

INTRODUCTION TO THE COURSE

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Financial Econometrics

Real-time decision making on the financial markets is based on the ability of the financial analyst to extract relevant information from financial data.

As financial analyst you should be able to answer questions like:

- How do I allocate my wealth in a portfolio of risky assets?
- How do I determine the risk of a financial investment?
- What is the probability of loosing more than $x\%$ on my investment?
- What is the fair-price of a derivative contract?

Financial Econometrics

In order to answer this kind of question you need:

- ① Definition of the problem
- ② Econometric model
- ③ Analyze the financial data
- ④ Interpretation of the results
- ⑤ Extract coherent strategies
- ⑥ Economic-financial theory
- ⑦ Programming skills

Points 6 and 7 are not the goals of this course.

Step 1: Analyse the data

- Read the original data file
- Make a first set of plots, look at it
- Transform as necessary (aggregate, logs, first differences, combine with other data sets)
- Calculate statistics
- Save a file in a convenient format for later analysis

Step 2: Analyse and Estimate the Model

- Can you simulate data from the model?
- Does it look 'similar' to empirical data?
- Is it 'the same' type of input?
- Take input (either empirical or simulated data)
- Implement model estimation
- Prepare useful outcome

Step 3: Extract results

- Use estimated model parameters
- Create tables/graphs

Specific to the content of this course:

- Take investment decisions
- Predict the amount of risk related to an investment
- Assign the correct probability to extreme events
- Policy?

Course Objectives

The main goals of this course are:

- Provide the students with the practical tools in quantitative finance that are in demand in the finance industry.
- Learn to use a computational tool (R) for the implementation of financial econometric methods.
- Work with real data to solve real world problems.
- Apply econometric methods in analyzing and predicting financial time series like asset prices and returns, volatility, and correlations.
- Evaluate and reflect upon empirical studies using financial data.

Course Content

- Learning programming in R
 - ① Basic Functions
 - ② Handling data sets
 - ③ Numerical optimization
 - ④ Simulation
- Implement econometric techniques
 - ① Maximum Likelihood
 - ② Multivariate distributions and tails
- Financial Modelling
 - ① Volatility Modelling
 - ② Correlation Modelling
 - ③ Risk Management
 - ④ Portfolio Allocation
 - ⑤ Systemic Risk

Not part of this course

- Review of basic probability concepts (random variables, probability distributions)
- Review of basic econometric concepts (OLS, GLS, p-value)
- Review of time series concepts (AR, MA, unit roots, cointegration)
- Review of financial theories (no-arbitrage, EMH)

Practicalities

Lectures take place in 2624-E1 on:

- Monday, 12:00 - 14:00
- Thursday, 12:00 - 14:00

Evaluation

The final exam is

- 4 days TAKE-HOME ASSIGNMENT

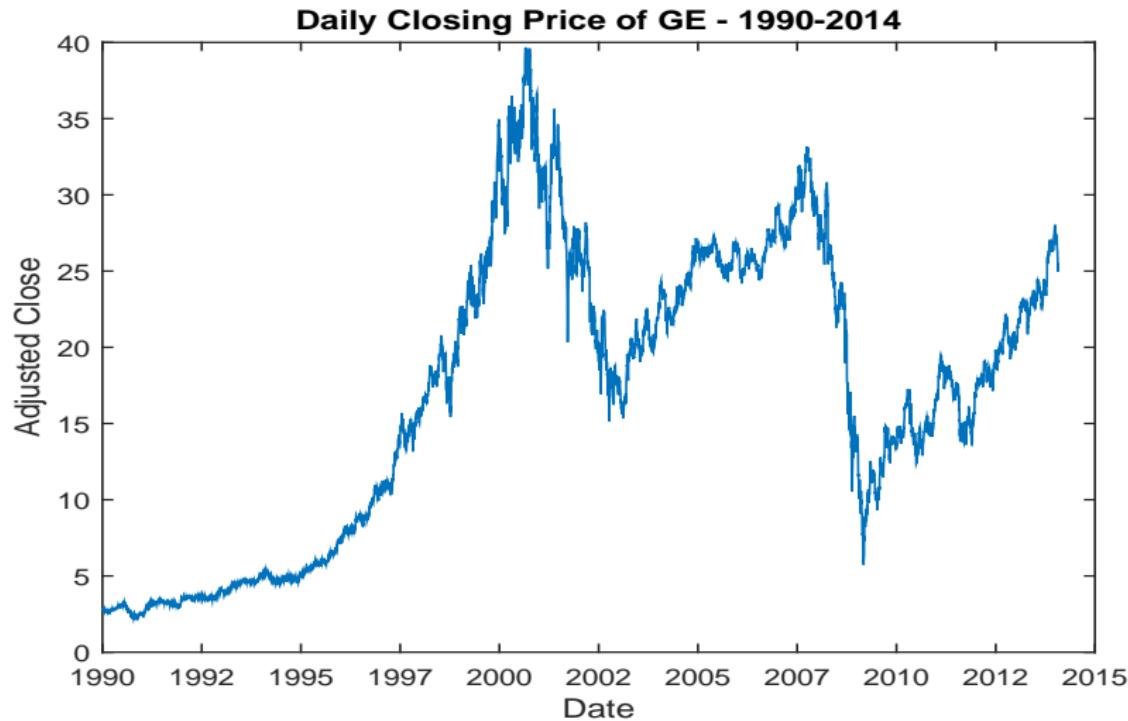
During the course, I will provide you with assignments. Solutions to the assignments will be provided. You are encouraged to solve the assignments before looking at the solutions.

Teaching Material

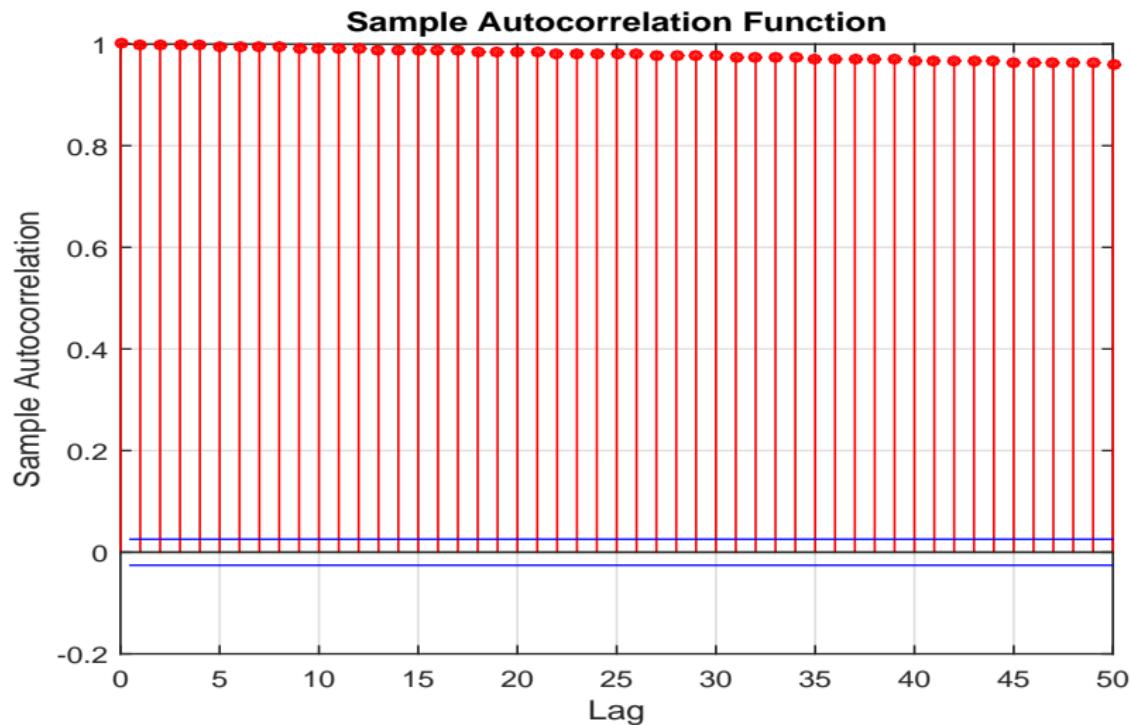
There is no book that matches 100% the topics of this course. The teaching material is

- Jondeau, Poon and Rockinger, Financial Modeling Under Non-Gaussian Distributions, (2007), Springer
- Published articles
- Slides
- Alexios Ghalanos: Introduction to the rugarch package. Free.

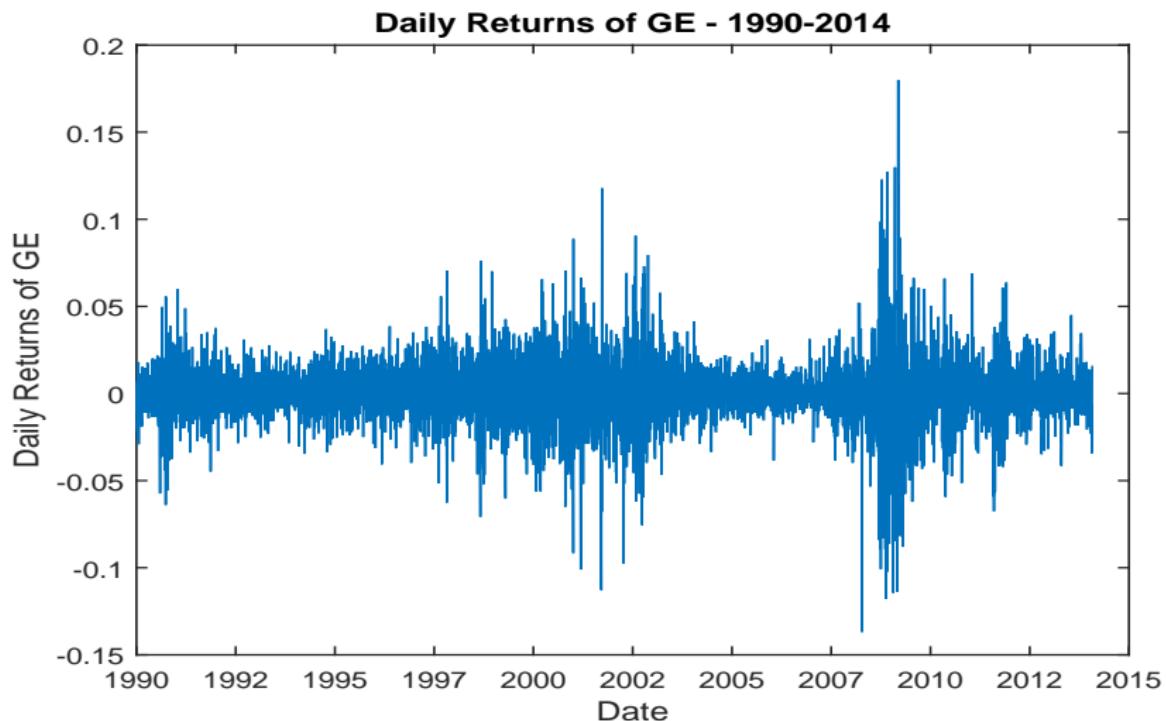
Let's take a look at the daily stock prices of General Electrics...



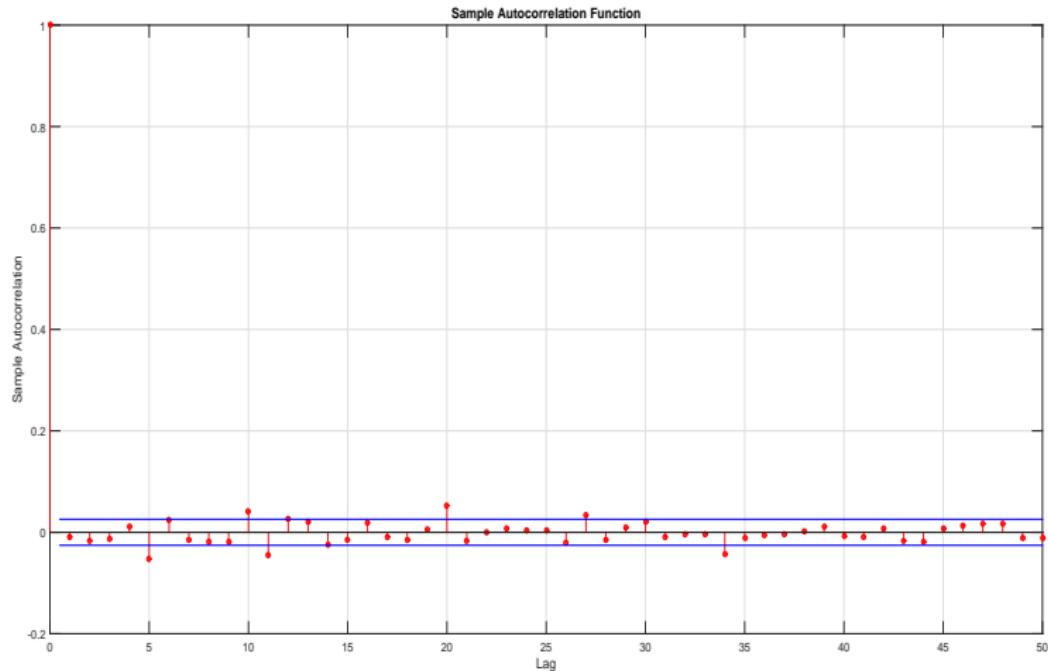
...their ACF...



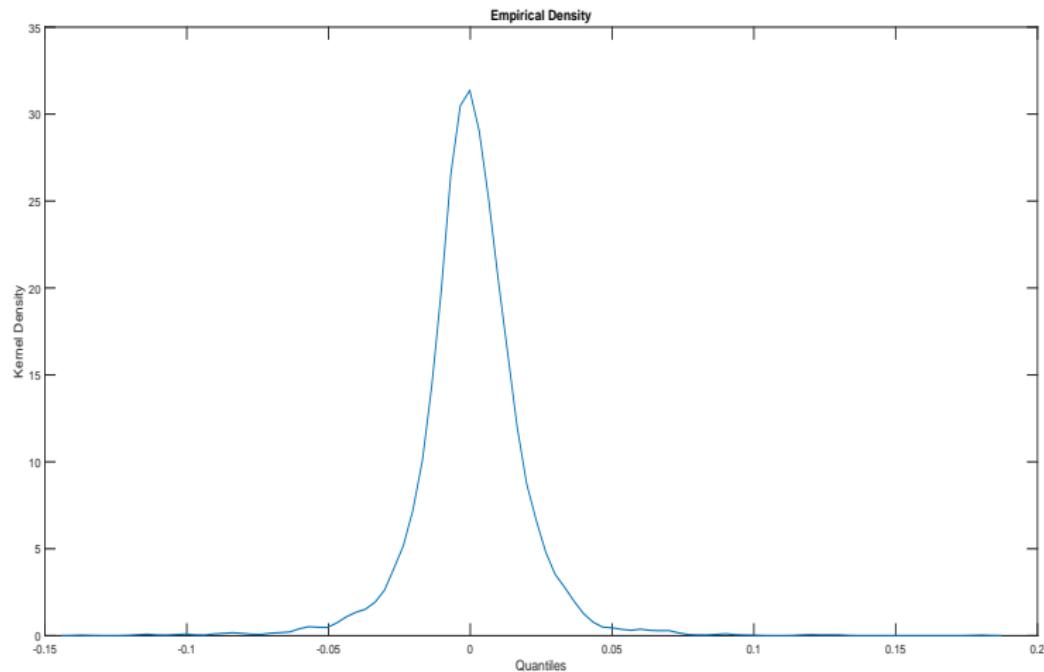
...and their associated log-returns...



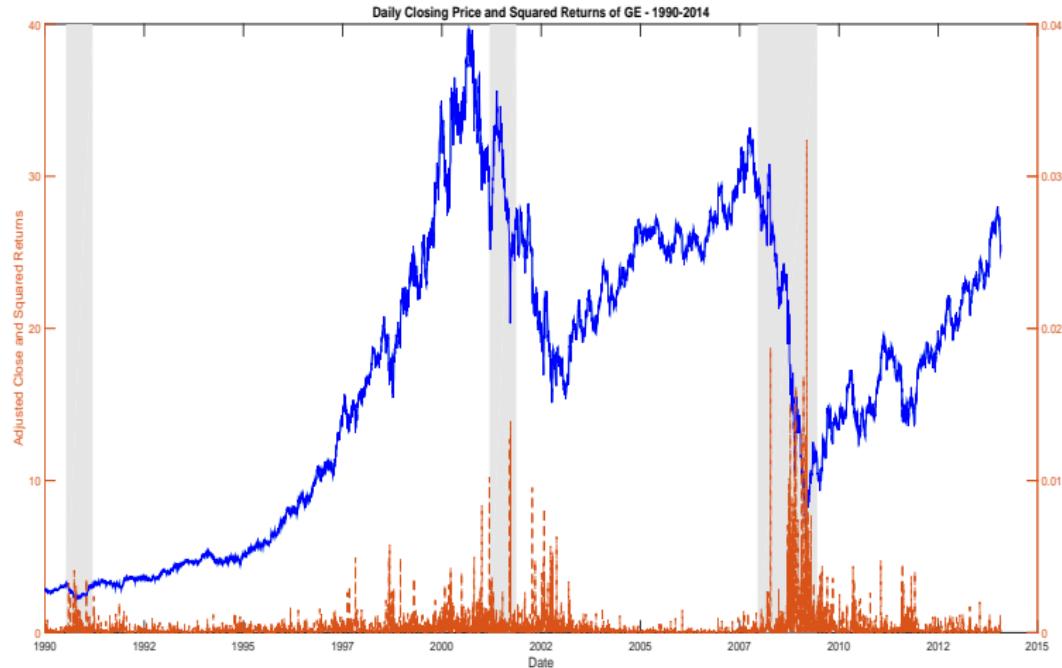
...the ACF of the log-returns...



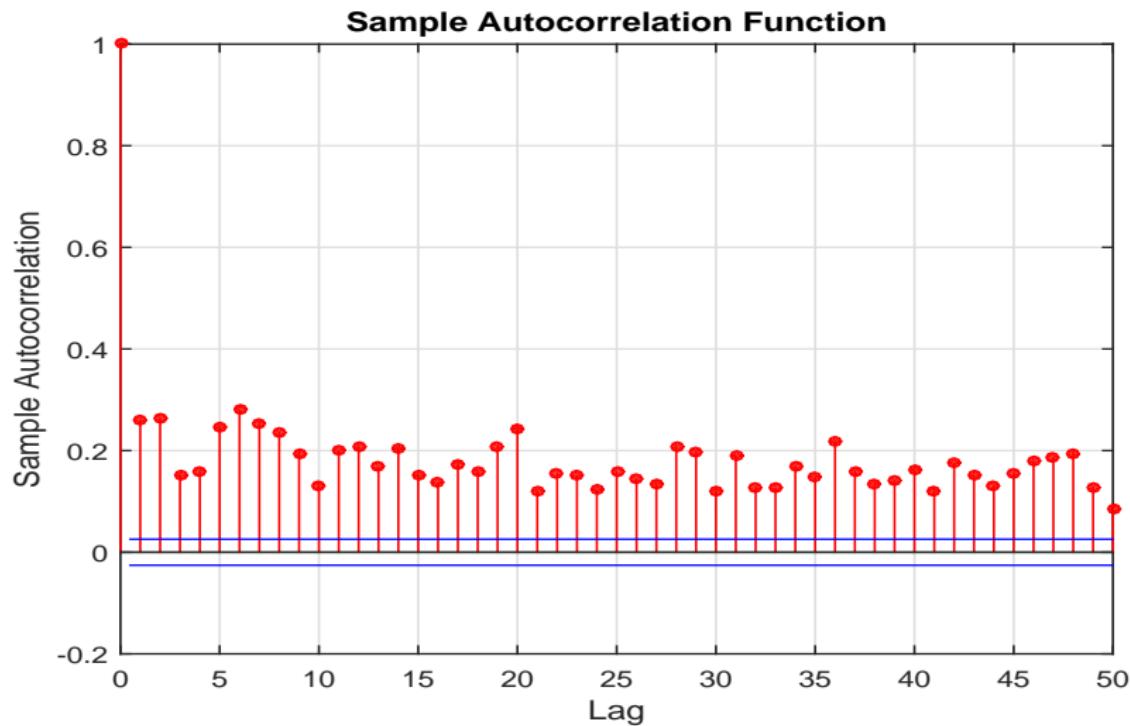
Their distribution?



What about the squares of log-returns?



...and their ACF?



Summarizing...

- If we look at the **dynamics**:
 - ① Daily stock prices are highly persistent (they are $I(1)$ variables)
 - ② Daily log-returns are **uncorrelated**...
 - ③ ... but their squares are **highly correlated**
 - ④ Variations are more pronounced and are **higher** during **financial crises**
- The empirical distribution of daily log-returns is **skewed** and has **fat-tails**
 - ① Under Gaussianity the probability of a return smaller than -10% is **1.9094e-08** (one event every 14500 years)
 - ② It happened **9** times in the sample (between 1990 and 2014)
- More stylized facts about the **multivariate** distribution of returns will also be discussed:
 - ① Correlation
 - ② Dependence in the **tail events**

REVIEW OF R

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Outline

This set of slides includes a review of R. It is divided in three parts:

- Introduction to R.
- Getting Started with R.
- Basic programming.

INTRODUCTION TO R

Leopoldo Catania

Aarhus University and CREATES

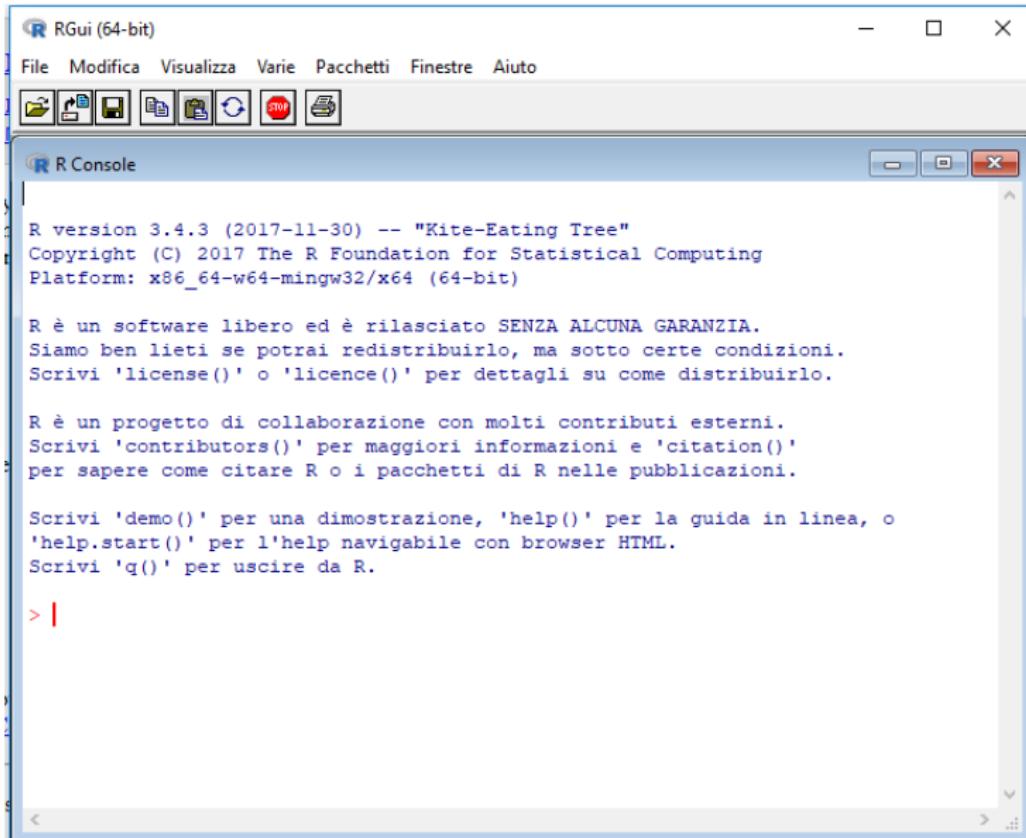
leopoldo.catania@econ.au.dk

The R programming language

- R is a free software environment for statistical computing and graphics.
- It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.
- R is maintained by its community! You can contribute to it's development.
- R has a modular structure: external libraries are freely (also) available from the R repository CRAN.

- Download: <https://cloud.r-project.org/>

Plain R



RStudio

- The plain R interface is quite minimalist...
- We will use the RStudio graphical interface. This is freely available from:
<https://www.rstudio.com/products/rstudio/download/>
- First install R, then install RStudio!

RStudio

The screenshot shows the RStudio interface with several tabs open:

- Code Editor:** Shows the content of `RcppFunctions.cpp`, which includes Rcpp code and calls to `fun_EM`.
- File:** Shows files like `OrganizeData.R`, `code.R`, `MonteCarlo.R`, `MonteCarlo_ExtractedResults.R`, `Ses.R`, and `DNA.R`.
- Environment:** Displays global variables and their values.
- Plots:** Shows a histogram of `mtab_low`.
- Packages:** Shows installed packages: `abind`, `adaptGMC`, `addressexamples`, `ADGofTest`, `assertthat`, `ATS`, `backports`, `base64enc`, and `BiocGenerics`.
- Help:** Shows help pages for `f_compute_stat`, `f_stat`, `f_trimmedmean`, and `f_trimmedtmean`.
- Console:** Shows R code execution, including the creation of matrices `mtab_low` and `mtab_medium`, and the writing of tables to disk.
- Terminal:** Shows the command `G:\Dropbox\Catania\GitHub\Projects\EM\main.R` being run.

```

76 iterate_EM <- function(mv = 10, fun_EM = EM_HMM_Pois_Multiv, cluster = NULL, ...){
77   fit_nonrandom = try(fun_EM(mv, randomInitialization = FALSE, ...), silent = TRUE)
78
79   if (!is(fit_nonrandom, "try-error")){
80     dLik_nonrandom = c(tail(fit_nonrandom$mlikSeries, 1))
81   } else {
82     iter = 0
83   }
84   while (is(fit_nonrandom, "try-error") | iter < 10) {
85
86     fit_nonrandom = try(fun_EM(mv, randomInitialization = FALSE, ...), silent = TRUE)
87     iter = iter + 1
88   }
89
90   if (!is(fit_nonrandom, "try-error")){
91     dLik_nonrandom = c(tail(fit_nonrandom$mlikSeries, 1))
92   } else {
93     dLik_nonrandom = NA
94   }
95
96   if (!is(dLik_nonrandom, "NA")){
97     dLik_nonrandom = c(tail(dLik_nonrandom$likSeries, 1))
98   }
99
100 }
101
102 if (is.null(cluster)){
103   fit_random = apply(1:10, function(b, mv, ...){
104     foo = try(fun_EM(mv = mv, RandomInitialization = TRUE, ...))
105
106     if (!is(foo, "try-error")){
107       foo = list(Likseries = NA)
108     }
109   })
110
111   return(foo)
112 }
113
114 return(mtab_low)
115
116 write.table(mtab_low, file = paste(spathSave, "mtab_low", ".txt", sep = ""),
117             sep = "\t", quote = FALSE, row.names = TRUE, col.names = FALSE)
118
119 write.table(mtab_medium, file = paste(spathSave, "mtab_medium", ".txt", sep = ""),
120             sep = "\t", quote = FALSE, row.names = TRUE, col.names = FALSE)
121
122 }
```

Much nicer...

