Exploring autocorrelation in time series (Assignment Sheet 11)

Introduction To Chaos Applied To Systems, Processes And Products (ETSIDI, UPM)

Alfonso Allen-Perkins, Juan Carlos Bueno, and Eduardo Faleiro

2026-02-03

Contents

Introduction

This assignment introduces the concepts of **autocorrelation (ACF)**, using a variety of time series ranging from deterministic systems to physiological and financial data.

The **autocorrelation function (ACF)** measures the linear relationship between a time series and a lagged version of itself. The formal definition for the autocorrelation at **lag** k is:

$r_k = \frac{\sum_{t=1}^{N-k}(x_t-\bar{x})(x_{t+k}-\bar{x})}{\sum_{t=1}^{N}(x_t-\bar{x})^2}$

This is essentially the Pearson correlation between the time series and its k-lagged version.

Example: Manually Estimating ACF at Lag 1

To understand the meaning of the above equation, let's (manually) estimate ACF at lag 1, in a simple series of 4 points:

```
# Load required libraries
library(tidyverse)
library(forecast)
x <- c(4, 7, 6, 5, 9)
```

We compute the lag-1 autocorrelation by correlating:

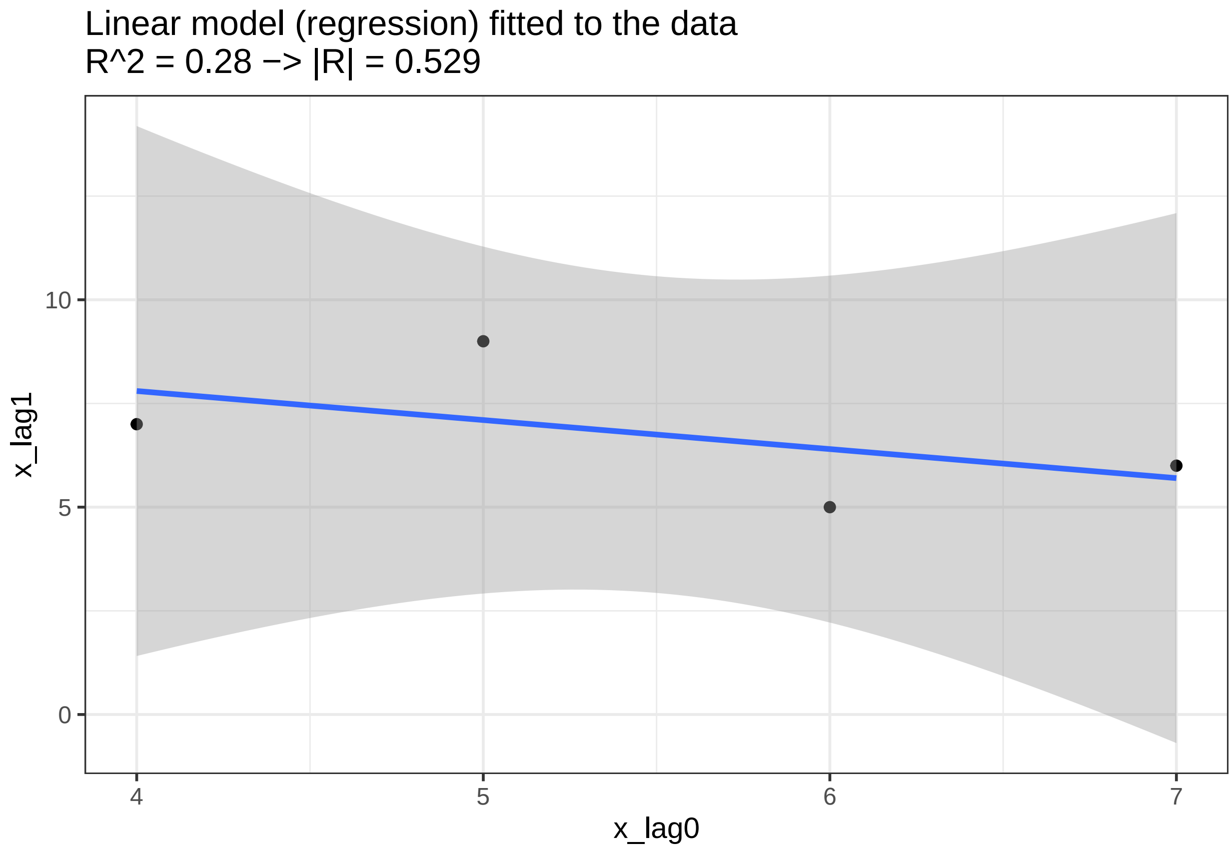
- $x_{1:(n-1)} = \{4, 7, 6, 5\}$
- $x_{2:n} = \{7, 6, 5, 9\}$

```
x_lag0 <- x[1:(length(x)-1)]
x_lag1 <- x[2:length(x)]

lag1_df <- data.frame(x_lag0 = x_lag0,
                      x_lag1 = x_lag1)

# Fit linear model
model <- lm(x_lag1 ~ x_lag0, data = lag1_df)
summary(model)$r.squared
```

```
## [1] 0.28
```

```
# Visualize x_lag0 VS x_lag1

ggplot(lag1_df, aes(x = x_lag0, y = x_lag1))+
  geom_point()+
  geom_smooth(method = "lm")+
  labs(title = paste0("Linear model (regression) fitted to the data\nR^2 = ",
                      round(summary(model)$r.squared,3)," -> |R| = ",
                      round(sqrt(summary(model)$r.squared),3)))+
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
acf_lag1 <- cor(x_lag0, x_lag1, method = "pearson") # Correlation coefficient
acf_lag1
```

```
## [1] -0.5291503
```

```
# Sanity check
round(acf_lag1^2,5) == round(summary(model)$r.squared,5)
```

