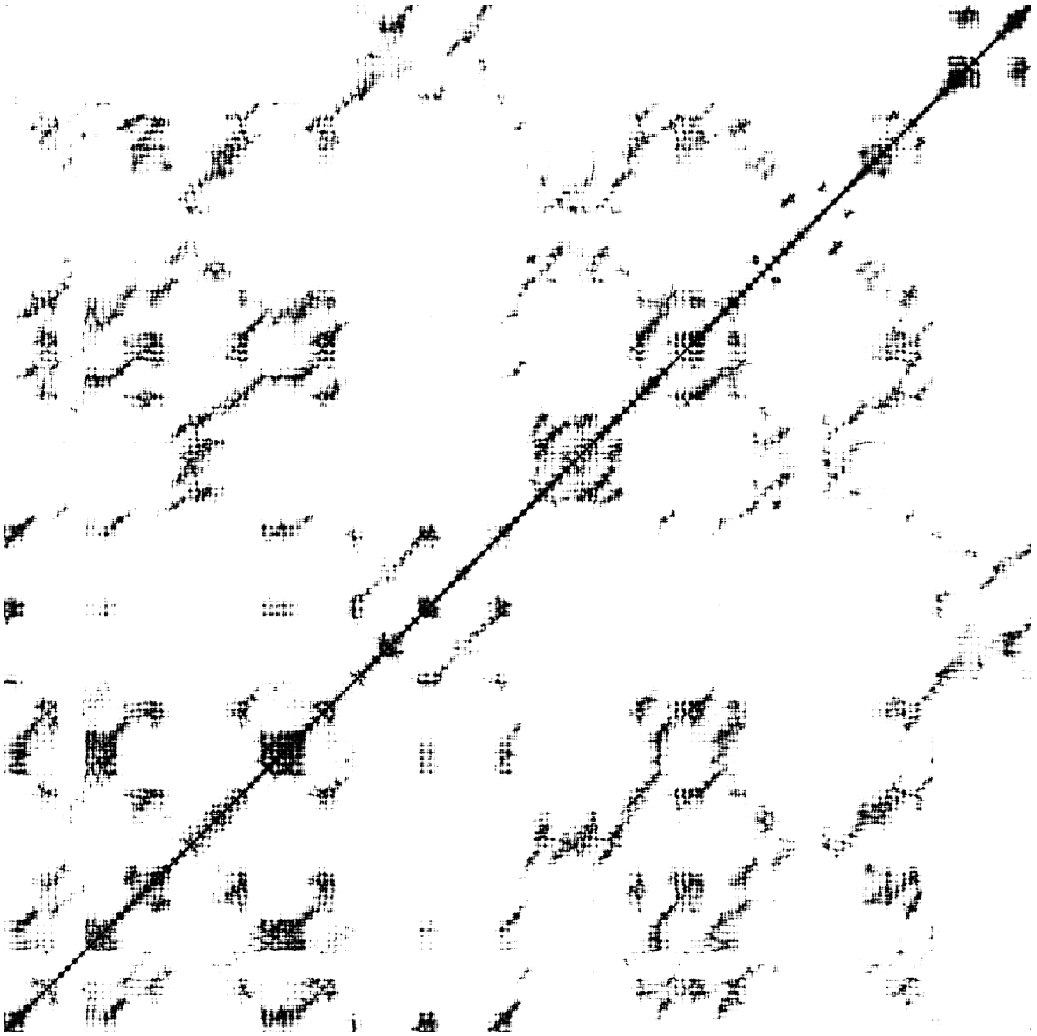


Figure 10.1 – RP produced by PyRQA

Chapter 11

Cross-Recurrence Analysis: Categorical Data

kjkjkjkjk

Chapter 12

Cross-Recurrence Analysis: Continuous Data

lklklklk

12.0.1 Diagonal Recurrence Profiles

Chapter 13

Other flavours of RQA

13.1 Lagged RQA: Sliding window analyses

13.2 Chromatic RQA

13.3 Anisotropic RQA

13.4 Multidimensional RQA

13.5 References

Part VI

Multivariate Timeseries Analysis

Chapter 14

Vector Auto Regression (VAR)

Chapter 15

Dynamic Complexity

Study Materials and Resources

Systems Innovation

The Systems Innovation platform has lots of resources on Complex Systems, Complex Networks and related topics. Their YouTube channel contains a wealth of informative videos.

Chapter 16

Graph Theory and Complex Network Analysis

kjkjkjkj

16.1 Recurrence Networks

kjkjkjk

16.2 Multiplex Recurrence Networks

16.3 Networks based on Multidimensional RQA

16.4 Transition Networks

kjkjkj

Appendix A

Some Notes on Using R

You have probably heard many people say they should invest more time and effort to learn to use the R software environment for statistical computing… *and they were right*. However, what they probably meant to say is: "I tried it, but it's so damned complicated, I gave up"… *and they were right*. That is, they were right to note that this is not a point and click tool designed to accommodate any user. It was built for the niche market of scientists who use statistics, but in that segment it's actually the most useful tool I have encountered so far.

A.1 New to R?

Now that your struggles with getting a grip on R are fully acknowledged in advance, let's try to avoid the 'giving up' from happening. Try to follow these steps to get started:

1. **Get R and add some user comfort:** Install the latest R software *and* install a user interface like RStudio... *It's all free!* An R interface will make some things easier, e.g., searching and installing packages from repositories. R Studio will also add functionality, like git/svn version control, project management and more, like the tools to create html pages like this one (`knitr` and `Rmarkdown`). Another source of user comfort are the `packages`. R comes with some basic packages installed, but you'll soon need to fit generalised linear mixture models, or visualise social networks using graph theory and that means you'll be searching for packages that allow you to do such things. A good place to start *package hunting* are the CRAN task view pages.

2. **Learn by running example `code`:** Copy the commands in the `code` blocks you find on this page, or any other tutorial or help files (e.g., Rob Kabacoff's Quick R). Paste them into an `.R` script file in the script (or, source) editor. In R Studio You can run code by pressing `cmd + enter` when the cursor is on a single single line, or you can run multiple lines at once by selecting them first. If you get stuck remember that there are expert R users who probably have answered your question already when it was posted on a forum. Search for example through the Stack overflow site for questions tagged with R)

3. **Examine what happens... when you tell R to make something happen:** R stores variables (anything from numeric data to functions) in an `Environment`. There are in fact many different environments, but we'll focus on the main workspace for the current R session. If you run the command `x <- 1+1`, a variable `x` will appear in the `Environment` with the value `2` assigned to it. Examining what happens in the `Environment` is not the same as examining the output of a statistical analysis. Output in R will appear in the `Console` window. Note that in a basic set-up each new R session starts with an empty `Environment`. If you need data in another session, you can save the entire `Environment`, or just some selected variables, to a file (`.RData`).

4. **Learn about the properties of R objects:** Think of objects as containers designed for specific content. One way to characterize the different objects

in R is by how picky they are about the content you can assign it. There are objects that hold `character` and `numeric` type data, a `matrix` for numeric data organised in rows and columns, a `data.frame` is a matrix that allows different data types in columns, and least picky of all is the `list` object. It can carry any other object, you can have a `list` of which item 1 is an entire `data.frame` and item 2 is just a `character` vector of the letter R. The most difficult thing to master is how to efficiently work with these objects, how to assign values and query contents.

5. **Avoid repeating yourself:** The R language has some amazing properties that allow execution of many repetitive algorithmic operations using just a few lines of code at speeds up to warp 10. Naturally, you'll need to be at least half Vulcan to master these features properly and I catch myself copying code when I shouldn't on a daily basis. The first thing you will struggle with are the `apply` functions. These functions pass the contents of a `list` object to a function. Suppose we need to calculate the means of column variables in 40 different SPSS `.sav` files stored in the folder DAT. With the `foreign` package loaded we can execute the following commands:

```
data <- lapply(dir("/DAT/",pattern=".sav$"),read.spss)
out  <- sapply(data,colMeans)
```

The first command applies read.spss to all files with a `.sav` extension found in the folder `/DAT`. It creates a data frame for each file which are all stored as elements of the list `data`. The second line applies the function `colMeans` to each element of `data` and puts the combined results in a matrix with dataset ID as columns (1-40), dataset variables as rows and the calculated column means as cells. This is just the beginning of the R magic, wait 'till you learn how to write functions that can create functions.

A.2 Getting started with R tutorials

- Tutorials on using **functions**:
  - Quick-R
  - Software Carpentry
  - Nicer Code
  - Advanced R
- Tutorials on using **conditionals** and **for loops**:
  - Quick-R
  - Software Carpentry
  - R-Bloggers
- Tutorials on the **-ply family** of functions: + R-bloggers + Nicer Code + R for Dummies
- **Plotting**, **plotting** and more **plotting**: + A Compendium of Clean Graphs in R + ggplot2 reference + ggplot2 extensions + patchwork, the ultimate ggplot2 extension + The R-graph gallery + Quick-R + Nicer Code
- Tutorial on **Effect Size Confidence Intervals** and more:
  - In this tutorial on estimating Effect Size Confidence Intervals (ESCI) there are a lot of examples on how to use R.
  - It was written as an addendum for a post on the **Open Science Collaboration Blog**, which contains many interesting entries on diverse subjects (like behavioural priming, theoretical amnesia and anonymous peer review)

A.3 Not new to R?

If you have been using R for a while, but do not consider yourself a master yet, I recommend learning to use the **tidyverse** packages and the accompanying web-book **R for data scientists**.

- Welcome to the **tidyverse**: + Install the tidyverse + Learn how to use the tidyverse + Learn how to use the tidyverse to do statistics + Learn how to use the tidyverse to create networks + How to make R purrr

A.4 Time series analyses in R

In this book you can find some tips on plotting time series (see section **Working with time series in R**) and we will be using package `casnet` as our main tool for analyses. However, if you really want a deep dive into everything related to time series in R be sure to check the CRAN task view on time series: https://cran.r-project.org/web/views/TimeSeries.html

Installing casnet

To install `casnet` you need to have package `devtools` or `remotes` installed and call the following code from the commands line:

```
library(devtools)
devtools::install_github("FredHasselman/casnet", dependencies = TRUE)

# or equivalently
library(devtools)
remotes::install_github("FredHasselman/casnet", dependencies = TRUE)
```

If all goes well this should install the package and all the packages it depends on. If the vignette build fails, don't worry, you can access them through the casnet website under *Articles*.

Appendix B

Working with time series in R

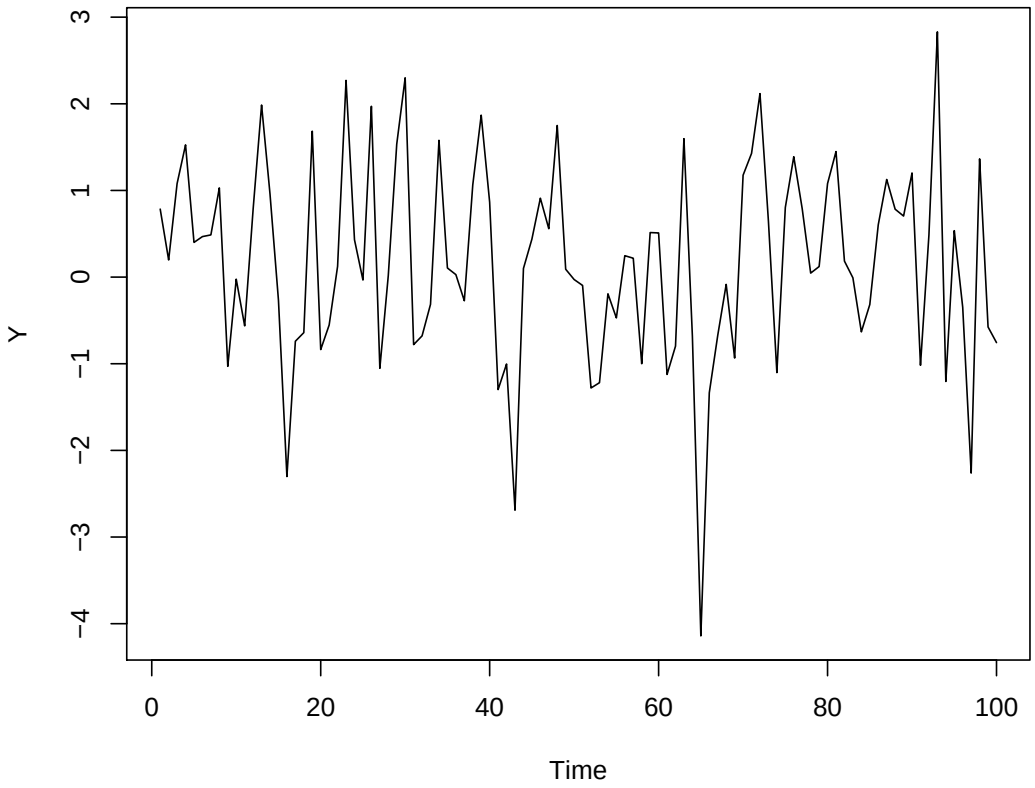
There are many ways to handle time series in `R`, this appendix provides some examples and suggest some best practices, based on the function `ts()`, which creates a time series object.

A time series object is expected to have a time-dimension on the x-axis. This is very convenient, because `R` will generate the time axis for you by looking at the *t*ime *s*eries *p*roperties attribute of the object. Even though we are not working with measurement outcomes, consider a value at a time-index in a time series object a **sample**:

- `Start` - The value of time at the first sample in the series (e.g., 0, or 1905)
- `End` - The value of time at the last sample in the series (e.g., 100, or 2005)
- `Frequency` - The amount of time that passed between two samples, or, the sample rate (e.g., 0.5, or 10)

Examples of using the time series object.

```
set.seed(2718282)
# Get a timeseries of 100 random numbers
Y <- ts(rnorm(100))
# plot.ts
plot(Y)
```




```
# Get sample rate info
tsp(Y)
```

```
> [1]   1 100   1
```

```
# Extract the time vector
time(Y)
```

```
> Time Series:
> Start = 1
> End = 100
> Frequency = 1
>  [1]  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 2
> [32] 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 5
> [63] 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 8
>  [94]  94  95  96  97  98  99 100
```

For now, these values are in principle all arbitrary units (`a.u.`). These settings only make sense if they represent the parameters of an actual measurement

procedure.