RStudio panels

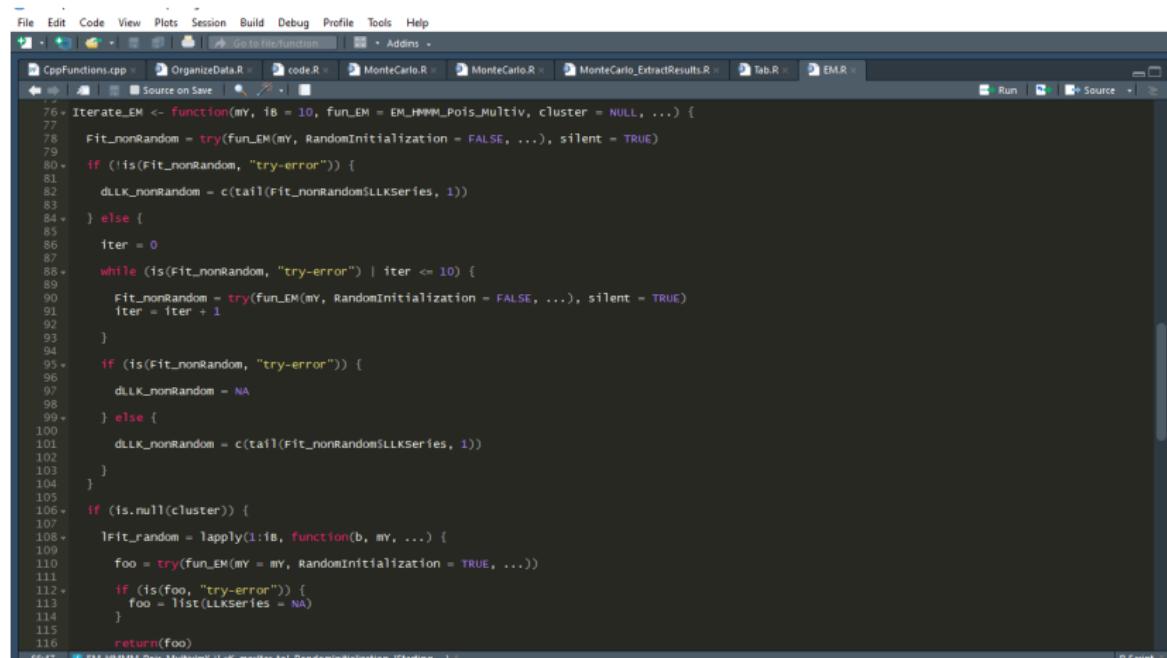
The screenshot shows the RStudio interface with several panels open:

- Source Panel:** Displays the code for `iterate_EM` and `write_DM`.
- Environment Panel:** Shows the global environment with objects like `tStat`, `trab`, `trab_low`, `trab_medium`, and `values`.
- Plots Panel:** Empty.
- Packages Panel:** Shows installed packages: abind, adagio, admb4MC, addressnames, RStudio Addins, ADGfT, assertthat, aTSA, backports, BB, and Bessel.
- Help Panel:** Empty.
- Viewers Panel:** Empty.

```

76+ iterate_EM <- function(mv, ik = 10, fun_EM = EM_HMM_Multiv, cluster = NULL, ...){
77+   Fit_norandom <- try(fun_EM(mv, randomInitialization = FALSE, ...), silent = TRUE)
78+   if (is(Fit_norandom, "try-error")) {
79+     iIter = 0
80+     dLk_norandom = c(tail(Fit_norandom$likSeries, 1))
81+   } else {
82+     iIter = 0
83+   }
84+   while (is(Fit_norandom, "try-error")) iIter <- 10L
85+   Fit_norandom = try(fun_EM(mv, randomInitialization = FALSE, ...), silent = TRUE)
86+   iIter = iIter + 1
87+   dLk_norandom = NA
88+ }
89+ if (is(Fit_norandom, "try-error")) {
90+   dLk_norandom = NA
91+ } else {
92+   dLk_norandom = c(tail(Fit_norandom$likSeries, 1))
93+ }
94+
95+ if (is.null(cluster)) {
96+   Fit_random = apply(1:ik, function(b, mv, ...) {
97+     foo = try(fun_EM(mv = mv, RandomInitialization = TRUE, ...))
98+     if (is(foo, "try-error")) {
99+       foo = list(likSeries = NA)
100+     }
101+     return(foo)
102+   })
103+   write_DM(0, sum_DM, cluster)
104+ }
105+
106+ trab_low = format(cround(trab_low, 2), scientific = FALSE)
107+ trab_medium = format(cround(trab_medium, 2), scientific = FALSE)
108+
109+ trab_low[, ncol(trab_low)] = paste(trab_low[, ncol(trab_low)], "\\\\"")
110+ trab_medium[, ncol(trab_medium)] = paste(trab_medium[, ncol(trab_medium)], "\\\\"")
111+
112+ write.table(trab_low, file = paste(trab_low, ".txt", sep = ""), sep = "\t", quote = FALSE, row.names = TRUE, col.names = FALSE)
113+
114+ write.table(trab_medium, file = paste(trab_medium, ".txt", sep = ""), sep = "\t", quote = FALSE, row.names = TRUE, col.names = FALSE)
115+
116+
117+ }
```

RStudio panels: source



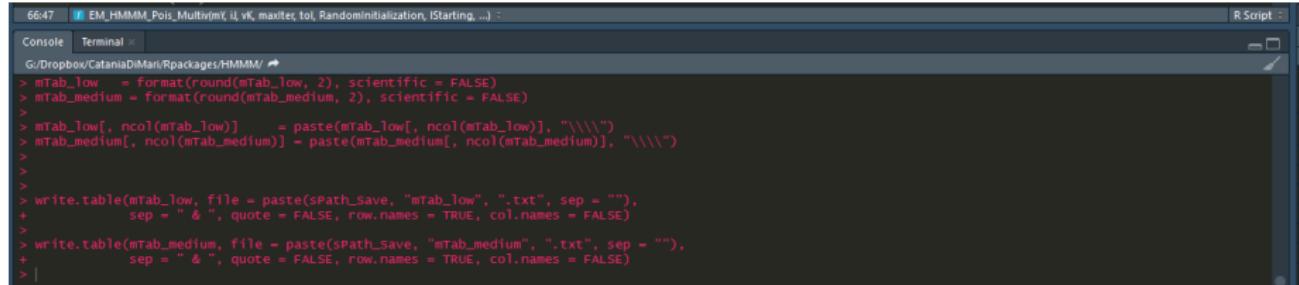
The screenshot shows the RStudio interface with the Source panel active. The code in the panel is as follows:

```

76+ Iterate_EM <- function(mY, iB = 10, fun_EM = EM_HMMML_Pois_Multiv, cluster = NULL, ...) {
77
78  fit_nonRandom = try(fun_EM(mY, RandomInitialization = FALSE, ...), silent = TRUE)
79
80+ if (is(fit_nonRandom, "try-error")) {
81
82  dLLK_nonRandom = c(tail(fit_nonRandom$LLKseries, 1))
83
84+ } else {
85
86  iter = 0
87
88+ while (is(fit_nonRandom, "try-error") | iter <= 10) {
89
90  fit_nonRandom = try(fun_EM(mY, RandomInitialization = FALSE, ...), silent = TRUE)
91  iter = iter + 1
92
93+ }
94
95+ if (is(fit_nonRandom, "try-error")) {
96
97  dLLK_nonRandom = NA
98+ } else {
99
100  dLLK_nonRandom = c(tail(fit_nonRandom$LLKseries, 1))
101
102+ }
103}
104}
105
106+ if (is.null(cluster)) {
107
108+ fit_random = lapply(1:iB, function(b, mY, ...) {
109
110  foo = try(fun_EM(mY = mY, RandomInitialization = TRUE, ...))
111
112+ if (is(foo, "try-error")) {
113  foo = list(LLKseries = NA)
114  }
115
116  return(foo)
117}
118
119  R Script : EM_HMMML_Pois_Multiv.mllk.R, maxIter, tol, RandomInitialization, (Starting, ...)
```

Where you write your code.

RStudio panels: Console



The screenshot shows the RStudio interface with the 'Console' tab selected. The title bar indicates the current file is 'EM_HMMM_Pois_Multiv(mY, lI, vK, maxiter, tol, RandomInitialization, lStarting, ...)' located at 'G:/Dropbox/CataniaDiMari/Rpackages/HMMM/'. The console window displays R code for generating two matrices, mTab_low and mTab_medium, by rounding their respective columns to 2 decimal places. It then pastes these columns into single strings separated by backslashes. Finally, it writes these strings to files named 'mTab_low.txt' and 'mTab_medium.txt' using the 'write.table' function. The code is as follows:

```
66:47 EM_HMMM_Pois_Multiv(mY, lI, vK, maxiter, tol, RandomInitialization, lStarting, ...)

Console Terminal R Script
G:/Dropbox/CataniaDiMari/Rpackages/HMMM/
> mTab_low = format(round(mTab_low, 2), scientific = FALSE)
> mTab_medium = format(round(mTab_medium, 2), scientific = FALSE)
>
> mTab_low[, ncol(mTab_low)] = paste(mTab_low[, ncol(mTab_low)], "\\\\"")
> mTab_medium[, ncol(mTab_medium)] = paste(mTab_medium[, ncol(mTab_medium)], "\\\\"")
>
>
> write.table(mTab_low, file = paste(sPath_save, "mTab_low", ".txt", sep = ""),
+             sep = " & ", quote = FALSE, row.names = TRUE, col.names = FALSE)
>
> write.table(mTab_medium, file = paste(sPath_save, "mTab_medium", ".txt", sep = ""),
+             sep = " & ", quote = FALSE, row.names = TRUE, col.names = FALSE)
> |
```

Where you run your code.

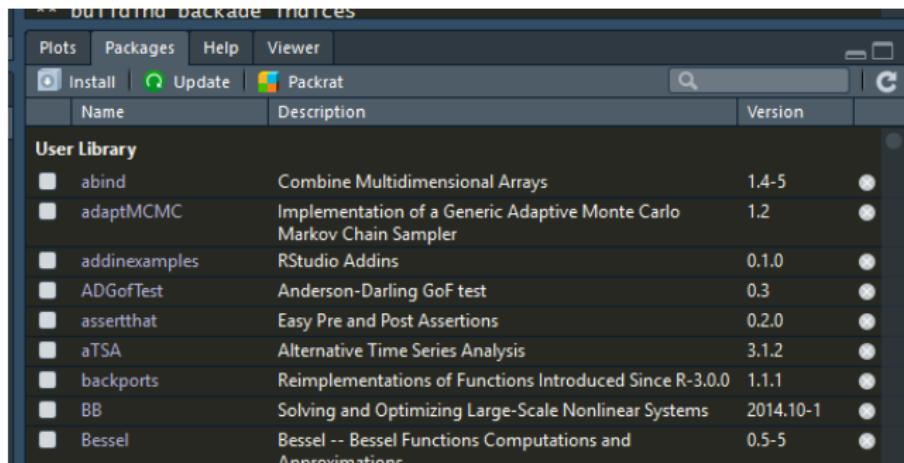
RStudio panels: Environment

The screenshot shows the RStudio interface with the 'Environment' tab selected in the top navigation bar. The main area displays a list of objects categorized into 'Data', 'values', and 'Functions'. Each entry includes the object name, its type, and a preview of its content.

lstat	List of 48	
lTab	List of 2	
mout	Large matrix (128000 elements, 1008.4 Kb)	
mTab_low	chr [1:12, 1:12] "-0.01" "-0.04" "-0.04" "-0.09..."	
mTab_medium	chr [1:12, 1:12] "-0.06" "-0.09" "-0.04" "-0.05..."	
values		
vFiles	chr [1:72] "T.1000.N.1.J.2.L.2.Sep.low.rdata" "T..."	
VJ	num [1:2] 2 4	
VL	num [1:2] 2 4	
VN	num [1:3] 1 3 6	
vRowNames	chr [1:12] "\$j = 2\$ \$k = 2\$" "\$j = 2\$ \$k = 4\$" "\$..."	
vSeparation	chr [1:2] "low" "medium"	
vspec	chr [1:2] "T.1000.N.6.J.4.L.4.Sep.medium" "T.5000..."	
VT	num [1:2] 1000 5000	
Functions		
f_Compute_stat	function (mout)	
f_stat	function (x, ...)	
f_trimmed_mean	function (vx, dLower = 0.025, dUpper = 0.975)	
f_trimmed_rmse	function (vx, dLower = 0.025, dUpper = 0.975)	

Where you see the object stored in memory

RStudio panels: Plots, Packages, Help



- Figures are displayed
- List of available packages
- Help

Standard functions in R

- R comes with a lot of standard functions
 - Printing stuff to the screen, inverting matrices, performing OLS, generating random numbers... no need to reinvent the wheel!
- These functions are also available in R packages. However these are 'special' packages which are included by default in R and don't need to be loaded. Examples are: 'stats', 'base', 'graphics'....

Standard functions in R

- These functions generally have sensible names:
 - `print` to print something to the screen
 - `plot` to plot figures
 - `solve` to invert a matrix
 - `lm` to perform OLS ('linear model' → 'lm')
 - `sum`, `mean`, `median`...
- Obviously, too many to list here. R manuals are freely available online!
<https://cran.r-project.org/>

External packages in R

More than 12'000 external packages are available on CRAN:

<https://cran.r-project.org/web/packages/>.

Knowing the name of a package, say ("rugarch"), this can be installed in R with:

```
install.packages("rugarch")
```

After a package has been installed, in order to load it into the R environment we need to run:

```
library("rugarch").
```

Types of variables

The function `class()` allows you to find the class of an R object. For instance:

```
> cfoo = "sun"
> class(cfoo)
[1] "character"
>
> bfoo = TRUE
> class(bfoo)
[1] "logical"
>
> dfoo = 1.785632458
> class(dfoo)
[1] "numeric"
```

The "foo" terminology stands for a variable which is not important, i.e. a variable where intermediate results are stored. We will later see that "c", "b", and "d" before the name "foo" does actually have a reason!

Vectors, Matrices, Arrays, Data frames

Data can be organized in different ways in R depending on the particular needs.

If we plan to do linear algebra we want to use:

```
> vY = c(1, 2, 5.48652) # vectors  
  
> mY = matrix(c(1, 2, 3,  
        4, 5, 6,  
        7, 8, 9.485), ncol = 3) # matrices  
  
> aY = array(1:27, dim = c(3, 3, 3))# arrays
```

If we want to organize data (numeric/character or mixed) we use:

```
d = data.frame(x = 1, y = 1:10, fac = LETTERS[1:10]) #data frames
```

vectors, matrices and arrays can be: "numeric" or "character" i.e., they can contain only two types or variables (we cannot have a vector with some elements "numeric" and others "character"). data.frames can contain both.

Access elements

To access elements of vectors, matrices and arrays we use the square brackets:

```
> vY[1]  
[1] 1  
> vY[1:2]  
[1] 1 2  
> vY[c(3, 1)]  
[1] 5.48652 1.00000  
> mY[1, 1]  
[1] 1  
> mY[1, ]  
[1] 1 4 7  
> mY[1, c(1, 3)]  
[1] 1 7  
> aY[1, 1, 1]  
[1] 1  
> aY[1, 1, ]  
[1] 1 10 19
```

HELP!

When you are in troubles the `help()` function is your friend! Whenever you want to understand the functioning of any function, say `list`, you type:

```
> help(list)
```

This is equivalent to:

```
> ?list
```

If you don't remember the function name but part of it, or something related to it, you can use the "`??`" operator, for example:

```
> ??download
```

GETTING STARTED WITH R

Leopoldo Catania

Aarhus University and CREATES

leopoldo.catania@econ.au.dk

Outline

- Arithmetic
- Variables, Functions
- Vectors
- Matrices
- Logical expressions
- Missing data: NA
- Lists
- Factors
- Dataframes

Working Directory

When you run R, it nominates one of the directories on your hard drive as a working directory, which is where it looks for user-written programs and data files.

You can determine the current working directory using the command `getwd`.

You can do this using the command `setwd("dir")`, where `dir` is the directory address.

```
> getwd()
[1] "C:/Users/leopo/Documents"
>
> dir = "G:/foo" #this path needs to exist
>
> setwd(dir)
>
> getwd()
[1] "G:/foo"
```

Arithmetic

R uses the usual symbols for addition +, subtraction -, multiplication *, division /, and exponentiation ^.

Parentheses () can be used to specify the order of operations.

```
> (1 + 1/100)^100  
[1] 2.704814
```

Notice that by default R prints 7 significant digits. You can change the display to x digits using `options(digits = x)`.

See `help(options)` for other options.

Build in Functions

R has a number of built-in functions, for example `sin(x)`, `cos(x)`, `tan(x)`, (all in radians), `exp(x)`, `log(x)`, and `sqrt(x)`. Some special constants such as `pi` (π) are also predefined.

```
> exp(1)
[1] 2.718282
> options(digits = 16)
> exp(1)
[1] 2.718281828459045
> pi
[1] 3.141592653589793
> sin(pi/6)
[1] 0.4999999999999999
```

The functions `floor(x)` and `ceiling(x)` round down and up, respectively, to the nearest integer.

Defining variables

To assign a value to a variable we use (almost equivalently) the assignment commands `=` and `<-`. For example:

```
> iX <- 100
```

and

```
> iX = 100
```

are equivalent. Later we see when this is not the case. We can perform operations with `iX`:

```
> iX  
[1] 100  
> (1 + 1/iX)^iX  
[1] 2.704814
```

Notice that:

```
> iX = iX + 1  is allowed.  
> iX  
[1] 101
```

Functions

In mathematics a function takes one or more arguments (or inputs) and produces one or more outputs (or return values). Functions in R work in an analogous way.

Consider the `seq` function. This function allows you to create sequence of numbers:

```
> seq(from = 1, to = 9, by = 2)  
[1] 1 3 5 7 9
```

Notice that by default (see `help(seq)`), the `by` argument is equal to 1, such that:

```
> seq(from = 1, to = 3)  
[1] 1 2 3
```

Arguments of functions

Every function has a default order for the arguments. For `seq` this is: `from`, `to`, `by`.

If you provide arguments in this order, then they do not need to be named:

```
> seq(1, 9, 2)
[1] 1 3 5 7 9
```

But you can choose to give the arguments out of order provided you give them names in the format `argument_name = expression`.

```
> seq(by = 2, to = 9, from = 1)
[1] 1 3 5 7 9
```

Which is of course different from:

```
> seq(2, 9, 1)
[1] 2 3 4 5 6 7 8 9
```

Vectors

We have already seen the `c` function. However, vectors can be created by any function that returns vectors as output. For instance `seq` and `rep`. The `rep(vX, n)` function replicates the vector `vX`, `n` times:

```
> vX = c(1, 3, 4)
> rep(vX, 3)
[1] 1 3 4 1 3 4 1 3 4
```

The function `c` also accepts vectors as inputs:

```
> vX <- seq(1, 20, by = 2)
[1] 1 3 5 7 9 11 13 15 17 19
```

```
> vY <- rep(3, 4)
[1] 3 3 3 3
```

```
> vZ <- c(vY, vX)
[1] 3 3 3 3 1 3 5 7 9 11 13 15 17 19
```

Vector operations

All algebraic operations are defined for vectors and act on each element separately, that is, elementwise:

```
> vX <- c(1, 2, 3)
> vY <- c(4, 5, 6)
> vX * vY
[1] 4 10 18
> vX + vY
[1] 5 7 9
> vY ^ vX
[1] 4 25 216
```

However care should be taken when vectors of unequal length are used:

```
> c(1, 2, 3) + c(1, 2)
[1] 2 4 4
```

Warning message:

In c(1, 2, 3) + c(1, 2) :

longer object length is not a multiple of shorter object length

Vector operations

When you apply an algebraic expression to two vectors of unequal length, R automatically repeats the shorter vector until it has something the same length as the longer vector.

```
> c(1, 2, 3, 4) + c(1, 2)
[1] 2 4 4 6
> (1:10) ^ c(1, 2)
[1] 1 4 3 16 5 36 7 64 9 100
```

This happens even when the shorter vector is of length 1, allowing the shorthand notation:

```
> 2 + c(1, 2, 3)
[1] 3 4 5
> 2 * c(1, 2, 3)
[1] 2 4 6
```

For example using the modulus operator: $a \bmod b$ (%):

```
1:20 %% 3
[1] 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2
```

Missing Data: NA

In real experiments it is often the case, for one reason or another, that certain observations are missing.

Depending on the statistical analysis involved, missing data can be ignored or invented (a process called imputation).

R represents missing observations through the data value NA. They can be mixed in with all other kinds of data:

```
> vA <- c(11, NA, 13)  
> vA  
[1] 11 NA 13
```

Performing analysis with NAs can be problematic:

```
> mean(vA) # NAs can propagate  
[1] NA
```

The logical argument `na.rm` is often specified to deal with NAs.

```
> mean(vA, na.rm = TRUE) # NAs can be removed  
[1] 12
```

Searching for NAs

Sometimes you want to search for NAs in your dataset to prevent misleading results at the end of your analysis. The function `is.na` searches for NAs inside vectors (also matrices and arrays) and returns a logical output of the same size of the input provided:

```
> vA <- c(11, NA, 13)
> is.na(vA) # identify missing elements
[1] FALSE TRUE FALSE
```

When `length(vA)` is very large we might want to write:

```
> any(is.na(vA)) # are any missing?
[1] TRUE
```

In order to remove NAs we can use:

```
> na.omit(vA)
[1] 11 13
attr(,"na.action")
[1] 2
attr(,"class")
[1] "omit"
Leopoldo Catania
```

Logical Expressions

A logical expression is formed using the comparison operators:

- `<` lower than
- `>` bigger than
- `≤` lower or equal than
- `≥` bigger or equal than
- `==` equal to
- `!=` not equal to
- `&` and
- `|` or
- `!` not

These operators return a logical output:

```
> vX = c(1, 2, 3, 4)
> vX == 2
[1] FALSE  TRUE FALSE FALSE
> vX != 2
[1]  TRUE FALSE  TRUE  TRUE
```

Logical Expressions

Vectors (but also matrices and arrays) can be accessed also using logical indicators:

```
> vX <- c(1, 3, 4, 18)
```

```
> vX > 2
```

```
[1] FALSE TRUE TRUE TRUE
```

```
> vX[vX > 2 & vX < 10]
```

```
[1] 3 4
```

The subset function can be used for a similar scope:

```
> subset(vX, subset = vX > 2)
```

```
[1] 3 4 18
```

Logical Expressions: & and |

Take two logical objects:

```
> vX = 4
> vY = vX > 2
> vY
[1] TRUE
>
> vZ = vX < 3
> vZ
[1] FALSE
```

The & operator returns TRUE if both vY and vZ are TRUE, FALSE otherwise:

```
> vY & vZ
[1] FALSE
```

The | operator returns TRUE if y or z are TRUE, FALSE otherwise:

```
> vY | vZ
[1] TRUE
```

Sequential && and ||

The logical operators `&&` and `||` are sequentially evaluated versions of `&` and `|`, respectively.

To evaluate `vX & vY`, R first evaluates `vX` and `vY`, then returns TRUE if `vX` and `vY` are both TRUE, FALSE otherwise.

To evaluate `vX && vY`, R first evaluates `vX`. If `vX` is FALSE then R returns FALSE without evaluating `vY`. If `vX` is TRUE then R evaluates `vY` and returns TRUE if `vY` is TRUE, FALSE otherwise.

Sequential evaluation of `vX` and `vY` is useful when `vY` is not always well defined, or when `vY` takes a long time to compute.

Sequential && and ||

As an example of the first instance, suppose we wish to know if $vX * \sin(1/vX) = 0$.

```
> vX <- 0  
> vX * sin(1/vX) == 0
```

```
[1] NA
```

Warning message:

In $\sin(1/vX)$: NaNs produced

```
> (vX == 0) | (sin(1/vX) == 0)
```

```
[1] TRUE
```

Warning message:

In $\sin(1/vX)$: NaNs produced

```
> (vX == 0) || (sin(1/vX) == 0)
```

```
[1] TRUE
```

Note that `&&` and `||` only work on scalars, whereas `&` and `|` work on vectors on an element-by-element basis.

Matrices

We have already seen how to build matrices in R:

```
> mA <- matrix(1:6, nrow = 2, ncol = 3, byrow = TRUE)
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 4 5 6
```

To retrieve the dimension of a matrix use dim:

```
> dim(mA)
[1] 2 3
```

Useful functions for matrices:

- diag, extract the diagonal elements and return a vector or create a diagonal matrix, see `help(diag)`!
- rbind, join matrices with rows of the same length (stacking vertically)
- cbind, join matrices with columns of the same length (stacking horizontally)
- solve, invert a matrix
- eigen, extract eigenvalues and associated eigenvectors of a matrix
- t, transpose of matrix

Multiplication with matrices

Define:

```
> mA <- matrix(c(3, 5, 2, 3), nrow = 2, ncol = 2)
> mB <- matrix(c(9, 4, 1, 2), nrow = 2, ncol = 2)
>
> mA
      [,1] [,2]
[1,]     3     2
[2,]     5     3
> mB
      [,1] [,2]
[1,]     9     1
[2,]     4     2
```

Multiplication with matrices

Element-wise multiplication, $A \circ B$ (Hadamard product):

```
> mA * mB
 [,1] [,2]
[1,]   27    2
[2,]   20    6
```

Matrix multiplication, AB :

```
> mA %*% mB
 [,1] [,2]
[1,]   35    7
[2,]   57   11
```

Kronecker product, $A \otimes B$:

```
> kronecker(mA, mB)
 [,1] [,2] [,3] [,4]
[1,]   27    3   18    2
[2,]   12    6    8    4
[3,]   45    5   27    3
[4,]   20   10   12    6
```

Lists

We have seen that all the elements of a vector have to be of the same type: numeric, character, or logical. The type of the vector is called: mode

A list is an indexed set of objects (and so has a length) whose elements can be of different types, including other lists! The mode of a list is list.

A list is just a generic container for other objects and the power and utility of lists comes from this generality.

In R lists are often used for collecting and storing complicated function output.

For example, the first element of a list can be a vector, the second can be another list and the third can be a matrix.

Lists

A list is created using the `list(...)` command, with comma-separated arguments:

```
> my.list <- list("one", TRUE, 3, c("f", "o", "u", "r"))
```

Double square brackets are used to extract a single element:

```
> my.list[[1]]  
[1] "one"  
> mode(my.list[[1]])  
[1] "character"
```

Single square brackets are used to select a sublist:

```
> my.list[1]  
[[1]]  
[1] "one"  
> mode(my.list[1])  
[1] "list"
```

Lists

When displaying a list, R uses double square brackets [[1]], [[2]], etc., to indicate list elements, then single square brackets [1], [2], etc., to indicate vector elements within the list.

The elements of a list can be named when the list is created, using arguments of the form name1 = x1, name2 = x2, etc.:

```
my.list <- list(first = "one", second = TRUE, third = 3,  
fourth = c("f","o","u","r"))
```

```
> names(my.list)  
[1] "first" "second" "third" "fourth"
```

```
> my.list$second  
[1] TRUE
```

Note the use of the \$ operator to access named lists.

Lists

Alternatively, a list elements can be named later by assigning a value to the `names` attribute:

```
> my.list <- list( "one", TRUE, 3, c("f", "o", "u", "r"))

> names(my.list) <- c("first", "second", "third", "fourth")
```

To flatten a list `x`, that is convert it to a vector, we use `unlist(x)`:

```
> x <- list(1, c(2, 3), c(4, 5, 6))
> unlist(x)
[1] 1 2 3 4 5 6
```

If the list object itself comprises lists, then these lists are also flattened, unless the argument `recursive = FALSE` is set in `unlist`.

Dataframes

We have already seen how to work in R with numbers, strings, and logical values.

We have also worked with homogeneous collections of such objects, grouped into numeric, character, or logical vectors.

The defining characteristic of the vector data structure in R is that all components must be of the same mode.

Obviously to work with datasets from real experiments we need a way to group data of differing modes!

Dataframes

Suppose to have the following dataset representing a forestry experiment in which we randomly selected a number of plots and then from each plot selected a number of trees.

Plot	Tree	Species	Diameter (cm)	Height (m)
2	1	DF	39	20.5
2	2	WL	48	33.0
3	2	GF	52	30.0
3	5	WC	36	20.7
3	8	WC	38	22.5
:	:	:	:	:

For each tree we measured its height and diameter (which are **numeric**), and also the species of tree (which is a **character string**).

Dataframes

As experimental data collated in a table look like an array, you may be tempted to represent it in R as a matrix.

But in R matrices cannot contain heterogeneous data (data of different modes, like numeric and character).

Lists and dataframes are able to store much more complicated data structures than matrices.

A dataframe is a list of vectors restricted to be of equal length. Each vector (column of the dataframe) can be of any of the basic modes of object. To create a dataframe we write:

```
mData <- data.frame(Plot = c(1, 2, 2, 5, 8, 8),  
                     Tree = c(2, 2, 3, 3, 3, 2),  
                     Species = c("DF", "WL", "GF", "WC", "WC", "GF"),  
                     Diameter = c(39, 48, 52, 35, 37, 30),  
                     Height = c(20.5, 33.0, 30.0, 20.7, 22.5, 20.1))
```

Dataframes: extract

Each column, or variable, in a dataframe has a unique name. We can extract that variable by means of the dataframe name, the column name, the a dollar sign, or as we do for a matrix:

```
> mData$Diameter  
[1] 39 48 52 35 37 30
```

```
> mData[["Diameter"]]  
[1] 39 48 52 35 37 30
```

```
> mData[[4]]  
[1] 39 48 52 35 37 30
```

```
> mData[, 4]  
[1] 39 48 52 35 37 30
```

```
> mData[, "Diameter"]  
[1] 39 48 52 35 37 30
```

Dataframes: assign

To assign a new variable to a dataframe we write:

```
mData$newdata <- c(1, 2, 3, 4, 5, 6)
```

If the new variable is the same across all the observations we can write:

```
mData$newdata2 <- TRUE
```

This also works as long as the number of rows of the dataframe is a multiple of the length of the new variable (if it is not a multiple we get an error):

```
mData$newdata3 <- c("one", "two")
```

```
> mData
```

	Plot	Tree	Species	Diameter	Height	newdata1	newdata2	newdata3
1	1	2	DF	39	20.5	1	TRUE	one
2	2	2	WL	48	33.0	2	TRUE	two
3	2	3	GF	52	30.0	3	TRUE	one
4	5	3	WC	35	20.7	4	TRUE	two
5	8	3	WC	37	22.5	5	TRUE	one
6	8	2	GF	30	20.1	6	TRUE	two

BASIC PROGRAMMING

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Outline

- The if statement
- The for loop
- The while loop
- Vector-based programming
- Load data in R
- Write data from R
- Plotting

Programming: Intro

This lecture introduces a set of basic programming constructs, which are the building blocks of most programs.

Some of these tools are used by practically all programming languages, for example, conditional execution by `if` statements, and looped execution by `for` and `while` statements.

Notice that, code that seems to be efficient in another language may not be efficient in R. For example, R is known to be quite slow in computing `for` and `while` loops. On the other hand, R is fast in doing vector-based operations.

Programming: Intro

A program or script is just a list of commands, which are executed one after the other.

A program is usually composed by three parts:

- input
- computations
- output

Recall that documentation also plays an important role!!!

Usually, we write the list of commands in separate files, called a scripts, which we can save in our Hard Drive.