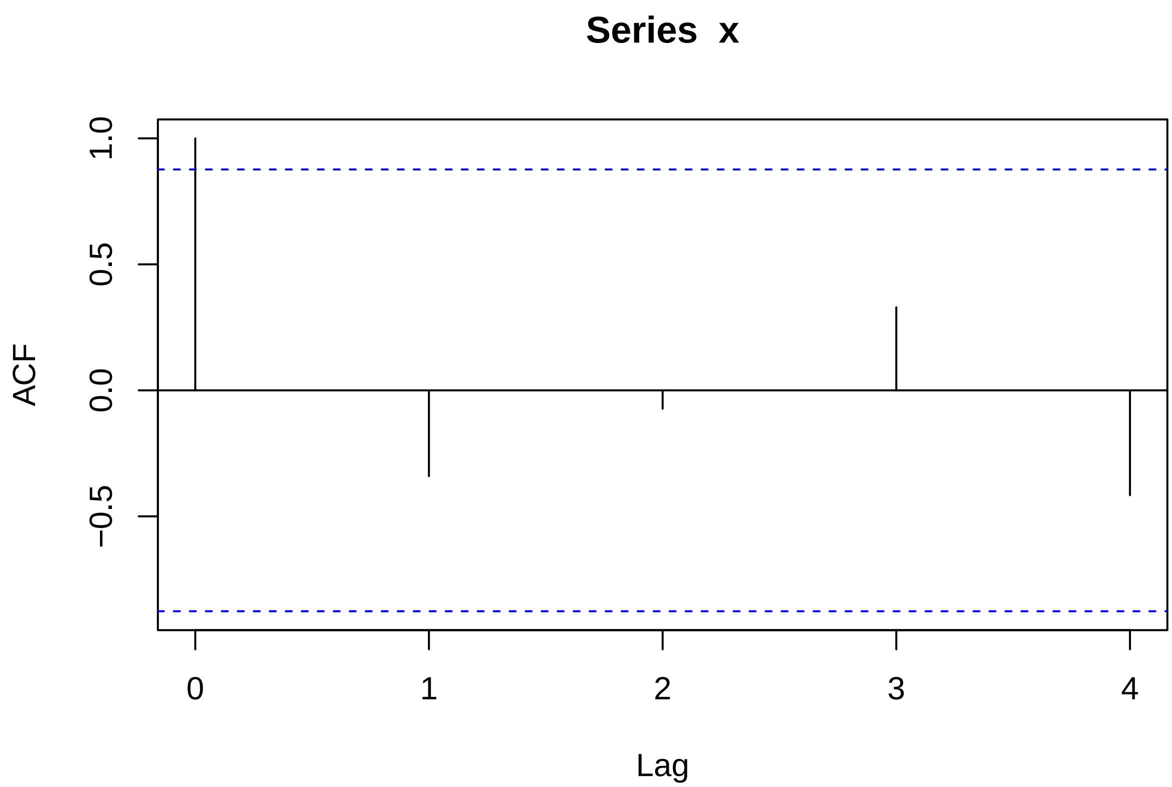
```
## [1] TRUE
```

This returns the value of the **autocorrelation at lag 1**.

Example: Estimating ACF at Lag 1 with a built-in function

We can also use `acf(x)`, the built-in ACF function in R, to compute that value:

```
acf(x, lag.max = 5, plot = TRUE)
```



Note on differences with `acf()` output:

When using `cor(x_lag0, x_lag1)` (manual estimation), we are calculating the Pearson correlation between two lagged vectors of length $n-1$. However, `acf(x)` uses a slightly different formula based on the full series, including a normalization step that adjusts for sample size and bias. For short time series, this leads to visible differences between the manual correlation and the ACF plot.

To retrieve the exact autocorrelation value at lag 1 used by `acf()`, use:

```
acf(x, lag.max = 1, plot = FALSE)$acf[2]
```

This returns the same value plotted in the ACF chart.

Why Does ACF Matter?

When examining the ACF of a periodic signal (e.g., a sine wave), you will see repeating peaks at regular lags, corresponding to the period of oscillation. This reflects long-term memory and predictability.

In contrast, the ACF of a chaotic series (e.g., the logistic map at $r = 3.7$) shows a rapid drop-off (sometimes within a few lags), indicating limited predictability and short memory.

Creating ACF plots in R from a data file

1. Load required libraries

```r
library(tidyverse)
library(forecast)
```

2. Load time series data

```r
setwd("C:/Users/alfon/Desktop/Projects/HandsOnChaos")
getwd()
```

```
## [1] "C:/Users/alfon/Desktop/Projects/HandsOnChaos"
```

