**Estimation and inference for measures of dependence**

μ, γ(h), and ρ(h) are usually unknown so we need to estimate them. To estimate these quantities, we need to assume the time series is weakly stationary.

Sample mean function

By the weakly stationary assumption, E(x1) = μ, E(x2) = μ,…, E(xn) = μ. Thus, a logical estimate of μ is



Note that this would not make sense to do if the weakly stationarity assumption did not hold!

Sample autocovariance function

Again with the weakly stationarity assumption, we only need to worry about the lag difference. The estimated autocovariance function is:



### Notes

* 
* What is this quantity if h = 0?
* What is this quantity if h =1?
* This is similar to the formula often used to estimate the covariance between two random variables x and y: . If you do not recognize this formula, look at the numerator of the estimated Pearson correlation coefficient.
* The sum goes up to n - h to avoid having negative subscripts in the x’s.
* This is NOT an unbiased estimate of γ(h)! However, as n gets larger, the bias will go to 0.

Sample autocorrelation function (ACF)



Question: What does ρ(h) = 0 mean and why would this be important to detect?

Because this is important, we conduct hypothesis tests for ρ(h) for all h ≠ 0! To do the hypothesis test, we need to find the sampling distribution for  under the null hypothesis of ρ(h) = 0.

Sampling distribution for : In summary, if ρ(h) = 0, xt is stationary, and the sample size is “large”, then  has an approximate normal distribution with mean 0 and standard deviation .

A proof is available in Shumway and Stoffer’s textbook and requires an understanding asymptotics (PhD level statistics course).

For a hypothesis test, we could check if  is within the bounds of 0 or not where P(Z < Z1-α/2) = 1 – α/2 for a standard normal random variable Z. If it is not, then there is sufficient evidence to conclude that ρ(h) ≠ 0. We will be using this result a lot for the rest of this course!

Example: xt = 0.7xt-1 + wt where wt ~ ind. N(0,1) and n = 100. (ar1\_0.7.R, AR1.0.7.txt)

The data are simulated using the above model and are different from the example earlier in the notes. Also, the data are read in from a file instead of simulated within R.

> ar1 <- read.table(file = "C:\\data\\AR1.0.7.txt", header

 = TRUE, sep = "")

> head(ar1)

 t x

1 1 0.04172680

2 2 0.37190682

3 3 -0.18545185

4 4 -1.38297422

5 5 -2.87593652

6 6 -2.60017605

The plot below is constructed in a similar manner as past plots.



The easiest way to find the autocorrelations in R is to use the acf() function.

> x <- ar1$x

> rho.x <- acf(x = x, type = "correlation", main =

 expression(paste("Data simulated from AR(1): ", x[t] ==

 0.7\*x[t-1] + w[t], " where ", w[t], "~N(0,1)")))



The horizontal lines on the plot are drawn at 0 where Z1-0.05/2 = 1.96. The location of the lines can be changed by using the ci argument. The default is ci = 0.95.

> rho.x

Autocorrelations of series 'x', by lag

 0 1 2 3 4 5 6 7

 1.000 0.674 0.401 0.169 -0.023 -0.125 -0.067 -0.064

 8 9 10 11 12 13 14

-0.058 0.005 -0.044 -0.041 -0.017 0.064 0.076

 15 16 17 18 19 20

 0.160 0.191 0.141 0.081 0.006 -0.132

> names(rho.x)

[1] "acf" "type" "n.used" "lag" "series" "snames"

> rho.x$acf

, , 1

 [,1]

 [1,] 1.000000000

 [2,] 0.673671871

 [3,] 0.400891188

 [4,] 0.168552826

 [5,] -0.023391129

 [6,] -0.124632501

 [7,] -0.067392830

 [8,] -0.064248086

 [9,] -0.057717749

[10,] 0.005312358

[11,] -0.044035976

[12,] -0.041121407

[13,] -0.017197132

[14,] 0.063864970

[15,] 0.075575696

[16,] 0.159665692

[17,] 0.191349965

[18,] 0.140967540

[19,] 0.080508273

[20,] 0.005584061

[21,] -0.131559629

> rho.x$acf[1:2]

[1] 1.0000000 0.6736719

Questions:

* What happens to the autocorrelations over time? Why do you think this happens?
* Is there a positive or negative correlation?
* At what lags is ρ(h) ≠ 0?

R plots  by default. This is unnecessary because  will be 1 for all time series data sets! To remove  from the plot, one can specify the x-axis limit to start at 1. Below is one way this can be done and also illustrates how to use the lag.max argument.

> par(xaxs = "i") # Remove default 4% extra space around

 min and max of x-axis

> rho.x2 <- acf(x = x, type = "correlation", xlim =

 c(0,30), lag.max = 30, main = expression(paste("Data

 simulated from AR(1): ", x[t] == 0.7\*x[t-1] + w[t], "

 where ", w[t], "~N(0,1)")))

> par(xaxs = "r") # Return to the default



Note that  is still present but the y-axis at x = 0 hides it.

While displaying  may seem minor, we will examine these autocorrelations later in the course to determine an appropriate model for a data set. Often, one will forget to ignore the line drawn at lag = 0 and choose an incorrect model.

The autocovariances can also be found using acf().