Suppose we have a program saved as `prog.r` in the working directory (recall the `getwd` and `setwd` functions). In order to run or execute the program we use the command:

```
source("prog.r")
```

The if statement

It is often useful to choose the execution of some or other part of a program to depend on a condition. The if function has the form

```
if (logical_expression) {  
expression_1  
...  
}
```

A natural extension of the if command includes an else part:

```
if (logical_expression) {  
expression_1  
...  
} else {  
expression_2  
...  
}
```

The if statement

When an if expression is evaluated, if logical_expression is TRUE then the first group of expressions is executed and the second group of expressions is not executed. Conversely if logical_expression is FALSE then only the second group of expressions is executed.

```
dX = rnorm(1)

if (dX > 0) {

  print("x is positive")

} else {

  print("x is nonpositive")

}
```

Nested if

Nested if statements are constructed using the `else if` command:

```
if (logical_expression_1) {  
  expression_1  
  ...  
} else if (logical_expression_2) {  
  expression_2  
  ...  
} else if (logical_expression_3) {  
  expression_3  
  ...  
} else {  
  expression_4  
  ...  
}
```

The for loop

The `for` command has the following form, where `x` is a simple variable and `vector` is a vector.

```
for (x in vector) {  
  expression_1  
  ...  
}
```

When executed, the `for` command executes the group of expressions within the braces `{}` once for each element of `vector`:

```
for (iX in 1:50) {  
  print(iX)  
}
```

Note that `vector` can be a `list`, which we cover later.

for loop example

Consider the following for loop example:

```
> vX <- seq(1, 9, by = 2)
[1] 1 3 5 7 9
> dSum_x <- 0
> for (iX in vX) {
  dSum_x <- dSum_x + iX
  cat("The current loop element is", iX, "\n")
  cat("The cumulative total is", dSum_x, "\n")
}
```

The current loop element is 1

The cumulative total is 1

...

The current loop element is 9

The cumulative total is 25

```
> sum(vX)
```

```
[1] 25
```

```
> cumsum(vX)
```

```
[1] 1 4 9 16 25
```

The while loop

Often we do not know beforehand how many times we need to go around a loop. That is, each time we go around the loop, we check some condition to see if we are done yet.

In this situation we use a `while` loop, which has the form:

```
while (logical_expression) {  
# This should have an impact on logical_expression  
expression_1  
...  
}
```

When a `while` command is executed, `logical_expression` is evaluated first. If it is `TRUE` then the group of expressions in braces `{}` is executed.

Until `logical_expression` becomes `FALSE`, the `while` command executes the group of expressions in braces `{}`.

Warning: while loop can run forever!

Difference between for and while loops

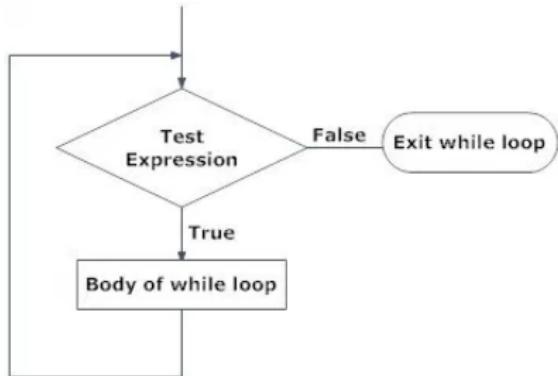
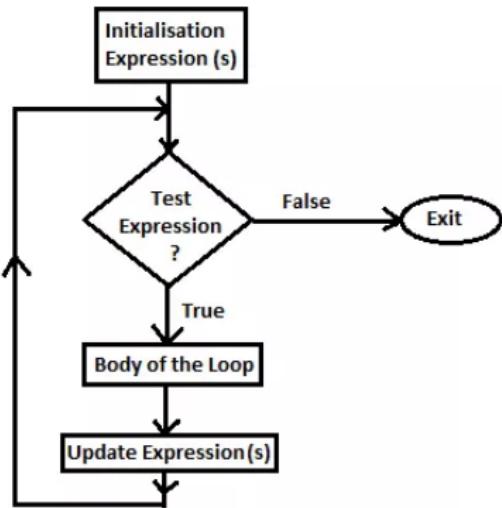


Figure: Flowchart of while loop

Vector-based programming

It is often necessary to perform an operation upon each of the elements of a vector.

R is set up so that such programming tasks can be accomplished using vector operations rather than looping.

Using vector operations is more efficient computationally, as well as more concise literally. For example, find the sum of the first n squares:

```
> iN <- 100  
> dS <- 0  
> for (i in 1:iN) {  
  dS <- dS + i^2  
}  
> dS  
[1] 338350  
  
> sum((1:iN)^2)  
[1] 338350
```

Computational time: loop vs vector-operation

```
> iN <- 1e8
> system.time({
  dS <- 0
  for (i in 1:iN) {
    dS <- dS + i^2
  }
  dS
})
      user  system elapsed
      3.41    0.00   3.43
> system.time({sum((1:iN)^2)})
      user  system elapsed
      0.44    0.09   0.53
```

In both approaches the vector `1:N` needs to be created. This occupies approximately 400MB of RAM. Can you think of an alternative procedure which does not require the evaluation of `1:iN`?

Load data in R

R implements functionalities to read from text files like: .txt and .csv. In order to read excel files external libraries are required such as: **excel**, **gdata**, **rodbc**, **XLConnect**, **xls**, **xlsReadWrite**, **xlsx** are needed.

R provides a number of ways to read data from a file: `scan`, `read.table`, `read.csv`. In this course we will mostly use `read.table`.

```
read.table(file, header = FALSE, sep = "", dec = ".",
           row.names, col.names, na.strings = "NA", skip = 0, ...)
```

... means that other arguments are available. See `help(read.table)`.

- **file**: the name of the file which the data are to be read from. Each row of the table appears as one line of the file. If it does not contain an absolute path, the file name is relative to the current working directory, `getwd()`.
- **header**: a logical value indicating whether the file contains the names of the variables as its first line.

The `read.table` function

- `sep`: the field separator character. Values on each line of the file are separated by this character. If `sep = ""` (the default) the separator is 'white space'.
- `dec`: the character used in the file for decimal points.
- `row.names`: a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row name.
- `col.names`: a vector of optional names for the variables. The default is to use "V" followed by the column number.
- `na.strings`: a character vector of strings which are to be interpreted as NA values.
- `skip`: the number of lines of the data file to skip before beginning to read data.

Load MAERSK historical prices

Load the MAERSK historical prices contained in the MAERSK-B.CO.csv file located in your working directory getwd().

```
mData = read.table(file = "MAERSK-B.CO.csv",
                    sep = ",",
                    dec = ".",
                    header = TRUE,
                    row.names = 1,
                    na.strings = "null")
```

```
> head(mData)
```

	Open	High	Low	Close	Adj.Close	Volume
2000-02-01	7640.00	7666.67	7333.33	7466.67	2267.703	4590
2000-02-02	7466.67	7666.67	7400.00	7566.67	2298.074	5640
2000-02-03	7741.07	7800.00	7659.00	7800.00	2368.939	3090
2000-02-04	7800.00	7800.00	7525.80	7600.00	2308.197	1185
2000-02-07	7660.00	7666.67	7566.67	7600.00	2308.197	1035
2000-02-08	7600.00	8200.00	7600.00	8110.80	2463.332	12990

```
> dim(mData)
```

```
[1] 4571     6
```

Descriptive Statistics of the MAERSK log-returns

First search for NAs:

```
> any(is.na(mData[, "Adj.Close"]))
[1] TRUE
```

how many ?

```
> length(which(is.na(mData[, "Adj.Close"])))
[1] 59
```

Compute financial log-returns omitting the NAs:

```
vY = diff(log(na.omit(mData[, "Adj.Close"])))
```

Compute descriptive statistics:

```
> c("mean" = mean(vY), "sd" = sd(vY), "median" = median(vY))
      mean          sd      median
0.0003439318 0.0220065696 0.0000000000
```

Output a file

R provides a number of commands for writing output to a file.

We will generally use `write.table` for writing numeric values and `cat` for writing text, or a combination of numeric and character values.

The command `write.table` has the form:

```
write.table(x, file = "", quote = TRUE, sep = " ",  
           na = "NA", dec = ".", row.names = TRUE,  
           col.names = TRUE, ...)
```

- `x`: the object to be written, preferably a `matrix` or `data.frame`.
- `file`: character string naming a file.

The write.table function

- quote: a logical value (TRUE or FALSE). If TRUE, any character columns will be surrounded by double quotes ("").
- sep and dec work as in read.table.
- na: the string to use for missing values in the data.
- row.names: either a logical value indicating whether the row names of x are to be written along with x, or a character vector of row names to be written.
- col.names: either a logical value indicating whether the column names of x are to be written along with x, or a character vector of column names to be written.

Example:

```
write.table(vY, file = "MAERSK_retuns.txt", row.names = FALSE,  
           col.names = FALSE, dec = ".")
```

Are you able to set row.names equal to the log-returns dates?

Plotting

R provides a huge number of plotting routines. The generic function for plotting of R objects is `plot`. See `help(plot)` to discover how many options are available.

External packages can be used for plotting. One of the most used is **ggplot2**, see <http://ggplot2.org/>.

`plot` can be used to represent a pair of data points `x` and `y`:

```
vX = seq(-10, 10, 0.1)
vY = sin(vX)
```

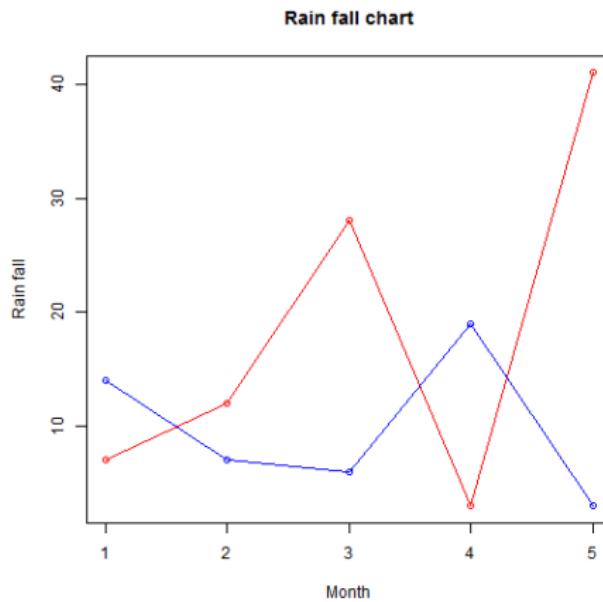
```
plot(vX, vY)
```

of a single vector of points:

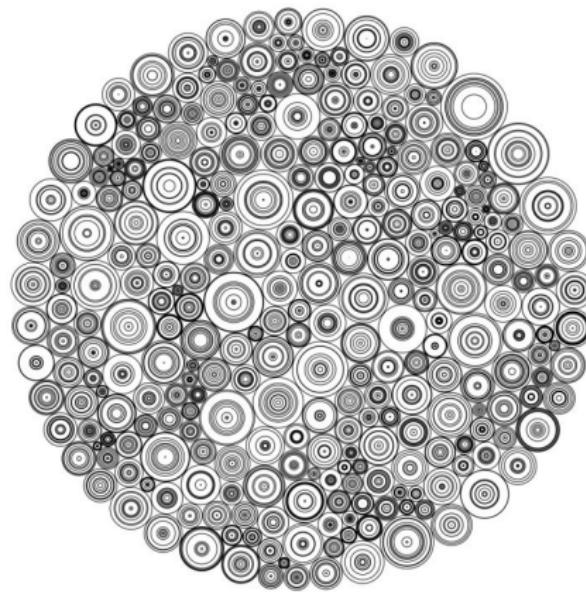
```
vZ = runif(10)
plot(vZ)
```

in this second case the `x`-axis is simply replaced by `1:length(vZ)`.

Plotting: ugly plot



Plotting: nice plot



see <https://www.r-graph-gallery.com/> for more nice plots!

Basic arguments for plot

- type, the type of plot: `type = "p"` for points, `type = "l"` for lines... see `help(plot)`.
- main, title of the plot.
- xlab, ylab: labels for the x- and y-axes, respectively.
- col, colour e.g. `col = "red"`.
- lty, if `type = "l"` determines the line type, e.g., 0=blank, 1=solid (default), 2=dashed, 3=dotted...
- lwd, if `type = "l"` determines line width, e.g., `lwd = 1` normal width, `lwd = 2` more width etc..

To add points `(x[1], y[1])`, `(x[2], y[2])`, ... to the current plot, use `points(x, y)`. To add lines instead use `lines(x, y)`. To add text use `text(x, y, "text")`

Vertical or horizontal lines can be drawn using `abline(v = xpos)` and `abline(h = ypos)`

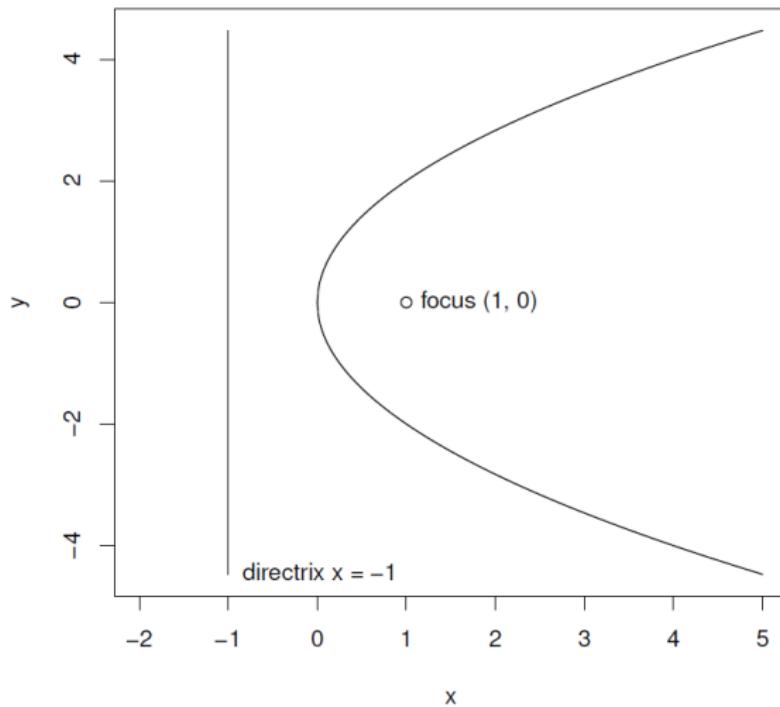
The functions `points`, `lines`, `abline` also accept the `col`, `lty`, `lwd` arguments.

Plot of a parabola

```
x <- seq(0, 5, by = 0.01)
y.upper <- 2*sqrt(x)
y.lower <- -2*sqrt(x)
y.max <- max(y.upper)
y.min <- min(y.lower)
plot(c(-2, 5), c(y.min, y.max), type = "n", xlab = "x", ylab = "y")
lines(x, y.upper)
lines(x, y.lower)
abline(v=-1)
points(1, 0)
text(1, 0, "focus (1, 0)", pos=4)
text(-1, y.min, "directrix x = -1", pos = 4)
title("The parabola  $y^2 = 4*x$ ")
```

Plot of a parabola

The parabola $y^2 = 4*x$



PROGRAMMING WITH FUNCTIONS

Leopoldo Catania

Aarhus University and CReATES

`leopoldo.catania@econ.au.dk`

John Chambers

"To understand computations in R, two slogans are helpful:

- *Everything that exists is an object.*
- *Everything that happens is a function call"*

– John Chambers

Outline

- Definitions of functions in R
- Function's variables
- Arguments of a function
- Ellipsis
- Vector-based programming using functions
- The `apply` family
- Recursive programming

Intro

In this lecture we cover the creation of functions, the rules that they must follow, and how they relate to and communicate with the environments from which they are called.

User-defined functions are now one of the main building blocks for developing sophisticated software. R packages are nothing more than a collection of user-defined functions.

Recall that all R packages available on CRAN are freely downloadable, i.e., you can use/extend/include their code for your analysis.

Functions

A function has the general form:

```
name <- function(argument_1, argument_2, ...) {  
  expression_1  
  expression_2  
<some other expressions>  
  return(output)  
}
```

- `argument_1`, `argument_2`, etc., are the names of variables.
- `expression_1`, `expression_2`, and `output` are all regular R expressions.
- `name` is the name of the function.

Note that some functions have no arguments: e.g.: `getwd()`.

Example: roots of a quadratic function

Assume we want to find the roots of:

$$a_2x^2 + a_1x + a_0 = 0.$$

The analytic solution is:

$$(x_1, x_2) = \frac{-a_1 \pm \sqrt{a_2^2 - 4a_2a_0}}{2a_2}.$$

The associated code is:

```
rootfinder <- function(dA0, dA1, dA2) {
  vOut = numeric(2)
  vOut[1] = (-dA1 + sqrt(dA2^2 - 4 * dA2 * dA0))/(2 * dA2)
  vOut[2] = (-dA1 - sqrt(dA2^2 - 4 * dA2 * dA0))/(2 * dA2)
  return(vOut)
}
```

Do you think this to be a good strategy?

Example: roots of a quadratic function

Problems related to the efficiency of the code:

- We are computing algebraic operations too many times.
- We are not exploiting vector-operations

Problems related to the user usage. We are assuming that a_2 , a_1 and a_0 always imply the coded solution.

- What if $a_2 = a_1 = a_0 = 0$?
- What if $a_2 = a_1 = 0$?
- What if $a_2 = 0$?
- What if $a_2^2 - 4a_2a_0 = 0$?
- What if $a_2^2 - 4a_2a_0 < 0$?

We need to modify our code!

Example: roots of a quadratic function

Reformulate the analytical solution as:

$$(x_1, x_2) = \begin{cases} \mathbb{R}, & \text{if } a_2 = a_1 = a_0 = 0 \\ \emptyset, & \text{if } a_2 = a_1 = 0 \wedge a_0 \neq 0 \\ -a_0/a_1 & \text{if } a_2 = 0 \wedge a_1 \neq 0 \wedge a_0 \neq 0 \\ \frac{-a_1 \pm \sqrt{|\Delta|} \sqrt{\operatorname{sgn}(\Delta)}}{2a_2}, & \text{otherwise.} \end{cases}$$

where $\Delta = a_2^2 - 4a_2a_0$.

Note that:

$$\sqrt{\operatorname{sgn}(\Delta)} = \begin{cases} 1, & \text{if } \Delta \geq 0 \\ i, & \text{if } \Delta < 0. \end{cases}$$

```
rootfinder <- function(dA0, dA1, dA2) {  
  if (dA2 == 0 && dA1 == 0 && dA0 == 0) {  
    vRoots <- Inf  
  } else if (dA2 == 0 && dA1 == 0) {  
    vRoots <- NULL  
  } else if (dA2 == 0) {  
    vRoots <- -dA0/dA1  
  } else {  
    # calculate the discriminant  
    dDelta <- dA1^2 - 4 * dA2 * dA0  
    if (dDelta > 0) {  
      vRoots <- (-dA1 + c(1, -1) * sqrt(dDelta))/(2*dA2)  
    } else if (dDelta == 0) {  
      vRoots <- rep(-dA1 / (2 * dA2), 2)  
    } else {  
      di <- complex(1, 0, 1)  
      vRoots <- (-dA1 + c(1,-1) * sqrt(-dDelta) * di)/(2 * dA2)  
    }  
  }  
  return(vRoots)  
}
```

Example: roots of a quadratic function

Suppose we have saved the function `rootfinder` in the script `rootfinder.R` which is located in the `script` folder inside our working directory.

To use the function we first load it (using `source` or by copying and pasting into R), then call it, supplying suitable arguments.

```
> source("../scripts/rootfinder.r")
> rootfinder(1, 1, 0)
[1] -1 -1
>
> rootfinder(1, 0, -1)
[1] -1  1
>
> rootfinder(1, -2, 1)
[1] 1
>
> rootfinder(1, 1, 1)
[1] -0.5+0.8660254i -0.5-0.8660254i
```

Advantages of coding with functions

Once a function it is loaded, it can be used again and again without having to reload it.

User-defined functions can be used in the same way as predefined functions are used in R. In particular they can be used within other functions.

The use of functions allows you to break down a programming task into smaller logical units.

Large programs are typically made up of a number of smaller functions, each of which does a simple well-defined task.

The mechanic of a function in R

When a function is executed the computer performs a series of operations:

- Set aside space in memory for the function.
- Make a copy of the function code.
- Transfer controls to the function.
- Run the function.
- Pass the output of the function back to the main program.
- Delete the copy of the function and all its variables.

Variables inside a function

Last point implies that arguments and variables that are defined within a function exist only within that function.

If you define and use a variable `x` inside a function, it does not exist outside the function.

```
> test <- function(x) {  
+ y <- x + 1  
+ return(y)  
+ }  
> test(1)  
[1] 2  
  
> x  
Error: Object "x" not found  
> y  
Error: Object "y" not found
```

The scope of a variable

If variables with the same name exist inside and outside a function, then they are separate and do not interact at all.

You can think of a function as a separate environment that communicates with the outside world only through the values of its arguments and its output expression.

That part of a program in which a variable is defined is called its scope.
Restricting the scope of variables within a function provides an assurance that calling the function will not modify variables outside the function, except by assigning the returned value.

Beware, however, the scope of a variable is not symmetric!

Variables defined inside a function cannot be seen outside, but variables defined outside the function can be seen inside the function, **provided there is not a variable with the same name defined inside.**

The scope of a variable

Consider for example:

```
> test2 <- function(x) {  
+ y <- x + z  
+ return(y)  
+ }  
> z <- 1  
> test2(1)  
[1] 2  
> z <- 2  
> test2(1)  
[1] 3
```

Best practice is to always pass variables to be used inside the function as additional arguments:

```
> test2 <- function(x, z) {  
+ y <- x + z  
+ return(y)  
+ }
```

Arguments of a function

The argument of a function can be mandatory or optional depending on the function specification.

The arguments of an existing function can be obtained by calling the `formals` function.

```
> formals(matrix)
$data
[1] NA
$nrow
[1] 1
$ncol
[1] 1
$byrow
[1] FALSE
$dimnames
NULL
```

Default arguments

In order to simplify calling functions, some arguments may be assigned default values.

Default values are used in case the argument is not provided in the call to the function:

```
> test3 <- function(x = 1) {  
+   return(x)  
+ }  
> test3(2)  
[1] 2  
> test3()  
[1] 1
```

Default arguments: partial matching

Sometimes you will want to define arguments so that they can take only a small number of different values, and the function will stop informatively if an inappropriate value is passed.

When writing the function, we include a vector of the permissible values for any such argument, and then check them using the `match.arg` function. For example:

```
> funk <- function(vibe = c("Do", "Be", "Dooby", "Dooo")) {  
+   vibe <- match.arg(vibe)  
+   return(vibe)  
+ }  
  
> funk("Dooby")  
[1] "Dooby"  
  
> funk("Dum")  
Error in match.arg(vibe) (from #2) :  
'arg' should be one of "Do", "Be", "Dooby", "Dooo"
```

Ellipsis . . .

R provides a very useful means of passing arguments, unaltered, from the function that is being called to the functions that are called within it.

These arguments do not need to be named explicitly in the outer function, hence providing great flexibility.

To use this facility you need to include . . . in your argument list.

These three dots (an ellipsis) act as a placeholder for any extra arguments given to the function.

Ellipsis ...

Consider for example the function 'square of the mean':

```
> SquareMean <- function(vX, ...) {  
+   dSM <- mean(vX, ...) ^ 2  
+   return(dSM)  
+ }  
>  
> SquareMean(c(1, 3, 5, NA))  
[1] NA  
>  
> SquareMean(c(1, 3, 5, NA), na.rm = TRUE)  
[1] 9
```

The use of ellipsis dramatically increases the flexibility of the R programming language.

Vector-based programming using functions

R provides a family of powerful and flexible functions that make it easier for user-defined functions to handle vector inputs these belong to the `apply` family of functions.

Here we cover: `apply`, `sapply`, `lapply`, and `mapply`.

Other functions belonging to this family are: `tapply`, `vapply`, and `eapply`, which require a bit of advanced R programming knowledge.

sapply

The effect of `sapply(vX, FUN)` is to apply the function `FUN` to every element of vector `vX`. Consider the function, `f`, "sum of all integers lower than X":

```
f <- function(dX) {  
  if (dX < 0) {  
    stop("dX need to be positive.")  
  }  
  dSum = 0.0  
  iC   = 0  
  while (iC <= dX) {  
    dSum = dSum + iC  
    iC = iC + 1  
  }  
  return(dSum)  
}
```

sapply and lapply

Suppose we want to apply the function `f` to the following vector of observations

`vX = c(1, 4, 9.478, 6, 75, 0.48)`. We can of course write a `for` loop:

```
> vX    = c(1, 4, 9.478, 6, 75, 0.48)
> vSum = numeric(length(vX))
> for (i in 1:length(vSum)) {
+   vSum[i] = f(vX[i])
+ }
> vSum
[1]    1   10   45   21 2850     0
```

Or we can use the `sapply` function:

```
> vSum = sapply(vX, f)
> vSum
[1]    1   10   45   21 2850     0
```

`lapply` does the same of `sapply` but always returns a list.

mapply

`mapply` does the same of `sapply` but allows you to iterate over multiple arguments. Consider for example the function `rep` which takes arguments: `x` and `times`. `mapply` allows you to write:

```
> mapply(rep, times = 1:4, x = 4:1)
```

```
[[1]]
```

```
[1] 4
```

```
[[2]]
```

```
[1] 3 3
```

```
[[3]]
```

```
[1] 2 2 2
```

```
[[4]]
```

```
[1] 1 1 1 1
```

apply

apply allows you to evaluate a function, FUN, over the margins of an array. Its formula is:

```
apply(X, MARGIN, FUN, ...)
```

where:

- X is an array
- MARGIN is the index/indices of the array to which apply FUN
- FUN is the function to apply
- ... are extra arguments for FUN

Note that MARGIN can be a vector of indices. If X is a matrix (an array of 2 dimensions), then MARGIN = 1 indicates rows, while MARGIN = 2 indicates columns.

apply example

To compute the cumulative sum over the columns of a matrix `mX` we write:

```
> mX = matrix(1:16, 4, 4)
> mX
      [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
>
> apply(mX, 2, cumsum) # cumulative sum over columns
      [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    3   11   19   27
[3,]    6   18   30   42
[4,]   10   26   42   58
```

Recursive programming

Many algorithms are recursive in nature. Consider for example the evaluation of $!n$, i.e., the factorial of the nonnegative integer n .

Obviously $!n = n(n - 1)!$, such that we can write a function like:

```
> myfactorial <- function(iN) {  
+   if ((iN == 0) | (iN == 1)) {  
+     return(1)  
+   } else {  
+     return(iN * myfactorial(iN - 1))  
+   }  
+ }
```

```
> myfactorial(5)  
[1] 120
```

```
> factorial(5)  
[1] 120
```

Multiple Outputs

- A function can generate multiple output values
- Example: parameter estimates, their standard errors, and the optimized log-likelihood value
- A strategy in R is to create a list with all your outputs and return it as a single output of a function

For example:

```
l0ut = list(Par = vParam,  
           SE  = vStdErr,  
           LLK = dLogLik)  
  
return(l0ut)
```

Comments

- Remember, you often want to re-use a code
- It is often hard to read code that someone else has written, or code that you have written yourself months ago. Using sensible function and variable names, and Hungarian notation helps... somewhat
- Solution: **comment** your code!
- In R, # signals a comment to the end of the line:
`vBeta = c(0, 1, -5) # Initial values`

Comments

- It is best practice to declare what a script does at its begin.
- Furthermore, at the start of *every* function, put something like

```
##  
##  FunctionName(Inputs)  
##  
##  Purpose:  
##    Description of what the function does  
##  
##  Input:  
##    List of inputs, describing what they represent  
##  
##  Output:  
##    List of outputs, describing what they represent  
##  
##  Return value:  
##    List of return values, describing what they represent  
##
```

In Rstudio to comment a selected portion of code we have the Ctrl + alt + C shortcut.

Comments

- In addition, add a general description of the program at the top of the .R file.
This should include your name and the date, for future reference
- May seem like a lot of work... but it will pay off in the end
- So: comment a lot!

MAXIMUM LIKELIHOOD ESTIMATION

Leopoldo Catania

Aarhus University and CREATES

leopoldo.catania@econ.au.dk

Program for today

Today we (briefly) review the maximum likelihood estimator and see its implementation in R

- Writing a log-likelihood function
- Maximizing the log-likelihood function
- Asymptotic covariance matrix of parameters
- Standard errors
- Hypothesis Testing with the LR test

The principle of maximum likelihood

The method of maximum likelihood (ML) is a fundamental and very general estimation method, that encompasses other estimation methods, such as least squares.

Maximum likelihood estimates of parameter vector, θ , are easily obtained under few hypothesis:

- Models are *parametric*, in the sense the probability distribution can be described by a finite number of parameters, θ ;
- The sample $\mathbf{X} = \{X_1, \dots, X_T\}$ is drawn from a family of probability distributions parametrized by an unknown parameter vector θ ;
- This probability distribution is known;
- Data drawn from this probability are *iid*.

The principle of maximum likelihood

The likelihood estimation is obtained, given the specification of the model, by the joint density associated with our dataset $f(\mathbf{X}; \theta)$. Given θ fixed, then $f(\mathbf{X}; \cdot)$ is the density of \mathbf{X} . Then, given that X_t is *iid*, we can compute the joint density of the entire sample as

$$f(\mathbf{X}; \theta) = \prod_{t=1}^T f(X_t; \theta) \quad (1)$$

The likelihood function is then defined as the joint density function, $f(\mathbf{X}; \theta)$, given \mathbf{X} fixed, as a function of θ , i.e. $L(\theta; \mathbf{X}) = f(\mathbf{X}; \theta)$. Note that $L(\theta; \mathbf{X})$ is a function of θ , while \mathbf{X} is fixed. We usually work with the log of the likelihood function:

$$\log L(\theta; \mathbf{X}) = \sum_{t=1}^T \log f(X_t; \theta) \quad (2)$$

The parameter vector, θ is identified if for any $\theta_1 \neq \theta$, $L(\theta; \mathbf{X}) \neq L(\theta_1; \mathbf{X})$.