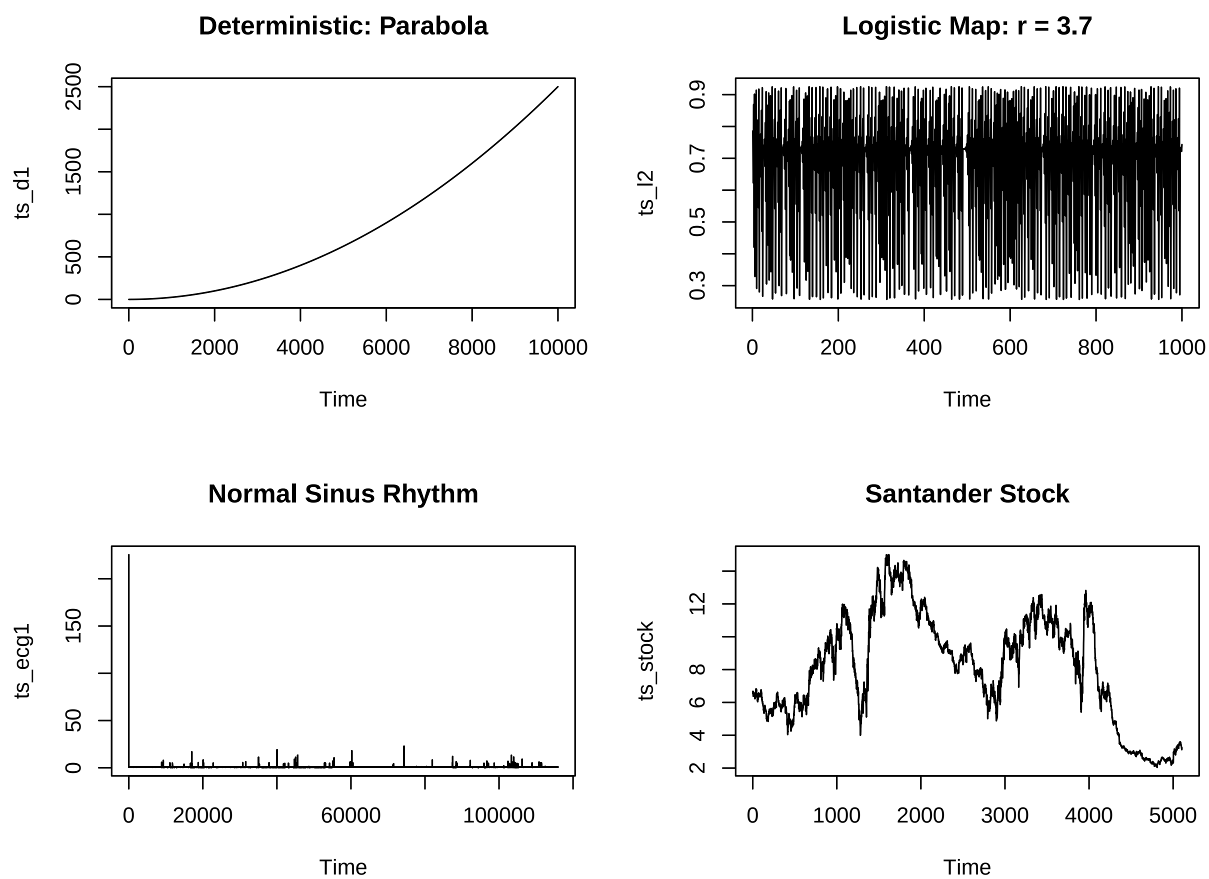
```r
# Make sure working directory is set correctly and data files are available
list.files("Raw-data/Time_series")
```

```
##  [1] "MIT-BIH_SupravArrhyth_801.txt"    "NormalSinusRithm011.txt"
##  [3] "SuddenCardiacDeath35.txt"         "TS_CongesHeartFail211.txt"
##  [5] "TS_deterministic_parabol.txt"     "TS_deterministic_sinx.txt"
##  [7] "TS_logistic.txt"                  "TS_logistic_3.7.txt"
##  [9] "TS_logistic_3.7_x0_0.1.txt"       "TS_logistic_3.7_x0_0.101.txt"
## [11] "TS_MIT-BIH_SupravArrhyth_801.txt" "TS_NormalSinusRithm011.txt"
## [13] "TS_santander.txt"                 "TS_SuddenCardiacDeath35.txt"
```

```r
load_ts <- function(filename) {
  scan(paste0("Raw-data/Time_series/", filename), quiet = TRUE) %>% ts()
}

# Deterministic
ts_d1 <- load_ts("TS_deterministic_parabol.txt")
ts_d2 <- load_ts("TS_deterministic_sinx.txt")

# Logistic (chaotic)
ts_l1 <- load_ts("TS_logistic.txt")
ts_l2 <- load_ts("TS_logistic_3.7.txt")
ts_l3 <- load_ts("TS_logistic_3.7_x0_0.1.txt")
ts_l4 <- load_ts("TS_logistic_3.7_x0_0.101.txt")

# Physiological
ts_ecg1 <- load_ts("NormalSinusRithm011.txt")
ts_ecg2 <- load_ts("SuddenCardiacDeath35.txt")

# Financial
ts_stock <- load_ts("TS_santander.txt")
```