It is easy to adjust the time vector, by assigning new values using `tsp()` (values have to be possible given the time series length). For example, suppose the sampling frequency was 0.1 instead of 1 and the Start time was 10 and End time was 1000.

```
# Assign new values
(tsp(Y) <- c(10, 1000, .1))
```

```
> [1] 1e+01 1e+03 1e-01
```

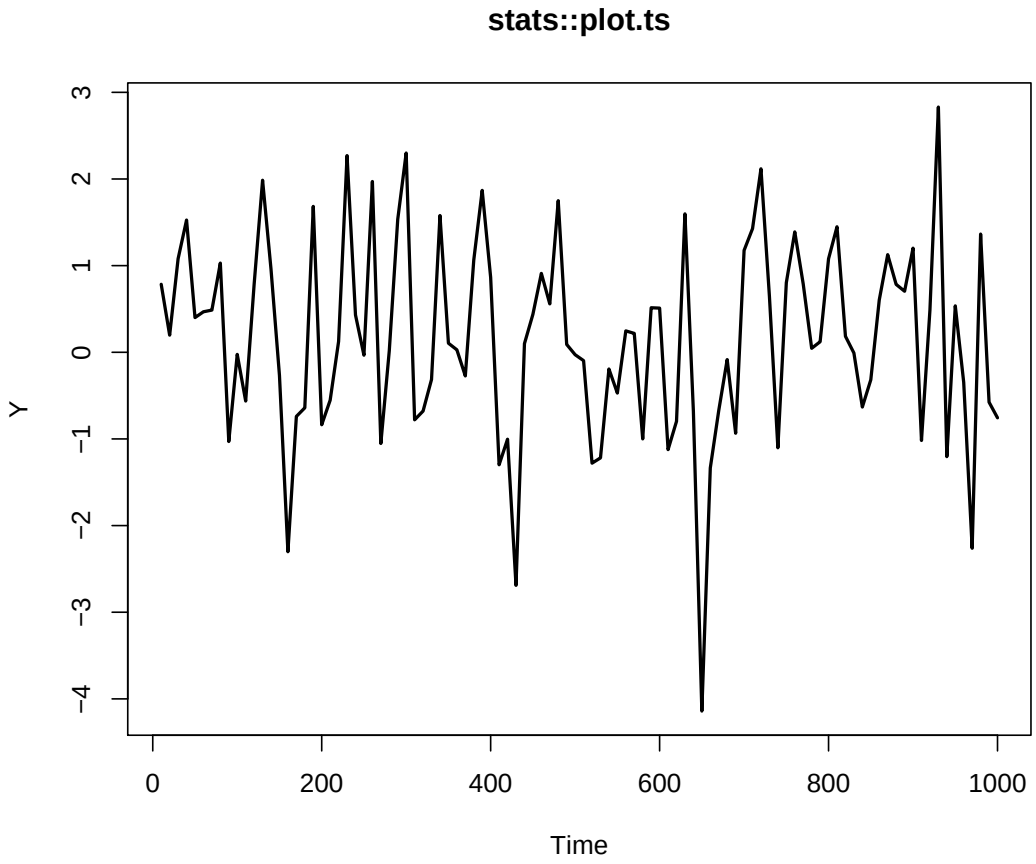
```
# Time axis is automatically adjusted
time(Y)
```

```
> Time Series:
> Start = 10
> End = 1000
> Frequency = 0.1
>  [1]  10  20  30  40  50  60  70  80  90 100 110 120 130 140 150 160 170 180 190 200 210 220 230
> [26] 260 270 280 290 300 310 320 330 340 350 360 370 380 390 400 410 420 430 440 450 460 470 4
> [51] 510 520 530 540 550 560 570 580 590 600 610 620 630 640 650 660 670 680 690 700 710 720 7
> [76] 760 770 780 790 800 810 820 830 840 850 860 870 880 890 900 910 920 930 940 950 960 970 9
```

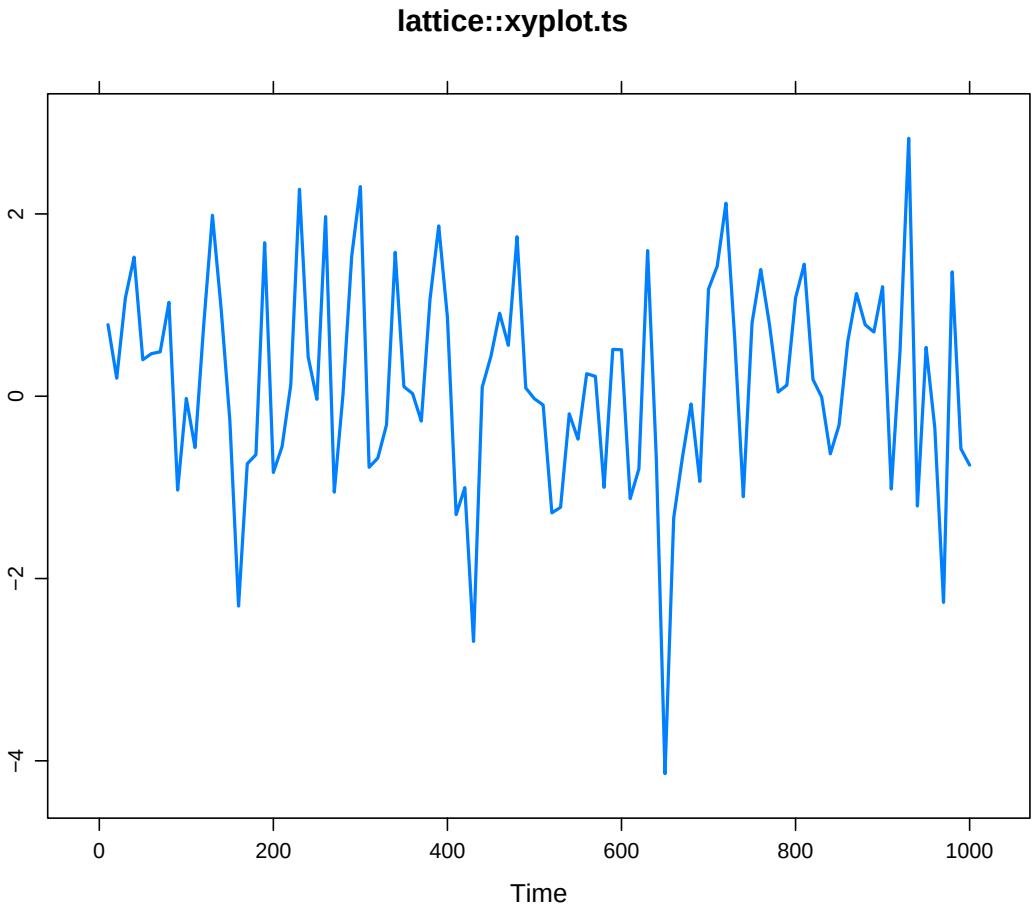
B.1 Plotting a ts object as a time series

Depending on which packages you use, there will be different settings applied to time series objects created by ts(). Below are some examples of differences between plotting routines.

```
require(lattice)       # Needed for plotting
require(latticeExtra)  # Needed for plotting
require(casnet)        # Need for ts_center()

# stats::plot.ts
plot(Y, lwd = 2, main = "stats::plot.ts")
```



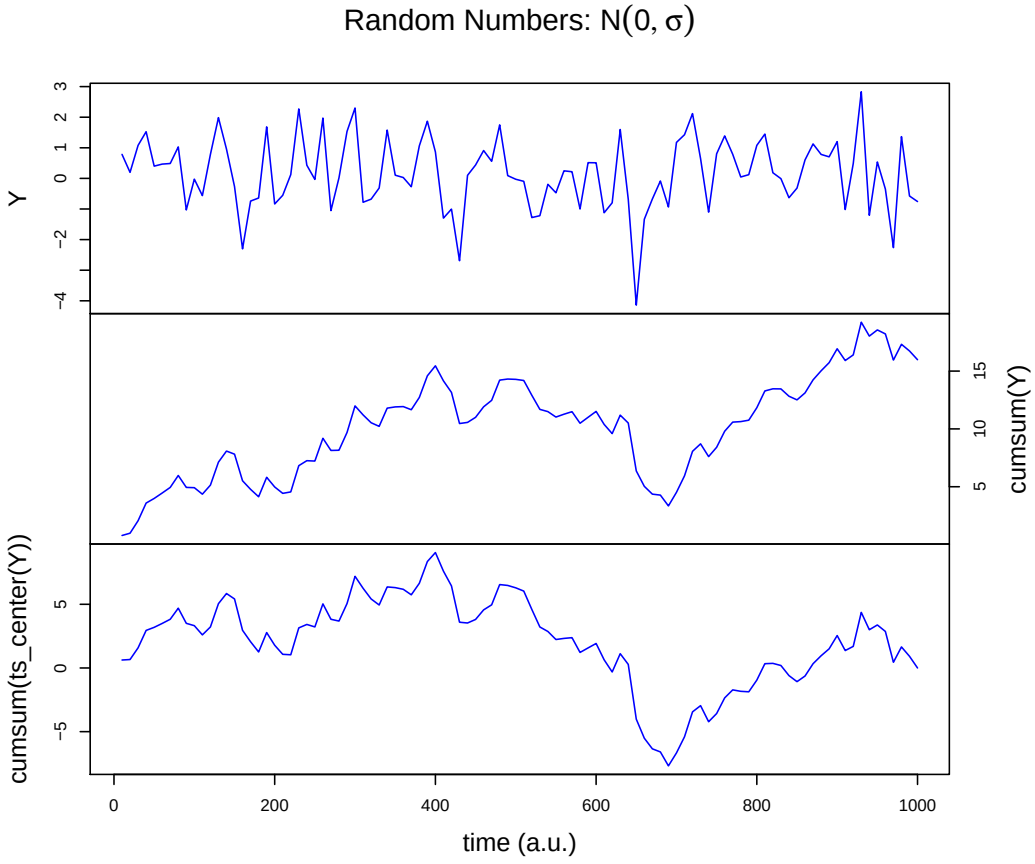

```
# lattice::xyplot.ts
xyplot(Y, lwd = 2, main = "lattice::xyplot.ts")
```


lattice::xyplot.ts
2
0
−2
−4
0
200
400
600
800
1000
Time

B.2 Plotting multiple time series in one figure

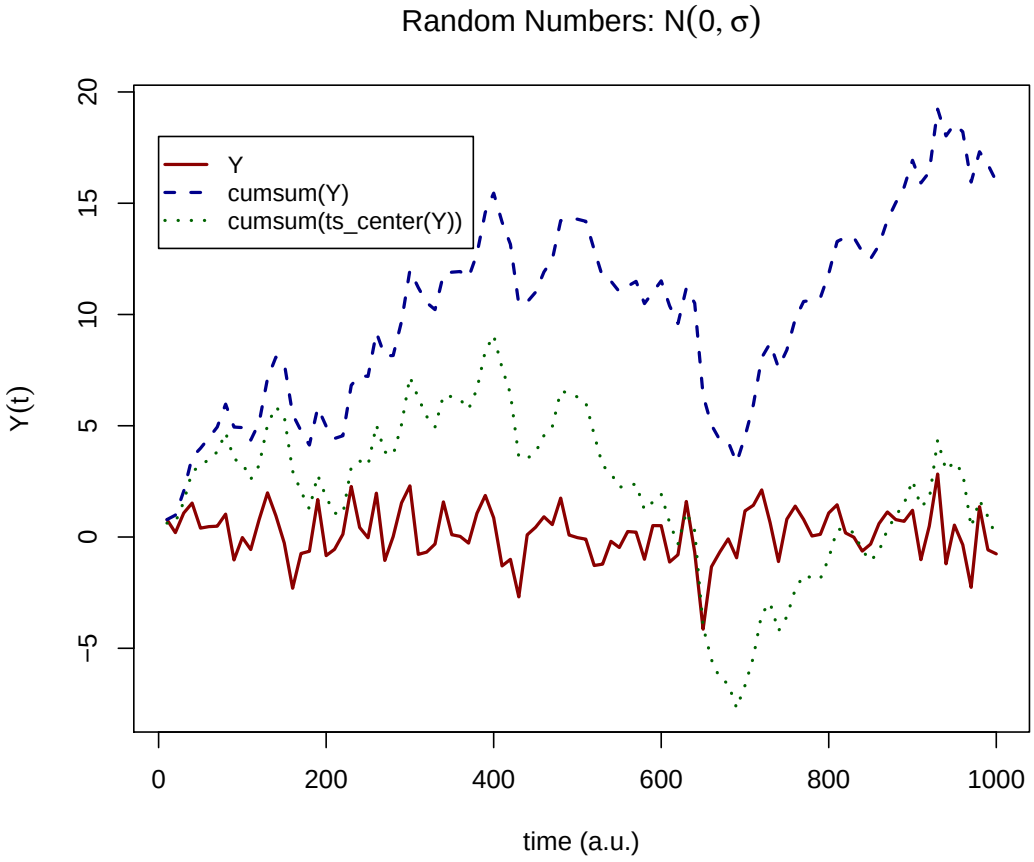
Plot multiple time series in frames with `plot.ts()` in `package::stats`. This function takes a matrix as input, here we use `cbind( ... )`.

```
# stats::plot.ts
plot(cbind(Y,
           cumsum(Y),
           cumsum(ts_center(Y))
           ),
     yax.flip = TRUE, col = "blue", frame.plot = TRUE,
     main = expression(paste("Random Numbers: ",N(0,sigma))), xlab = "time (a.u.)")
```



Plot multiple time series in one graph with `ts.plot()` in `package::graphics`. This function can handle multiple `ts` objects as arguments.

```r
# graphics::ts.plot
ts.plot(Y,
        cumsum(Y),
        cumsum(ts_center(Y)),
        gpars = list(xlab = "time (a.u.)",
                     ylab = expression(Y(t)),
                     main = expression(paste("Random Numbers: ",N(0,sigma))),
                     lwd = rep(2,3),
                     lty = c(1:3),
                     col = c("darkred","darkblue","darkgreen")
                     )
        )
legend(0, 18, c("Y","cumsum(Y)", "cumsum(ts_center(Y))"), lwd = rep(2,3), lty = c(1:3), col = c("darkred
```

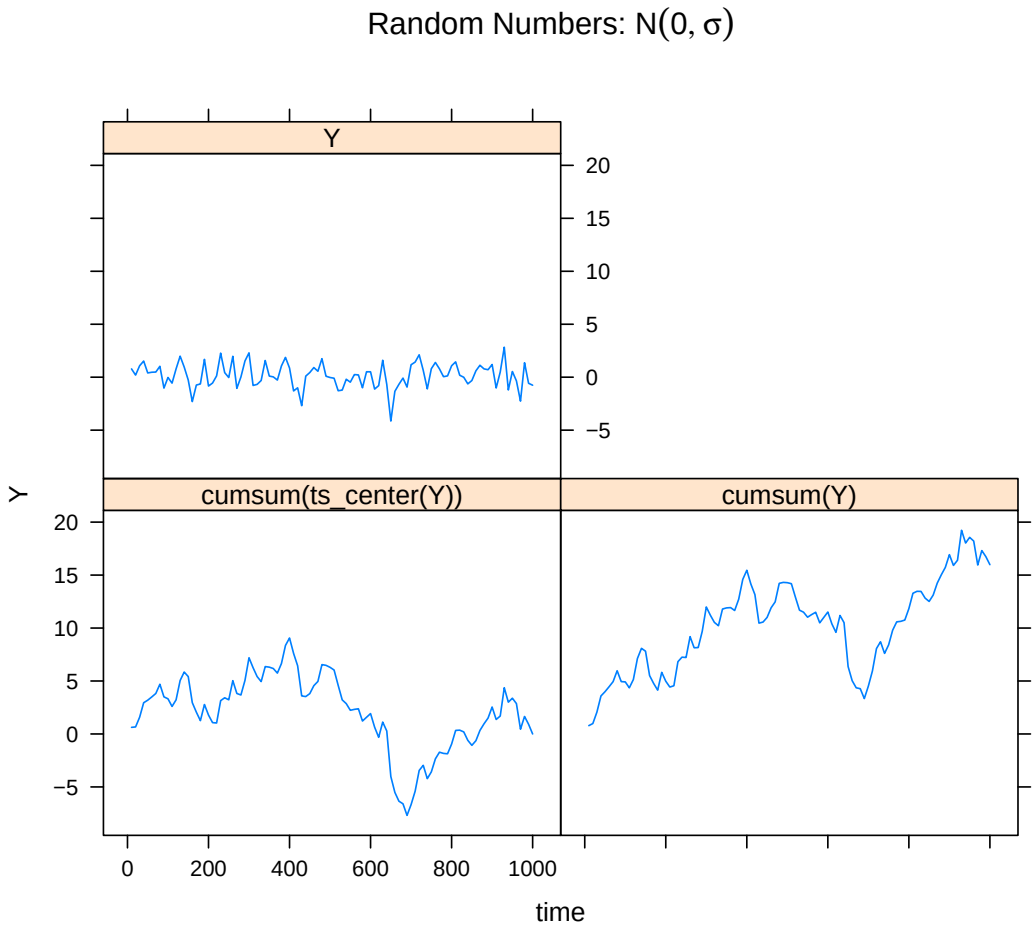


Use `xyplot()` in `package::lattice` to create a plot with panels. The easiest way to

do this is to create a dataset in so-called “long” format. This means the variable to plot is in 1 column and other variables indicate different levels, or conditions under which the variable was observed or simulated.