The principle of maximum likelihood

- The principle of maximum likelihood means of choosing an asymptotically efficient estimator for a set of parameters, θ , by maximizing the likelihood function with respect to this set of parameters.
- The estimation entails the calculation of the first and second derivatives of the likelihood function with respect to the parameter vector.
- In some cases, derivatives are in closed form, but, in general, the derivatives must be computed numerically.
- The necessary first-order condition for maximizing $\log L(\theta; \mathbf{X})$ is

$$\frac{\partial \log L(\theta; \mathbf{X})}{\partial \theta} \Big|_{\theta=\hat{\theta}} = 0 \quad (3)$$

Writing and maximizing a likelihood function in R

In R the steps are:

- Write a function that computes the negative of log-likelihood function
- The inputs of this function are:
 - ① A parameter vector, θ ;
 - ② The data, X ;
 - ③ Additional optional inputs
- The output is the negative of the sum of the log-likelihood.
- Set initial values for θ .
- Maximize the log-likelihood function using the optimization routines of R such as `optim()`.

Example: Exponential i.i.d. observations

Suppose that we have a sample, $\{x_t\}_{t=1,\dots,T}$, of i.i.d. observations extracted from an exponential distribution. We denote $X_t \stackrel{iid}{\sim} Exp(\lambda)$ with

$$f_X(x_t, \lambda) = \lambda e^{-\lambda x_t}, \quad (4)$$

where $x_t \geq 0$.

- We want to estimate the coefficient λ by ML using the information coming from the i.i.d. sample, $\{x_t\}_{t=1,\dots,T}$.
- Let's write an R code for this!

ML in the case of dependent observations

In financial time-series observations are rarely *iid* that is:

$$f(X_t, X_{t-s}) \neq f(X_t)f(X_{t-s}). \quad (5)$$

for $s = 1, 2, \dots$

However, recall that every joint distribution can be decomposed in the product of the conditional and marginal distribution:

$$f(X_t, X_{t-s}) = f(X_t|X_{t-s})f(X_{t-s}). \quad (6)$$

ML in the case of dependent observations

Suppose to have a time-series of T observations: X_1, \dots, X_T . Its joint distribution can be factorized as:

$$f(X_1, \dots, X_T) = f(X_1) \prod_{t=2}^T f(X_t | X_1, \dots, X_{t-1})$$

In general we identify with \mathcal{F}_t with all the information contained in the observations up to time t , that is $\mathcal{F}_t = \{X_1, \dots, X_t\}$, with $\mathcal{F}_0 = \{\emptyset\}$. In this way we can write:

$$f(X_1, \dots, X_T) = \prod_{t=1}^T f(X_t | \mathcal{F}_{t-1}).$$

In financial econometrics we usually make a parametric assumption on $f(X_t | \mathcal{F}_{t-1})$ and $f(X_1)$ in order to estimate models.

Under the assumption of correct model specification, identification of the parameters, continuity and finiteness of the first three derivatives of $\log(f(\theta, X_t))$ wrt θ , the ML estimator has the following properties:

- ① Consistency, $\hat{\theta} \xrightarrow{P} \theta_0$.
- ② Asymptotic Normality:

$$\sqrt{T}(\hat{\theta} - \theta_0) \xrightarrow{d} N(0, \Sigma(\theta_0))$$

- ③ $\Sigma(\theta_0) = I(\theta_0)^{-1}$, with

$$I(\theta_0) = -E_{\theta_0} \left[\frac{\partial^2 L(\theta; X)}{\partial \theta \partial \theta'} \Big|_{\theta=\theta_0} \right]$$

which is called Fisher information matrix. Detailed derivation in Newey and McFadden (1994), M-estimators.

- ④ ML estimator is the most efficient unbiased estimator, as the inverse of the Fisher information matrix is the lower bound on the variance of any estimator of θ (Cramer-Rao bound).
- ⑤ Invariance: The maximum likelihood estimator of $\gamma_0 = c(\theta_0)$ is $c(\hat{\theta})$ if $c(\cdot)$ is a continuous and continuously differentiable function. (By an application of the continuous mapping theorem)

Asymptotic Variance I

Given the previous results, the asymptotic covariance matrix of the MLE can be computed as

$$\Sigma(\hat{\theta}) = -H(\hat{\theta})^{-1}$$
$$H(\hat{\theta}) = \left. \frac{\partial^2 \log L(\theta; X)}{\partial \theta \partial \theta'} \right|_{\theta=\hat{\theta}}$$

where $H(\hat{\theta})$ is evaluated numerically in $\hat{\theta}$.

Alternative Estimators of the asymptotic covariance matrix

Other estimators of the asymptotic covariance matrix are

- BHHH (Berndt-Hall-Hall-Hausman) or gradient outer-product:

$$\Sigma(\hat{\theta}) = \left[\sum_{i=1}^N \hat{g}_i \hat{g}_i' \right]^{-1}$$

where \hat{g}_i is the gradient for the i -th observation computed in $\hat{\theta}$.

- Sandwich estimator (QML estimator):

$$\Sigma(\hat{\theta}) = [-H(\hat{\theta})^{-1}] \left(\sum_{i=1}^N \hat{g}_i \hat{g}_i' \right) [-H(\hat{\theta})^{-1}]$$

Delta Method

Delta method is an approximate method to derive the standard errors of functions of the parameter estimates. Given that:

$$\frac{\hat{\theta} - \theta}{S.E.(\theta)} \rightarrow N(0, 1) \quad (7)$$

then

$$\frac{f(\hat{\theta}) - f(\theta)}{f'(\theta)S.E.(\theta)} \rightarrow N(0, 1) \quad (8)$$

The multivariate version implies an asymptotic covariance matrix of the parameters, Λ ,

$$\Lambda = J(\theta)' \Sigma(\theta) J(\theta) \quad (9)$$

where $J(\theta)'$ is the $p \times p$ matrix with the first derivatives, $f'(\theta)$ with respect to the p parameters in θ , and Σ is the asymptotic covariance matrix of θ .

Hypothesis testing in the ML context

We now focus on one of the most common testing procedure related to ML estimation: the Likelihood-ratio (LR) test.

The LR test is based on the idea that under the null hypothesis values of the log-likelihood of the unrestricted model, $\log L_U$, and the model under \mathcal{H}_0 , $\log L_R$, must be close. The test takes the form

$$LR = 2 \cdot (\log L_U - \log L_R)$$

As $N \rightarrow \infty$,

$$LR \rightarrow \chi^2(p)$$

where p is the number of restrictions imposed.

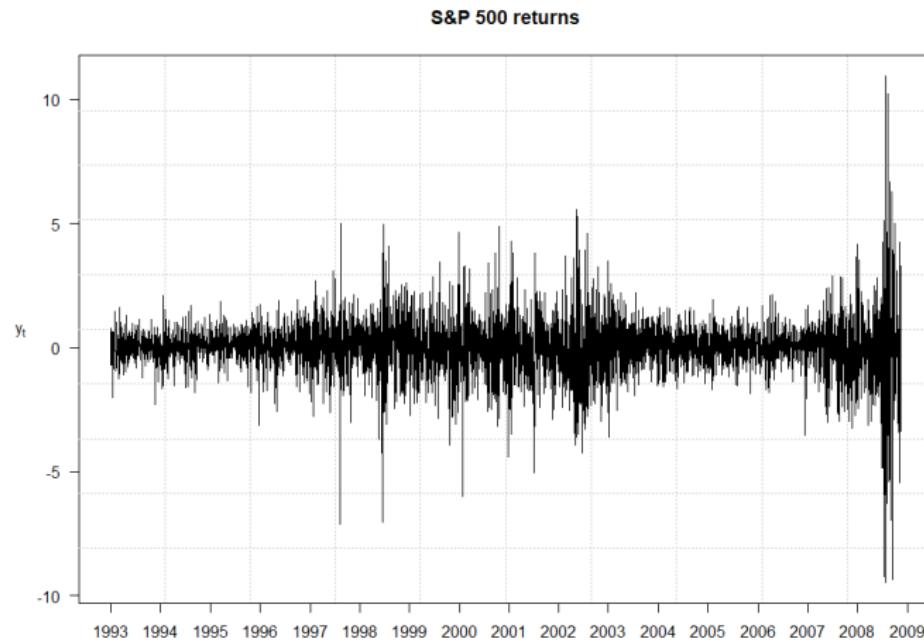
UNIVARIATE VOLATILITY MODELING

Leopoldo Catania

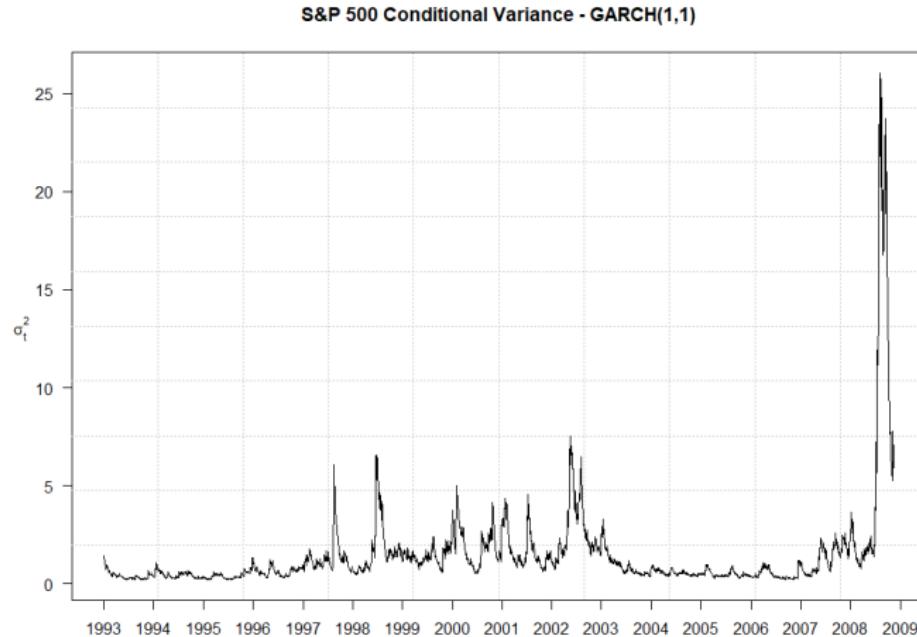
Aarhus University and CReATES

leopoldo.catania@econ.au.dk

S&P500 Returns



S&P 500 Conditional Variance - GARCH(1,1)



Engle (1982)

Econometrica, Vol. 50, No. 4 (July, 1982)

AUTOREGRESSIVE CONDITIONAL HETEROSCEDASTICITY WITH ESTIMATES OF THE VARIANCE OF UNITED KINGDOM INFLATION¹

BY ROBERT F. ENGLE

Traditional econometric models assume a constant one-period forecast variance. To generalize this implausible assumption, a new class of stochastic processes called autoregressive conditional heteroscedastic (ARCH) processes are introduced in this paper. These are mean zero, serially uncorrelated processes with nonconstant variances conditional on the past, but constant unconditional variances. For such processes, the recent past gives information about the one-period forecast variance.

A regression model is then introduced with disturbances following an ARCH process. Maximum likelihood estimators are described and a simple scoring iteration formulated. Ordinary least squares maintains its optimality properties in this set-up, but maximum likelihood is more efficient. The relative efficiency is calculated and can be infinite. To test whether the disturbances follow an ARCH process, the Lagrange multiplier procedure is employed. The test is based simply on the autocorrelation of the squared OLS residuals.

This model is used to estimate the means and variances of inflation in the U.K. The ARCH effect is found to be significant and the estimated variances increase substantially during the chaotic seventies.

GARCH models: a class of models

- The family of ARCH-GARCH models has been introduced by Engle (1982) and Bollerslev (1986) to capture the features of the financial log-returns.
- The idea is to assume that the variance at time t can be represented as a function of past observations, $\mathcal{F}_{t-1} = \{y_{t-s}, s = 1, 2, \dots\}$.
- The original ARCH(1) model of Engle (1982) assumes that:

$$\begin{aligned} r_t &= \mu_t + \varepsilon_t, \quad \varepsilon_t = \sigma_t z_t \\ \sigma_t^2 &= \omega + \alpha \varepsilon_{t-1}^2 \end{aligned}$$

where $\mathbb{E}[z_t | \mathcal{F}_{t-1}] = 0$ and $\mathbb{E}[z_t^2 | \mathcal{F}_{t-1}] = 1$ and ε_{t-1}^2 is called the 'forcing variable' of the volatility process.

- An ARCH(q) model can be easily specified as $\sigma_t^2 = \omega + \sum_{j=1}^q \alpha_j \varepsilon_{t-j}^2$

ARCH model

Assume that $\mu_t = \mu = 0$ such that $r_t = \varepsilon_t = \sigma_t z_t$. The unconditional variance of an ARCH(q) model is:

$$\begin{aligned}\mathbb{E}[r_t^2] &= \mathbb{E}[\sigma_t^2 z_t^2] \\ &= \mathbb{E}[\sigma_t^2] = \frac{\omega}{1 - \sum_{j=1}^q \alpha_j}\end{aligned}$$

In the simpler ARCH(1) case, assuming gaussianity for z_t , the forth moment is

$$\begin{aligned}\mathbb{E}[r_t^4] &= \mathbb{E}[\sigma_t^4 z_t^4] \\ &= \mathbb{E}[\sigma_t^4] \mathbb{E}[z_t^4] = 3 \frac{\omega^2 (1 + \alpha)}{(1 - \alpha)(1 - 3\alpha^2)}\end{aligned}$$

such that the kurtosis coefficient is

$$\kappa = 3 \frac{1 - \alpha^2}{1 - 3\alpha^2} > 3$$



AR(p) representation

Let $v_t = r_t^2 - \sigma_t^2$, such that $\sigma_t^2 = r_t^2 - v_t$. We can write

$$\sigma_t^2 = \omega + \sum_{j=1}^q \alpha_j r_{t-j}^2$$

$$r_t^2 - v_t = \omega + \sum_{j=1}^q \alpha_j r_{t-j}^2$$

$$r_t^2 = \omega + \sum_{j=1}^q \alpha_j r_{t-j}^2 + v_t,$$

where we know that $\mathbb{E}[v_t] = 0$ for all t and $\mathbb{E}[v_t^2] = 2\mathbb{E}[\sigma_t^4]$, i.e. $\{v_t\}$ is a martingale difference sequence with constant variance. So the ARCH(p) model has an AR(p) representation for r_t^2 .

GARCH models: a class of models

- Bollerslev (1986) has generalized the ARCH model to GARCH as follow:

$$\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2.$$

- Note that now the volatility also linearly depends from its past value.
- GARCH(p, q) is constructed as follow:

$$\sigma_t^2 = \omega + \sum_{j=1}^q \alpha_j r_{t-j}^2 + \sum_{i=1}^p \beta_i \sigma_{t-i}^2,$$

Properties of the model: stationarity

The unconditional variance of r_t is

$$\text{Var}(r_t) = E(\sigma_t^2) = \omega + \alpha E(r_{t-1}^2) + \beta E(\sigma_{t-1}^2)$$

under the assumption of weak stationarity (that we will check later) we have $E(\sigma_{t-1}^2) = E(r_{t-1}^2) = E(\sigma_t^2)$ and we note that $\text{Var}(r_t)$ has a parametric expression

$$\text{Var}(r_t) = \frac{\omega}{1 - \alpha - \beta}$$

Condition for weak stationarity is $\alpha + \beta < 1$. Conditions for positivity are $\omega > 0$, $\alpha > 0$ and $\beta > 0$, see Nelson and Cao (1992).

Properties of the model: Excess unconditional kurtosis

From the previous result, it follows that the second moment of returns ($\mu = 0$) is

$$E(r_t^2) = \frac{\omega}{1 - \alpha - \beta}$$

The fourth moment is

$$\begin{aligned} E(r_t^4) &= E(z_t^4)E(\sigma_t^4) \\ &= 3\omega^2(1 + \alpha + \beta)[(1 - \alpha - \beta)(1 - \beta^2 - 2\alpha\beta - 3\alpha^2)]^{-1} \end{aligned}$$

Such that the kurtosis

$$\begin{aligned} K(r_t) &= \frac{E(r_t^4)}{E(r_t^2)^2} \\ &= \frac{3[(1 + \alpha + \beta)(1 - \alpha - \beta)]}{1 - \beta^2 - 2\alpha\beta - 3\alpha^2} > 3 \end{aligned}$$

which is larger than 3.

ARMA representation

Consider the GARCH(1,1) model. Let $v_t = r_t^2 - \sigma_t^2$, such that $\sigma_t^2 = r_t^2 - v_t$. We can write

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2 \\ r_t^2 - v_t &= \omega + \alpha r_{t-1}^2 + \beta(r_{t-1}^2 - v_{t-1}) \\ r_t^2 &= \omega + (\alpha + \beta)r_{t-1}^2 - \beta v_{t-1} + v_t,\end{aligned}$$

where we know that $\mathbb{E}[v_t] = 0$ for all t and $\mathbb{E}[v_t^2] = const$, i.e. $\{v_t\}$ is a martingale difference sequence with constant variance. So the GARCH(1, 1) model has an ARMA(1, 1) representation for r_t^2 .

It can be shown that GARCH(p,q) has an ARMA(max(p,q),p) representation for r_t^2 .

ARCH(∞) representation

The GARCH(1,1) model has an ARCH(∞) representation. To see this write:

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2 \\ \sigma_t^2 (1 - \beta L) &= \omega + \alpha r_{t-1}^2 \\ \sigma_t^2 &= \frac{\omega}{1 - \beta} + \alpha \sum_{s=0}^{\infty} \beta^s r_{t-s-1}^2 \\ \sigma_t^2 &= \omega^* + \sum_{s=0}^{\infty} \phi_s r_{t-s-1}^2,\end{aligned}$$

where $\omega^* = \frac{\omega}{1-\beta}$ and $\phi_s = \alpha\beta^s$.

Stationarity of GARCH(1,1)

Assume that r_t follows the GARCH(1,1) process:

$$\begin{aligned} r_t &= \sigma_t z_t \\ \sigma_t^2 &= \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2, \end{aligned}$$

We have that $\{r_t\}$ is a Martingale difference sequence: $\mathbb{E}[r_t | \mathcal{F}_{t-1}] = 0$ for all t , where \mathcal{F}_{t-1} represents the whole history of the process up to time $t - 1$.

Since $\mathbb{E}[r_t | \mathcal{F}_{t-1}] = 0$ for all t , we also have that $\mathbb{E}[r_t] = 0$ for all t , such that $\text{Var}(r_t) = \mathbb{E}[r_t^2] = \mathbb{E}[\sigma_t^2]$. To have weak stationarity we need $\mathbb{E}[\sigma_t^2] = \sigma^2 < \infty$ for all t . We have that

$$\mathbb{E}[\sigma_t^2] = \frac{\omega}{1 - \alpha - \beta},$$

such that $\alpha + \beta < 1$ is required for weak stationarity of r_t (proof later).

Stationarity of GARCH(1,1)

Many people believe that since GARCH(1,1) has an ARMA(1,1) representation for r_t^2 the condition $\alpha + \beta < 1$ is also necessary to have strong stationarity of r_t .

This is wrong!

The reason is that GARCH processes are thick tailed (we saw evidence for this looking at the kurtosis coefficient) such that the conditions for weak stationarity are often more stringent than the conditions for strict stationarity.

A GARCH(1,1) model can be written as:

$$\sigma_t^2 = \omega \left[1 + \sum_{k=1}^{\infty} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right]$$

Note that we wrote σ_t^2 only as a function of the z_t variables. Thus, we can now formally derive

$$\mathbb{E}[\sigma_t^2] = \omega \left[1 + \sum_{k=1}^{\infty} \mathbb{E} \left[\prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right] \right] = \frac{\omega}{1 - \alpha - \beta}$$

Stationarity of GARCH(1,1)

Indeed, by substituting r_{t-1}^2 with its definition $r_{t-1}^2 = \sigma_{t-1}^2 z_{t-1}^2$ we obtain:

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2 \\ &= \omega + \alpha \sigma_{t-1}^2 z_{t-1}^2 + \beta \sigma_{t-1}^2 \\ &= \omega + \sigma_{t-1}^2 (\alpha z_{t-1}^2 + \beta),\end{aligned}$$

which holds for all t , i.e. $\sigma_{t-s}^2 = \omega + \sigma_{t-s-1}^2 (\alpha z_{t-s-1}^2 + \beta)$. By repeated substitutions we obtain

$$\sigma_t^2 = \omega \left[1 + \sum_{k=1}^{\infty} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right].$$

Stationarity of GARCH(1,1)

Nelson (1990) shows that when $\omega > 0$ we have $\sigma_t^2 < \infty$ almost surely and the joint process $\{r_t, \sigma_t^2\}$ is strictly stationary if and only if $\mathbb{E}[\ln(\beta + \alpha z_t^2)] < 0$. Note that, by Jensen inequality we have

$$\mathbb{E}[\ln(\beta + \alpha z_t^2)] < \ln(\mathbb{E}[(\beta + \alpha z_t^2)]) = \ln(\alpha + \beta).$$

Exploiting this, we have the (at first glance counterintuitive) result that when $\alpha + \beta = 1$ the model is strictly stationary but not weakly stationary.

Indeed, the condition $\mathbb{E}[\ln(\beta + \alpha z_t^2)]$ is weaker than $\alpha + \beta < 1$.

As a byproduct we have that the ARCH(1) model with $\alpha = 1$ is also strictly stationary but not weekly stationary.

Stationarity of GARCH(1,1): about $\mathbb{E}[\ln(\beta + \alpha z_t^2)]$

So far we have assumed that the process goes back to the infinite past. In reality this is not true such that there is always a starting point $0 < \sigma_0^2 < \infty$. Let's define:

- σ_t^2 as the process initialized at time $t = 0$ at σ_0^2 , and
- $_u\sigma_t^2$ as the process that goes back to the infinite past (u for unconditional).

It is easy to show that

$$\begin{aligned}\sigma_t^2 &= \sigma_0^2 \prod_{i=1}^t (\beta + \alpha z_{t-i}^2) + \omega \left[1 + \sum_{k=1}^{t-1} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right] \\ _u\sigma_t^2 &= \omega \left[1 + \sum_{k=1}^{\infty} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right]\end{aligned}$$

Stationarity of GARCH(1,1): about $\mathbb{E}[\ln(\beta + \alpha z_t^2)]$

Note that the following theorems assume that $\mathbb{E}[\ln(\beta + \alpha z_t^2)]$ exists.

Theorem (1 of Nelson 1990)

If $\omega = 0$

- (a) ${}_u\sigma_t^2 = 0$ almost surely (a.s.) for all t .
- (b) $\sigma_t^2 \rightarrow \infty$ a.s. if and only if (iff) $\mathbb{E}[\ln(\beta + \alpha z_t^2)] > 0$.
- (c) $\sigma_t^2 \rightarrow 0$ a.s. iff $\mathbb{E}[\ln(\beta + \alpha z_t^2)] < 0$.
- (d) If $\mathbb{E}[\ln(\beta + \alpha z_t^2)] = 0$, $\log(\sigma_t^2)$ is a driftless random walk (RW) such that $\limsup \ln(\sigma_t^2) = +\infty$ and $\liminf \ln(\sigma_t^2) = -\infty$

Proof.

Point (a) is straightforward. Points (b)-(c) follows by computing the log and realizing that $\ln(\sigma_t^2)$ is a RW with drift $\mathbb{E}[\ln(\beta + \alpha z_t^2)]$ and by direct application of the strong law of large numbers for iid random variables. Point (d) is proven by an application of Corollary 6.1.1 and Theorem 6.1.4 of Stout (1974). □

Stationarity of GARCH(1,1): about $\mathbb{E}[\ln(\beta + \alpha z_t^2)]$

Theorem (2 of Nelson 1990)

If $\omega > 0$ and $\mathbb{E}[\ln(\beta + \alpha z_t^2)] \geq 0$

- (a) $\sigma_t^2 \rightarrow \infty$ a.s.
- (b) ${}_u\sigma_t^2 \rightarrow \infty$ a.s. for all t .

If $\omega > 0$ and $\mathbb{E}[\ln(\beta + \alpha z_t^2)] < 0$

- (c) $\frac{\omega}{1-\beta} \leq {}_u\sigma_t^2 < \infty$ for all t a.s.
- (d) ${}_u\sigma_t^2 - \sigma_t^2 \rightarrow 0$ a.s.

We need to be in this situation to ensure good results.

Proof.

To prove point (a) note that:

$$\begin{aligned}\sigma_t^2 &= \sigma_0^2 \prod_{i=1}^t (\beta + \alpha z_{t-1}^2) + \omega \left[1 + \sum_{k=1}^{t-1} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right] \\ &\geq \omega \sup_{1 \leq k \leq t-1} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2)\end{aligned}$$

such that

$$\log \sigma_t^2 \geq \log \omega + \sup_{1 \leq k \leq t-1} \sum_{i=1}^k \log(\beta + \alpha z_{t-i}^2) \rightarrow \infty,$$

because from Theorem 1 we have:

$$\sum_{i=1}^k \log(\beta + \alpha z_{t-i}^2) \rightarrow \infty \quad \text{iff} \quad \mathbb{E}[\ln(\beta + \alpha z_t^2)] > 0$$

$$\lim_{t \rightarrow \infty} \sup_{1 \leq k \leq t-1} \sum_{i=1}^k \log(\beta + \alpha z_{t-i}^2) = \infty \quad \text{iff} \quad \mathbb{E}[\ln(\beta + \alpha z_t^2)] = 0$$



Proof.

To prove point (b) we use similar arguments to:

$$\begin{aligned}_u\sigma_t^2 &= \omega \left[1 + \sum_{k=1}^{\infty} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right] \\ &\geq \omega \sup_{1 \leq k \leq \infty} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2)\end{aligned}$$



Proof.

To prove the lower bound of point (c) we note that:

$$_u\sigma_t^2 = \omega \left[1 + \sum_{k=1}^{\infty} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right]$$

and its infimum is achieved for $z_{t-1}^2 = 0$ for all t , such that

$$\inf _u\sigma_t^2 = \omega \left[1 + \sum_{k=1}^{\infty} \prod_{i=1}^k \beta \right] = \omega \left[1 + \sum_{k=1}^{\infty} \beta^k \right] = \omega \sum_{k=0}^{\infty} \beta^k = \frac{\omega}{1 - \beta},$$

since $0 < \beta < 1$. The proof of the upper bound is more intricate and relies on the fact that each element $\prod_{i=1}^k (\beta + \alpha z_{t-i}^2)$ is $O(\exp(-\lambda k))$ for some $\lambda > 0$. □

Proof.

To prove point (d) we note that

$$\sigma_t^2 - \sigma_t^2 = \sigma_0^2 \prod_{i=1}^t (\beta + \alpha z_{t-i}^2) - \omega \left[1 + \sum_{k=t}^{\infty} \prod_{i=1}^k (\beta + \alpha z_{t-i}^2) \right]$$

the first term goes to zero according to Theorem 1. It can be shown that the second term also goes to zero because each element $\prod_{i=1}^k (\beta + \alpha z_{t-i}^2)$ is $O(\exp(-\lambda k))$ for some $\lambda > 0$. □

Forecasting Volatility: GARCH(1,1)

Forecasting with a GARCH(1,1) (Engle and Bollerslev 1986).

For the GARCH(1,1) we have:

$$\sigma_t^2 = \omega + \alpha_1 r_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

and

$$\begin{aligned} E_t[\sigma_{t+k}^2] &= \sum_{i=0}^{k-2} [(\alpha_1 + \beta_1)^i \omega] + (\alpha_1 + \beta_1)^{k-1} \sigma_{t+1}^2 \\ &= \sigma^2 [1 - (\alpha_1 + \beta_1)^{k-1}] + (\alpha_1 + \beta_1)^{k-1} \sigma_{t+1}^2 \\ &= \sigma^2 + (\alpha_1 + \beta_1)^{k-1} [\sigma_{t+1}^2 - \sigma^2], \end{aligned}$$

where $\sigma^2 = \frac{\omega}{1-\alpha-\beta}$. When $\alpha_1 + \beta_1 < 1$, i.e. the process is stationary,
 $E_t[\sigma_{t+k}^2] \rightarrow \sigma^2$ as $k \rightarrow \infty$.

Forecasting Volatility

For the GARCH(p,q) we have:

$$\sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i r_{t-i}^2 + \sum_{i=1}^p \beta_i \sigma_{t-i}^2$$

we can write the process in two parts, before and after time t:

$$\sigma_{t+k}^2 = \omega + \sum_{i=1}^n [\alpha_i r_{t+k-i}^2 + \beta_i \sigma_{t+k-i}^2] + \sum_{i=k}^m [\alpha_i r_{t+k-i}^2 + \beta_i \sigma_{t+k-i}^2]$$

where $m = \max[p; q]$, and $n = \min[m; k - 1]$. It follows that

$$E_t[\sigma_{t+k}^2] = \omega + \sum_{i=1}^n [(\alpha_i + \beta_i) E_t(\sigma_{t+k-i}^2)] + \sum_{i=k}^m [\alpha_i r_{t+k-i}^2 + \beta_i \sigma_{t+k-i}^2]$$

Testing for ARCH effects

- A widely used test for ARCH effects is the Lagrange-Multiplier (LM) test of Engle (1982).
- Null hypothesis: $r_t | \mathcal{F}_{t-1} \sim N(0, \sigma^2)$
- Alternative hypothesis: $r_t \sim \text{ARCH}(q)$
- The LM test exploits the following auxiliary regression

$$r_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \dots + \alpha_q r_{t-q}^2 + \nu_t$$

and test that $\alpha_1 = \dots = \alpha_q = 0$ vs $\alpha_0 > 0, \alpha_1 > 0, \dots, \alpha_q > 0$.