3. Plot Example Time Series

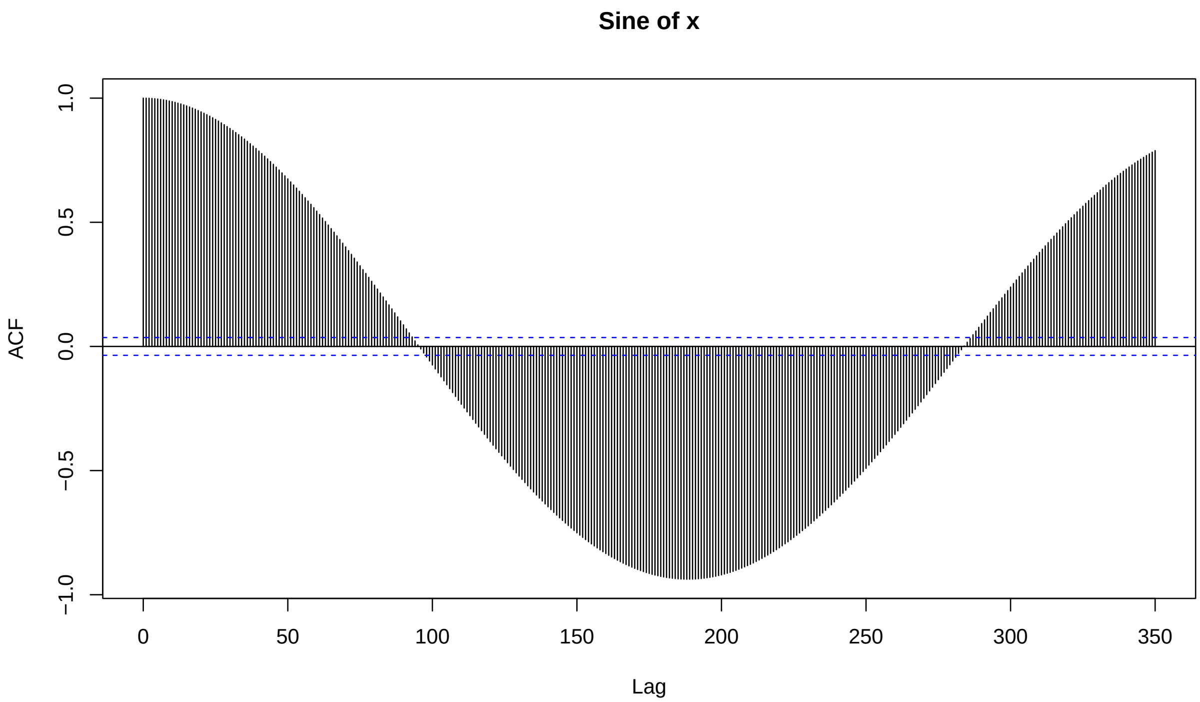
```r
par(mfrow = c(2, 2))
plot(ts_d1, main = "Deterministic: Parabola")
plot(ts_l2, main = "Logistic Map: r = 3.7")
plot(ts_ecg1, main = "Normal Sinus Rhythm")
plot(ts_stock, main = "Santander Stock")
```