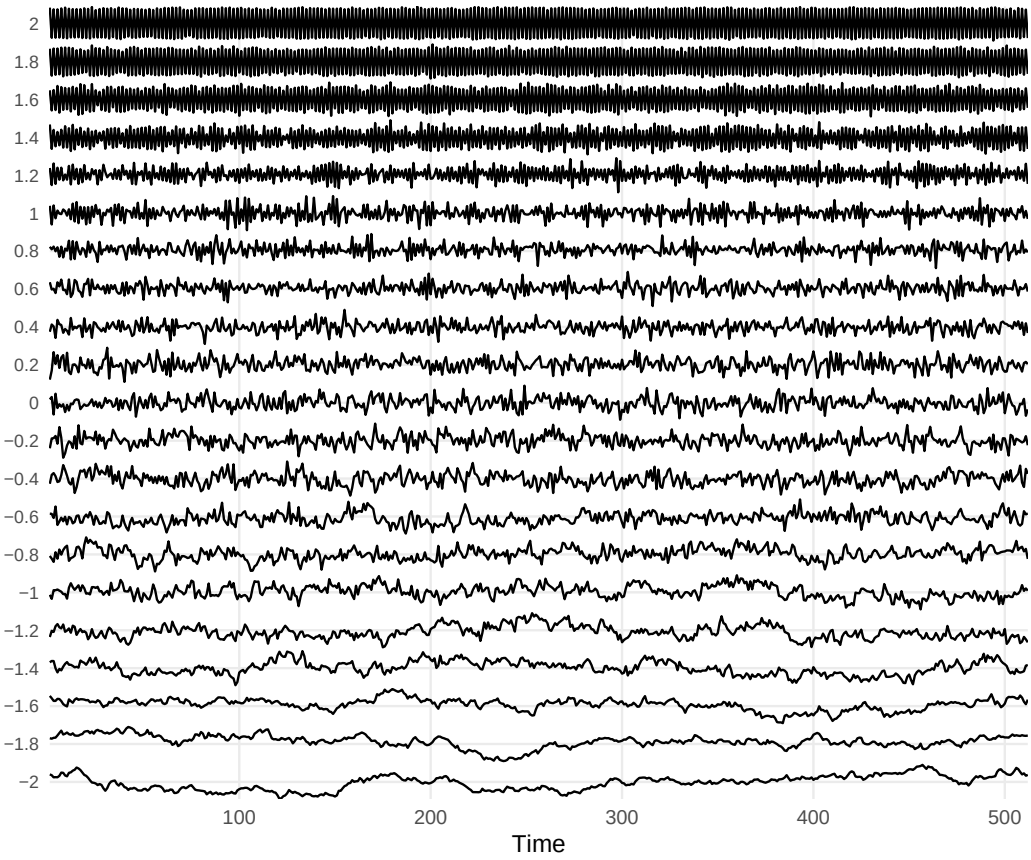
Function `ldply()` is used to generate Y for three different settings of r. The values of r are passed as a **l**ist and after a function is applied the result is returned as a **d**ataframe.

```
require(plyr)          # Needed for function ldply()

# Create a long format dataframe for various values for `r`
data <- cbind.data.frame(Y     = c(as.numeric(Y), cumsum(Y), cumsum(ts_center(Y))),
                         time  = c(time(Y), time(Y), time(Y)),
                         label = factor(c(rep("Y",length(Y)),  rep("cumsum(Y)",length(Y)), rep(
                         )
# Plot using the formula interface
xyplot(Y ~ time | label, data = data, type = "l", main = expression(paste("Random Numbers: ",N(
```



Or, if you have very may time series, you can use the function `PLOT()` in `casnet`.

```
# Create a data frame with time series
# Generate some coloured noise
N <- 512
noises <- seq(-2,2,by=.2)
y <- data.frame(matrix(rep(NA,length(noises)*N), ncol=length(noises)))

for(c in seq_along(noises)){y[,c] <- noise_powerlaw(N=N, alpha = noises[c])}
colnames(y) <- paste0(noises)

plotTS_multi(y)
```

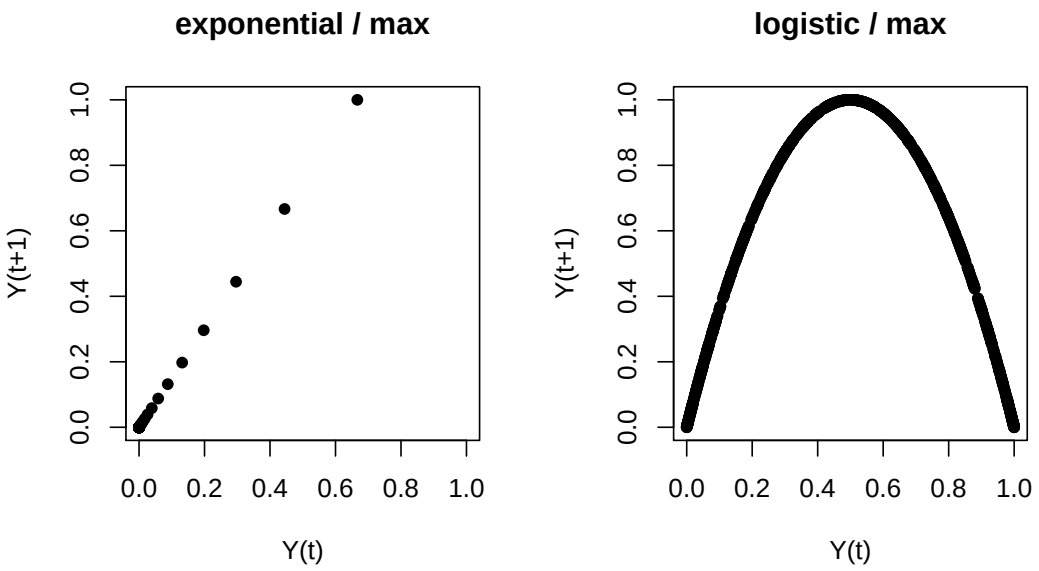


Note that the y-axis is rescaled for each series and does not reflect magnitude differences between the series.

B.3 The return plot

To create a return plot the values of Y have to be shifted by a certain lag. The functions `lead()` and `lag()` in `package::dplyr` are excellent for this purpose (note that `dplyr::lag()` behaves different from `stats::lag()`).

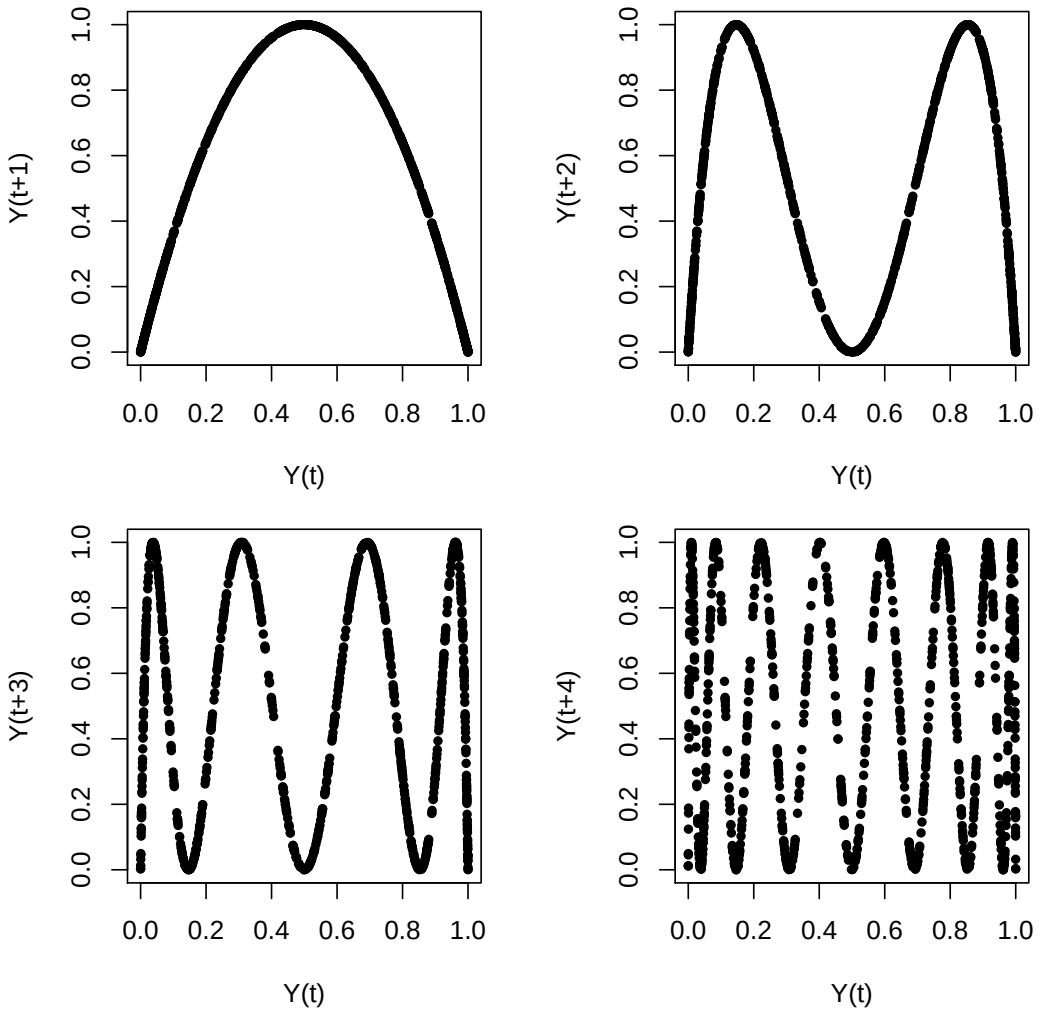
```
# Function lag() and lead()
library(dplyr)
library(casnet)

# Get exponential growth
YY <- growth_ac(N=1000,r=1.5,type = "driving")
Y1 <- as.numeric(YY/max(YY))
# Get logistic growth in the chaotic regime
Y2 <- as.numeric(growth_ac(N=1000,r=4,type = "logistic"))
# Use the `lag` function from package `dplyr`
op <- par(mfrow = c(1,2), pty = "s")
plot.ts(dplyr::lag(Y1), Y1, xy.labels = FALSE, pch = 16, xlim = c(0,1), ylim = c(0,1), xlab = "Y(t)", yl
     main = "exponential / max")
plot.ts(dplyr::lag(Y2), Y2, xy.labels = FALSE, pch = 16, xlim = c(0,1), ylim = c(0,1), xlab = "Y(t)", yl
     main = "logistic / max")
```



```
par(op)
```

Use `l_ply()` from `package::plyr` to create return plots with different lags. The **l_** before **ply** means the function will take a **l**ist as input to a function, but it will not expect any data to be returned, for example in the case of a function that is used to plot something.

```
# Explore different lags
op <- par(mfrow = c(1,2), pty = "s")
plyr::l_ply(1:4, function(l) plot.ts(dplyr::lag(Y2, n = l), Y2, xy.labels = FALSE, pch = 16, xl
```



```
par(op)
```

B.4 Using ggplot2

Becoming proficient at ggplot2 can take some time, but it does pay off. One of the problems with plotting time series data is that ggplot2 wants tidy data in *long* format. Tidy data is:

> Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types. In tidy data: 1. Each variable forms a column. 2. Each observation forms a row. 3. Each type of observational unit forms a table.
>
> ---**?**

So if we have a set of time series as in the previous examples, we need to change it to long format.

```
library(tidyverse)

# A wide data frame
df.wide <- data.frame(rnormY        = Y,
                      cumsumY       = cumsum(Y),
                      centercumsumY = cumsum(ts_center(Y)),
                      time          = seq_along(Y)
                      )

glimpse(df.wide)
```

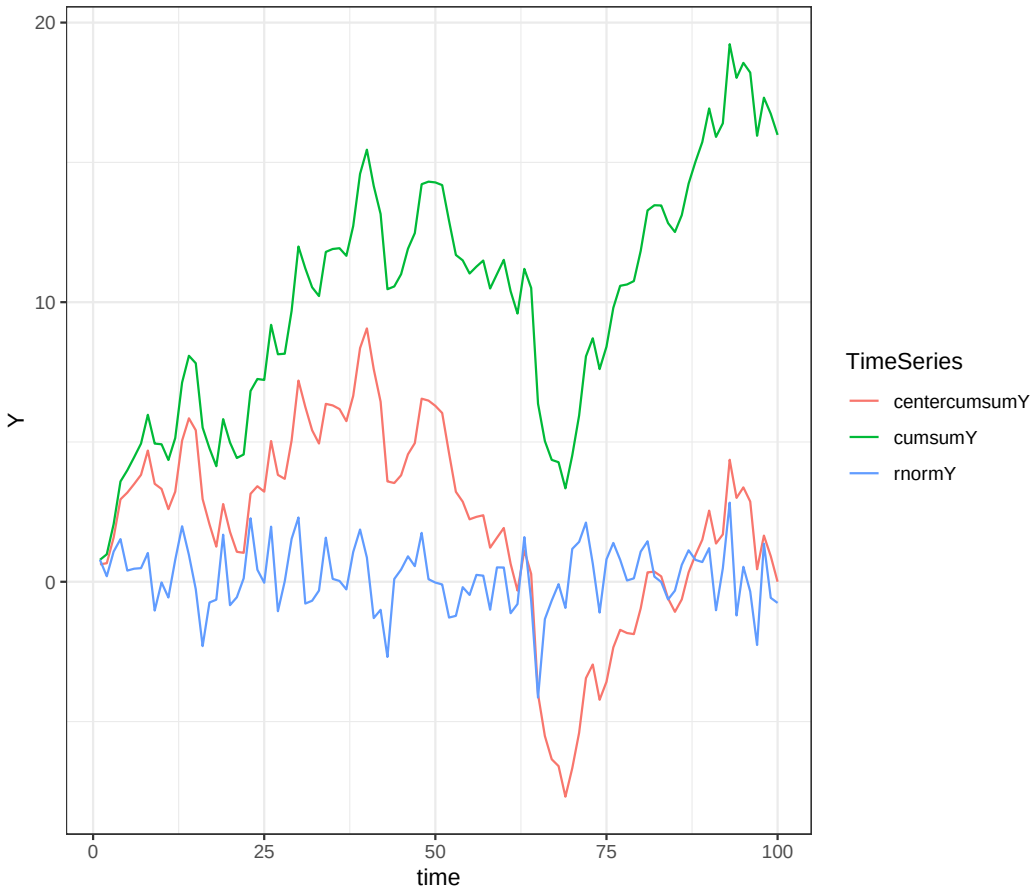
```
> Rows: 100
> Columns: 4
> $ rnormY      <dbl> 0.78482166, 0.19776074, 1.07957851, 1.52605836, 0.40026016, 0.46755371, 0.486
1.029885...
> $ cumsumY     <dbl> 0.7848217, 0.9825824, 2.0621609, 3.5882193, 3.9884794, 4.4560331, 4.9427715,
> $ centercumsumY <dbl> 0.6249966, 0.6629322, 1.5826857, 2.9489189, 3.1893540, 3.4970826, 3.8239959
> $ time       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
```

```
# Create a long dataframe using gather()
df.long <- df.wide %>%
  gather(key=TimeSeries,value=Y,-"time")
```