- The test statistic is a $T \times R^2$ type (it is an OLS regression) and the asymptotic distribution is $\chi^2(q)$.
- One-sided test proposed by Demos and Sentana (1998).

Non-stationary volatility: the IGARCH

The IGARCH(1,1) is characterized by

$$\alpha_1 + \beta_1 = 1$$

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha_1 r_{t-1}^2 + (1 - \alpha_1)\sigma_{t-1}^2 \\ &= \omega + \sigma_{t-1}^2 + \alpha_1(r_{t-1}^2 - \sigma_{t-1}^2)\end{aligned}$$

In this case, the conditional variance k steps in the future is:

$$E_t[\sigma_{t+k}^2] = (k-1)\omega + \sigma_{t+1}^2 \quad (1)$$

Riskmetrics (1996): $\alpha_1 = 0.06$, $\beta_1 = 0.94$ and $\omega = 0$.

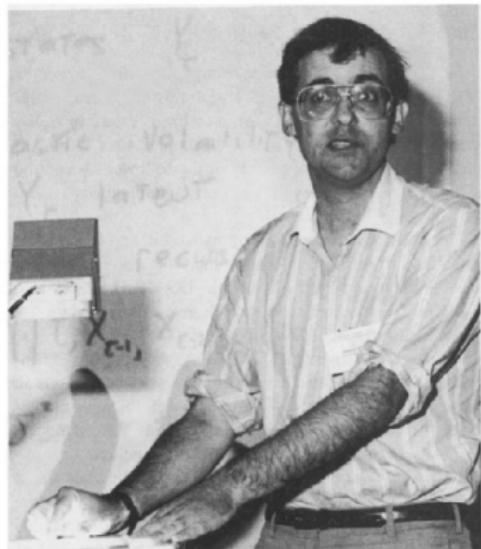
Stationarity of IGARCH(1,1)

IGARCH(1,1) with $\omega > 0$ is strictly stationary because $\mathbb{E}[\ln(1 - \alpha + \alpha z_t^2)] < 0$ (Remember the result for GARCH(1,1)).

We have the following behaviour of the IGARCH(1,1) process (Nelson, 1990):

- In the IGARCH(1,1) model with $\omega = 0$ (the "Riskmetrics approach"), σ_t^2 collapses to zero almost surely.
- In the IGARCH(1,1) model with $\omega > 0$, σ_t^2 is strictly stationary and thus does not behave like a random walk! (Remember that random walk diverge almost surely).

Nelson (1991)



Econometrica, Vol. 59, No. 2 (March, 1991), 347–370

CONDITIONAL HETROSKEDEASTICITY IN ASSET RETURNS: A NEW APPROACH

BY DANIEL B. NELSON¹

GARCH models have been applied in modelling the relation between conditional variance and asset risk premia. These models, however, have at least three major drawbacks in asset pricing applications: (i) Researchers beginning with Black (1976) have found a negative correlation between current returns and future returns volatility. GARCH models rule this out by assumption. (ii) GARCH models impose parameter restrictions that are often violated by estimated coefficients and that may unduly restrict the dynamics of the conditional variance process. (iii) Interpreting whether shocks to conditional variance “persist” or not is difficult in GARCH models, because the usual norms measuring persistence often do not agree. A new form of ARCH is proposed that meets these objections. The method is used to estimate a model of the risk premium on the CRSP Value-Weighted Market Index from 1962 to 1987.

E-GARCH

- There exists a negative correlation between stock returns and changes in returns volatility, i.e. volatility tends to rise in response to "bad news", (excess returns lower than expected) and to fall in response to "good news" (excess returns higher than expected).
- The GARCH models are not able to explain the observed covariance between r_t^2 and r_{t-j} .
- GARCH models essentially specify the behavior of the square of the data. In this case a few large observations can dominate the sample.

E-GARCH

In the EGARCH(p,q) model (Exponential GARCH(p,q)) put forward by Nelson the σ_{t-1}^2 process depends on both size and the sign of lagged residuals. This is the first example of asymmetric model:

$$\log(\sigma_t^2) = \omega + \sum_{i=1}^p \beta_i \log(\sigma_{t-i}^2) + \sum_{i=1}^q \alpha_i [\phi z_{t-i} + \psi(|z_{t-i}| - E|z_{t-i}|)].$$

Note that $\alpha_1 \equiv 1$, $E(|z_t|) = (2/\pi)^{1/2}$ when $z_t \sim NID(0, 1)$ and for any value of the parameters α_i , ω ; and β_i we have $\sigma_t > 0$. Let's define

$$g(z_t) \equiv \phi z_t + \psi [|z_t| - E |z_t|]$$

by construction $\{g(z_t)\}_{t=-\infty}^\infty$ is a zero-mean, i.i.d. random sequence.

The EGARCH(p,q) Model

- The components of $g(z_t)$ are ϕz_t and $\psi [|z_t| - E |z_t|]$, each with mean zero.
- If the distribution of z_t is symmetric, the components are orthogonal, but not independent.
- Over the range $0 < z_t < \infty$, $g(z_t)$ is linear in z_t with slope $\psi + \phi$, and over the range $-\infty < z_t \leq 0$, $g(z_t)$ is linear with slope $\psi - \phi$.
- The term $\psi [|z_t| - E |z_t|]$ represents a magnitude effect.

To see the different slope of $g(z_t)$ note that $z_t = sgn(z_t)|z_t|$, where sgn is the sign function¹ and

$$\begin{aligned} g(z_t) &= \phi z_t + \psi(|z_t| - E|z_t|) \\ &= \phi sgn(z_t)|z_t| + \psi|z_t| - \psi E|z_t| \\ &= |z_t|(\phi sgn(z_t) + \psi) - \psi E|z_t| \end{aligned}$$

¹ $sgn(x) = 1$ if $x > 0$ and $sgn(x) = -1$ if $x < 0$.

The EGARCH(p,q) Model

- If $\psi > 0$ and $\phi = 0$, the innovation in $\ln(\sigma_{t+1}^2)$ is positive (negative) when the magnitude of z_t is larger (smaller) than its expected value.
- If $\psi = 0$ and $\phi < 0$, the innovation in conditional variance is now positive (negative) when returns innovations are negative (positive).
- A negative shock to the returns which would increase the debt to equity ratio and therefore increase uncertainty of future returns could be accounted for when $\alpha_i > 0$ and $\phi < 0$.

The EGARCH(p,q) Model

Nelson assumes that z_t has a GED distribution (exponential power family). The density of a GED random variable normalized is:

$$f(z; v) = \frac{v \exp\left[-\left(\frac{1}{2}\right)|z/\lambda|^v\right]}{\lambda 2^{(1+1/v)} \Gamma(1/v)} \quad -\infty < z < \infty, 0 < v \leq \infty$$

The EGARCH(p,q) Model

where $\Gamma(\cdot)$ is the gamma function, and

$$\lambda \equiv \left[2^{(-2/v)} \Gamma(1/v) / \Gamma(3/v) \right]^{1/2}$$

v is a tail thickness parameter.

	z's distribution
$v = 2$	standard normal distribution
$v < 2$	thicker tails than the normal
$v = 1$	double exponential distribution
$v > 2$	thinner tails than the normal
$v = \infty$	uniformly distributed on $[-3^{1/2}, 3^{1/2}]$

With this density, $E|z_t| = \frac{\lambda 2^{1/v} \Gamma(2/v)}{\Gamma(1/v)}$.

More on $E|z_t|$

The term $E|z_t|$ depends on the distributional assumption of z_t . Some examples are

- If $z_t \sim N(0, 1)$ then $E|z_t| = \sqrt{\frac{2}{\pi}}$
- If $z_t \sim GED(0, 1, v)$ then $E|z_t| = \frac{\lambda 2^{1/v} \Gamma(2/v)}{\Gamma(1/v)}$
- If $z_t \sim \mathcal{T}(0, 1, v)$ then $E|z_t| = \frac{2\sqrt{v-2}\Gamma((v+1)/2)}{(v-1)\Gamma(v/2)\sqrt{\pi}}$

In general, if a closed form is not available, it can be evaluated by numerical integration

$$E|z_t| = \int_{-\infty}^{\infty} |z_t| p_z(z_t) dz_t$$

or by simulation

$$E|z_t| \approx \frac{1}{B} \sum_{i=1}^B |z_t^{(i)}|,$$

where $z_t^{(i)}$, $i = 1, \dots, B$ are B independent draws from the distribution of z_t .

Volatility prediction with EGARCH

Consider an EGARCH(1,1)

$$r_t = \sigma_t z_t$$

$$\log \sigma_t^2 = \omega + \beta \log \sigma_{t-1}^2 + \phi z_{t-1} + \psi(|z_{t-1}| - \mathbb{E}|z_{t-1}|)$$

As in GARCH, the one step ahead prediction ($h = 1$) is deterministic and given by

$$\begin{aligned}\sigma_{t+1}^2 &= \exp \left[\omega + \beta \log \sigma_t^2 + \phi z_t + \psi(|z_t| - \mathbb{E}|z_t|) \right] \\ &= \exp(\omega) \sigma_t^{2\beta} \exp(\phi z_t + \psi(|z_t| - \mathbb{E}|z_t|))\end{aligned}$$

Volatility prediction with EGARCH

The h step ahead prediction for the log-variance $\widehat{\log \sigma}_{t+h|t}^2 = \mathbb{E}[\log \sigma_{t+h}^2 | \mathcal{F}_t]$ is

$$\widehat{\log \sigma}_{t+h|t}^2 = \omega + \sum_{i=0}^{h-2} \beta^i + \beta^{h-1} \log \sigma_{t+1}^2$$

However, note that by the Jensen's inequality

$$\mathbb{E}_t[\sigma_{t+h}^2] > \exp(\mathbb{E}[\log(\sigma_{t+h}^2)]),$$

such that, if we wrongly compute the h step ahead variance prediction by taking $\exp(\widehat{\log \sigma}_{t+h|t}^2)$, we will always (for $h > 1$) underestimate the future volatility!

In practice we simulate the process - especially when the distribution is different than that of the Normal.

Volatility prediction with EGARCH

To compute the h step ahead prediction of the EGARCH(1,1) model we need to solve

$$\mathbb{E}_t[\sigma_{t+h}^2] = \mathbb{E}_t \left[\exp \left[\omega + \beta \log \sigma_{t+h-1}^2 + \phi z_{t+h-1} + \psi(|z_{t+h-1}| - \mathbb{E}|z_{t+h-1}|) \right] \right]$$

First note that for $h \geq 2$

$$\begin{aligned} \log \sigma_{t+h}^2 &= \omega + \beta \log \sigma_{t+h-1}^2 + g(z_{t+h-1}) \\ &= \omega \sum_{i=0}^{h-2} \beta^i + \beta^{h-1} \log \sigma_{t+1}^2 + \sum_{i=0}^{h-2} \beta^i g(z_{t+h-i-1}), \end{aligned}$$

thus

$$\sigma_{t+h}^2 = \exp \left(\omega \sum_{i=0}^{h-2} \beta^i \right) \sigma_{t+1}^{2\beta^{h-1}} \prod_{i=0}^{h-2} \exp(\beta^i g(z_{t+h-i-1})),$$

where $g(z_t) = \phi z_t + \psi(|z_t| - \mathbb{E}|z_t|)$.

Volatility prediction with EGARCH

It follows that the variance prediction is given by:

$$\hat{\sigma}_{t+h|t}^2 = \exp \left(\omega \sum_{i=0}^{h-2} \beta^i \right) \sigma_{t+1}^{2\beta^{h-1}} \mathbb{E}_t \left[\prod_{i=0}^{h-2} \exp(\beta^i g(z_{t+h-i-1})) \right],$$

however, since z_t is iid we have that

this assumption is truly required.

$$\mathbb{E}_t \left[\prod_{i=0}^{h-2} \exp(\beta^i g(z_{t+h-i-1})) \right] = \prod_{i=0}^{h-2} \mathbb{E}_t \left[\exp(\beta^i g(z_{t+h-i-1})) \right]$$

and that

Conditional expectations are equal to unconditional expectations with an iid process.

$$\mathbb{E}_t \left[\exp(\beta^i g(z_{t+h-i-1})) \right] = \mathbb{E} \left[\exp(\beta^i g(z_{t+h-i-1})) \right] = \mathbb{E} \left[\exp(\beta^i g(z_t)) \right]$$

Volatility prediction with EGARCH

The computation of $\mathbb{E}[\exp(bg(z_t))]$ depends on the distributional assumption of z_t . If $z_t \sim N(0, 1)$ we have

$$\mathbb{E}[e^{bg(z_t)}] = e^{-b\psi\sqrt{\frac{2}{\pi}}} \left[\Phi(b\psi + b\phi) \exp\left(\frac{b^2(\psi + \phi)^2}{2}\right) + \Phi(b\psi - b\phi) \exp\left(\frac{b^2(\psi - \phi)^2}{2}\right) \right]$$

where $\Phi(\cdot)$ is the cdf of a standard Gaussian distribution, i.e. $\Phi(z) = P(z_t \leq z)$.

If $z_t \sim GED(0, 1, v)$ this expression is more intricate and can be found in the appendix of Nelson (1991).

Volatility prediction with EGARCH

So, the h -step ahead variance prediction in the EGARCH(1,1) model with Gaussian shocks is

$$\begin{aligned}\hat{\sigma}_{t+h|t}^2 &= \exp \left(\omega \sum_{i=0}^{h-2} \beta^i \right) \sigma_{t+1}^{2\beta^{h-1}} \mathbb{E}_t \left[\prod_{i=0}^{h-2} \exp(\beta^i g(z_{t+h-i-1})) \right] \\ &= \exp \left[\left(\omega - \psi \sqrt{\frac{2}{\pi}} \right) \sum_{i=0}^{h-2} \beta^i \right] \sigma_{t+1}^{2\beta^{h-1}} \prod_{i=0}^{h-2} \left[\Phi(\beta^{2i}(\psi + \phi)) \exp \left(\frac{\beta^{2i}(\psi + \phi)^2}{2} \right) + \right. \\ &\quad \left. + \Phi(\beta^{2i}(\psi - \phi)) \exp \left(\frac{\beta^{2i}(\psi - \phi)^2}{2} \right) \right]\end{aligned}$$

GJR model

The Glosten - Jagannathan - Runkle model (1993):

$$\sigma_t^2 = \omega + \sum_{i=1}^p \beta_i \sigma_{t-i}^2 + \sum_{i=1}^q \left(\alpha_i r_{t-i}^2 + \gamma_i S_{t-i}^- r_{t-i}^2 \right) \text{ gamma is usually positive.}$$

where

$$S_t^- = \begin{cases} 1 & \text{if } r_t < 0 \\ 0 & \text{if } r_t \geq 0 \end{cases} \quad \text{Pretty much GARCH with an indicator function.}$$

If $p = q = 1$, we have

$$\sigma_t^2 = \omega + r_{t-1}^2 (\alpha + \gamma S_{t-1}^-) + \beta \sigma_{t-1}^2$$

The unconditional variance is

$$\mathbb{E}[\sigma_t^2] = \frac{\omega}{1 - \alpha - \kappa\gamma - \beta},$$

where $\kappa = P(z_t < 0)$ such that $\kappa = \frac{1}{2}$ if z_t is symmetrically distributed. Necessary conditions for covariance stationarity are $\omega > 0$ and $0 < \alpha + \kappa\gamma + \beta < 1$

ARCH-in-mean

The ARCH-in-mean model of Engle et al. (1987) was designed to capture the so-called volatility risk premium. It assumes that the volatility enters linearly the conditional mean of the returns. The model is formulated as:

$$\begin{aligned}r_t &= \delta\phi(\sigma_t) + \varepsilon_t, & \varepsilon_t &= \sigma_t z_t \\ \sigma_t^2 &= \text{some ARCH specification}\end{aligned}$$

The function $\phi(\cdot)$ is usually chosen between

$$\begin{aligned}\phi(\sigma_t) &= \sigma_t \\ \phi(\sigma_t) &= \sigma_t^2 \\ \phi(\sigma_t) &= \log \sigma_t\end{aligned}$$

The News Impact Curve

- The news have asymmetric effects on volatility.
- In the asymmetric volatility models good news and bad news have different predictability for future volatility.
- The news impact curve characterizes the impact of past return shocks on the return volatility which is implicit in a volatility model. It has been introduced by Engle and Ng (1993).
- Holding constant the information dated $t - 2$ and earlier, we can examine the implied relation between r_{t-1} and σ_t^2 , with $\sigma_{t-i}^2 = \sigma^2$ $i = 1, \dots, p$.
- This impact curve relates past return shocks (news) to current volatility.
- This curve measures how new information is incorporated into volatility estimates.

For the GARCH model the News Impact Curve (NIC) is centered on $r_{t-1} = 0$.
GARCH(1,1):

$$\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2$$

The news impact curve has the following expression:

$$\sigma_t^2 = A + \alpha r_{t-1}^2$$

$$A \equiv \omega + \beta \sigma^2$$

In the case of EGARCH model the curve has its minimum at $r_{t-1} = 0$ and is exponentially increasing in both directions but with different parameters.

EGARCH(1,1):

$$\ln(\sigma_t^2) = \omega + \beta \ln(\sigma_{t-1}^2) + \phi z_{t-1} + \psi(|z_{t-1}| - E|z_{t-1}|)$$

where $z_t = r_t / \sigma_t$. The news impact curve is

$$\sigma_t^2 = \begin{cases} A \exp\left[\frac{\phi + \psi}{\sigma} r_{t-1}\right] & \text{for } r_{t-1} > 0 \\ A \exp\left[\frac{\phi - \psi}{\sigma} r_{t-1}\right] & \text{for } r_{t-1} < 0 \end{cases}$$

$$\begin{aligned} A &\equiv \sigma^{2\beta} \exp[\omega - \psi E|z_{t-1}|] \\ \phi &< 0 \quad \phi + \psi > 0 \end{aligned}$$

- The EGARCH allows good news and bad news to have different impact on volatility, while the standard GARCH does not.
- The EGARCH model allows big news to have a greater impact on volatility than GARCH model. EGARCH would have higher variances in both directions because the exponential curve eventually dominates the quadrature.

The Asymmetric GARCH(1,1) (Engle, 1990)

$$\sigma_t^2 = \omega + \alpha (r_{t-1} + \gamma)^2 + \beta \sigma_{t-1}^2$$

the NIC is

$$\sigma_t^2 = A + \alpha (r_{t-1} + \gamma)^2$$

$$A \equiv \omega + \beta \sigma^2$$

$$\omega > 0, 0 \leq \beta < 1, \sigma > 0, 0 \leq \alpha < 1.$$

is asymmetric and centered at $r_{t-1} = -\gamma$.

The Glosten-Jagannathan-Runkle model

$$\sigma_t^2 = \omega + \alpha r_t^2 + \beta \sigma_{t-1}^2 + \gamma S_{t-1}^- r_{t-1}^2$$

$$S_{t-1}^- = \begin{cases} 1 & \text{if } r_{t-1} < 0 \\ 0 & \text{otherwise} \end{cases}$$

The NIC is

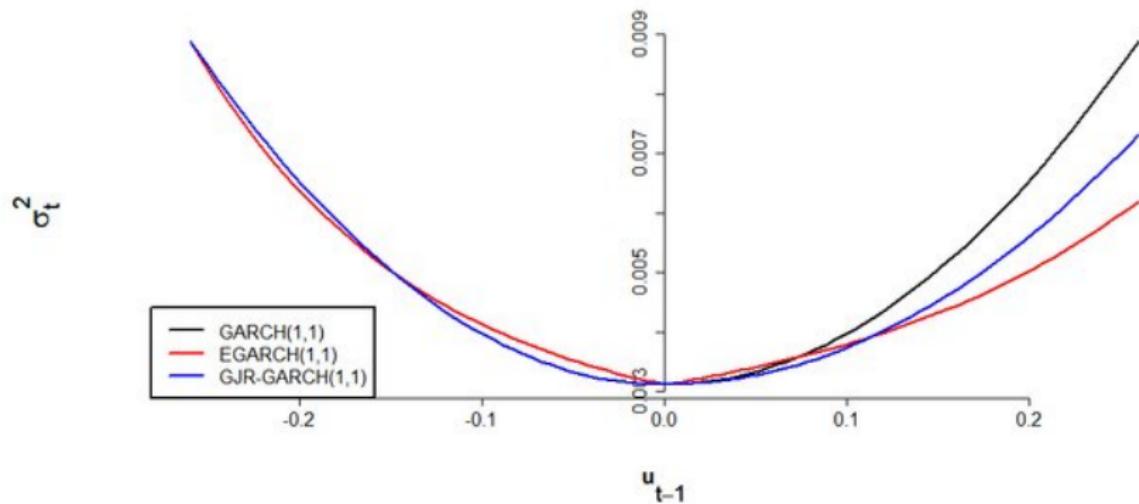
$$\sigma_t^2 = \begin{cases} A + \alpha r_{t-1}^2 & \text{if } r_{t-1} > 0 \\ A + (\alpha + \gamma) r_{t-1}^2 & \text{if } r_{t-1} < 0 \end{cases}$$

$$A \equiv \omega + \beta \sigma^2$$

$$\omega > 0, 0 \leq \beta < 1, \sigma > 0, 0 \leq \alpha < 1, \alpha + \beta < 1$$

is centered at $r_{t-1} = -\gamma$.

News Impact Curve



Likelihood function

Assume the model:

$$\begin{aligned} r_t &= \sigma_t z_t \quad z_t \stackrel{iid}{\sim} p_z(z_t; \eta) \\ \sigma_t^2 &= \sigma^2(\psi, r_{1:t-1}) \end{aligned}$$

Let $\theta = (\psi, \eta)$ be the vector of fixed parameters to be estimated. The likelihood function is

$$\begin{aligned} \log L_T(\theta | y_{1:T}) &= \sum_{t=1}^T \log p_r(r_t | \mathcal{F}_{t-1}) \\ &\quad \text{this step is explained in the video.} \\ &= \sum_{t=1}^T \log p_z(z_t(\psi); \eta) - \frac{1}{2} \log \sigma_t^2(\psi), \\ &= \sum_{t=1}^T l_t(\theta) \end{aligned}$$

where $l_t(\theta) = \log p_z(z_t(\psi); \eta) - \frac{1}{2} \log \sigma_t^2(\psi)$ and $z_t(\theta) = \frac{r_t}{\sigma_t(\psi)}$.

Likelihood function

Consider the Gaussian case where

$$p_z(z_t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z_t^2}{2}},$$

note that in this case $\theta = \psi$ and $\eta = \{\emptyset\}$.

The log likelihood becomes

$$l_t(\theta) \propto -\frac{z_t(\theta)^2}{2} - \frac{1}{2} \log \sigma_t^2(\theta),$$

where " \propto " means "proportional to" a constant ($-\frac{1}{2} \log 2\pi$).

Likelihood function: Gradient and Hessian

The gradient (or score vector) is

$$\begin{aligned}\frac{\partial l_t(\theta)}{\partial \theta} &= -\frac{1}{2} \frac{\partial z_t(\theta)^2}{\partial \theta} - \frac{1}{2} \frac{\partial \log \sigma_t^2(\theta)}{\partial \theta} \\ &= \frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} \left[\frac{r_t^2}{\sigma_t^2(\theta)} - 1 \right].\end{aligned}$$

The Hessian matrix is

$$\begin{aligned}\frac{\partial^2 l_t(\theta)}{\partial \theta \partial \theta'} &= \left[\frac{r_t^2}{\sigma_t^2(\theta)} - 1 \right] \frac{\partial}{\partial \theta'} \left(\frac{1}{2} \frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} \right) - \frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} \frac{r_t^2}{\sigma_t^4(\theta)} \frac{\partial \sigma_t^2(\theta)}{\partial \theta'} \\ &= \left[\frac{r_t^2}{\sigma_t^2(\theta)} - 1 \right] \frac{\partial}{\partial \theta'} \left(\frac{1}{2} \frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} \right) - \frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{\partial \sigma_t^2(\theta)}{\partial \theta'} \frac{r_t^2}{2\sigma_t^6(\theta)}\end{aligned}$$

Likelihood function: Gradient and Hessian

Note that the expected value of the score is zero:

$$\begin{aligned}
 \mathbb{E} \left[\frac{\partial l_t(\theta)}{\partial \theta} \right] &= \mathbb{E} \left[\frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} \left(\frac{r_t^2}{\sigma_t^2(\theta)} - 1 \right) \right] \\
 &= \mathbb{E} \left[\frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} (z_t^2 - 1) \right] \\
 &= \mathbb{E} \left[\frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} \right] \mathbb{E} [z_t^2 - 1] = 0
 \end{aligned}$$

where we exploited the fact that $r_t = \sigma_t(\theta)z_t$ (in the second line), the independence of the z_t (to split the expectation in the third line), and the fact that $\mathbb{E}[z_t^2] = 1$.

Likelihood function: Gradient and Hessian

From ML theory we know that the Fisher information matrix is related to the expected value of the Hessian matrix in this way:

$$\mathcal{I}(\theta) = -\mathbb{E} \left[\frac{\partial^2 l_t(\theta)}{\partial \theta \partial \theta'} \right]$$

such that, for a Gaussian likelihood we have

$$\begin{aligned}\mathcal{I}(\theta) &= -\mathbb{E} \left[\left[\frac{r_t^2}{\sigma_t^2(\theta)} - 1 \right] \frac{\partial}{\partial \theta'} \left(\frac{1}{2} \frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} \right) - \frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{\partial \sigma_t^2(\theta)}{\partial \theta'} \frac{r_t^2}{2\sigma_t^6(\theta)} \right] \\ &= -\mathbb{E} \left[(z_t^2 - 1) \frac{\partial}{\partial \theta'} \left(\frac{1}{2} \frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{1}{2\sigma_t^2(\theta)} \right) \right] + \mathbb{E} \left[\frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{\partial \sigma_t^2(\theta)}{\partial \theta'} \frac{z_t^2}{2\sigma_t^4(\theta)} \right] \\ &= \mathbb{E} \left[\frac{\partial \sigma_t^2(\theta)}{\partial \theta} \frac{\partial \sigma_t^2(\theta)}{\partial \theta'} \frac{1}{2\sigma_t^4(\theta)} \right]\end{aligned}$$

Likelihood function: Gradient and Hessian

For a GARCH(1,1) model we have

$$\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2$$

Jondeau et al. (2007) propose to estimate $\mathcal{I}(\theta)$ as

$$\widehat{\mathcal{I}}(\theta) = \frac{1}{2T} \sum_{t=1}^T \frac{\hat{\zeta}'_t \hat{\zeta}_t}{\hat{\sigma}_t^4},$$

where $\hat{\zeta}_t = (1, r_{t-1}^2, \sigma_{t-1}^2)'$.² However, it is easily seen that

$$\begin{aligned}\frac{\partial \sigma_t^2(\theta)}{\partial \omega} &= 1 + \beta \frac{\partial \sigma_{t-1}^2(\theta)}{\partial \omega} \\ \frac{\partial \sigma_t^2(\theta)}{\partial \alpha} &= r_{t-1}^2 + \beta \frac{\partial \sigma_{t-1}^2(\theta)}{\partial \alpha} \\ \frac{\partial \sigma_t^2(\theta)}{\partial \beta} &= \sigma_{t-1}^2(\theta) + \beta \frac{\partial \sigma_{t-1}^2(\theta)}{\partial \beta}\end{aligned}$$

²Note a typo in their derivations at page 90-91. Instead of σ_t^4 they write σ_t^2 which is wrong.

Regression GARCH

For models of the type

$$r_t = X'_t \delta + \varepsilon_t, \quad \varepsilon_t = \sigma_t z_t$$

$$\sigma_t^2 = \omega + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^q \beta_i \sigma_{t-i}^2$$

it can be shown that

$$\mathbb{E} \left[\frac{\partial^2 I_t(\theta; \delta)}{\partial \theta \partial \delta'} \right] = 0,$$

where $\theta = (\omega, \alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q)'$. This result implies that the Fisher information matrix is block diagonal and that the two set of parameters δ and θ can be estimated in two steps.

Two words of caution:

- 1) For other ARCH models (like EGARCH) this is not true. The condition need to be checked on a case by case basis.
- 2) For the ARCH-in-mean specification the condition is never satisfied.

Quasi-maximum likelihood estimation

Quasi-maximum likelihood estimation (QML) = the method based on the maximization of the log likelihood assuming conditional normality.

- Thus, we do as if the conditionally standardized process z_t follows a normal distribution.
- Even if the normality assumption does not hold (i.e., the true distribution is not conditionally normal), the estimator - then called quasi-maximum likelihood estimator - is consistent and asymptotically normal

Quasi-maximum likelihood estimation

Under regularity conditions, the QML estimator is asymptotically normal distributed with

$$\sqrt{T}(\hat{\varphi}_n - \varphi_0^*) \xrightarrow{d} N(0, A^{-1}BA^{-1})$$

The matrices A and B are, respectively, equal to:

$$A = -\frac{1}{T} E_0 \left[\frac{\partial^2 \log L(\varphi)}{\partial \varphi \partial \varphi'} \right]$$

$$B = \frac{1}{T} E_0 \left[\frac{\partial \log L(\varphi)}{\partial \varphi} \frac{\partial \log L(\varphi)}{\partial \varphi'} \right]$$

The matrices A and B are not, in general, equal when specification errors are present. Thus comparing estimates of the matrices A and B can be useful for detecting specification errors.