> acf(x = x, type = "covariance", main =

 expression(paste("Data simulated from AR(1): ", x[t]

 == 0.7\*x[t-1] + w[t], " where ", w[t], "~N(0,1)")))



To help understand autocorrelations and their relationship with the correlation coefficient better, I decided to look at the “usual” estimated Pearson correlation coefficients between xt, xt-1, xt-2, and xt-3.

> x.ts <- ts(x)

> set1 <- ts.intersect(x.ts, x.ts1 = lag(x = x.ts, k =

 -1), x.ts2 = lag(x = x.ts, k = -2), x.ts3 =

 lag(x = x.ts, k = -3))

> set1

Time Series:

Start = 4

End = 100

Frequency = 1

 x.ts x.ts1 x.ts2 x.ts3

 4 -1.38297422 -0.18545185 0.37190682 0.04172680

 5 -2.87593652 -1.38297422 -0.18545185 0.37190682

 6 -2.60017605 -2.87593652 -1.38297422 -0.18545185

 7 -1.10401719 -2.60017605 -2.87593652 -1.38297422

 8 -0.46385116 -1.10401719 -2.60017605 -2.87593652

 9 0.80339069 -0.46385116 -1.10401719 -2.60017605

Output edited

 97 -1.36418639 -1.63175408 -1.56008530 -0.40824385

 98 -0.37209392 -1.36418639 -1.63175408 -1.56008530

 99 -0.65833401 -0.37209392 -1.36418639 -1.63175408

100 2.03705932 -0.65833401 -0.37209392 -1.36418639

> cor(set1)

 x.ts x.ts1 x.ts2 x.ts3

x.ts 1.0000000 0.6824913 0.4065326 0.1710145

x.ts1 0.6824913 1.0000000 0.6929638 0.4108375

x.ts2 0.4065326 0.6929638 1.0000000 0.6935801

x.ts3 0.1710145 0.4108375 0.6935801 1.0000000

> library(car) #scatterplot.matrix is in this package

> scatterplotMatrix(formula = ~x.ts + x.ts1 + x.ts2 +

 x.ts3, data = set1, diagonal = list(method =

 "histogram"), col = "red")



> set2 <- ts.intersect(x.ts, x.ts1 = lag(x = x.ts, k = 1),

 x.ts2 = lag(x = x.ts, k = 2), x.ts3 = lag(x = x.ts, k =

 3))

> set2

Time Series:

Start = 1

End = 97

Frequency = 1

 x.ts x.ts1 x.ts2 x.ts3

 1 0.04172680 0.37190682 -0.18545185 -1.38297422

 2 0.37190682 -0.18545185 -1.38297422 -2.87593652

 3 -0.18545185 -1.38297422 -2.87593652 -2.60017605

 4 -1.38297422 -2.87593652 -2.60017605 -1.10401719

 5 -2.87593652 -2.60017605 -1.10401719 -0.46385116

Output edited

94 -0.40824385 -1.56008530 -1.63175408 -1.36418639

95 -1.56008530 -1.63175408 -1.36418639 -0.37209392

96 -1.63175408 -1.36418639 -0.37209392 -0.65833401

97 -1.36418639 -0.37209392 -0.65833401 2.03705932

> cor(set2)

 x.ts x.ts1 x.ts2 x.ts3

x.ts 1.0000000 0.6935801 0.4108375 0.1710145

x.ts1 0.6935801 1.0000000 0.6929638 0.4065326

x.ts2 0.4108375 0.6929638 1.0000000 0.6824913

x.ts3 0.1710145 0.4065326 0.6824913 1.0000000

* The ts() function converts the time series data to an object that R recognizes as a time series.
* The lag() function is used to find xt-1, xt-2, and xt-3. The k argument specifies how many time periods to go *back*. Run lag(x.ts, k = -1) and lag(x.ts, k = 1) to see what happens. To get everything lined up as I wanted with ts.intersect(), I chose to use k = -1.
* The ts.intersect() function finds the intersection of the four different “variables”.
* The cor()function finds the estimated Pearson correlation coefficients between all variable pairs. Notice how close these correlations are to the autocorrelations!
* The scatterplotMatrix() function finds a scatter plot matrix. The function is in the car package.

Example: OSU enrollment data (osu\_enroll.R, osu\_enroll.csv)

The code used to find the autocorrelations is:

> x <- osu.enroll$Enrollment

> rho.x <- acf(x = x, type = "correlation", main = "OSU

 Enrollment series")

> rho.x

Autocorrelations of series 'x', by lag

 0 1 2 3 4 5 6 7 8 9 10

 1.000 -0.470 -0.425 0.909 -0.438 -0.395 0.822 -0.403 -0.358 0.739 -0.367

 11 12 13 14 15 16

-0.327 0.655 -0.337 -0.297 0.581 -0.309

> rho.x$acf[1:9]

[1] 1.0000000 -0.4702315 -0.4253427 0.9087421 -0.4377336

 -0.3946048 0.8224660 -0.4025871 -0.3584216



Notes:

* There are some large autocorrelations. This is a characteristic of a nonstationary series. We will examine this more later.
* Because the series is not stationary, the hypothesis test for ρ(h) = 0 should not be done here using the methods discussed earlier.
* There is a pattern among the autocorrelations. What does this correspond to?