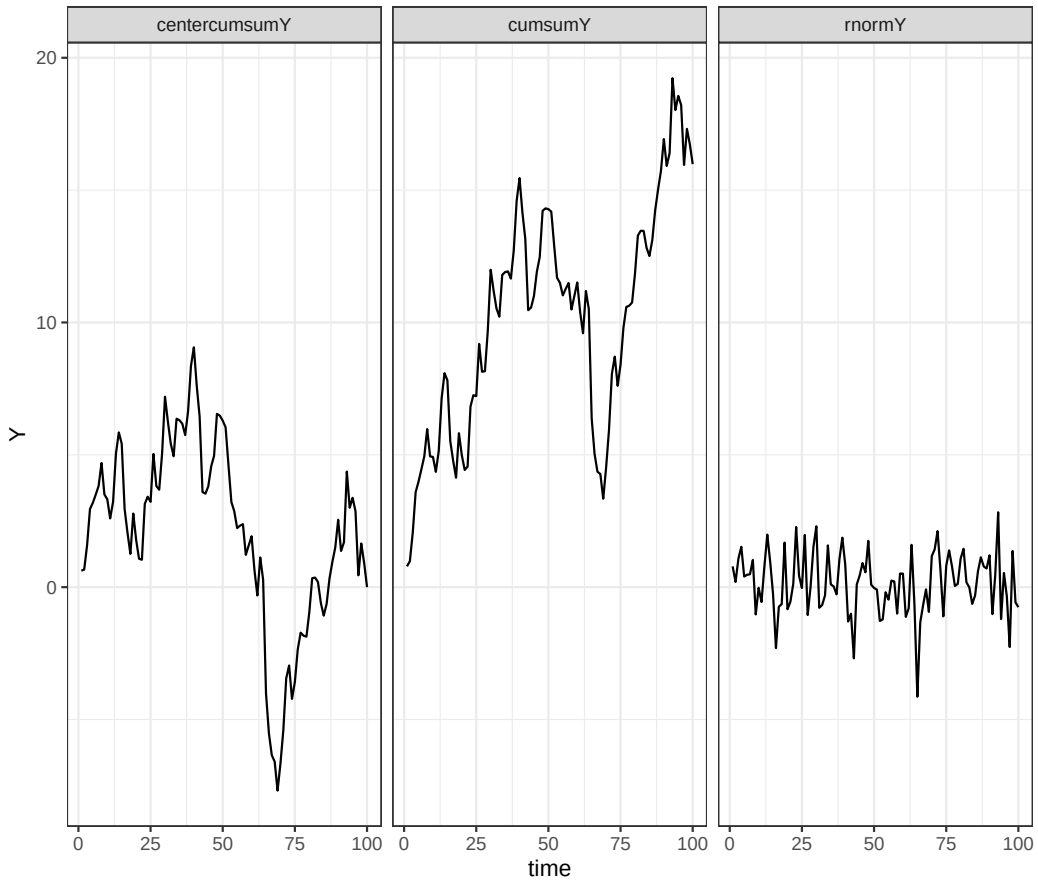
```
glimpse(df.long)
```

```
> Rows: 300
> Columns: 3
> $ time     <int> 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,
> $ TimeSeries <chr> "rnormY", "rnormY", "rnormY", "rnormY", "rnormY", "rnormY", "rnormY", "rnormY", "rnormY
> $ Y        <dbl> 0.78482166, 0.19776074, 1.07957851, 1.52605836, 0.40026016, 0.46755371, 0.48673832, 1.0295
1.02988553,...
```

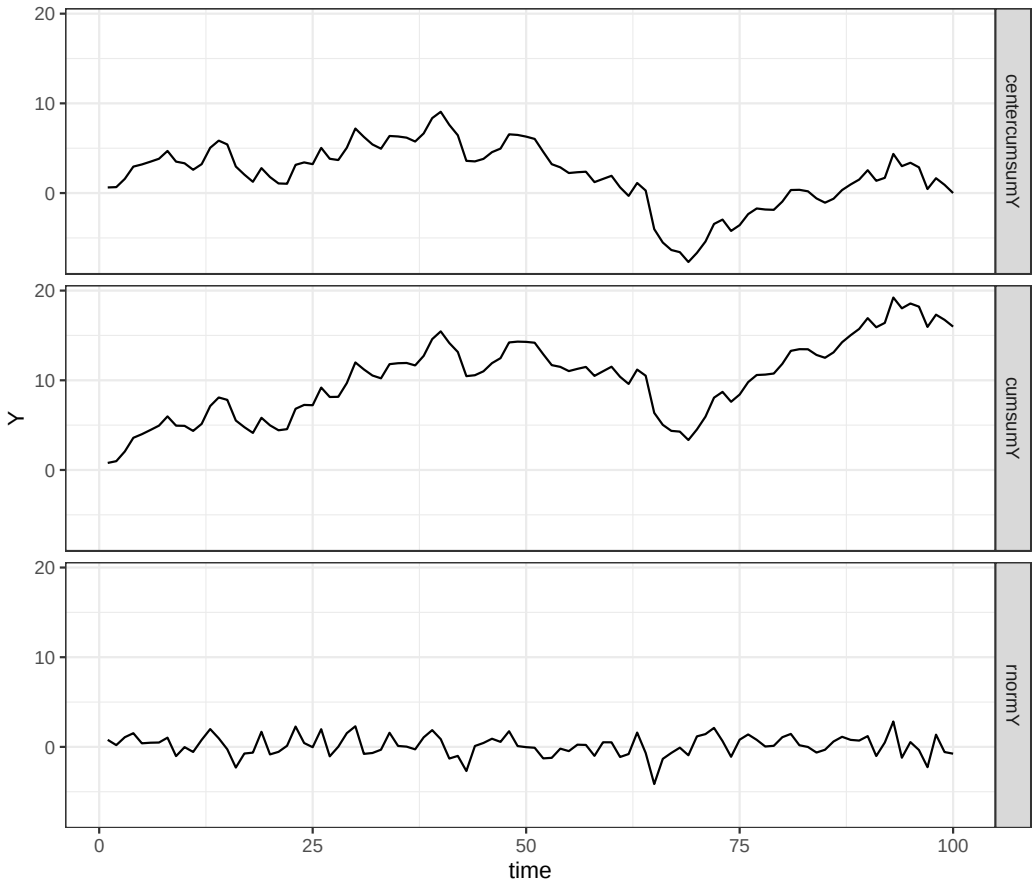
```
# 1 plot
ggplot(df.long, aes(x=time, y=Y, colour=TimeSeries)) +
  geom_line() +
  theme_bw()
```



```
# using facets
ggplot(df.long, aes(x=time,y=Y)) +
  geom_line() +
  facet_wrap(~TimeSeries) +
  theme_bw()
```

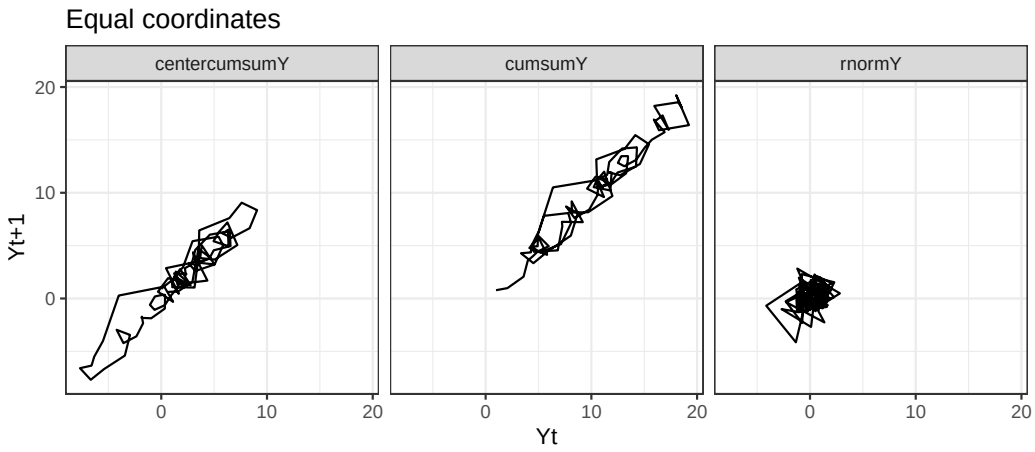



```
# using facets
ggplot(df.long, aes(x=time,y=Y)) +
  geom_line() +
  facet_grid(TimeSeries~.) +
  theme_bw()
```

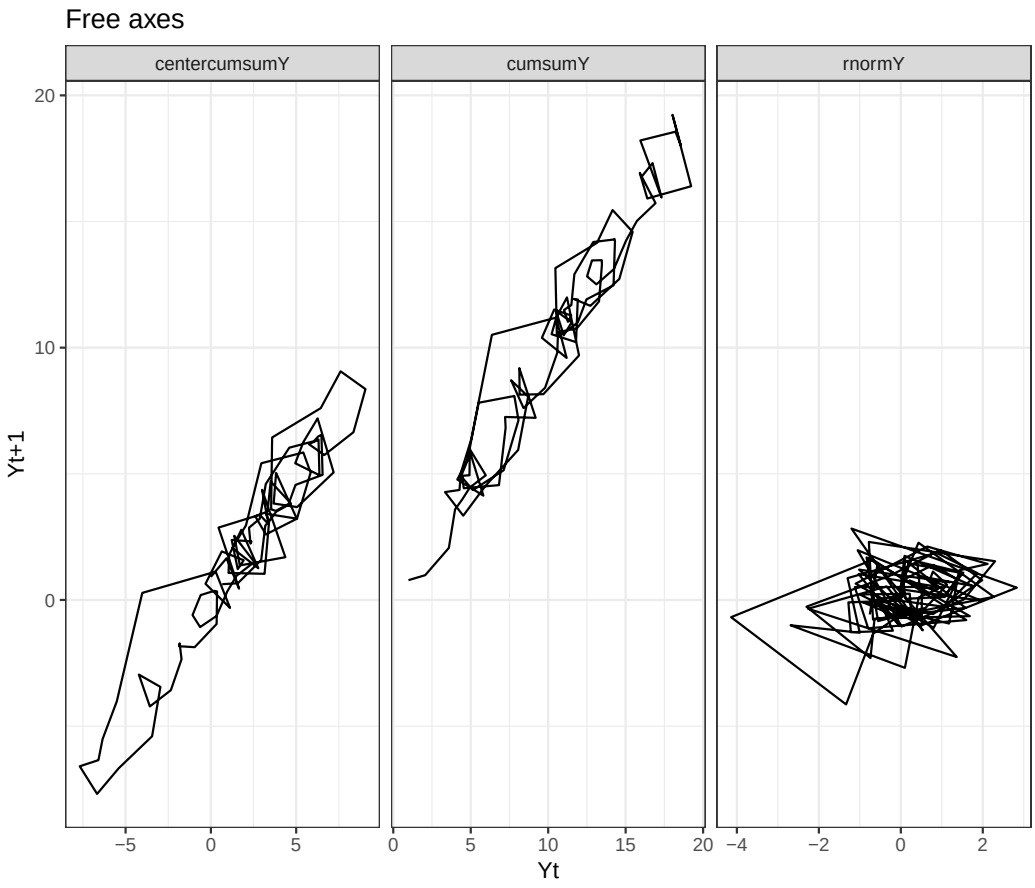


To create a return plot you can use `geom_path()` instead of `geom_line()` and make the area square using `coord_equal()`.

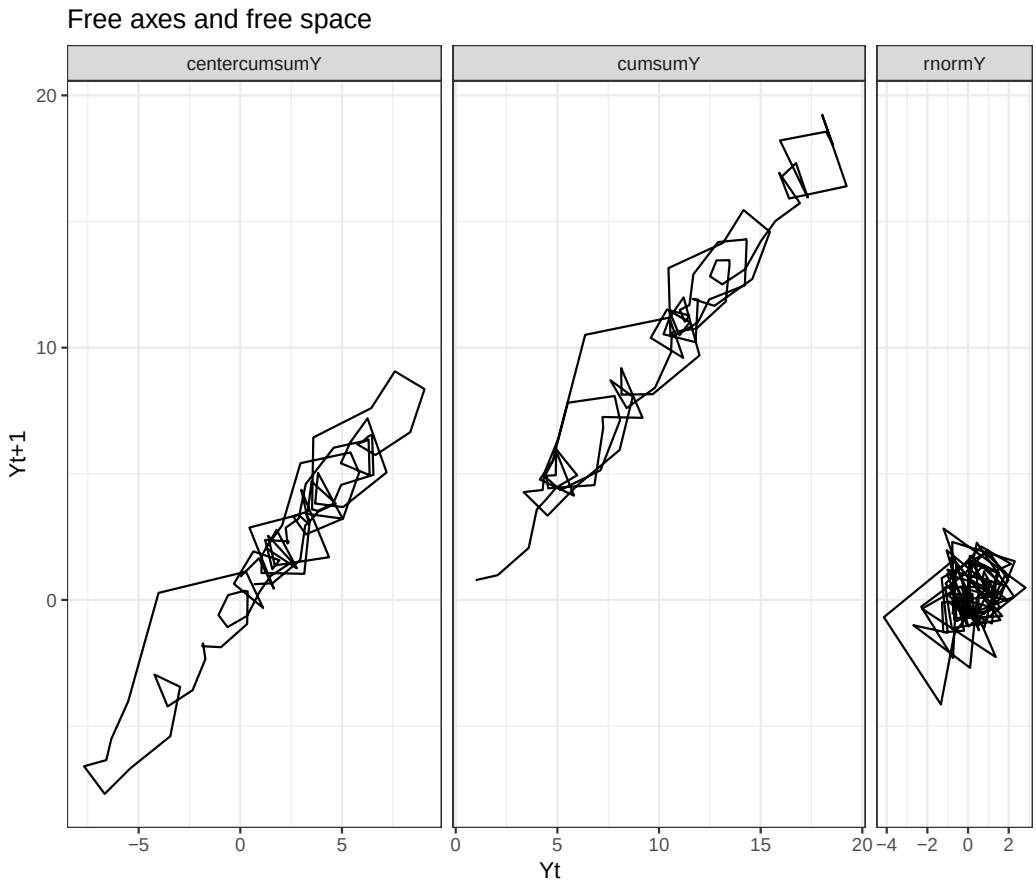
```
# Add a lagged variable
df.long <- df.long %>%
  group_by(TimeSeries) %>%
  mutate(Ylag = dplyr::lag(Y))

# Use geom-path()
ggplot(df.long, aes(x=Y,y=Ylag,group=TimeSeries)) +
  geom_path() +
  facet_grid(.~TimeSeries) +
  theme_bw() +
  labs(title = "Equal coordinates", x="Yt",y="Yt+1") +
  coord_equal()
```




```
# You could also have free axes
ggplot(df.long, aes(x=Y,y=Ylag,group=TimeSeries)) +
  geom_path() +
  facet_grid(.~TimeSeries, scales = 'free') +
  labs(title="Free axes", x="Yt",y="Yt+1") +
  theme_bw()
```




```
# Or free axes and a free space
ggplot(df.long, aes(x=Y,y=Ylag,group=TimeSeries)) +
  geom_path() +
  facet_grid(.~TimeSeries, scales = 'free', space = 'free') +
  labs(title="Free axes and free space", x="Yt",y="Yt+1") +
  theme_bw()
```


Free axes and free space
centercumsumY
cumsumY
rnormY
Yt+1
Yt
20
10
0
−5
0
5
0
5
10
15
20
−4
−2
0
2

Appendix C

Dealing with Missing Data

In an article called "Much ado about nothing"? several different approaches, methods and best practices for dealing with missing data are discussed. The methods are diverse, both in number and in their effect on the results of analyses. Therefore, the first rule of dealing with missing data is: **Always report analysis results for the imputed data as well as the data with missing values removed!**

The CRAN taskview on missing data is a good starting point for finding what you may need. In this vignette we will specifically discuss package imputeTS for *Time Series Missing Value Imputation* and mice *Multivariate Imputation by Chained Equations*.