What we have learned in this lecture?

- 1) Several ARCH specifications exist which are able to represent two of the main features of financial returns: i) volatility clustering, and ii) excess of kurtosis in the unconditional distribution ($\kappa > 3$).
- 2) The methodology to study the properties of an ARCH model: i) strong stationarity, ii) weak stationarity, iii) ARMA representations, and iv) how to make predictions.
- 3) The effect of the initialization of the volatility in the properties of the GARCH and IGARCH models.
- 4) The ARCH, GARCH, IGARCH, EGARCH, GJR-GARCH models, and the ARCH-in-mean specification.
- 5) The asymmetric reaction of volatility to past positive/negative returns (leverage effect), and the News Impact Curve.
- 6) Model estimation via the Maximum Likelihood Estimator.

Which tools we used from math/stat/econometrics?

1) Probability concepts:

- i) conditional/unconditional/joint distributions,
- ii) taking the expectation of a random variable,
- iii) moments of a random variable,
- iv) independence of random variables,
- v) transformation of a random variable,
- vi) strong/weak stationarity,
- vii) the strong law of large numbers for iid random variables.

2) Econometrics tools:

- i) Maximum Likelihood Estimation,
- ii) OLS,
- iii) hypothesis testing via the Lagrange Multiplier test,
- iv) linear models: Random Walks and ARMA.

3) Math tools:

- i) convergence of geometric series,
- ii) computing derivatives: gradient and Hessian of a function,
- iii) maximization of a function,
- iv) Jensen's inequality.

References I

- Engle, R. F., Lilien, D. M., and Robins, R. P. (1987). Estimating time varying risk premia in the term structure: The arch-m model. *Econometrica: journal of the Econometric Society*, pages 391–407.
- Engle, R. F. and Ng, V. K. (1993). Measuring and testing the impact of news on volatility. *The journal of finance*, 48(5):1749–1778.
- Jondeau, E., Poon, S.-H., and Rockinger, M. (2007). *Financial modeling under non-Gaussian distributions*. Springer Science & Business Media.
- Nelson, D. B. (1990). Stationarity and persistence in the garch (1, 1) model. *Econometric theory*, 6(3):318–334.
- Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the Econometric Society*, pages 347–370.
- Stout, W. F. (1974). Almost sure convergence. Technical report.

UNIVARIATE VOLATILITY MODELING: STOCHASTIC VOLATILITY

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

SV models: a class of models

So far we have discussed models of the form $y_t = \sigma_t \varepsilon_t$, where $\sigma_t = \sigma(\mathcal{F}_{t-1}, \theta)$, that is: models for which the variance at time t given \mathcal{F}_{t-1} is known.

Note that the only source of error in this class of models is ε_t . Indeed, models of this kind are classified as “Single Source of Error” according to Snyder (1985).

However, it might be reasonable to believe that different shocks update models parameters. To this end, Taylor (1986) has introduced the class of Stochastic Volatility (SV) models.

Stochastic volatility

Consider the model

$$y_t = \sigma_t u_t,$$

where

- σ_t is a positive random variable with $Var(\sigma_t | \mathcal{F}_{t-1}) > 0$.
- $\{\sigma_t\}$ is stationary with $E[\sigma_t^4] < \infty$ and $\rho_{\tau,\sigma^2} = cor(\sigma_t^2, \sigma_{t+\tau}^2) > 0$.
- $u_t \stackrel{iid}{\sim} D(0, 1)$.
- $\{u_t\}$ and $\{\sigma_t\}$ are independent.

Note that this model does not nest GARCH models, why?

Stochastic volatility

Since $\{u_t\}$ and $\{\sigma_t\}$ are independent we have that for any function f_1 and f_2

$$E[f_1(\sigma_t, \sigma_{t-1}, \dots) f_2(u_t, u_{t-1}, \dots)] = E[f_1(\sigma_t, \sigma_{t-1}, \dots)] E[f_2(u_t, u_{t-1}, \dots)]$$

Let's evaluate:

- $E[y_t]$
- $var(y_t)$
- $skew(y_t)$
- $kurt(y_t)$

Stochastic volatility: moments

Note that $y_t = \sigma_t u_t$, hence:

$$E[\sigma_t u_t] = E[\sigma_t]E[u_t] = 0$$

$$\text{var}(\sigma_t u_t) = E[\sigma_t^2 u_t^2] - (E[\sigma_t u_t])^2 = E[\sigma_t^2]E[u_t^2] = E[\sigma_t^2]$$

$$E[\sigma_t^3 u_t^3] = E[\sigma_t^3]E[u_t^3] = E[\sigma_t^3]\text{skew}(u_t)$$

$$E[\sigma_t^4 u_t^4] = E[\sigma_t^4]E[u_t^4] = E[\sigma_t^4]\text{kurt}(u_t)$$

hence

$$\text{skew}(\sigma_t u_t) = \frac{E[\sigma_t^3 u_t^3]}{E[(\sigma_t u_t)^2]^{3/2}} = \frac{E[\sigma_t^3]}{E[\sigma_t^2]^{3/2}} \text{skew}(u_t)$$

$$\text{kurt}(\sigma_t u_t) = \text{kurt}(u_t) \left(1 + \frac{\text{var}(\sigma_t^2)}{E[\sigma_t^2]^2} \right) > \text{kurt}(u_t)$$

Stochastic volatility: autocovariance

Autocovariance of y_t :

$$\begin{aligned}\gamma_{\tau,y} &= \text{cov}(y_t, y_{\tau+t}) = \text{cov}(\sigma_t u_t, \sigma_{\tau+t} u_{\tau+t}) \\ &= E[\sigma_t u_t \sigma_{\tau+t} u_{\tau+t}] - E[\sigma_t u_t] E[\sigma_{\tau+t} u_{\tau+t}] \\ &= E[\sigma_t \sigma_{\tau+t}] E[u_t u_{\tau+t}] - E[\sigma_t u_t] E[\sigma_{\tau+t} u_{\tau+t}] = 0\end{aligned}$$

Autocovariance of $y_t^2 = \sigma_t^2 u_t^2 = s_t$:

$$\begin{aligned}\gamma_{\tau,s} &= \text{cov}(s_t, s_{t+\tau}) = \text{cov}(\sigma_t^2 u_t^2, \sigma_{t+\tau}^2 u_{t+\tau}^2) \\ &= E[\sigma_t^2 u_t^2 \sigma_{t+\tau}^2 u_{t+\tau}^2] - E[\sigma_t^2 u_t^2] E[\sigma_{t+\tau}^2 u_{t+\tau}^2] \\ &= E[\sigma_t^2 \sigma_{t+\tau}^2] E[u_t^2 u_{t+\tau}^2] - E[\sigma_t^2] E[u_t^2] E[\sigma_{t+\tau}^2] E[u_{t+\tau}^2] \\ &= E[\sigma_t^2 \sigma_{t+\tau}^2] - E[\sigma_{t+\tau}^2] E[\sigma_t^2] \\ &= \text{cov}(\sigma_t^2, \sigma_{t+\tau}^2) = \gamma_{\tau,\sigma^2}\end{aligned}$$

Independent random volatility

Consider the model:

$$\begin{aligned}y_t &= \sigma_t u_t \\&= \exp(w_t/2) u_t\end{aligned}$$

where:

$$\begin{pmatrix} u_t \\ w_t \end{pmatrix} \stackrel{iid}{\sim} N \left(\begin{pmatrix} 0 \\ \zeta \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & \sigma_w^2 \end{pmatrix} \right).$$

How would you estimate it?

Independent random volatility

We have several alternatives to estimate the model's parameters ζ and σ_w . For instance:

- Maximum Likelihood
- Method of Moments
- Generalized Method of Moments
- Simulation techniques

Consider for example the first two possibilities.

Independent random volatility: ML

In order to perform Maximum Likelihood estimation of the model's parameter we need to derive the log-likelihood function. Assume to observe a sequence of T observations $y_{1:T} = \{y_1, \dots, y_T\}$:

$$\begin{aligned}\mathcal{L}(\zeta, \sigma_w | y_{1:T}) &= \log P(y_{1:T}) \\ &= \sum_{t=1}^T \log p(y_t),\end{aligned}$$

since in this model $y_t \perp\!\!\!\perp y_{t+\tau}$ for all $\tau \neq 0$. However:

$$\begin{aligned}p(y_t) &= \int_{\Re} p(y_t | w_t) p(w_t) dw_t \\ &= \frac{1}{2\pi\sigma_w} \int_{\Re} \exp \left[-\frac{1}{2} \left(w_t + \frac{y_t^2}{\exp(w_t)} + \frac{(w_t - \zeta)^2}{\sigma_w^2} \right) \right] dw_t,\end{aligned}$$

does not admit a closed form solution. Let's code it in R!

Intermezzo: Log Normal Distribution

If $X \sim N(a, b^2)$, then $\exp X = Z \sim LN(a, b^2)$ such that:

$$p(z) = \frac{1}{zb\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{(\log z - a)^2}{b^2}\right)\right), \quad z > 0$$

$$E[Z^n] = \exp\left(na + \frac{1}{2}n^2b^2\right)$$

$$\text{var}(Z) = \exp(2a + b^2)(\exp(b^2) - 1),$$

Independent random volatility: MM

We note that since $w_t \sim N(\zeta, \sigma_w^2)$, then $\sigma_t^2 = \exp(w_t) \sim LN(\zeta, \sigma_w^2)$. Also note that $E[\sigma_t^2] = \exp(\zeta + \frac{\sigma_w^2}{2})$

$$E[y_2^2] = E[\sigma_t^2 u_t^2] = E[\sigma_t^2] E[u_t^2] = E[\exp(w_t)] = \exp\left(\zeta + \frac{\sigma_w^2}{2}\right)$$

$$E[y_t^4] = E[\sigma_t^4 u_t^4] = E[\sigma_t^4] E[u_t^4] = 3E[\exp(2w_t)] = 3\exp(2(\zeta + \sigma_w^2))$$

Let $\hat{\mu}_2 = T^{-1} \sum_{t=1}^T y_t^2$ and $\hat{\mu}_4 = T^{-1} \sum_{t=1}^T y_t^4$. By equating empirical and theoretical moments we find:

$$\sigma_w^2 = \log\left(\frac{\hat{\mu}_4}{3\hat{\mu}_2^2}\right), \quad \zeta = \log\left(\frac{\hat{\mu}_2^2 \sqrt{3}}{\sqrt{\hat{\mu}_4}}\right)$$

Let's code it in R!

SV models: a class of models

Stochastic Volatility models belong to the general class of nonlinear non gaussian state space models:

$$y_t = m(\theta_t, \varepsilon_t), \quad (1)$$

$$\theta_{t+1} = q(\theta_t, \eta_t) \quad (2)$$

where ε_t and η_t are (possibly dependent) random variables.

The Log-Normal AR(1) Stochastic Volatility Model

In the original SV model we have:

$$y_t = \sigma_t u_t \quad (3)$$

$$\log \sigma_{t+1} - \alpha = \phi(\log \sigma_t - \alpha) + \zeta_t, \quad (4)$$

where

$$\begin{pmatrix} u_t \\ \zeta_t \end{pmatrix} \stackrel{iid}{\sim} N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & \sigma_\zeta^2 \end{pmatrix} \right), \quad (5)$$

that is: $\log \sigma_t$ follows a first order autoregressive process. Hence $\log \sigma_t \sim N(\alpha, \beta^2)$, where $\beta^2 = \frac{\sigma_\zeta^2}{1-\phi^2}$.

The Log-Normal AR(1) Stochastic Volatility Model

The model is also written as:

$$y_t = \exp(w_t/2) u_t \quad (6)$$

$$w_{t+1} = \omega + \phi w_t + \eta_t, \quad (7)$$

where $w_t = \log \sigma_t^2$, and

$$\begin{pmatrix} u_t \\ \eta_t \end{pmatrix} \stackrel{iid}{\sim} N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & \sigma_\eta^2 \end{pmatrix} \right), \quad (8)$$

Note that to ensure stationarity of w_t we have to impose $|\phi| < 1$ like in usual AR(1) models. In this case $E[w_t] = \frac{\omega}{1-\phi}$

The Log-Normal AR(1) Stochastic Volatility Model

Basic properties

- $\{y_t\}$ is strictly stationary
- All moments of y_t are finite
- $kurt(y_t) = 3 \exp(\beta^2)$, where $\beta^2 = \text{var}(w_t)$.
- $\text{cov}(y_t, y_{t+\tau}) = 0$
- $\text{cov}(s_t, s_{t+\tau}) > 0$, when $\psi > 0$, $s_t = y_t^2$
- ACF of $|y_t|^p$ behaves like ACF of s_t .

SV Model: estimation

Suppose to observe a sequence of T observations from the SV model with zero mean $y_{1:T} = \{y_1, \dots, y_T\}$. The likelihood is:

$$\begin{aligned} L(\alpha, \phi, \sigma_\eta | y_{1:T}) &= \log p(y_{1:T}) \\ &= \log \int p(y_{1:T} | \sigma_{1:T}) p(\sigma_{1:T}) d\sigma_{1:T} \\ &= \log \int \prod_{t=1}^T p(y_t | \sigma_t) p(\sigma_t | \sigma_{t-1}) d\sigma_t, \end{aligned}$$

where the distribution of σ_1 can be set to the unconditional distribution of the process. Note that direct numerical integration is not feasible. Can you derive a GMM estimator?

SV Model: extensions

- Fat tailed distribution for u_t (E.g. Student's t)
- Dependence between u_t and η_t (Leverage effect)
- Long memory in $\log \sigma_t$
- Multivariate formulations

Not straightforward to implement!

References

- Snyder, R. (1985). Recursive estimation of dynamic linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 272–276.
- Taylor, S. J. (1986). *Modelling financial time series*. world scientific.

GENERALIZED METHOD OF MOMENTS WITH APPLICATION TO THE SV MODEL

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Introduction

- GMM has become one of the main statistical tools for the analysis of economic and financial data.
- GMM was first introduced by L.P. Hansen in 1982. Since then it has been widely applied to analyze economic and financial data.
- GMM has been applied to time series, cross sectional, and panel data.
- Optimality of MLE stems from its basis on the joint p.d. of the data. However, in some circumstances, this dependence becomes a weakness.

Problems of MLE:

- Sensitivity of statistical properties to the distributional assumptions
- The likelihood function is not always available...
- ... for instance if we have latent (unobserved) variables in the model (SV models).
- Computational burden.

The Method of Moments

- Population moments implied by the family of distributions are functions of the unknown parameter vector.
- Pearson (1895) proposed estimating the parameter vector by the value implied by the corresponding sample moments.
- Normal distribution with parameters μ_0 and σ_0^2

$$\begin{aligned} E[x_t] - \mu_0 &= 0 \\ E[x_t^2] - (\sigma_0^2 + \mu_0^2) &= 0 \end{aligned}$$

- Pearson's method involves replacing the population moments by the sample moments:

$$\begin{aligned} E_N[x_t] - \hat{\mu} &= 0 \\ E_N[x_t^2] - (\hat{\sigma}^2 + \hat{\mu}_0^2) &= 0 \end{aligned}$$

The Method of Moments: Orthogonality Condition and Regression

In linear regression model we assume the following condition:

$$E(x_t \epsilon_t) = 0 \quad t = 1, \dots, T$$

Let β_0 denote the true value of β , where the latter denotes a generic value of the parameter. Then

$$E[x_t(y_t - x_t\beta_0)] = 0$$

The Method of Moments: Orthogonality Condition and Regression

The moment condition can be written as

$$g_t(\beta) = x_t(y_t - x_t\beta)$$

such that

$$E[g_t(\beta_0)] = 0$$

The empirical analogue of $E[g_t(\beta_0)]$ is

$$\bar{g}_t = \frac{1}{T} \sum_{t=1}^T x_t(y_t - x_t\beta) = \frac{1}{T}(X'y - X'X\beta).$$

Which implies

$$(X'y - X'X\hat{\beta}) = 0,$$

which are called *normal equations*.

The Method of Moments: the SV model

Consider the SV model

$$y_t = \exp(h_t/2) u_t$$
$$h_{t+1} - \alpha = \phi(h_t - \alpha) + \eta_t$$

where

$$\begin{pmatrix} u_t \\ \eta_t \end{pmatrix} \stackrel{iid}{\sim} N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & \sigma_\eta^2 \end{pmatrix} \right).$$

We want to estimate $\theta = (\alpha, \phi, \sigma_\eta^2)$, or the reparameterization $\theta = (\alpha, \phi, \beta^2)$ where $\beta^2 = \sigma_\eta^2 / (1 - \phi^2)$.

The Method of Moments: the SV model

Consider the population parameters α, ϕ, β^2 . Three population moments are (for example):

$$E[|y_t|] = \sqrt{2/\pi} \exp\left(\frac{\alpha}{2} + \frac{1}{8}\beta^2\right)$$

$$E[y_t^2] = \exp\left(\alpha + \frac{1}{2}\beta^2\right)$$

$$E[y_t^4] = 3 \exp\left(2\alpha + 2\beta^2\right)$$

The Method of Moments: the SV model

One possible solution for α and β^2 is obtained using the empirical counterparts of $E(y_t^2)$ and $E(y_t^4)$, that we indicate by $\hat{\mu}_2 = \frac{1}{T} \sum_{t=1}^T y_t^2$ and $\hat{\mu}_4 = \frac{1}{T} \sum_{t=1}^T y_t^4$, respectively. The MM estimator of (α, β^2) , indicated by $(\hat{\alpha}, \hat{\beta})$, is given by:

$$\hat{\alpha} = \log \left(\hat{\mu}_2^2 \sqrt{\frac{3}{\hat{\mu}_4}} \right), \quad \hat{\beta}^2 = \log \left(\frac{\hat{\mu}_4}{3\hat{\mu}_2^2} \right)$$

Note that $\hat{\mu}_4$ is very sensitive to extreme observations!

The Method of Moments: the SV model

One second solution for α and β^2 is obtained using $E[|y_t|]$ and $E[y_t^2]$:

$$\hat{\alpha} = \log \left(\frac{\pi^2 \hat{\kappa}^4}{4\mu_2} \right), \quad \hat{\beta}^2 = \log \left(\frac{16\hat{\mu}_2^4}{\pi^4 \hat{\kappa}^8} \right),$$

where $\hat{\kappa} = \frac{1}{T} \sum_{t=1}^T |y_t|$ is the empirical counterpart of $E[|y_t|]$.

- No simple method of moments estimator for ϕ
- Method of moments estimators are not unique!

The Generalized Method of Moments (GMM)

The idea of GMM is to optimally combine moment conditions to estimate population parameters.

Let $\{w_t\}$ be a covariance stationary and ergodic vector process representing the underlying data. Let the $p \times 1$ vector θ denote the population parameters. The moment conditions $g(w_t, \theta)$ are $K \geq p$ possibly nonlinear functions satisfying:

$$E[g(w_t, \theta_0)] = 0,$$

where θ_0 represents the true parameter vector.

The Generalized Method of Moments: identification

Global identification of θ_0 requires that:

$$\begin{aligned} E[g(w_t, \theta_0)] &= 0 \\ E[g(w_t, \theta)] &\neq 0 \quad \text{for } \theta \neq \theta_0, \end{aligned}$$

Local identification requires that the $K \times p$ matrix

$$G = E \left[\frac{\partial g(w_t, \theta_0)}{\partial \theta'} \right],$$

has full column rank p .

The Generalized Method of Moments: estimation

The sample moment conditions for an arbitrary θ is:

$$g_T(\theta) = T^{-1} \sum_{t=1}^T g(w_t, \theta).$$

If $K = p$, then θ_0 is apparently just identified and the GMM objective function is:

$$J(\theta) = T g_T(\theta)' g_T(\theta),$$

which does not depend on a weight matrix. The corresponding GMM estimator is then:

$$\hat{\theta} = \arg \min_{\theta} J(\theta)$$

and satisfies $g_T(\hat{\theta}) = 0$.

The Generalized Method of Moments: estimation

If $K > p$, then θ_0 is apparently overidentified. We thus denote with \hat{W} a $K \times K$ symmetric and positive definite weight matrix, possibly dependent on the data, such that $\hat{W} \rightarrow W$ as $T \rightarrow \infty$ with W symmetric and positive definite.

The GMM estimator of θ_0 , denoted $\hat{\theta}(\hat{W})$, is defined as:

$$\hat{\theta} = \arg \min_{\theta} J(\theta, \hat{W}) = T g_T(\hat{\theta})' \hat{W} g_T(\hat{\theta}),$$

whose first order conditions are:

$$\frac{\partial J(\hat{\theta}(\hat{W}), \hat{W})}{\partial \theta} = 2 G_T(\hat{\theta}(\hat{W}))' \hat{W} g_T(\hat{\theta}(\hat{W})) = 0$$

$$G_T(\hat{\theta}(\hat{W})) = \frac{\partial g_T(\hat{\theta}(\hat{W}))}{\partial \theta'}$$

The Generalized Method of Moments: Asymptotic Properties

Under standard regularity conditions, it can be shown that:

$$\begin{aligned}\hat{\theta}(\hat{W}) &\xrightarrow{P} \theta_0 \\ \sqrt{T}(\hat{\theta}(\hat{W}) - \theta_0) &\xrightarrow{d} N(0, avar(\hat{\theta}(\hat{W}))),\end{aligned}$$

where

$$avar(\hat{\theta}(\hat{W})) = (G'WG)^{-1}G'WSWG(G'WG)^{-1},$$

and

$$\begin{aligned}G &= E \left[\frac{\partial g(w_t, \theta_0)}{\partial \theta'} \right] \\ S &= avar(\sqrt{T}g_T(\theta_0))\end{aligned}$$

The Generalized Method of Moments: About W

The efficient GMM estimator uses a weight matrix W that minimizes $\text{avar}(\hat{\theta}(\hat{W}))$. Hansen (1982) showed that the optimal weight matrix is $W = S^{-1}$, that is:

$$\text{avar}(\hat{\theta}(\hat{W})) = (G' S^{-1} G)^{-1},$$

if $\{g_t(w_t, \theta_0)\}$ is an ergodic stationarity martingale difference sequence then:

$$S = E[g_t(w_t, \theta_0)g_t(w_t, \theta_0)'],$$

and a consistent estimator of S takes the form:

$$\hat{S}_{HC} = T^{-1} \sum_{t=1}^T g_t(w_t, \hat{\theta})g_t(w_t, \hat{\theta})',$$

The Generalized Method of Moments: About W

If $\{g_t(w_t, \theta_0)\}$ is a mean-zero serially correlated ergodic stationary process then

$$S = \Gamma_0 + \sum_{j=1}^{\infty} (\Gamma_j + \Gamma'_j),$$
$$\Gamma_j = E[g_t(w_t, \theta_0)g_t(w_{t-j}, \theta_0)]$$

and a consistent estimator has the form:

$$S_{HAC} = \hat{\Gamma}_0(\hat{\theta}) + \sum_{j=1}^{q(T)} k\left(\frac{j}{q(T)+1}\right) (\hat{\Gamma}_j(\hat{\theta}) + \hat{\Gamma}'_j(\hat{\theta}))'$$
$$\hat{\Gamma}_j(\hat{\theta}) = \frac{1}{T-j} \sum_{t=j+1}^T g_t(w_t, \hat{\theta})g_t(w_{t-j}, \hat{\theta})',$$

for a proper kernel function $k(\cdot)$. The usual choice is the triangular kernel $k(x) = 1 - |x|$. We usually set $q(T) = \lfloor T^{1/3} \rfloor$

The Generalized Method of Moments: SV

Consider the alternative parameterization of the simple log-normal stochastic volatility (SV) model assuming:

$$y_t = \exp(h_t/2) u_t$$
$$h_{t+1} = \omega + \phi h_t + \eta_t,$$

where

$$\begin{pmatrix} u_t \\ \eta_t \end{pmatrix} \stackrel{iid}{\sim} N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & \sigma_\eta^2 \end{pmatrix} \right)$$

and in this case $\theta = (\omega, \phi, \sigma_\eta^2)$. Note that according to this model y_t is stationary and unconditional moments of all orders exists.

The Generalized Method of Moments: SV

The GMM estimation of the SV model is surveyed in Andersen and Sørensen (1996).

They recommended using moment conditions for GMM estimation based on lower-order moments of y_t , since higher-order moments tend to exhibit erratic finite sample behavior.

They considered a GMM estimation based on (subsets) of 24 moments considered by Jacquier et al. (1994). To describe these moment conditions, first define:

$$\alpha = \frac{\omega}{1 - \phi}, \quad \beta^2 = \frac{\sigma_\eta^2}{1 - \phi^2},$$

and $\theta_h = (\alpha, \phi, \beta^2)$, which is just a reparameterization of $\theta = (\omega, \alpha, \sigma_\eta^2)$.

The Generalized Method of Moments: SV

The moment conditions, which follow from properties of the log-normal distribution and the Gaussian AR(1) model, are expressed as:

$$E[|y_t|] = (2/\pi)^{1/2} E[\sigma_t]$$

$$E[y_t^2] = E[\sigma_t^2]$$

$$E[|y_t^3|] = 2\sqrt{2/\pi} E[\sigma_t^3]$$

$$E[y_t^4] = 3E[\sigma_t^4]$$

$$E[|y_t y_{t-j}|] = (2/\pi) E[\sigma_t \sigma_{t-j}], \quad j = 1, \dots, 10$$

$$E[y_t^2 y_{t-j}^2] = E[\sigma_t^2 \sigma_{t-j}^2], \quad j = 1, \dots, 10$$

where for any positive integer j and positive constants p and s ,

$$E[\sigma_t^p] = \exp\left(\frac{p\alpha}{2} + \frac{p^2\beta^2}{8}\right)$$

$$E[\sigma_t^p \sigma_{t-j}^s] = E[\sigma_t^p] E[\sigma_{t-j}^s] \exp\left(\frac{ps\phi^j\beta^2}{4}\right)$$

The Generalized Method of Moments: SV

We set $w_t = (|y_t|, y_t^2, |y_t^3|, y_t^4, |y_t y_{t-1}|, \dots, |y_t y_{t-10}|, y_t^2 y_{t-1}^2, \dots, y_t^2 y_{t-10}^2)',$ and define the 24×1 vector

$$g(w_t, \theta_h) = \begin{pmatrix} |y_t| - (2/\pi)^{1/2} \exp\left(\frac{\alpha}{2} + \frac{\beta^2}{8}\right) \\ y_t^2 - \exp\left(\alpha + \frac{\beta^2}{2}\right) \\ \vdots \\ y_t^2 y_{t-10}^2 - \exp\left(\alpha + \frac{\beta^2}{2}\right)^2 \exp(\phi^{10} \beta^2) \end{pmatrix}$$

Then, $E[g(w_t, \theta_{h0})] = 0$ is the population moment condition used for the GMM estimation of the model parameters $\theta_h = (\alpha, \phi, \beta^2).$

Since the elements of w_t are serially correlated, the efficient weight matrix $S = \text{avar}(\sqrt{T}g_T(\theta_{h0}))$, must be estimated using an HAC estimator.

References I

- Andersen, T. G. and Sørensen, B. E. (1996). Gmm estimation of a stochastic volatility model: a monte carlo study. *Journal of Business & Economic Statistics*, 14(3):328–352.
- Jacquier, E., Polson, N. G., and Rossi, P. E. (1994). Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, 20(1):69–87.

STATE-SPACE MODELS AND THE KALMAN FILTER

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

A brief introduction

- State-space models are models that use state variables to describe a system by a set of difference equations;
- The object of the methodology is to infer relevant properties of the state variables α_t from knowledge of the observations y_1, \dots, y_t
- Most time series models can be written in state-space form.
- The state-space form representation allows for a straightforward modelling of additive feature of the data such as missing values, seasonal components, measurement errors and outliers.

State Space form: univariate model

Measurement equation:

$$y_t = Z\alpha_t + D\varepsilon_t, \quad t = 1, 2, \dots, T, \quad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2), \quad (1)$$

where Z is a $1 \times m$ matrix and D is a selection matrix.

Transition equation:

$$\alpha_{t+1} = T\alpha_t + H\eta_t, \quad \eta_t \sim NID(0, Q), \quad (2)$$

where T is $m \times m$ and H is $m \times g$ selection matrix, and η_t is a $g \times 1$ disturbance vector. Finally Q is a $g \times g$ variance covariance matrix.

Examples: local level model

Measurement equation:

$$y_t = \alpha_t + \varepsilon_t, \quad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2),$$

Transition equation:

$$\alpha_{t+1} = \alpha_t + \eta_t, \quad \eta_t \sim NID(0, \sigma_\eta^2),$$

which gives a random-walk plus noise model.

Examples: trend model

The model is

$$\begin{aligned}y_t &= \mu_t + \varepsilon_t \quad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2) \\ \mu_{t+1} &= \mu_t + \nu_t + \zeta_t \quad \zeta_t \sim NID(0, \sigma_\zeta^2) \\ \nu_{t+1} &= \nu_t + \xi_t \quad \xi_t \sim NID(0, \sigma_\xi^2)\end{aligned}$$

if $\sigma_\xi^2 = \sigma_\zeta^2 = 0$, then we get

$$\begin{aligned}y_t &= \mu_t + \varepsilon_t \quad t = 1, 2, \dots, T, \quad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2) \\ \mu_{t+1} &= \mu_t + \nu \quad t = 1, 2, \dots, T,\end{aligned}$$

which is a deterministic trend plus noise.