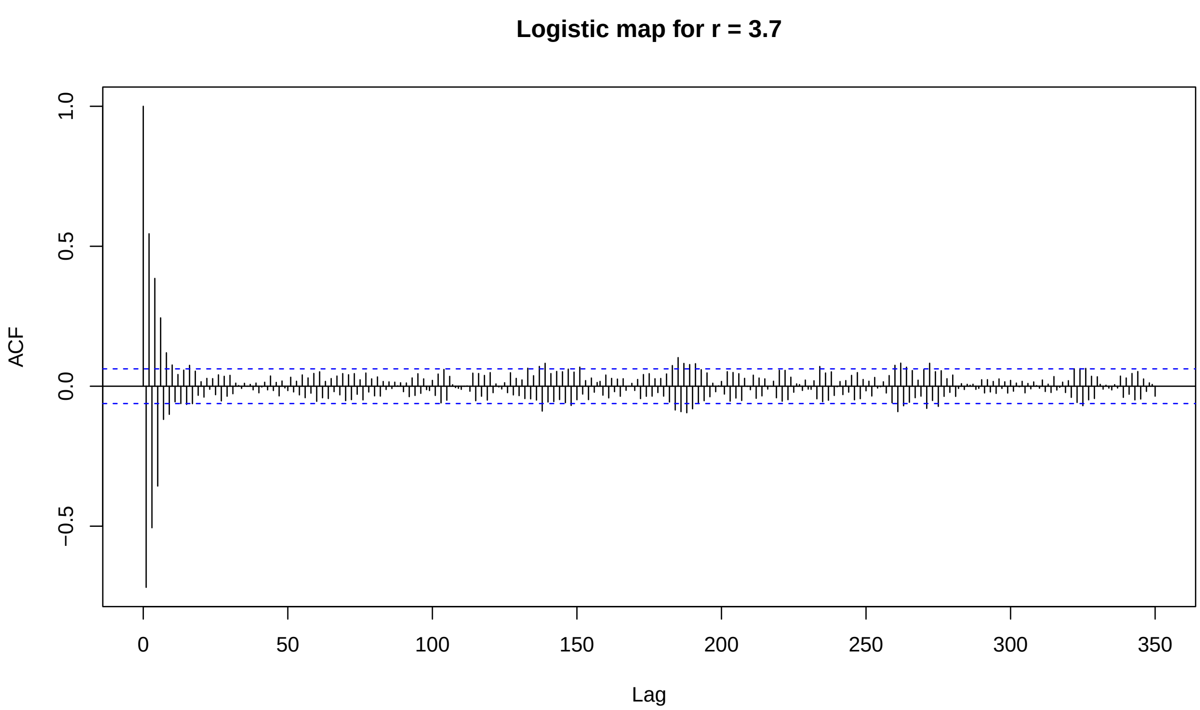

```r
par(mfrow = c(1, 1))
```

4. Autocorrelation

```r
acf(ts_d2, lag.max = 350, main = "Sine of x")
```




```r
acf(ts_l2, lag.max = 350, main = "Logistic map for r = 3.7")
```



- The ACF of the deterministic sine signal (`ts_d2`) shows clear, regular oscillations, indicating strong periodicity and long-range dependence.

- In contrast, the ACF of the chaotic logistic map (`ts_l2`) decays rapidly and irregularly, reflecting short memory and unpredictability.

Note on differences with `acf()` output:

We use the `ts_l2` series to show that differences between the manual correlation and the values in the built-in function are similar with longer time series.

```
ts_l2_lag0 <- ts_l2[1:(length(ts_l2)-1)]
ts_l2_lag1 <- ts_l2[2:length(ts_l2)]

cor(ts_l2_lag0, ts_l2_lag1, method = "pearson") # Correlation coefficient
```

```
## [1] -0.7186455
```

To retrieve the exact autocorrelation value at lag 1 used by `acf()`, use:

```
acf(ts_l2, lag.max = 1, plot = FALSE)$acf[2]
```

```
## [1] -0.7184755
```

Task 1: Physiological Signals

- Compare `ts_ecg1` and `ts_ecg2`.
- What do the ACF plots suggest about regularity or randomness?

Task 2: Stock Market Data

- Analyze `ts_stock`. Is there autocorrelation at any lags?
- How might this affect forecasting?