Data with missing values

We'll create some variables from which we artifically remove datapoints. This allows us to evaluate how well the imputation methods perform in recovering the true values.

```
set.seed(54321)
# Random normally distributed numbers
zscore <- rnorm(n = 122)
df_vars <- data.frame(zscore = zscore)
# Random discrete uniform numbers
df_vars$unif_discrete <- unif_discrete  <- round(runif(NROW(df_vars),min = 0,max = 6))
```

```
df_vars$unif_discrete[c(5,10:15,74:78,102,111,120)] <- NA
# Unordered catagorical
df_vars$cat_unordered <- cat_unordered  <- factor(round(runif(NROW(df_vars),min = 1,max = 7)))
df_vars$cat_unordered[c(5,10:15,74:78,102,111,120)] <- NA
# Ordered categroical
df_vars$cat_ordered <- cat_ordered <- ordered(round(runif(NROW(df_vars),min = 1,max = 20)))
```

We'll also load the data analysed by Bastiaansen et al. (2019) and select some variables which have missing values.

```
# # Load data from OSF https://osf.io/tcnpd/
# require(osfr)
# manyAnalystsESM <- rio::import(osfr::osf_download(osfr::osf_retrieve_file("tcnpd") , overwrit

# Or use the internal data
data(manyAnalystsESM)

# We want to use these variables
# Note: the infix function '%ci%' is from package 'invctr'
vars    <- c("angry"%ci%manyAnalystsESM,"ruminate"%ci%manyAnalystsESM,"hours"%ci%manyAnalystsES

df_vars <-  cbind(df_vars,manyAnalystsESM[,vars])

# Give zscore and ordered categorical gthe same NAs as variable 'angry'
df_vars$zscore[is.na(df_vars$angry)] <- NA
df_vars$cat_ordered[is.na(df_vars$angry)] <- NA
```

Function `imputeTS::statsNA()` can produce some helpful statistics on the `NA`s that might be present in your data.

```
require(imputeTS)

# The variable 'angry'
imputeTS::statsNA(df_vars$angry)
```

```
> [1] "Length of time series:"
> [1] 122
> [1] "-------------------------"
> [1] "Number of Missing Values:"
```

```
> [1] 9
> [1] "-------------------------"
> [1] "Percentage of Missing Values:"
> [1] "7.38%"
> [1] "-------------------------"
> [1] "Number of Gaps:"
> [1] 8
> [1] "-------------------------"
> [1] "Average Gap Size:"
> [1] 1.125
> [1] "-------------------------"
> [1] "Stats for Bins"
> [1] "  Bin 1 (31 values from 1 to 31) :      1 NAs (3.23%)"
> [1] "  Bin 2 (31 values from 32 to 62) :      0 NAs (0%)"
> [1] "  Bin 3 (31 values from 63 to 93) :      3 NAs (9.68%)"
> [1] "  Bin 4 (29 values from 94 to 122) :      5 NAs (17.2%)"
> [1] "-------------------------"
> [1] "Longest NA gap (series of consecutive NAs)"
> [1] "2 in a row"
> [1] "-------------------------"
> [1] "Most frequent gap size (series of consecutive NA series)"
> [1] "1 NA in a row (occurring 7 times)"
> [1] "-------------------------"
> [1] "Gap size accounting for most NAs"
> [1] "1 NA in a row (occurring 7 times, making up for overall 7 NAs)"
> [1] "-------------------------"
> [1] "Overview NA series"
> [1] "  1 NA in a row: 7 times"
> [1] "  2 NA in a row: 1 times"
```

```
# Uniform discrete numbers
imputeTS::statsNA(df_vars$unif_discrete)
```

```
> [1] "Length of time series:"
> [1] 122
> [1] "-------------------------"
> [1] "Number of Missing Values:"
> [1] 15
```

```
> [1] "-------------------------"
> [1] "Percentage of Missing Values:"
> [1] "12.3%"
> [1] "-------------------------"
> [1] "Number of Gaps:"
> [1] 6
> [1] "-------------------------"
> [1] "Average Gap Size:"
> [1] 2.5
> [1] "-------------------------"
> [1] "Stats for Bins"
> [1] "  Bin 1 (31 values from 1 to 31) :      7 NAs (22.6%)"
> [1] "  Bin 2 (31 values from 32 to 62) :      0 NAs (0%)"
> [1] "  Bin 3 (31 values from 63 to 93) :      5 NAs (16.1%)"
> [1] "  Bin 4 (29 values from 94 to 122) :      3 NAs (10.3%)"
> [1] "-------------------------"
> [1] "Longest NA gap (series of consecutive NAs)"
> [1] "6 in a row"
> [1] "-------------------------"
> [1] "Most frequent gap size (series of consecutive NA series)"
> [1] "1 NA in a row (occurring 4 times)"
> [1] "-------------------------"
> [1] "Gap size accounting for most NAs"
> [1] "6 NA in a row (occurring 1 times, making up for overall 6 NAs)"
> [1] "-------------------------"
> [1] "Overview NA series"
> [1] "  1 NA in a row: 4 times"
> [1] "  5 NA in a row: 1 times"
> [1] "  6 NA in a row: 1 times"
```

C.1 Univariate imputation

In addition to useful summary and visualisation tools, package `imputeTS` contains a number of imputation methods that are commonly used. If you have installed the package, run `vignette("imputeTS-Time-Series-Missing-Value-Imputation-in-R", package = "imputeTS")` from the console ans learn about all the options.

Linear interpolation

One of the most straightforward inputation methods is linear interpolation. This is a relatively sensible method if there is just one time point missing. However, when several values are missing in a row, the linear interpolation might be unrealistic. Other methods that will give less plausible results for imputation of multiple missing values in a row are *last observation carried forward* and *next observation carried backward*, also available in `imputeTS` as `na_locf(type = "locf")`, and `na_locf(type = "nocb")` respectively.

We'll generate a data set with linear interpolation (also available are `spline` and `stine` interpolation), to compare to the more advanced multiple imputation methods discussed below.

```
out.linear <- t(laply(1:NCOL(df_vars), function(c){
  y  <- as.numeric(as.numeric_discrete(x = df_vars[,c], keepNA = TRUE))
  idNA <- is.na(y)
  yy <- cbind(imputeTS::na_interpolation(y,option = "linear"))
  if(all(is.wholenumber(y[!idNA]))){
    return(round(yy))
  } else {
      return(yy)
    }
  }))
colnames(out.linear) <- colnames(df_vars)
```

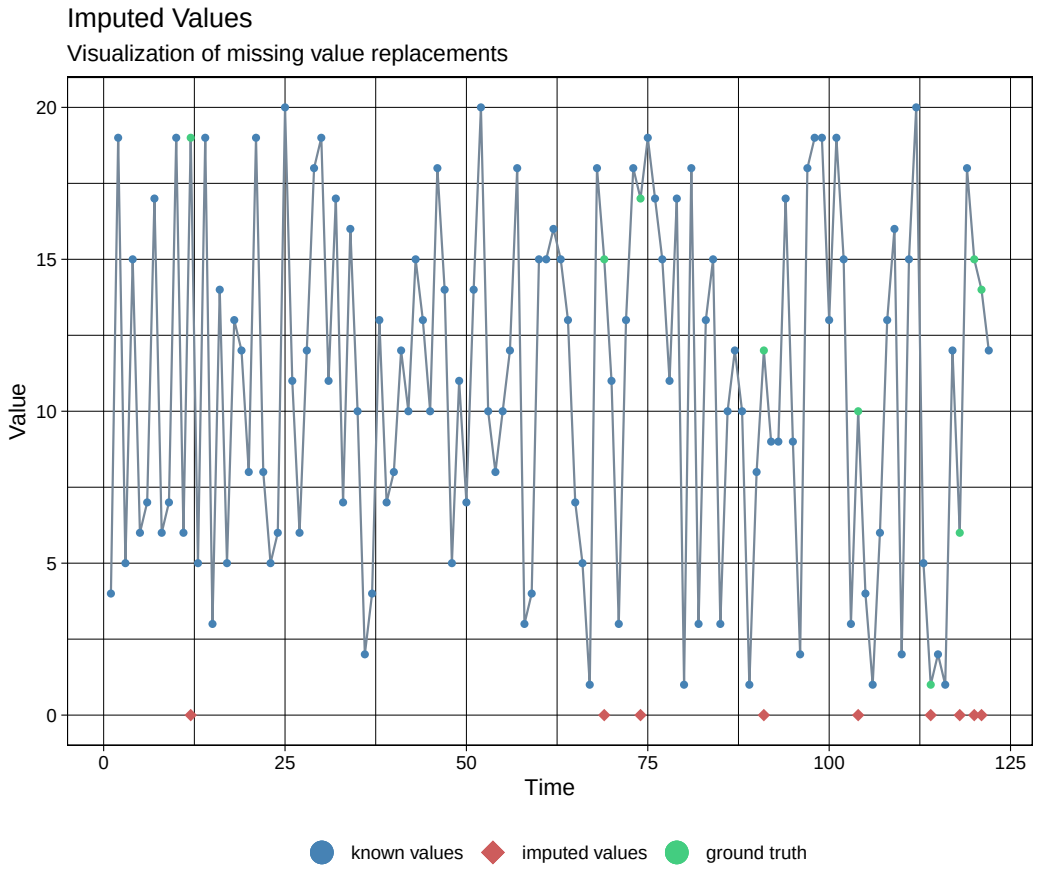
Note that we need to round the imputed values to get discrete values if the original variable was discrete.

Kalman filter

Imputation by using the Kalman filter is a powerful method for imputing data. However, when dealing with discrete data, one has to take some additional steps in order to get meaningful results.

For example, with uniform discrete numbers and/or scales that are bounded (eg. visual analog scale from `0-100`), the Kalman method will not correctly impute the data and might go outsdide the bounds of the scale.

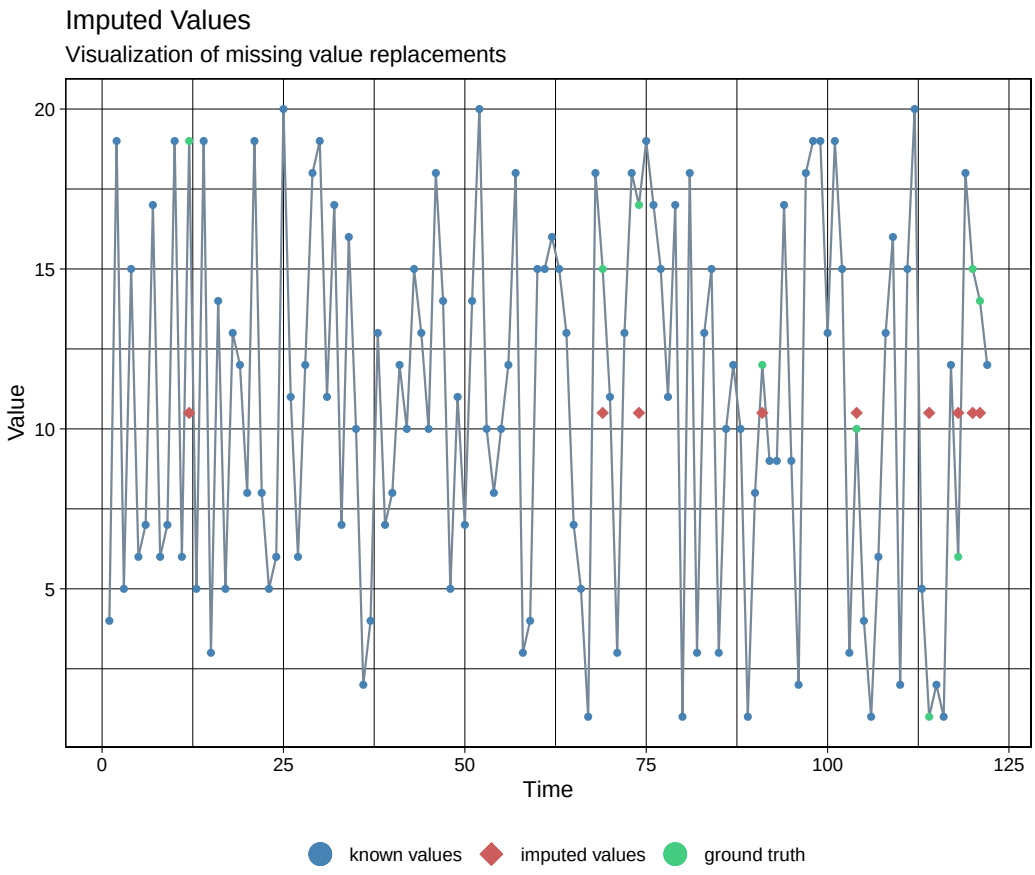
```
# Use casnet::as.numeric_discrete() to turn a factor or charcter vector into a named numeric ve
ca <- as.numeric_discrete(df_vars$cat_ordered, keepNA = TRUE)
imputations <- imputeTS::na_kalman(ca, model = "auto.arima")
imputeTS::ggplot_na_imputations(x_with_na = ca, x_with_truth = as.numeric_discrete(cat_ordered)
```



There is a way to adjust the imputation procedure by transforming the data

(thanks to Steffen Moritz, author of `imputeTS`, for suggesting this method). The ordered categorical series was created with bounds `1` and `20`.

```
# Bounds
lo <- 1
hi <- 20
# Transform data, take care of dividsion by 0
ca_t <- log(((ca-lo)+.Machine$double.eps)/((hi-ca)+.Machine$double.eps))
imputations <- imputeTS::na_kalman(ca_t, model = "auto.arima")
# Plot the result
# Back-transform the imputed forecasts
imputationsBack <- (hi-lo)*exp(imputations)/(1+exp(imputations)) + lo

imputeTS::ggplot_na_imputations(x_with_na = ca, x_with_truth = as.numeric_discrete(cat_ordered), x_with_
```



C.2 Multiple imputation

Package mice implements a method called: *Multivariate Imputation by Chained Equations*. The main function `mice()` will try to select an appropriate method based on the type of variable (discrete, continuous, etc.). In general, the advantage of using `mice()` with discrete data is that it has a number of methods that will actually return discrete values.

Check the manual page for mice (e.g. type `?mice` in the console), to see the 25 methods that are available. On that manual page you can also find links to a number of vignettes that provide a very thorough explanation of all the functions the package has to offer.

In this vignette, we will focus on a simple demonstration of just a few of the methods in `mice()`.