Examples: TVP models

Measurement equation:

$$y_t = X_t \alpha_t + \varepsilon_t \quad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2),$$

Transition equation:

$$\alpha_{t+1} = \alpha_t + \eta_t, \quad \eta_t \sim NID(0, Q),$$

X_t is a $1 \times g$ set of observable regressors at time t , the states in this case are the time-varying parameters. Note the matrix Q is a full $g \times g$ matrix of parameters that need to be estimated.

Examples: AR(2)

- The model is
 $y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \xi_t$ with $\xi_t \sim \mathcal{NID}(0, \sigma_\epsilon^2)$
- Can be put into state-space form: let $\alpha_t = \begin{pmatrix} y_t \\ y_{t-1} \end{pmatrix}$

Measurement equation:

$$y_t = (1, 0)\alpha_t \quad t = 1, 2, \dots, T,$$

Transition equation:

$$\begin{pmatrix} y_t \\ y_{t-1} \end{pmatrix} = \begin{pmatrix} \varphi_1 & \varphi_2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} y_{t-1} \\ y_{t-2} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \xi_t$$

Examples: AR(2) – alternative formulation

- The model is

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \xi_t \text{ with } \xi_t \sim \mathcal{NID}(0, \sigma_\epsilon^2)$$

- Can be put into state-space form: let $\alpha_t = \begin{pmatrix} y_t \\ \varphi_2 y_{t-1} \end{pmatrix}$

Measurement equation:

$$y_t = (1, 0)\alpha_t \quad t = 1, 2, \dots, T,$$

Transition equation:

$$\begin{pmatrix} y_t \\ \varphi_2 y_{t-1} \end{pmatrix} = \begin{pmatrix} \varphi_1 & 1 \\ \varphi_2 & 0 \end{pmatrix} \begin{pmatrix} y_{t-1} \\ \varphi_2 y_{t-2} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \xi_t$$

Examples: MA(1)

- The model is $y_t = \xi_t + \theta \xi_{t-1}$ with $\xi_t \sim \mathcal{NID}(0, \sigma_\epsilon^2)$
- Can be put into state-space form: let $\alpha_t = \begin{pmatrix} y_t \\ \theta \xi_t \end{pmatrix}$

Measurement equation:

$$y_t = (1, 0)\alpha_t \quad t = 1, 2, \dots, T,$$

Transition equation:

$$\begin{pmatrix} y_t \\ \theta \xi_t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_{t-1} \\ \theta \xi_{t-1} \end{pmatrix} + \begin{pmatrix} 1 \\ \theta \end{pmatrix} \xi_t$$

Examples: ARMA(1, 1)

- The model is
 $y_t = \varphi y_{t-1} + \xi_t + \theta \xi_{t-1}$ with $\xi_t \sim \mathcal{NID}(0, \sigma_\epsilon^2)$
- Can be put into state-space form: let $\alpha_t = \begin{pmatrix} y_t \\ \theta \xi_t \end{pmatrix}$

Measurement equation:

$$y_t = (1, 0)\alpha_t \quad t = 1, 2, \dots, T,$$

Transition equation:

$$\begin{pmatrix} y_t \\ \theta \xi_t \end{pmatrix} = \begin{pmatrix} \varphi & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_{t-1} \\ \theta \xi_{t-1} \end{pmatrix} + \begin{pmatrix} 1 \\ \theta \end{pmatrix} \xi_t$$

Examples: ARMA(p, q)

- The model is

$$y_t = \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} + \xi_t + \theta_1 \xi_{t-1} + \cdots + \theta_q \xi_{t-q} \text{ with } \xi_t \sim \mathcal{NID}(0, \sigma_\epsilon^2)$$

- Can be put into state-space form. Let $m = \max(p, q + 1)$ and re-write the ARMA(p,q) model as:

$$y_t = \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} + \xi_t + \theta_1 \xi_{t-1} + \cdots + \theta_{m-1} \xi_{t-m+1}$$

where some of the AR or MA coefficients will be zero unless $p = q + 1$.

- Define:

$$\alpha_t = \begin{pmatrix} y_t \\ \varphi_2 y_{t-1} + \cdots + \varphi_p y_{t-m+1} + \theta_1 \eta_t + \cdots + \theta_{m-1} \eta_{t-m+2} \\ \vdots \\ \varphi_m y_{t-1} + \theta_m \eta_t \end{pmatrix}$$

Examples: ARMA(p, q)

Measurement equation:

$$y_t = (1, \mathbf{0}_{m-1})\alpha_t \quad t = 1, 2, \dots, T,$$

Transition equation:

$$\alpha_t = \begin{pmatrix} \phi_1 & 1 & 0 & 0 & 0 \\ \phi_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{m-1} & 0 & 0 & \cdots & 1 \\ \phi_m & 0 & 0 & \cdots & 0 \end{pmatrix} \alpha_{t-1} + \begin{pmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{m-2} \\ \theta_{m-1} \end{pmatrix} \eta_t$$

Kalman filter

The Kalman filter is a recursive method, developed by Rudolf Kalman in 1960, to produce conditional expectations of the states variables given a linear Gaussian state-space.

- Kalman filter routine can be used to compute the log-likelihood function of the model when the parameters are unknown.
- Under Gaussianity and linearity the Kalman filter is the *optimal* filter, in the sense it produces the best estimate, (minimum variance), of the states.

Assumptions for standard Kalman Filter

The crucial assumptions for the Kalman filter method are:

- Linearity
- Gaussianity
- Independence between measurement errors and innovations to states. (This one can be removed after a proper reparameterization of the model.)

Deviations from these assumptions generate non-optimal filters and hence biases in the parameter estimates.

Intermezzo: the regression lemmas

Lemma 1

Consider the following two vectors, \mathbf{x} , and \mathbf{y} jointly normally distributed

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}; \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}$$

Then

$$E[\mathbf{x}|\mathbf{y}] = E[\mathbf{x}] + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\mathbf{y} - \boldsymbol{\mu}_y)$$
$$Var[\mathbf{x}|\mathbf{y}] = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$$

Intermezzo: the regression lemma2

Lemma 2

Consider the following three vectors, \mathbf{x} , \mathbf{y} , and \mathbf{z} jointly normally distributed

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} & \boldsymbol{\Sigma}_{xz} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} & \mathbf{0} \\ \boldsymbol{\Sigma}_{zx} & \mathbf{0} & \boldsymbol{\Sigma}_{zz} \end{bmatrix}; \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \\ \mathbf{0} \end{bmatrix}$$

In the multivariate normal regression we have that

$$\begin{aligned} E(\mathbf{x}|\mathbf{y}, \mathbf{z}) &= E(\mathbf{x}|\mathbf{y}) + \boldsymbol{\Sigma}_{xz}\boldsymbol{\Sigma}_{zz}^{-1}\mathbf{z} \\ Var(\mathbf{x}|\mathbf{y}, \mathbf{z}) &= Var(\mathbf{x}|\mathbf{y}) - \boldsymbol{\Sigma}_{xz}\boldsymbol{\Sigma}_{zz}^{-1}\boldsymbol{\Sigma}'_{xz} \end{aligned}$$

Derivation of the Kalman filter (local-level model)

The model is

$$\begin{aligned}y_t &= \alpha_t + \varepsilon_t, \quad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2) \\ \alpha_{t+1} &= \alpha_t + \eta_t, \quad \eta_t \sim NID(0, \sigma_\eta^2).\end{aligned}$$

The object of the filtering is to update the knowledge of the state each time a new observation is brought in. Indeed, conditional on the information set up to $t - 1$,

$$\alpha_t | Y_{t-1} \sim N(a_t, P_t) \tag{3}$$

If a_t and P_t are known then we can calculate a_{t+1} and P_{t+1} when y_t is brought in. In the local level model,

$$\begin{aligned}a_{t+1} &= E[\alpha_{t+1} | Y_t] = E[\alpha_t + \eta_t | Y_t] = E[\alpha_t | Y_t] \\ P_{t+1} &= Var[\alpha_{t+1} | Y_t] = Var[\alpha_t + \eta_t | Y_t] = Var[\alpha_t | Y_t] + \sigma_\eta^2\end{aligned}$$

The starting values a_1 and P_1 must be fixed.

Derivation of the Kalman filter (local-level model)

Define

$$v_t = y_t - a_t = y_t - E[\alpha_t | Y_{t-1}]$$

and $F_t = \text{Var}[v_t | Y_{t-1}]$ with $E[v_t | Y_{t-1}] = 0$ and $E[v_t y_{t-j}] = 0$ for $j = 1, \dots, t-1$.

Hence,

$$\begin{aligned} E[\alpha_t | Y_t] &= E[\alpha_t | Y_{t-1}, v_t] \\ \text{Var}[\alpha_t | Y_t] &= \text{Var}[\alpha_t | Y_{t-1}, v_t] \end{aligned}$$

Since all variables are normally distributed, the $E[\alpha_t | Y_t]$ and $\text{Var}[\alpha_t | Y_t]$ are given by standard formulae from multivariate normal regression theory.

Derivation of the Kalman filter (local-level model)

It follows that

$$E[\alpha_t | Y_t] = E(\alpha_t | Y_{t-1}) + Cov(\alpha_t, v_t) Var(v_t)^{-1} v_t$$

where

$$\begin{aligned} Cov[\alpha_t, v_t | Y_{t-1}] &= E[\alpha_t v_t | Y_{t-1}] = E[\alpha_t (\alpha_t + \epsilon_t - a_t) | Y_{t-1}] \\ &= E[\alpha_t^2 | Y_{t-1}] - E[\alpha_t | Y_{t-1}] a_t \\ &= E[\alpha_t^2 | Y_{t-1}] - a_t^2 = Var(\alpha_t | Y_{t-1}) = P_t. \end{aligned}$$

For the variance we have

$$\begin{aligned} F_t &= Var[v_t | Y_{t-1}] \\ &= Var[\alpha_t + \epsilon_t - a_t | Y_{t-1}] \\ &= Var[\alpha_t | Y_{t-1}] + Var(\epsilon_t) = P_t + \sigma_\epsilon^2 \end{aligned}$$

Derivation of the Kalman filter (local-level model)

Thus

$$E[\alpha_t | Y_t] = a_t + \frac{P_t}{F_t} v_t$$

where $K_t := \frac{P_t}{F_t}$ is the *Kalman gain*. Similarly

$$\begin{aligned} \textcolor{red}{Var}[\alpha_t | Y_t] &= Var[\alpha_t | Y_{t-1}] - Cov(\alpha_t, v_t | Y_{t-1})^2 Var[v_t | Y_{t-1}]^{-1} \\ &= P_t - \frac{P_t^2}{F_t} \\ &= P_t(1 - K_t) \end{aligned}$$

The Kalman filter for the local level model

Finally, the set of recursions of the Kalman filter for the local level model is

$$\begin{aligned}v_t &= y_t - a_t, & F_t &= P_t + \sigma_\epsilon^2 \\K_t &= \frac{P_t}{F_t} \\a_{t+1} &= a_t + K_t v_t, & P_{t+1} &= P_t(1 - K_t) + \sigma_\eta^2\end{aligned}$$

The Kalman filter (General Formula)

Finally, the set of recursions of the Kalman filter for the state space model

$$y_t = Z\alpha_t + D\varepsilon_t, \quad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2), \quad (4)$$

$$\alpha_{t+1} = T\alpha_t + H\eta_t, \quad \eta_t \sim NID(0, Q), \quad (5)$$

is

$$\begin{aligned} v_t &= y_t - Za_t, & F_t &= ZP_tZ' + \sigma_\varepsilon^2 DD', \\ K_t &= TP_tZ'F_t^{-1}, & L_t &= T - K_tZ \\ a_{t+1} &= Ta_t + K_tv_t, & P_{t+1} &= TP_tL_t' + HQH'. \end{aligned}$$

Constructing the log-likelihood

The log-likelihood function of a model represented in state-space form can be computed within the Kalman filter routine. Indeed, at each iteration, after the prediction step, we obtain the so called *one-step ahead prediction errors* and *one-step ahead prediction variance*:

$$\begin{aligned}v_t &= Y_t - Za_t \\F_t &= ZP_tZ' + \sigma_\epsilon^2 DD'\end{aligned}$$

We can compute the conditional Gaussian log-likelihood function, $\log \mathcal{L}_t = \sum_{t=1}^T \ell_t$, where ℓ_t is the likelihood contribution at time t :

$$\log \mathcal{L}_t = -\frac{T_p}{2} \log 2\pi - \frac{1}{2} \sum_{t=1}^T (\log |F_t| + v_t' F_t^{-1} v_t)$$

Smoothing

Smoothing means estimation of $\alpha_1, \dots, \alpha_N$ based on the entire sample.

The conditional density

$$\alpha_t | Y_N \sim N(\hat{\alpha}_t, V_t) \quad (6)$$

where the smoothed state

$$\hat{\alpha}_t = E[\alpha_t | Y_N]$$

and the smoothed state variance

$$V_t = \text{Var}[\alpha_t | Y_N]$$

The operation of calculating $\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_N$ is called state smoothing.

Smoothed state

- The one-step ahead errors v_1, \dots, v_N are mutually independent and are a linear transformation of y_1, \dots, y_N .
- The errors (v_t, \dots, v_N) are independent of (y_1, \dots, y_{t-1}) with zero means.
- When y_1, \dots, y_N are fixed, then Y_{t-1} (v_t, \dots, v_N) are fixed and viceversa.
- By the properties of the multivariate normal

$$E[x|y, z] = E[x|y] + \Sigma_{xz} \Sigma_{zz}^{-1} z \quad (7)$$

- It follows that $\hat{\alpha}_t = E[\alpha_t | Y_N]$ is given by

$$\begin{aligned}\hat{\alpha}_t &= E[\alpha_t | Y_N] = E[\alpha_t | Y_{t-1}, v_t, \dots, v_N] \\ &= E[\alpha_t | Y_{t-1}] + Cov[\alpha_t, (v_t, \dots, v_N)'] Var[(v_t, \dots, v_N)']^{-1} (v_t, \dots, v_N)' \\ &= a_t + \begin{bmatrix} Cov[\alpha_t, v_t] \\ Cov[\alpha_t, v_{t+1}] \\ \vdots \\ Cov[\alpha_t, v_N] \end{bmatrix} \begin{bmatrix} F_t & & & \\ & \ddots & & \\ & & F_N & \end{bmatrix}^{-1} \begin{bmatrix} v_t \\ v_{t+1} \\ \vdots \\ v_N \end{bmatrix}\end{aligned}$$

Smoothing

Hence, by the properties of the multivariate normal regression, we get

$$\hat{\alpha}_t = E[\alpha_t | Y_N] = E[\alpha_t | Y_{t-1}] + \sum_{j=t}^N \text{Cov}(\alpha_t, v_j) F_j^{-1} v_j$$

where

$$\begin{aligned}\text{Cov}(\alpha_t, v_t) &= P_t \\ \text{Cov}(\alpha_t, v_{t+1}) &= L_t P_t \\ \text{Cov}(\alpha_t, v_N) &= L_t L_{t+1} \dots L_{N-1} P_t\end{aligned}$$

Smoothing

Therefore, by substituting

$$\begin{aligned}\hat{\alpha}_t &= a_t + P_t \frac{v_t}{F_t} + P_t L_t \frac{v_{t+1}}{F_{t+1}} + \dots \\ &= a_t + P_t r_{t-1}\end{aligned}$$

where

$$r_{t-1} = \frac{v_t}{F_t} + L_t \frac{v_{t+1}}{F_{t+1}} + \dots + L_t L_{t+1} \dots L_{N-1} \frac{v_N}{F_N}$$

is a weighted sum of innovations after time $t - 1$ and needs to be computed by backward recursion (smoothing state recursion)

$$r_{t-1} = \frac{v_t}{F_t} + L_t r_t$$

Smoothing States Variance

With a similar argument, we can derive the recursive formula for the smoothed state variance, $V_t = \text{Var}[\alpha_t | Y_N]$.

$$\begin{aligned} V_t &= \text{Var}[\alpha_t | Y_{t-1}] - \text{Cov}[\alpha_t, (v_t, \dots, v_N)'] \text{Var}[(v_t, \dots, v_N)']^{-1} \text{Cov}[\alpha_t, (v_t, \dots, \\ &= P_t - \sum_{j=t}^N \text{Cov}(\alpha_t, v_j)^2 F_j^{-1} \end{aligned}$$

so that

$$V_t = P_t - P_t^2 N_{t-1}$$

and N_t is again given by the following backward recursion

$$\text{Var}(r_{t-1}) \equiv N_{t-1} = \frac{1}{F_t} - L_t^2 N_t$$

Given V_t , we can construct confidence intervals around the smoothed states $\hat{\alpha}_t$.

Missing Values

The Kalman filter in case of missing values

$$\nu_t = y_t - Za_t,$$

$$K_t = TP_t Z' F_t^{-1},$$

$$a_{t+1} = \begin{cases} Ta_t + K_t \eta_t, & \text{No MV} \\ Ta_t & \text{MV} \end{cases}$$

$$P_{t+1} = \begin{cases} TP_t L'_t + HQ_t H' & \text{No MV} \\ TP_t T' + HQ_t H' & \text{MV} \end{cases}$$

$$F_t = ZP_t Z' + \sigma_\varepsilon^2 DD',$$

$$L_t = T - K_t Z$$

Forecast

Forecasting is an operation that comes at no-costs from the Kalman filter. We regard forecasting as filtering observations $(y_1, \dots, y_t, y_{t+1}, \dots, y_{t+J})$ using the Kalman filter and treating the last J observations y_{t+1}, \dots, y_{t+J} as missing. The prediction of the states is

$$\begin{aligned}\bar{a}_{t+j|n} &= E(\alpha_{t+j} | Y_t) = T^j a_{t|t-1} \\ \bar{P}_{t+j|n} &= TP_{t|t-1} L'_t + HQ_t H'\end{aligned}$$

Estimating, Filtering and Smoothing Stochastic Volatility

Following Harvey et al. (1994) we have that the following ARSV model

$$\begin{aligned}r_t &= \sigma \exp(w_t/2) z_t \\w_t &= \rho w_{t-1} + \eta_t\end{aligned}$$

can be estimated with the Kalman filter by working with $\zeta_t = \log(r_t^2)$

$$\zeta_t = \mu + w_t + \epsilon_t$$

where $\mu = \log(\sigma^2) + E(\log(z_t^2))$, and ϵ_t is treated as $NID(0, \sigma_\epsilon^2)$. If the model is well specified, $\sigma_\epsilon^2 = \pi^2/2$.

The State-Space Form for ARSV

Following Harvey et al. (1994) we have the following ARSV model

$$\begin{aligned}\log(r_t^2) &= \mu + w_t + \epsilon_t \\ w_t &= \rho w_{t-1} + \eta_t\end{aligned}$$

which is an AR(1) plus noise process.

- The estimation can be carried out on $\bar{\zeta}_t = \log(r_t^2) - \hat{\mu}$ where $\hat{\mu}$ is the sample mean of $\log(r_t^2)$, such that we don't have to estimate σ .
- The parameter set is $\theta = [\rho, \sigma_\epsilon^2, \sigma_\eta^2]$.
- The parameter σ_ϵ^2 can also be restricted to $\pi^2/2$

PARTICLE FILTERING FOR THE LOG-NORMAL SV MODEL

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Filtering in State Space models

- 1) Filtering with linear and Gaussian State Space models is routinely done via a run of the Kalman filter.
- 2) Filtering with linear non-Gaussian State Space models can be still done with the Kalman filter, however filtered estimates are not optimal.
- 3) Filtering with non-linear (possibly) non-Gaussian state space models does not have a standard solution. Possibilities are:
 - 3a) Linearize the system and apply 2) as we have done with the SV model.
 - 3b) In some cases a modified Kalman filter can be developed, for instance: i) The Unscented Kalman Filter, and ii) the Extended Kalman Filter.
 - 3c) Develop a simulation based filtering procedure.

Today we focus on 3c) and apply the Bootstrap particle filter of Gordon et al. (1993) to the Log-Normal Stochastic Volatility model.

Law of Large Numbers

Let Y_1, Y_2, \dots, Y_n be a sequence of n iid random variables with finite mean $E[Y_1] = E[Y_2] = \dots = E[Y_n] = \mu$. Let

$$\bar{Y}_n = \frac{1}{n}(Y_1 + Y_2 + \dots + Y_n),$$

be the sample average of the sequence of random variables.
There are two versions of the Law of Large Numbers (LLN):

- The weak LLN implies that:

$$\lim_{n \rightarrow \infty} \bar{Y}_n \xrightarrow{P} \mu,$$

i.e. $\lim_{n \rightarrow \infty} P(|\bar{Y}_n - \mu| > \varepsilon) = 0$ for any $\varepsilon > 0$.

- The strong LLN implies that:

$$\lim_{n \rightarrow \infty} \bar{Y}_n \xrightarrow{a.s.} \mu,$$

i.e. $P(\lim_{n \rightarrow \infty} \bar{Y}_n = \mu) = 1$.

Importance Sampling

Suppose that the random variable Y is distributed with density $p(y)$. Assume we are interested in calculating some moment of Y , for instance $E[x(Y)]$ which we are not able to compute. Of course it is true that:

$$E[x(Y)] = \int_{y \in \mathcal{Y}} x(y)p(y)dy,$$

Evidently, if the dimension of Y is large (say 5 or 6) this integral cannot be computed numerically. By the law of large number we could for example approximate the integral by sampling random draws $y^{(i)}$, for $i = 1, \dots, N$, from the distribution of Y and calculate:

$$E[x(Y)] = \frac{1}{N} \sum_{i=1}^N x(y^{(i)}),$$

However, if sampling $y^{(i)}$ is not possible (or too costly from a computational point of view) Monte Carlo approximation cannot be done.

Importance Sampling

Consider now another density $g(y)$ which can be evaluated and is chosen to resemble $p(y)$ as closely as is reasonably possible while being easy to sample from. We can write:

$$\begin{aligned} E[x(Y)] &= \int_{y \in \mathcal{Y}} x(y)p(y)dy, \\ &= \int_{y \in \mathcal{Y}} x(y) \frac{p(y)}{g(y)} g(y)dy \\ &= E_g \left[x(y) \frac{p(y)}{g(y)} \right], \end{aligned}$$

where $E_g[\cdot]$ means that expectation is taken with respect to the **importance** density $g(y)$.

Importance Sampling

Since:

$$E[x(Y)] = E_g \left[x(y) \frac{p(y)}{g(y)} \right],$$

we can now approximate by Monte Carlo simulation $E_g \left[x(y) \frac{p(y)}{g(y)} \right]$, that is:

$$E_g \left[x(y) \frac{p(y)}{g(y)} \right] = \frac{1}{N} \sum_{i=1}^N x(y^{(i)}) \frac{p(y^{(i)})}{g(y^{(i)})},$$

where observations $y^{(i)}$ have been sampled according to the density $g(y)$.

Filtering in State Space models

Filtering in state space models means recovering the (unobserved) signal from the data. For instance in a SV model it means recovering the latent volatility from the data.

Consider the non-linear possibly non-Gaussian state space model:

$$p(y_t | Z(\alpha_t), \varepsilon_t), \quad \varepsilon_t \sim p(\varepsilon_t)$$
$$\alpha_{t+1} = T(\alpha_t) + R(\alpha_t)\eta_t, \quad \eta_t \sim p(\eta_t),$$

for $t = 1, 2, \dots$, where Z , T , and R are known functions of α_t and where ε_t and η_t are disturbance series. Here y_t represents the observed variable and α_t the (unobserved) state variable.

Filtering coincides with the evaluation of:

$$\bar{x}_t = E[x_t(\alpha_{1:t}) | y_{1:t}], \quad \text{for all } t \text{ and some function } x_t(\cdot).$$

Filtering in State Space models

Having a closer look at the filtered estimate $\hat{\alpha}_{1:t}$ reveals the main problem we face with nonlinear State Space models:

$$E[x_t(\alpha_{1:t})|y_{1:t}] = \int x_t(\alpha_{1:t}) p(\alpha_{1:t}|y_{1:t}) d\alpha_{1:t},$$

which is numerically unfeasible to solve.

Filtering by direct Importance Sampling

We can try to apply Importance Sampling in order to solve $E[x_t(\alpha_{1:t})|y_{1:t}]$. Consider an importance density $g(\alpha_{1:t}|y_{1:t})$, we can now write:

$$\begin{aligned} E[x_t(\alpha_{1:t})|y_{1:t}] &= \int x_t(\alpha_{1:t}) \frac{p(\alpha_{1:t}|y_{1:t})}{g(\alpha_{1:t}|y_{1:t})} g(\alpha_{1:t}|y_{1:t}) d\alpha_{1:t} \\ &= E_g \left[x_t(\alpha_{1:t}) \frac{p(\alpha_{1:t}|y_{1:t})}{g(\alpha_{1:t}|y_{1:t})} | y_{1:t} \right], \end{aligned}$$

where E_g denotes expectation with respect to the importance density $g(\alpha_{1:t}|y_{1:t})$.

Filtering by direct Importance Sampling

Since $p(\alpha_{1:t}, y_{1:t}) = p(\alpha_{1:t}|y_{1:t})p(y_{1:t})$ we can write:

$$\bar{x}_t = \frac{1}{p(y_{1:t})} E_g [x_t(\alpha_{1:t}) \tilde{\omega}_t | y_{1:t}],$$

where

$$\tilde{\omega}_t = \frac{p(\alpha_{1:t}, y_{1:t})}{g(\alpha_{1:t} | y_{1:t})}.$$

Note that generally we don't know how to evaluate $p(y_{1:t})$! However, setting $x_t(\alpha_{1:t}) = 1$ we see that:

$$1 = \frac{1}{p(y_{1:t})} E_g [\tilde{\omega}_t | y_{1:t}] \Rightarrow p(y_{1:t}) = E_g [\tilde{\omega}_t | y_{1:t}],$$

thus:

$$\bar{x}_t = \frac{E_g [x_t(\alpha_{1:t}) \tilde{\omega}_t | y_{1:t}]}{E_g [\tilde{\omega}_t | y_{1:t}]},$$

Filtering by direct Importance Sampling

Having a sample of draws $\alpha_{1:t}^{(1)}, \dots, \alpha_{1:t}^{(N)}$ from the importance density $g(\alpha_{1:t}|y_{1:t})$, we can approximate \bar{x}_t by:

$$\hat{x}_t = \frac{N^{-1} \sum_{i=1}^N x_t(\alpha_{1:t}^{(i)}) \tilde{\omega}_t^{(i)}}{N^{-1} \sum_{i=1}^N \tilde{\omega}_t^{(i)}},$$

that is:

$$\hat{x}_t = \sum_{i=1}^N x_t(\alpha_{1:t}^{(i)}) \omega_t^{(i)},$$

where:

$$\tilde{\omega}_t^{(i)} = \frac{p(\alpha_{1:t}^{(i)}, y_{1:t})}{g(\alpha_{1:t}^{(i)}|y_{1:t})}, \quad \omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$

The values $\tilde{\omega}_t^{(i)}$ are called “importance weights” and the values $\omega_t^{(i)}$ are called “normalised importance weights”. Note that $p(\alpha_{1:t}^{(i)}, y_{1:t}) = p(y_{1:t}|\alpha_{1:t}^{(i)})p(\alpha_{1:t}^{(i)})$ which is easy to compute.

Filtering by direct Importance Sampling

Consider the SV model:

$$y_t = \exp(\alpha_t/2)\varepsilon_t$$
$$\alpha_t = \omega + \phi\alpha_{t-1} + \tau\eta_t,$$

where ε_t and η_t are uncorrelated iid normally distributed shocks.

Suppose we have consistently estimated ω , ϕ , and τ by the GMM or QML estimators using a sample of T observations.

We now want to recover $\hat{\sigma}_t = E[\exp(\alpha_t/2)|y_{1:t}]$ for all $t = 1, \dots, T$.

Filtering by direct Importance Sampling

To apply the IS we specify $x_t(\alpha_{1:t}) = \exp(\alpha_t/2)$. For the importance density $g(\alpha_{1:t}|y_{1:t})$ we choose $p(\alpha_{1:t})$, where $p(\alpha_{1:t})$ is computed according to the SV model. We thus have:

$$\tilde{\omega}_t^{(i)} = \frac{p(\alpha_{1:t}^{(i)}, y_{1:t})}{g(\alpha_{1:t}^{(i)}|y_{1:t})} = \frac{p(\alpha_{1:t}^{(i)}, y_{1:t})}{p(\alpha_{1:t}^{(i)})} = \frac{p(y_{1:t}|\alpha_{1:t}^{(i)})p(\alpha_{1:t}^{(i)})}{p(\alpha_{1:t}^{(i)})} = p(y_{1:t}|\alpha_{1:t}^{(i)}),$$

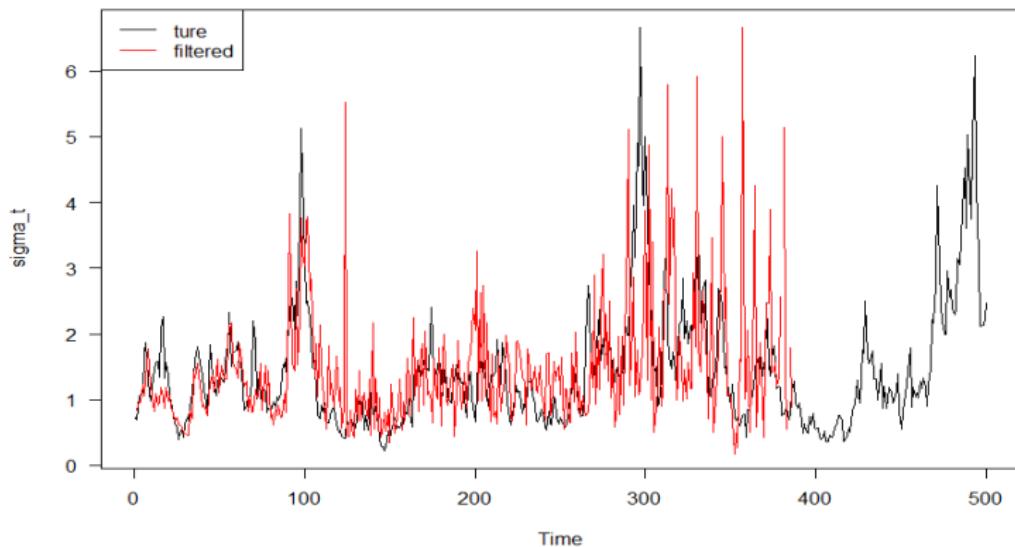
and

$$\omega_t^{(i)} = \frac{p(y_{1:t}|\alpha_{1:t}^{(i)})}{\sum_{j=1}^N p(y_{1:t}|\alpha_{1:t}^{(j)})}$$

Note that:

- 1) This is very demanding from a computational point of view.
- 2) Our importance density does not include the information contained in the data: $g(\alpha_{1:t}|y_{1:t}) = p(\alpha_{1:t})$. This means that the chosen importance density can be “far” from the target density thus reducing the efficiency of the sampler. Lower efficiency implies a higher number of draws N .