Auto-select method

We can just provide the `mice()` function our data set and it will take care of analysing the variables and selecting an appropriate imputation method.

```
require(mice)
# auto choice by mice algorithm
imp.mice <- mice::mice(df_vars)
```

```
>
>  iter imp variable
>  1  1 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  1  2 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  1  3 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  1  4 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  1  5 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  2  1 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  2  2 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  2  3 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours*
>  2  4 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  2  5 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  3  1 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours*
>  3  2 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  3  3 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
```

```
>  3  4 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  3  5 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  4  1 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  4  2 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  4  3 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  4  4 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  4  5 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  5  1 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  5  2 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  5  3 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  5  4 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
>  5  5 zscore unif_discrete cat_unordered cat_ordered angry ruminate hours
```

The algorithm chooses methods `pmm`, `polyreg` and `polr`:

```
imp.mice$method
```

```
>      zscore unif_discrete cat_unordered   cat_ordered       angry     ruminate       hours
>       "pmm"       "pmm"    "polyreg"      "polr"       "pmm"       "pmm"       "pmm"
```

By default `mice()` will generate `5` iterations of each time series, that is, argument `maxit = 5`. If you inspect the `imp.mice` object you can see it is a list with several fields, the field `imp` is another list with fields named after the columns in our data set. Each field contains `5` iterations for the variable.

```
lengths(imp.mice$imp)
```

```
>      zscore unif_discrete cat_unordered   cat_ordered       angry     ruminate       hours
>           5           5           5           5           5           5           5
```

To generate replacements for the missing values from those `5` iterations we need to call the function `complete()`.

```
out.auto <- mice::complete(imp.mice)
```

Check the `complete()` manual entry for some other interesting options.

Classification & regression trees

We also choose an imputation method for all variables, one based on classification and regression trees (`cart`), it will give the same results as the method based on random forest imputation (`rf`).

```r
# RF and CART return (identical) discrete numbers
imp.cart  <- mice(df_vars, meth = 'cart', printFlag = FALSE)
out.cart  <- complete(imp.cart)

# imp.rf  <- mice(df_vars, meth = 'rf')
# out.rf  <- complete(imp.cart)
```

C.3 Compare different imputation methods

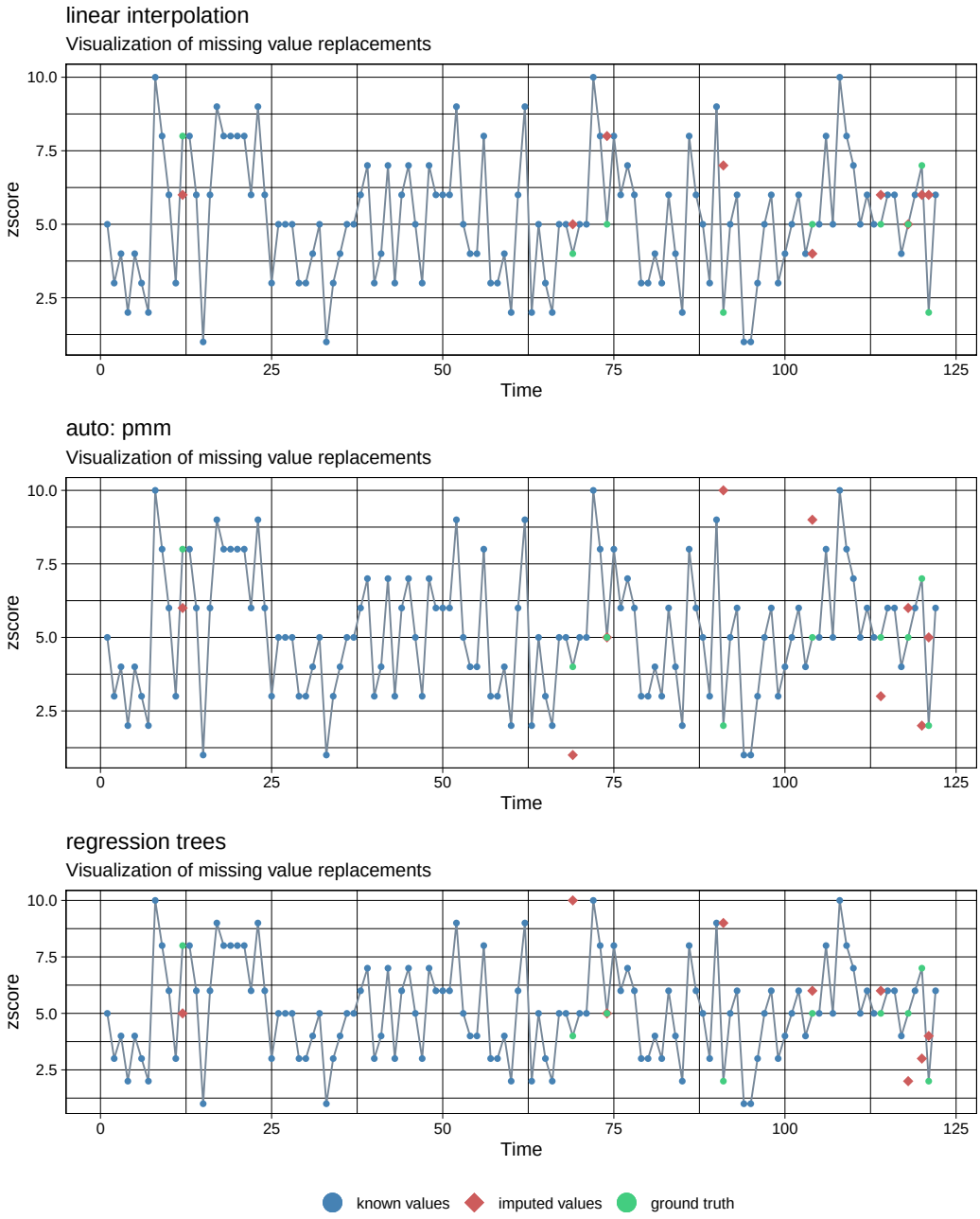
We can check "truth" values for the variables we created, which obviously cannot be done for the empirical data.

Visual inspection

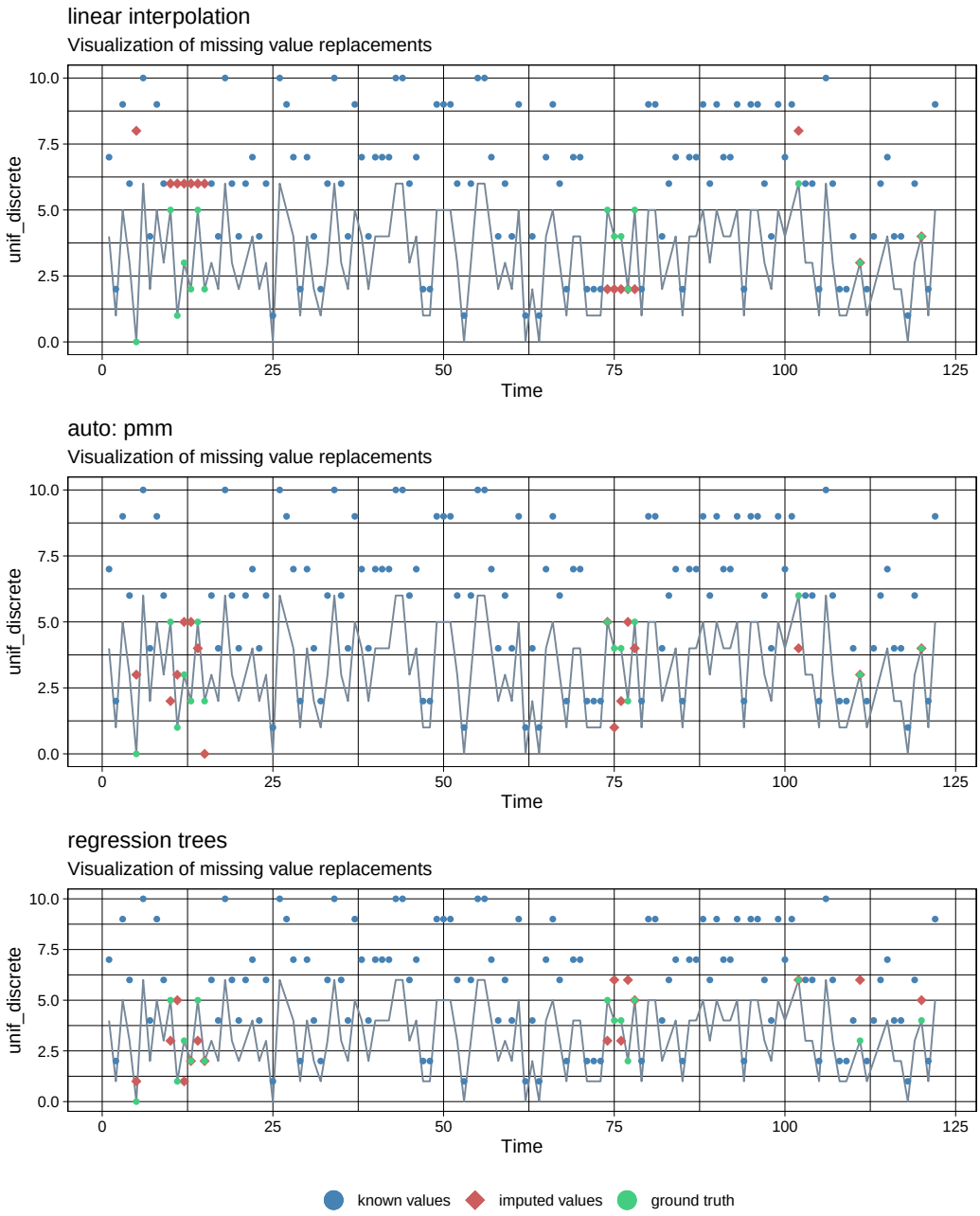
Function `imputeTS::ggplot_na_imputations()` is an excellent way to visualise the imputation result.

```
truth <- list(zscore, unif_discrete, cat_unordered, cat_ordered, df_vars$angry, df_vars$ruminate, df_var
g1 <- g2 <- g3 <- list()

for(c in 1:NCOL(df_vars)){

withNA  <- as.numeric_discrete(df_vars[,c], keepNA = TRUE)
Truth   <- as.numeric_discrete(truth[[c]], keepNA = TRUE)

g1[[c]] <- ggplot_na_imputations(x_with_na = withNA,
                 x_with_imputations = out.linear[,c],
                   x_with_truth = Truth,
                   title = "linear interpolation",
                   ylab = colnames(df_vars)[c], legend = FALSE)

g2[[c]] <- ggplot_na_imputations(x_with_na = withNA,
                 x_with_imputations = as.numeric_discrete(out.auto[,c]),
                   x_with_truth = Truth,
                   title = paste("auto:",imp.mice$method)[c],
                   ylab = colnames(df_vars)[c], legend = FALSE)

g3[[c]] <- ggplot_na_imputations(x_with_na = withNA,
                   x_with_imputations = as.numeric_discrete(out.cart[,c]),
                     x_with_truth =Truth,
                     title="regression trees",
                     ylab = colnames(df_vars)[c])

print(colnames(df_vars)[c])
print(cowplot::plot_grid(g1[[c]],g2[[c]],g3[[c]], ncol = 1))
```

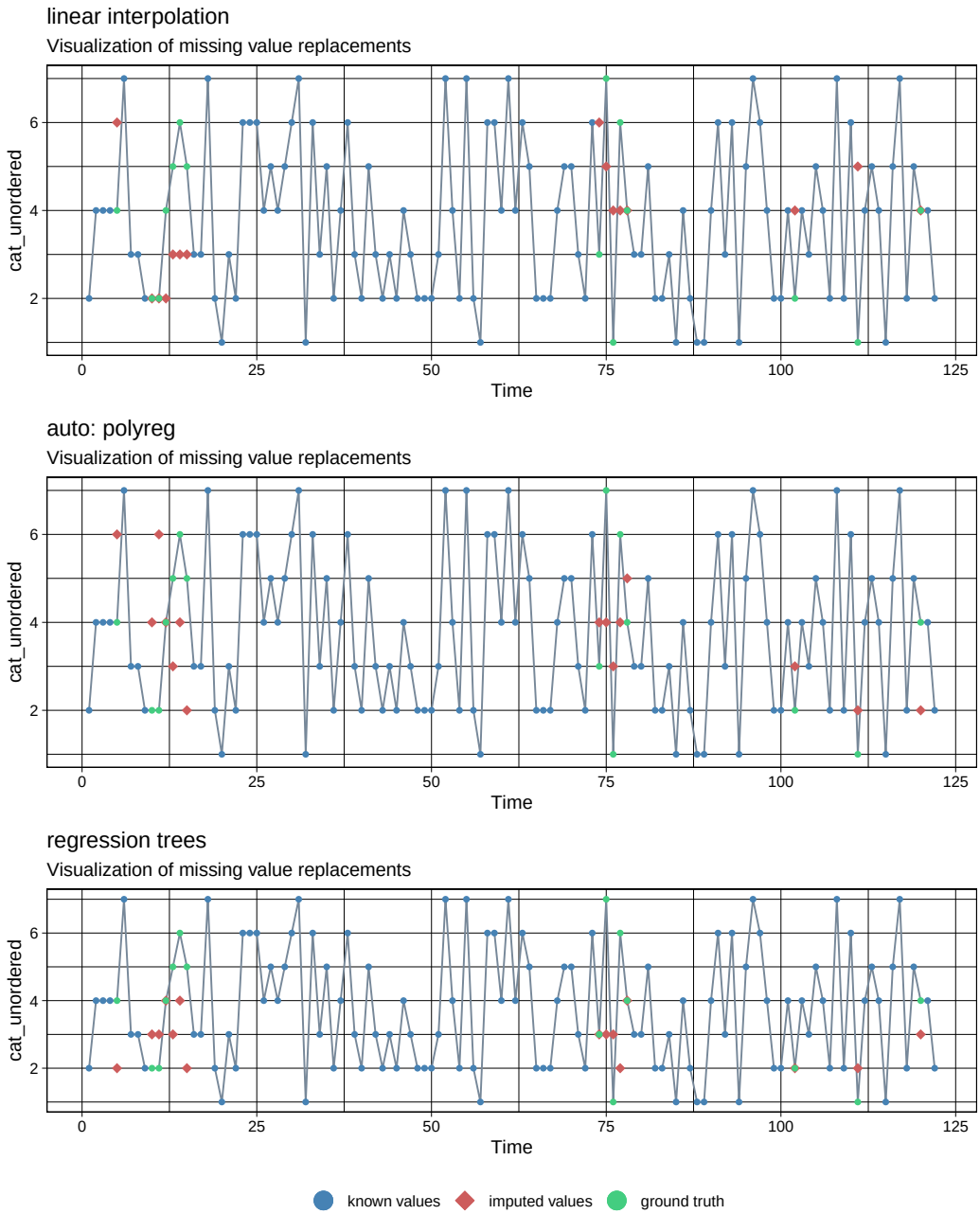
```
}
> [1] "zscore"
```


linear interpolation
Visualization of missing value replacements
zscore
Time
auto: pmm
Visualization of missing value replacements
zscore
Time
regression trees
Visualization of missing value replacements
zscore
Time
known values
imputed values
ground truth

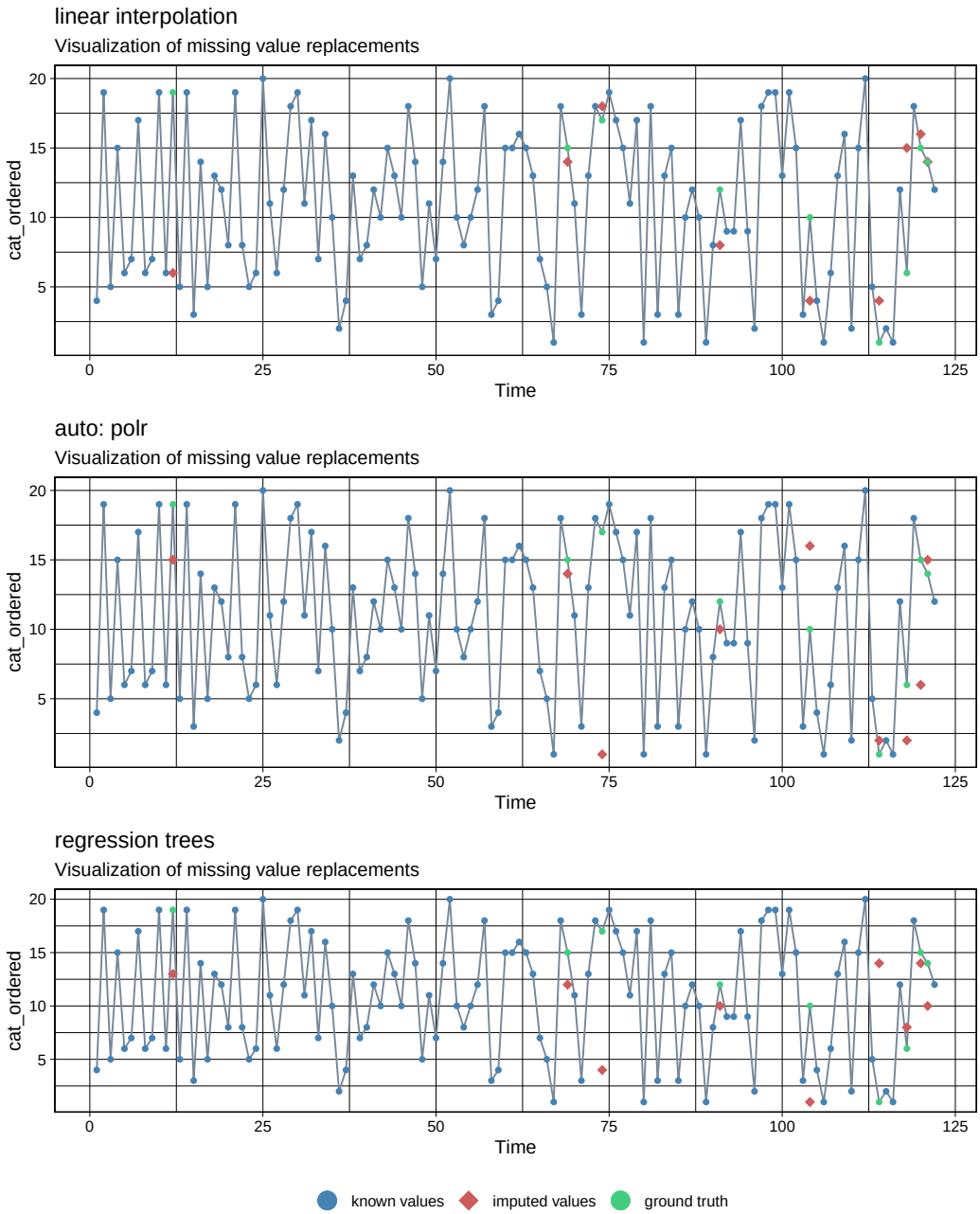
```
> [1] "unif_discrete"
```



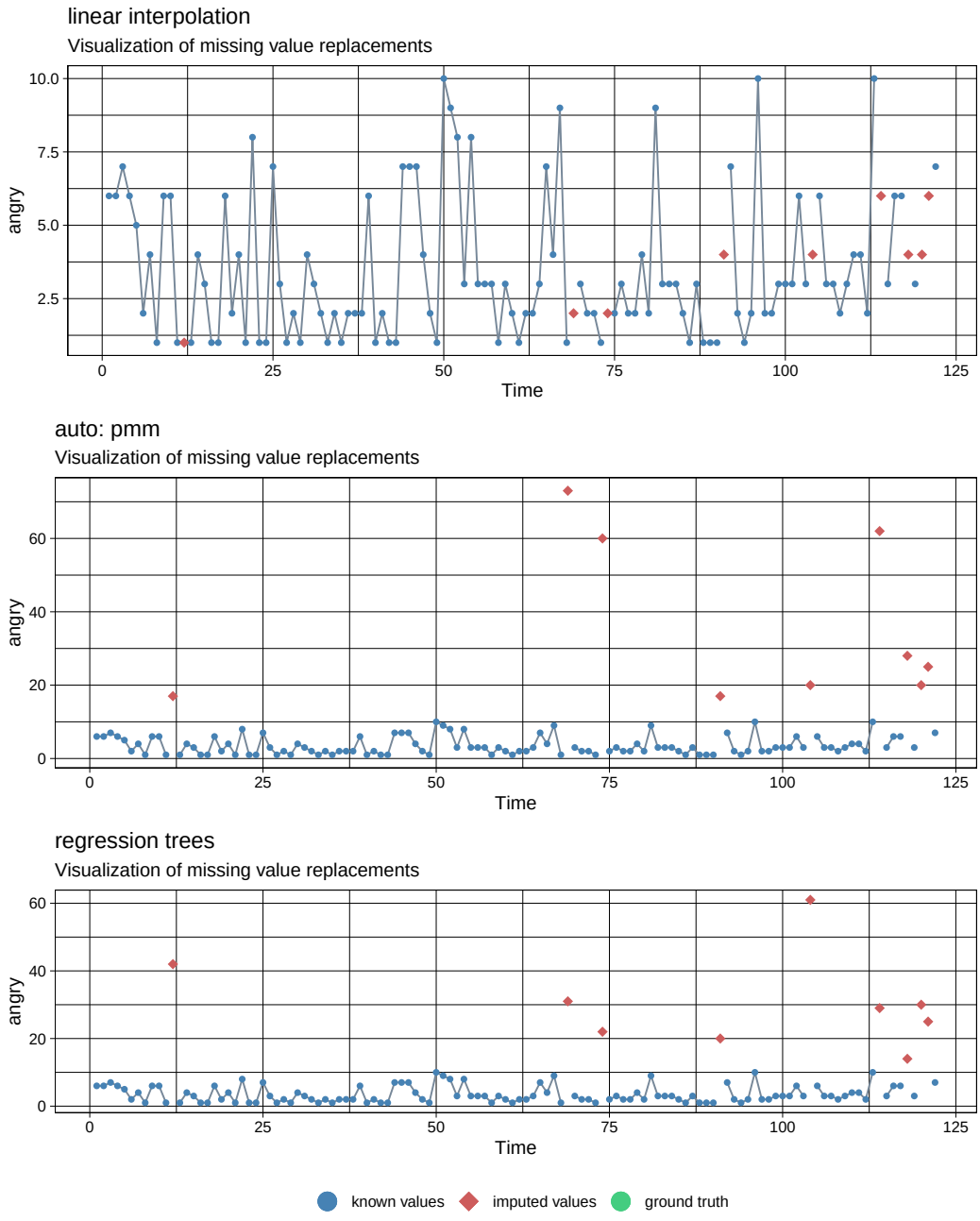

```
> [1] "cat_unordered"
```



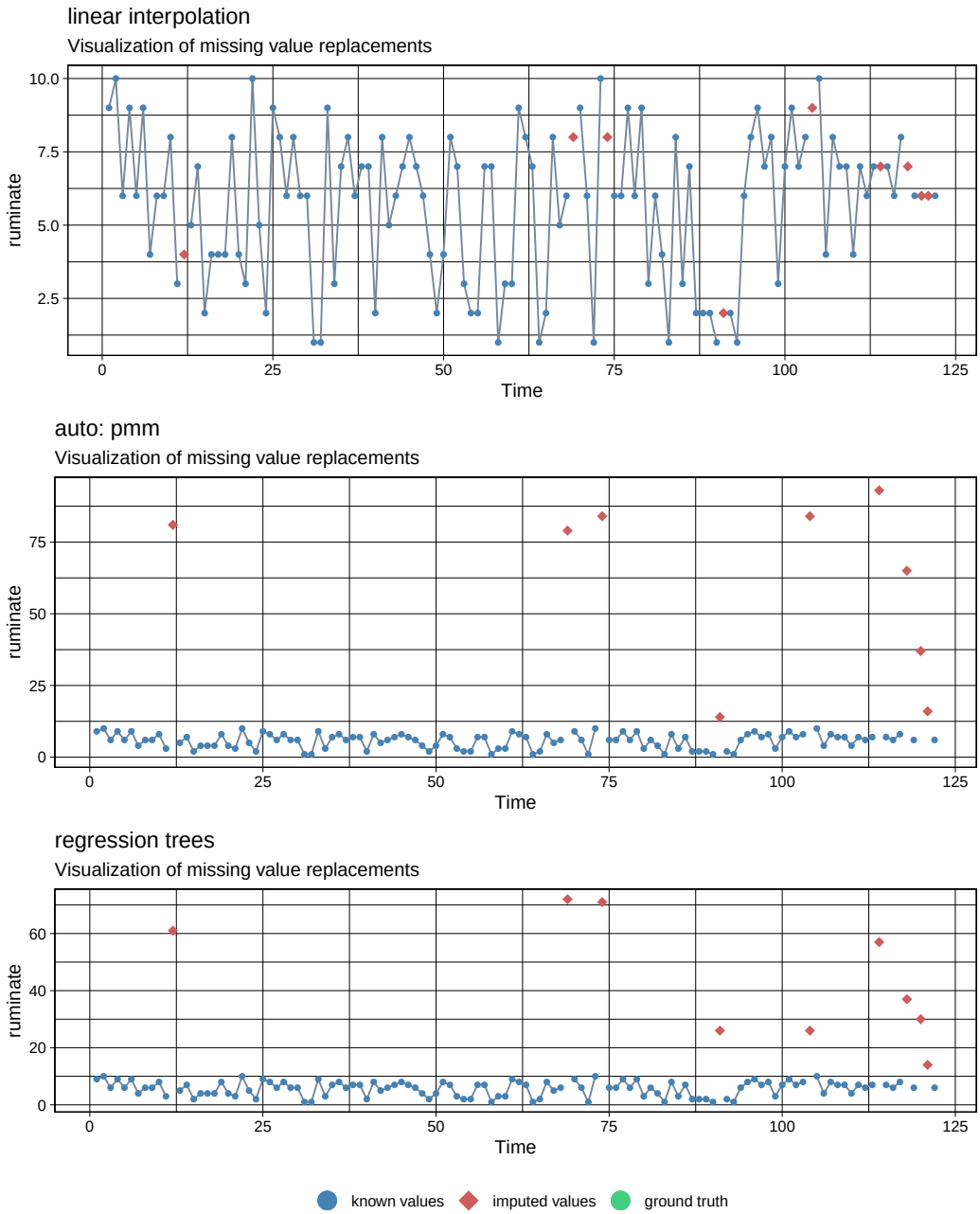

```
> [1] "cat_ordered"
```


linear interpolation
Visualization of missing value replacements
cat_ordered
Time
auto: polr
Visualization of missing value replacements
cat_ordered
Time
regression trees
Visualization of missing value replacements
cat_ordered
Time
known values
imputed values
ground truth

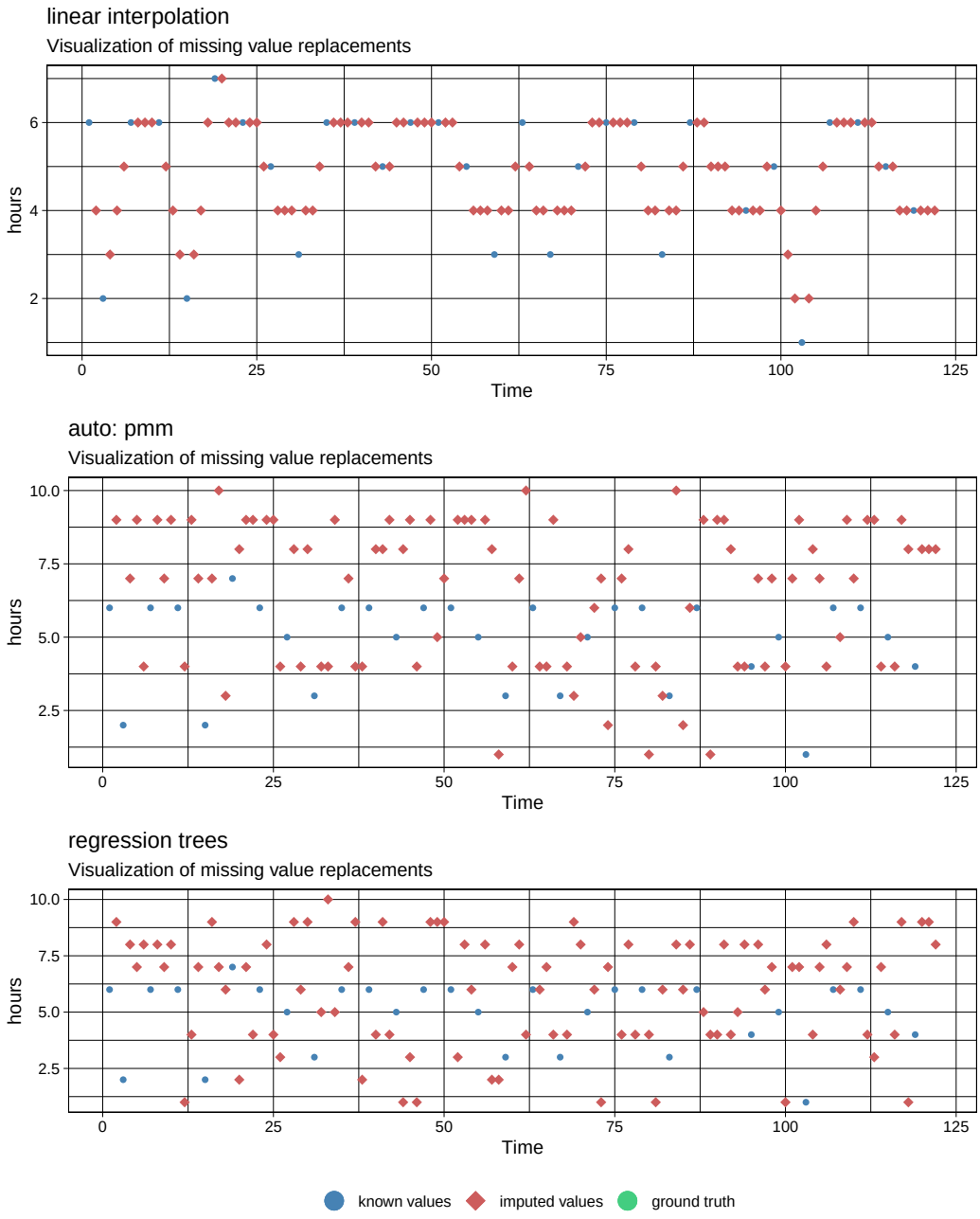
```
> [1] "angry"
```


linear interpolation
Visualization of missing value replacements
angry
Time
auto: pmm
Visualization of missing value replacements
angry
Time
regression trees
Visualization of missing value replacements
angry
Time
known values
imputed values
ground truth

```
> [1] "ruminate"
```


linear interpolation
Visualization of missing value replacements
ruminate
Time
auto: pmm
Visualization of missing value replacements
ruminate
Time
regression trees
Visualization of missing value replacements
ruminate
Time
known values
imputed values
ground truth

```
> [1] "hours"
```



Effect on analysis results

Finally, we compare the effect of different methods on the results of analyses.

```
df <- list()

for(c in 1:NCOL(df_vars)){

withNA  <- as.numeric_discrete(df_vars[,c], keepNA = FALSE)
Truth   <- as.numeric_discrete(truth[[c]], keepNA = FALSE)
LINEAR  <- as.numeric_discrete(unname(out.linear[,c]))
AUTO    <- as.numeric_discrete(out.auto[,c])
CART    <- as.numeric_discrete(out.cart[,c])



df[[c]] <- data.frame(
                NAremoved = c(mean(withNA, na.rm = TRUE), sd(withNA,na.rm = TRUE)),
                Truth     = c(mean(Truth,  na.rm = TRUE), sd(Truth,na.rm = TRUE)),
                LINEAR    = c(mean(LINEAR, na.rm = TRUE), sd(LINEAR,na.rm = TRUE)),
                AUTO      = c(mean(AUTO,   na.rm = TRUE), sd(AUTO,na.rm = TRUE)),
                CART      = c(mean(CART,   na.rm = TRUE), sd(CART,na.rm = TRUE)))
rownames(df[[c]]) <- c("Mean","SD")
}
```

zscore

NAremoved

Truth

LINEAR

AUTO

CART

Mean

5.186

5.156

5.238

5.189

5.213

SD

2.140

2.124

2.089

2.198

2.167

unif_discrete

NAremoved

Truth

LINEAR

AUTO

CART

Mean

5.888

3.213

5.730

3.205

3.262

SD

2.738

1.687

2.709

1.666

1.714

cat_unordered

NAremoved

Truth

LINEAR

AUTO

CART

Mean

3.766

3.762

3.770

3.762

3.656

SD

1.783

1.786

1.729

1.725

1.714

cat_ordered

NAremoved

Truth

LINEAR

AUTO

CART

Mean

10.841

10.934

10.852

10.705

10.746

SD

5.700

5.684

5.662

5.739

5.617

angry

NAremoved

Truth

LINEAR

AUTO

CART

Mean

3.478

3.478

3.492

36.336

35.943

SD

2.446

2.446

2.396

20.345

19.893

ruminate

NAremoved

Truth

LINEAR

AUTO

CART

Mean

5.690

5.690

5.738

55.598

54.295

SD

2.564

2.564

2.535

21.198

20.580

hours

NAremoved

Truth

LINEAR

AUTO

CART

Mean

4.867

4.867

4.852

6.639

6.156

SD

1.548

1.548

1.190

2.426

2.496

References

Bastiaansen, J. A., Kunkels, Y. K., Blaauw, F., Boker, S. M., Ceulemans, E., Chen, M., … Bringmann, L. F. (2019, March 21). Time to get personal? The impact of researchers' choices on the selection of treatment targets using the experience sampling methodology. https://doi.org/10.31234/osf.io/c8vp7

Appendix D

List of terms

Adaptive Behaviour

- System behaviour that appears to be (partially) coordinated by previously 'experienced events'

Analytic solution

- The solution to a difference or differential equation allows one to find any state of the system without the need to iterate the model starting from some initial condition. There are very few (systems of) equations for which analytical solutions exist

Attractor

- The status that a dynamic system eventually "settles down to". An attractor is a set of values in the phase space to which a system migrates over time, or iterations. Attractors can have as many dimensions as the number of variables that influence its system

Basin of attraction

- A region in phase space associated with a given attractor. The basin of attraction of an attractor is the set of all (initial) points that eventually end

up in that attractor

Behaviour (of a dynamic system)

- The temporal evolution of states of a system according to one or more rules (also known as state propagation rules, or, iterative processes). Models of the behaviour of dynamic systems use difference or differential equations to describe the iterative processes hypothesized to underlie the temporal evolution

Bifurcation

- A clearly observable qualitative change in the behavioural mode (attractor state) of a dynamic system associated with continuous change in one or more control parameters (also known as Phase-, State-, or Order- Transition). The value of a control parameter at which a bifurcation occurs is often non-specific, or trivial

Bifurcation diagram

- Visual summary of the succession of period-doubling bifurcations produced by gradual changes in the control parameter(s)

Catastrophe flags

- Markers indicative for a physical system that is described by a catastrophe. There are 5 'classical' flags and 3 'diagnostic' flags. Classical: bimodality, sudden jumps, inaccessibility, sensitivity & hysteresis. Diagnostic: divergence from linear response, critical slowing down and critical fluctuations. Diagnostic flags can be used as early-warning signals.

Catastrophe theory

- Mathematical research program describing how gradual change in some parameters can lead to disproportionately large changes in another parameter, called catastrophes (similar to bifurcations, Phase-, State-, or Order-transitions). 'This kind of behaviour has been summarized succinctly in the phrase "the straw that broke the camel's back".' (Gilmore, 1992).

Complex Network

- A network with of many nodes and likely many substructures depending on that nature and distribution of connections between nodes

Complex system

- Spatially and/or temporally extended nonlinear systems characterized by emergent properties and self-organised behavioural modes at a global, or, macro-level (the system as a whole), that is often different from the characteristic behaviour at a local, or, micro-level (behaviour of the individual parts that constitute the whole)

Complexity science

- Complexity science studies how systems that consist of many components can generate relatively simple and stable (non-random) behaviour. Important behavioural phenomena studied in Complexity Science are synchronisation, adaptation and coordination of behaviour across many different temporal and spatial scales, emergent properties and collective behaviour, holism and self-organisation

Component dominant dynamics

- A causal ontology in which observed behaviour is explained by assuming it is the result of a chain of independent efficient causes (components)

Control parameter

- A variable that controls the global behaviour of a dynamic system. For certain values of the parameter, transitions between qualitatively different behavioural modes (orders) can occur.

Critical fluctuations

- An early warning signal for a phase transition that is characterised by an increase in fluctuations (variability) of the behaviour of the system. The

increase occurs because the self-organised transition from one state to another relaxes the constraints on the degrees of freedom a system has available to generate its behaviour, allowing states and behavioural modes to appear that were previously inaccessible.

Critical slowing down

- An early warning signal for a phase transition that is characterised by an increase in the duration of relaxation times. If it takes longer for the system to return to the state it was perturbed from, this implies the emergence of a new stable state is imminent

Deterministic Chaos

- Behaviour of a dynamic system that "looks random, but is not" (Lorenz, 1973). The dynamics can be characterised as follows: 1) A-periodic, no point or trajectory in state space will exactly recur; 2) Sensitive dependence in initial conditions; 3) Bounded, not all theoretically possible degrees of freedom are available to the system; 4) The origin of this behaviour is deterministic, not stochastic

Difference equation

- A function specifying the underlying change process in a variable from one discrete point in time to another

Differential equation

- A function specifying the underlying change process of a variable in continuous time

Dimension

- See embedding dimension, box-counting dimension, correlation dimension, information dimension, dimension of a system

Dimensions of a system

- The set of variables that define a system. Iterative processes operate on the dimensions of a system

Dynamic system

- A set of equations specifying how certain variables change over time. The equations specify how to determine (compute) the new values as a function of their current values and control parameters. The functions, when explicit, are either difference equations or differential equations. Dynamic systems may be stochastic or deterministic. In a stochastic system, new values come from a probability distribution. In a deterministic system, a single new value is associated with any current value

Early warning signals

- Critical slowing down and critical fluctuations. Early-warning signals indicate instability in the existing state which may result in a qualitative shift towards a new state (phase transition / catastrophe). Early-warning signals are similar to diagnostic catastrophe flags.

Effective Complexity

- "The effective complexity of an entity is the length of a highly compressed description of its regularities." (Gell-man & Lloyd, 2004)

Embedding Dimension

- Successive N-tuples of points in a time series are treated as points in N dimensional space. The points are said to reside in embedding dimensions of size N, for N = 1, 2, 3, 4, … etc.

Emergence

- A complex system can generate emergent behaviour or display emergent properties that are novel and unexpected, that is, they are not predictable from the behaviour and properties of the components of the system

Entropy

- Relative absence of order/redundancy in a system. The degrees of freedom a system has available for generating its behaviour: Possibility

Epigenetic landscape (potential landscape)

- A hypothetical landscape describing the relative stability of behavioural modes of a system over time

Experienced event

- An interaction of a system with its environment that changed the internal structure/organization of the system such that it can be said to display adaptive behaviour. “Interaction with after-effects”. Random behaviour is “Interaction without after-effects”.

flow ~

- A differential equation

Fractal

- An irregular shape with self-similarity. It has infinite detail, and cannot be differentiated. “Wherever chaos, turbulence, and disorder are found, fractal geometry is at play” (Briggs and Peat, 1989).

Fractal Dimension

- A measure of a geometric object that can take on fractional values. At first used as a synonym to Hausdorff dimension, fractal dimension is currently used as a more general term for a measure of how fast length, area, or volume increases with decrease in scale. (Peitgen, Jurgens, & Saupe, 1992a).

Graph theory

- Models in which associations between mathematical objects are defined as edges (connections) between vertices (nodes)

Hausdorff Dimension

- A measure of a geometric object that can take on fractional values. (see fractal dimension).

Holism (epistemic)

- "some property of a whole would be holistic if, according to the theory in question, there is no way we can find out about it using only local means, i.e., by using only all possible non-holistic resources available to an agent." (Seevinck, 2002)

Idiographic approach

- Scientific explanation in which the goal is to generate knowledge about specific facts, events or entities. The goal is not to generalize to universal laws and first principles.

Information (quantity)

- A measurable quantity that resolves uncertainty about the state of a system by assigning a value to the uncertainty.

Initial condition

- The starting point of a dynamic system, the initial state of a system from which it evolved to the current state.

Interaction dominant dynamics

- A causal ontology in which observed behaviour is explained by assuming it is the result of interactions between processes across many temporal and spatial scales

Iteration

- The repeated application of a function, using its output from one application as its input for the next.

Iterative function

- A function used to calculate the new state of a dynamic system.

Iterative system

- A system in which one or more functions are iterated to define the system.

Largest Lyapunov exponent

- The value of the largest exponent in a spectrum of exponents (the Lyapunov spectrum), coefficients of time, that reflect the rate of departure (divergence) of dynamic orbits of a system. The largest exponent indicates the extent to which the behaviour of a system is sensitive to initial conditions.

Limit cycle

- An attractor that is periodic in time, that is, that cycles periodically through an ordered sequence of states.

Limit points

- Points in phase space. There are three kinds: attractors, repellors, and saddle points. A system moves away from repellors and towards attractors. A saddle point is both an attractor and a repellor, it attracts a system in certain regions, and repels the system to other regions.

Linear function of predictors

- A linear equation is of predictors is of the form y=a*x(i)+b, in which variable y varies 'linearly' with other variables x(i). In this equation, 'a' determines the slope of the line and 'b' reflects the y-intercept, the value y obtains when all x(i) equal zero.

Linear function of time

- A linear function of time is of the form $\hat{y}(t) = a*y(t) + b$, in which variable y varies 'linearly' with time 't', that is, with itself at an earlier moment in time.

In this equation 'a' determines the rate with which 'y' will change as time passes, 'b' reflects the initial condition, the value y obtains when t equals zero.

map …

- A difference equation

Nonlinear dynamics

- The study of dynamic systems whose functions specify that change is not a linear function of time.

Orbit (trajectory)

- A sequence of coordinates (a path) through the phase space of a system.

Order

- "order is essentially the arrival of redundancy in a system, a reduction of possibilities"(Von Förster, 2003). Any form of non-random association or dependency that exists between parts of a system, its behaviour over time and/or its environment is a form of order. In scientific explanation of behaviour, the presence of order in non-artificial systems must be explained and should not be (implicitly) assumed.

Order Parameter

- A nominal variable that indexes qualitatively different behavioural modes of a system, for example the phases of matter (gas, liquid, solid, plasma)

Period-doubling

- The change in dynamics in which a N-point attractor is replaced by a 2N-point attractor.

Phase portrait

- The collection of all trajectories from all possible starting points in the phase space of a dynamic system.

Phase space

- An abstract space used to represent the behaviour of a system. Its dimensions are the variables of the system. Thus a point in the phase space defines a potential state of the system. The points actually achieved by a system depend on its iterative function and initial condition (starting point).

Phase transition

- A transition between qualitatively different behavioural modes

Potential function

- A function that describes the order parameter of a system, that is, it describes the relative stability of the potential end-states (attractor states) a system can settle into. The parameters of the potential function include the control parameter.

Power-law scaling

- A relationship between two variables that is linear on doubly logarithmic coordinates, meaning the law is expressed in increments that represent 'power'

Recursive process

- For our purposes, "recursive" and "iterative" are synonyms. Thus recursive processes are iterative processes, and recursive functions are iterative functions.

Relaxation time

- The time it takes for a system to return to a stable state after it was perturbed enough to leave that state. A characteristic warning signal of an imminent phase transition is an increase relaxation times, also known as critical slowing down.

Repellors

- One type of limit point. A point in phase space that a system moves away from.

Return map

- Plot of time series values vs. a delayed copy of itself. A return plot can be used to get an idea about the functional form of the iterative process, it is a simple variant of delay embedding.

Saddle point

- A point, usually in three dimensional state space, that both attracts and repels, attracting in one dimension and repelling to another.

Scale free network

- A network in which the distribution of the number of connections of a node and their frequency of occurrence follows a power-law in which there are just a few nodes with many connections and many nodes with just a few connections

Self-affinity

- An infinite nesting of characteristic structure on all scales. Strict self-affinity refers to a form of which all substructures are affine transformation, which means the different dimensions of the system can be scaled by their own exponent. Statistical self-affinity refers to an approximate equivalence of form at all scales.

Self-similarity

- An infinite nesting of characteristic structure on all scales. Strict self-similarity refers to a form of which all substructures can be considered scaling transformations, larger or smaller copies scaled by a single exponent for all dimensions of the structure. Statistical self-similarity refers to an approximate equivalence of scaled structure.

Sensitive dependence on initial conditions

- A property of chaotic systems. A dynamic system has sensitivity to initial conditions when very small differences in starting values result in very different behaviour. If the orbits of nearby starting points diverge, the system has sensitivity to initial conditions.

Small world network

- Many real-world networks have a small average shortest path length, but also a clustering coefficient that is significantly higher than expected by chance. These networks are extremely efficient, each node in a very large network can still be reach in just a few steps (the 'six degrees of separation' phenomenon).

State

- A coordinate in state space designating the current status of a dynamic system. The elements of the coordinates are values on the dimensions of the system that span the state space.

State space

- A hypothetical space spanned by the dimensions of the system. Each combination of values of variables that represent the dimension is a state of the system, it is a coordinate in state space.

State space (phase space)

- An abstract space used to represent the behaviour of a system. Its dimensions are the variables of the system. Thus a point in the phase space defines a potential state of the system.

Strange attractor

- An attractor state representing chaotic dynamics: a-periodic, bounded, and sensitive dependence on initial conditions

System

- An entity that can be described as a composition of components according to some organising principle. Organising principles describe how parts of the system relate to the whole and.

Time series

- A record of observations (data points) of behaviour over time.

Trajectory (orbit)

- A sequence of positions (path) of a system in its phase space. The path from its starting point (initial condition) to and within its attractor.

Transient time (transient behaviour)

- The time it takes for a system to transition from one stable state (behavioural mode, attractor state) into another, during which the system displays transient behaviour