Filtering by direct Importance Sampling



Filtering of $\sigma_t = \exp(\alpha_t/2)$ with direct importance sampling using $g(\alpha_{1:t}|y_{1:t}) = p(\alpha_{1:t})$ and $N = 5000$. The algorithm broke before observation at time $t = 400$. Note how the filtering procedure becomes less accurate as long as t increases. The computational time is about 25 minutes.

Sequential Importance Sampling

To improve the simple method of filtering, it seems more natural in the context of filtering to retain the previous selection of $\alpha_{1:t-1}^{(i)}$ for each i and to confine the new sampling at time t to the selection of $\alpha_t^{(i)}$ only.

We call this sequential process for choosing $\alpha_{1:t}^{(i)}$ and the estimation based on it “particle filtering”; the resulting sets of values $\alpha_{1:t}^{(1)}, \dots, \alpha_{1:t}^{(N)}$ are called “particles”.

Thus, the i th particle at time t is defined by the relation:

$$\alpha_{1:t}^{(i)} = (\alpha_{1:t-1}^{(i)}, \alpha_t^{(i)}),$$

where $\alpha_{1:t-1}^{(i)}$ is the i th particle at time $t - 1$.

Sequential Importance Sampling

Consider the importance density $g(\alpha_{1:t}|y_{1:t})$. We can write:

$$\begin{aligned} g(\alpha_{1:t}|y_{1:t}) &= \frac{g(\alpha_{1:t}, y_{1:t})}{g(y_{1:t})} \\ &= \frac{g(\alpha_t|\alpha_{1:t-1}, y_{1:t})g(\alpha_{1:t-1}, y_{1:t})}{g(y_{1:t})} \\ &= g(\alpha_t|\alpha_{1:t-1}, y_{1:t})g(\alpha_{1:t-1}|y_{1:t}). \end{aligned}$$

Now suppose that $\alpha_{1:t-1}$ is selected using knowledge only of $y_{1:t-1}$. Moreover, given the realised values of $\alpha_{1:t-1}$ and $y_{1:t-1}$, the value of the observable y_t has already been selected by a process which does not depend on the simulated sequence $\alpha_{1:t-1}$. Under these circumstances, the density $g(\alpha_{1:t-1}|y_{1:t})$ is not affected by including y_t in its set of conditional variables $y_{1:t}$. Hence,
 $g(\alpha_{1:t-1}|y_{1:t}) \equiv g(\alpha_{1:t-1}|y_{1:t-1})$.

Sequential Importance Sampling

If $g(\alpha_{1:t-1}|y_{1:t}) \equiv g(\alpha_{1:t-1}|y_{1:t-1})$, then:

$$g(\alpha_{1:t}|y_{1:t}) = g(\alpha_t|\alpha_{1:t-1}, y_{1:t})g(\alpha_{1:t-1}|y_{1:t-1}),$$

such that we can specify an updating recursion from $g(\alpha_{1:t-1}|y_{1:t-1})$ to $g(\alpha_{1:t}|y_{1:t})$.

Note that it is assumed that sampling $\alpha_t^{(i)}$ from $g(\alpha_t|\alpha_{1:t-1}, y_{1:t})$ is practical and inexpensive.

Sequential Importance Sampling

We now develop the recursion for the importance weights $\tilde{\omega}_t^{(i)}$ which are required to approximate $E[x_t(\alpha_{1:t})|y_{1:t}]$.

We have that:

$$\begin{aligned}\tilde{\omega}_t &= \frac{p(\alpha_{1:t}, y_{1:t})}{g(\alpha_{1:t}|y_{1:t})} \\ &= \frac{p(\alpha_{1:t-1}, y_{1:t-1})p(\alpha_t, y_t|\alpha_{1:t-1}, y_{1:t-1})}{g(\alpha_{1:t-1}|y_{1:t-1})g(\alpha_t|\alpha_{1:t-1}, y_{1:t})},\end{aligned}$$

Due to the Markovian nature of the model we are considering we have:

$$p(\alpha_t, y_t|\alpha_{1:t-1}, y_{1:t-1}) = p(\alpha_t|\alpha_{t-1})p(y_t|\alpha_t),$$

such that for the i th particle we have:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} \frac{p(\alpha_t^{(i)}|\alpha_{t-1}^{(i)})p(y_t|\alpha_t^{(i)})}{g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y_{1:t})}, \quad \text{and} \quad \omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$

Sequential Importance Sampling

The recursion is initialized by $\tilde{\omega}_0^{(i)} = 1$ for $i = 1, \dots, N$. In most practical situation we have that $x_t(\alpha_{1:t}) = x(\alpha_t)$, for example $x(\alpha_t) = \exp(\alpha_t/2)$ in the case of volatility filtering. Let's continue with this convention. To filter $\exp(\alpha_t/2)$ we apply the Monte Carlo approximation:

$$\widehat{\exp(\alpha_t/2)} = \sum_{i=1}^N \exp(\alpha_t^{(i)}/2) \omega_t^{(i)},$$

where $\alpha_t^{(i)}$ is selected from the importance density $g(\alpha_t | \alpha_{1:t-1}, y_{1:t})$.

Importance sampling based on this approach is called “sequential importance sampling” (SIS). It was originally developed by Hammersley and Morton (1954) and applied to state space models by Handschin and Mayne (1969) and Handschin (1970).

Sequential Importance Sampling: Algorithm

(i) Sample α_t : draw N values $\alpha_t^{(i)}$ from $g(\alpha_t | \alpha_{t-1}^{(i)}, y_{1:t})$ and store $\alpha_{t-1:t}^{(i)} = \{\alpha_{t-1}^{(i)}, \alpha_t^{(i)}\}$.

(ii) Weights: compute the corresponding weights $\tilde{\omega}_t^{(i)}$:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} \frac{p(\alpha_t^{(i)} | \alpha_{t-1}^{(i)}) p(y_t | \alpha_t^{(i)})}{g(\alpha_t^{(i)} | \alpha_{1:t-1}^{(i)}, y_{1:t})}, \quad i = 1, \dots, N,$$

and normalize the weights to obtain $\omega_t^{(i)}$.

(iii) Compute the variable of interest $x_t(\alpha_t)$: given the set of particles $\{\alpha_t^{(1)}, \dots, \alpha_t^{(N)}\}$, compute

$$\widehat{x_t(\alpha_t)} = \sum_{i=1}^N x_t(\alpha_t^{(i)}) \omega_t^{(i)}.$$

Degeneracy

As t increases, the distribution of the weights $\omega_t^{(i)}$ based on the recursion:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} \frac{p(\alpha_t^{(i)} | \alpha_{t-1}^{(i)}) p(y_t | \alpha_t^{(i)})}{g(\alpha_t^{(i)} | \alpha_{1:t-1}^{(i)}, y_{1:t})}, \quad \text{and} \quad \omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}},$$

becomes highly skewed. It is possible for all but one particle to have negligible weights, for t large. We then say that the sample has become "degenerate".

The problem of degeneracy typically occurs when the likelihood function $p(y_t | \alpha_t)$ is highly peaked relatively to the density $p(\alpha_t | \alpha_{t-1})$, with the effect that few of the values $\alpha_t^{(1)}, \dots, \alpha_t^{(N)}$ lead to non-negligible values of $\omega_t^{(i)}$. It is obviously wasteful to retain particles in the recursion that are contributing negligible weights to the estimate $E[x(\alpha_t) | y_{1:t}]$.

Resampling

A way to combat degeneracy is to resample N particles $\tilde{\alpha}_t^{(i)}$, $i = 1, \dots, N$ with replacement from the set of particles $\{\alpha_t^{(1)}, \dots, \alpha_t^{(N)}\}$ according to probabilities $\omega_t^{(1)}, \dots, \omega_t^{(N)}$ at the end of each iteration of the Sequential Importance Sampling.

Although resampling is a technique that combats degeneracy in particle filtering, it clearly introduces additional Monte Carlo variation into the estimate of $E[x(\alpha_t)|y_{1:t}]$. A solution to mitigate the increased variability is to resample only at some iterations of the algorithm, see Liu and Chen (1998) for further details.

Sequential Importance Sampling algorithms that incorporates a resampling step are called “Sequential Importance Sampling Resampling” (SISR).

After a step of resampling the importance weights are set to $\tilde{\omega}_t^{(i)} = 1$ for all i .

Sequential Importance Sampling Resampling: Algorithm

- (i) Sample α_t : draw N values $\alpha_t^{(i)}$ from $g(\alpha_t | \tilde{\alpha}_{t-1}^{(i)}, y_{1:t})$ and store $\alpha_{t-1:t} = \{\tilde{\alpha}_{t-1}^{(i)}, \alpha_t^{(i)}\}$.
- (ii) Weights: compute the corresponding weights $\tilde{\omega}_t^{(i)}$:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} \frac{p(\alpha_t^{(i)} | \tilde{\alpha}_{t-1}^{(i)}) p(y_t | \alpha_t^{(i)})}{g(\alpha_t^{(i)} | \tilde{\alpha}_{1:t-1}^{(i)}, y_{1:t})}, \quad i = 1, \dots, N,$$

and normalize the weights to obtain $\omega_t^{(i)}$. (note that $\tilde{\omega}_{t-1}^{(i)} = 1$)

- (iii) Compute the variable of interest $x_t(\alpha_t)$: given the set of particles $\{\alpha_t^{(1)}, \dots, \alpha_t^{(N)}\}$, compute

$$\widehat{x_t(\alpha_t)} = \sum_{i=1}^N x_t(\alpha_t^{(i)}) \omega_t^{(i)}.$$

- (iv) Resample: draw N new independent particles $\tilde{\alpha}_t^{(i)}$ from $\{\alpha_t^{(1)}, \dots, \alpha_t^{(N)}\}$ with replacement and with corresponding probabilities $\{\omega_t^{(1)}, \dots, \omega_t^{(N)}\}$. Set $\tilde{\omega}_t^{(i)} = 1$ for all i .

The bootstrap particle filter

The first particle filter to be developed was the bootstrap filter of Gordon et al. (1993). It is sometimes called “Sampling Importance Resampling” algorithm.

The bootstrap filter uses the proposal distribution
 $g(\alpha_t^{(i)} | \tilde{\alpha}_{1:t-1}^{(i)}, y_{1:t}) = p(\alpha_t^{(i)} | \tilde{\alpha}_{t-1}^{(i)})$. This implies that:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} p(y_t | \alpha_t^{(i)}),$$

in the step (ii) of the Sequential Importance Sampling Resampling algorithm.

At first sight this looks crude since it neglects relevant information in y_t but when used with resampling and with N large enough, it can work well in many cases of interest and it is widely used in practice.

The bootstrap particle filter: Algorithm

- (i) Sample α_t : draw N values $\alpha_t^{(i)}$ from $p(\alpha_t | \tilde{\alpha}_{t-1}^{(i)})$ and store $\alpha_{t-1:t}^{(i)} = \{\tilde{\alpha}_{t-1}^{(i)}, \alpha_t^{(i)}\}$.
- (ii) Weights: compute the corresponding weights $\tilde{\omega}_t^{(i)}$:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} p(y_t | \alpha_t^{(i)}), \quad i = 1, \dots, N,$$

and normalize the weights to obtain $\omega_t^{(i)}$. (note that $\tilde{\omega}_{t-1}^{(i)} = 1$)

- (iii) Compute the variable of interest $x_t(\alpha_t)$: given the set of particles $\{\alpha_t^{(1)}, \dots, \alpha_t^{(N)}\}$, compute

$$\widehat{x_t(\alpha_t)} = \sum_{i=1}^N x_t(\alpha_t^{(i)}) \omega_t^{(i)}.$$

- (iv) Resample: draw N new independent particles $\tilde{\alpha}_t^{(i)}$ from $\{\alpha_t^{(1)}, \dots, \alpha_t^{(N)}\}$ with replacement and with corresponding probabilities $\{\omega_t^{(1)}, \dots, \omega_t^{(N)}\}$. Set $\tilde{\omega}_t^{(i)} = 1$ for all i .

The bootstrap particle filter: Algorithm

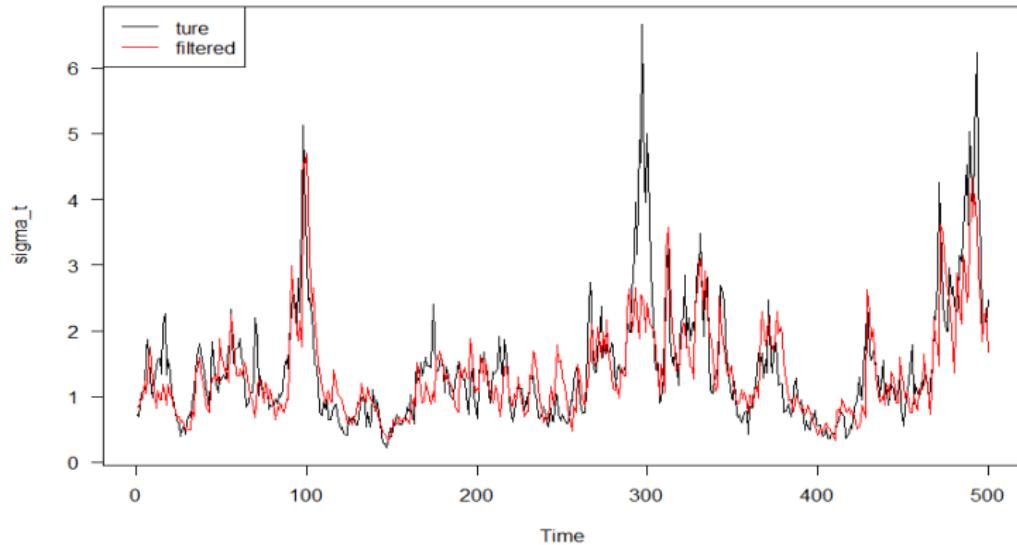
Note that, since we are resampling at each iteration step (ii) of the Bootstrap algorithm coincides with $\tilde{\omega}_t^{(i)} = p(y_t | \alpha_t^{(i)})$ (because $\tilde{\omega}_{t-1}^{(i)} = 1$).

Since resampling at each iteration increases the Monte Carlo variation in the computation of $E[x_t(\alpha_{1:t}|y_{1:t})]$, Carpenter et al. (1999) and the Liu and Chen (1998) have proposed different resampling schemes. The most popular one is based on the evaluation of the *effective sample size*:

$$ESS = \left(\sum_{i=1}^N \omega_t^{(i)2} \right)^{-1}.$$

Then resampling occurs only when $ESS < gN$ with $g = 0.75$ or $g = 0.5$.

Filtering with the bootstrap particle filter



Filtering of $\sigma_t = \exp(\alpha_t/2)$ with the bootstrap particle filter and $N = 10000$. The computational time is about 1.6 seconds.

Extensions

There are many ways to improve the filtering procedure of a non-linear non-Gaussian state space models. Most of these extensions aim at finding a better importance distribution $g(\alpha_t | \alpha_{1:t-1}, y_{1:t})$. Further readings can be find in Durbin and Koopman (2012), Creal (2012), Andrieu et al. (2001), and Doucet et al. (2001).

References I

- Andrieu, C., Doucet, A., and Punskaia, E. (2001). Sequential monte carlo methods for optimal filtering. In *Sequential Monte Carlo Methods in Practice*, pages 79–95. Springer.
- Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7.
- Creal, D. (2012). A survey of sequential monte carlo methods for economics and finance. *Econometric reviews*, 31(3):245–296.
- Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.
- Durbin, J. and Koopman, S. J. (2012). *Time series analysis by state space methods*, volume 38. Oxford University Press.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET.

References II

- Hammersley, J. M. and Morton, K. W. (1954). Poor man's monte carlo. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 23–38.
- Handschin, J. (1970). Monte carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6(4):555–563.
- Handschin, J. E. and Mayne, D. Q. (1969). Monte carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International journal of control*, 9(5):547–559.
- Liu, J. S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044.

GENERALIZED AUTOREGRESSIVE SCORE MODELS

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Time Varying Parameter models

Time varying parameter models are widely used for filtering and prediction of quantities of interest in finance such as:

- The volatility of a stock
- The number of trades in a given time period
- The waiting time before the next trade of a stock
- Changes in the risk level of a financial instrument

What we usually do is:

- 1) Formulate a parametric model which describes the evolution of the underlying stochastic process.
- 2) Estimate the model
- 3) Draw our conclusions

Today we focus on 1).

Types of Parametric Models

Sir David Cox in a very influential classical paper (Cox, 1981) classified dynamic models in two categories:

- Parameter driven models
- Observation driven models

Let ψ_t be the parameter of interest at time t , and let Y_t be an observed random variable at time t , define $\mathbf{Y}_{1:t-1} = (Y_{t-1}, Y_{t-2}, \dots)$. Even though Cox used a different formulation, nowadays we say that:

- When $\psi_t = \psi(g(\mathbf{Y}_{1:t-1}))$ that is ψ_t is known with certainty given $\mathcal{F}_{t-1} = \sigma(\mathbf{Y}_{1:t-1})$, we have an observation driven model.
- When $\psi_t = \psi^\dagger(\psi_{t-1}, \eta_t^\dagger)$ and η_t^\dagger is not known with certainty given \mathcal{F}_{t-1} we have a parameter driven model.

Types of Parametric Models

- Example of observation driven models are:
 - GARCH models (Engle, 1982; Bollerslev, 1987)
 - ACD models (Engle and Russell, 1998)
 - Multiplicative Error Models (Engle and Gallo, 2006)
 - Poisson Autoregression (Fokianos et al., 2009)
 - Time Varying Copulas (Patton, 2006)
 - ...
- Example of parameter driven models are:
 - SV models (Taylor, 1986)
 - DSGE models (DeJong and Dave, 2011)
 - Dynamic Linear Models (West and Harrison, 1997)
 - Hidden Markov Models (Frühwirth-Schnatter, 2006)
 - Stochastic Copula Models (Hafner and Manner, 2012)
 - General Non-linear Non-Gaussian State Space models (Durbin and Koopman, 2012).
 - ...

Types of Parametric Models

Most of the times we end up with the following characteristics for the two class of models:

- Observation driven models:
 - The conditional density $p(y_t | \mathbf{y}_{1:t-1})$, is usually available in closed form:
 - The likelihood is available via the prediction error decomposition.
 - One step ahead predictions are easy to compute.
 - Model tractability is easy.
 - Frequentist inference.
- Parameter driven models:
 - The conditional density $p(y_t | \mathbf{y}_{1:t-1})$, is available after integration of the latent parameter $p(y_t | \mathbf{y}_{1:t-1}) = \int_{\Psi} p(y_t, \psi_t | \mathbf{y}_{1:t-1}) d\psi_t$:
 - The likelihood is difficult to obtain and sometimes needs to be simulated.
 - Predictions are generally available only via simulation.
 - Model tractability is difficult.
 - Bayesian inference.

Observation driven models

The usual critiques to observations driven models are:

- They are too simplistic.
- Do not adequately describe the real properties of the underlying true data generating process.
- Assume that parameters at time t are deterministic given \mathcal{F}_{t-1} .

The general problem is that the filter constructed for the parameter(s) of interest most of the times lack of any statistical reasoning.

Assume that we have a good reason to specify $\psi_t = c_t + g(y_{t-1})$, where c_t is \mathcal{F}_{t-1} -measurable. The econometrician faces the problem: How to select $g(\cdot)$? The answer is generally: Lets try: $y_{t-1}, y_{t-1}^2, |y_{t-1}|, \dots$, and pick the one which allows me to: a) derive some theory, b) find reasonable filtered values for ψ_t .

Observation driven models

Of course the econometrician does not only try different parameterization of $g(y_{t-1})$, sometimes she searches for a “correct” functional which satisfies some moment conditions, for example $\mathbb{E}[g(Y_t)] = \mathbb{E}[\psi_t]$.

Consider for example the GARCH(1,1) model:

$$y_t = \sigma_t \varepsilon_t, \quad \varepsilon_t \sim \mathcal{D}(0, 1, \theta)$$
$$\sigma_t^2 = \omega + \alpha y_{t-1}^2 + \beta \sigma_{t-1}^2.$$

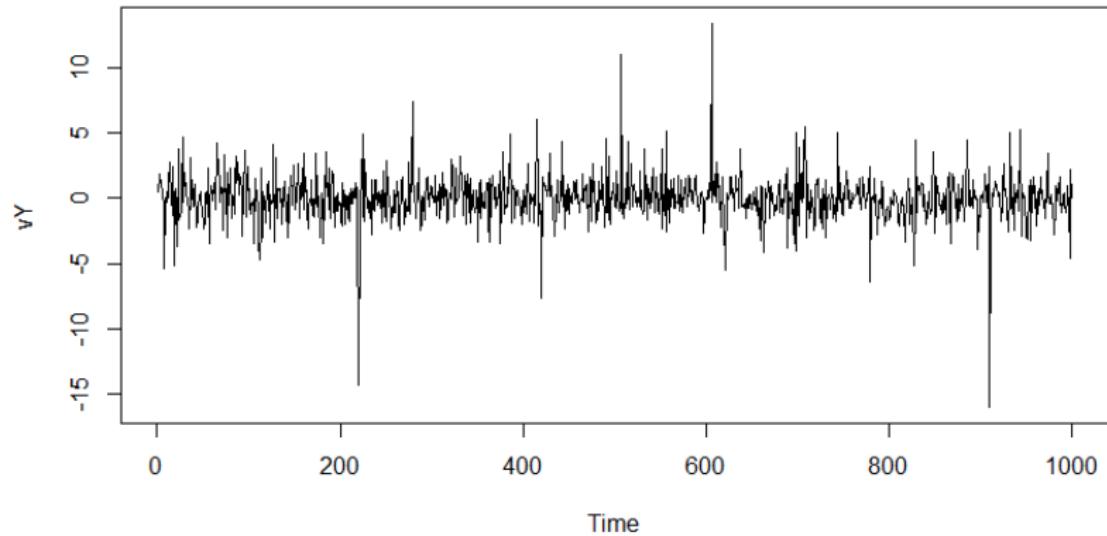
In this case $\mathbb{E}[Y_t^2] = \mathbb{E}[\sigma_t^2] = \sigma^2$ (provided that it exists).

Observation driven models

Few questions arises in this context:

- What if $\mathbb{E}[Y_t^2]$ is not defined?
- Is the *forcing variable* y_t^2 always adequate for σ_t^2 ? Is it always true that a large $|y_t|$ is signal of increased conditional variance?
- What if instead of the conditional variance σ_t^2 we have, say, the skewness parameter of a Skew–Student's t distribution? Which is in this case the “correct” forcing variable?

Observation driven models



One thousand iid draws from a Student's t distribution with $\nu = 3$

Generalized Autoregressive Score models

The class of Generalized Autoregressive Score (GAS) models provides coherent statistical answers to these questions.

GAS has been developed by Creal et al. (2013) and Harvey (2013). Harvey has named this class of models: Dynamic Conditional Score (DCS). He writes:

“Rather than the term dynamic conditional score (DCS) models, ..., Creal, Koopman and Lucas prefer the name generalized autoregressive score (GAS). However, despite the attraction of the acronym, the term “autoregressive” seems to me to convey a more limited dynamic structure than is actually the case.”, (Harvey, 2013).

The first articles about GAS are: Creal et al. (2008) and Harvey and Chakravarty (2008). The first use of the score to robustify the Kalman filter was originally proposed by Masreliez (1975).

Intermezzo: The Newton-Raphson method for root finding

Suppose the function f is differentiable with continuous derivative f' and a root a .

Let $x_0 \in \mathbb{R}$ our guess for a . Now the straight line through the point $(x_0, f(x_0))$ with slope $f'(x_0)$ is the best straight line approximation to the function $f(x)$ at the point x_0 . The equation of this straight line is given by:

$$y = f(x_0) + f'(x_0)(x - x_0) \quad (1)$$

This line crosses the x -axis at a point x_1 , which is a better approximation than x_0 to a . We find x_1 solving (1) for x with $y = 0$:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)},$$

i.e., we update our guess x_0 by a factor $\frac{f(x_0)}{f'(x_0)}$ which gives the direction to the root of the function.

Intermezzo: The Newton-Raphson method for root finding

If we iterate this procedure we end up with the following algorithm:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

Under conditions on f , f' and f'' it is possible to show that $x_{n+1} \rightarrow a$.

Since we are expecting $f(x_n) \rightarrow 0$, a good stopping condition for the Newton-Raphson algorithm is $|f(x_n)| \leq \varepsilon$ for some tolerance ε .

Intermezzo: The Newton-Raphson method for optimization

Since we deal with a problem of optimization we have $f'(x) = 0$. The Newton-Raphson method for optimization replaces $f(x)$ with $f'(x)$. Hence, the k -th Newton-Raphson step is

$$x_k = x_{k-1} - \frac{f'(x_{k-1})}{f''(x^{k-1})} \quad (2)$$

If $f(\cdot)$ is function of more than one variable, the step becomes

$$x_k = x_{k-1} - H^{-1}(x_{k-1}) \nabla f(x_{k-1}) \quad (3)$$

where $H^{-1}(x_{k-1})$ is the inverse of the Hessian evaluated in x_{k-1} and $\nabla f(x_{k-1})$ is a column-vector.

The NR method is such that the convergence is *local* and *quadratic*. Near the solution, the convergence is very fast! The NR method is really suitable when the first and second-order information are readily and easily calculated.

Generalized Autoregressive Score Models

Assume to observe the time-series $\mathbf{y}_{1:T} = (y_1, \dots, y_T)$, its joint density is given by:

$$p(\mathbf{y}_{1:T}; \psi) = p(y_1; \psi) \prod_{t=2}^T p(y_t | \mathbf{y}_{1:t-1}; \psi) \quad (4)$$

and the log likelihood for ψ reads:

$$\mathcal{L}(\psi; \mathbf{y}_{1:T}) = \log p(y_1; \psi) + \sum_{t=2}^T \log p(y_t | \mathbf{y}_{1:t-1}; \psi) \quad (5)$$

$$= \sum_{t=1}^T \ell_t, \quad (6)$$

hence, the log likelihood contribution of observation y_t is:

$$\ell_t = \log p(y_t | \mathbf{y}_{1:t-1}; \psi)$$

Generalized Autoregressive Score Models

Assume to specify an observation driven model for the time-series $\mathbf{y}_{1:T}$ parameterized in terms of the dynamic parameter $\psi_t = \psi(\mathbf{y}_{1:t-1})$.

GAS implements this filter as:

$$\begin{aligned}\psi_t &= \psi(\mathbf{y}_{1:t-1}) \\ &= \omega + \alpha u_{t-1} + \beta \psi_{t-1},\end{aligned}$$

where $u_t = S_t \nabla_t$ and:

$$\begin{aligned}\nabla_t &= \frac{\partial \log p(y_t | \mathbf{y}_{1:t-1}; \psi)}{\partial \psi_t} = \frac{\partial \ell_t}{\partial \psi_t} \\ S_t &= \mathcal{I}_t^{-1} = E_{t-1}[\nabla_t^2]^{-1}\end{aligned}$$

Generalized Autoregressive Score Models

The quantity $\nabla_t = \nabla(y_t; \psi_t)$ is the score of the conditional distribution $p(y_t | \mathbf{y}_{1:t-1}; \psi_t)$ and, as in the Newton–Raphson algorithm, gives the direction of the update.

The quantity $S_t = \mathcal{I}_t^{-1} = \mathcal{I}(\psi_t)^{-1}$ is the inverse of the Fisher information matrix of ψ_t evaluated with respect to the information set available at time $t - 1$. This quantity scales the score in order to account for the curvature of the likelihood at time t .

Note that ∇_t depends on y_t and the parameter value ψ_t . However, $\psi_t = \psi(\mathbf{y}_{1:t-1})$ such that $\nabla_t = \nabla(\mathbf{y}_{1:t})$, i.e., the forcing variable u_t is a (generally non-linear) function of the data $\mathbf{y}_{1:t}$. Differently, y_t has been integrated out in the computation of \mathcal{I}_t since the expectation is taken with respect to the information up to time $t - 1$.

Properties of the score

The score ∇_t has zero expectation:

$$\begin{aligned}\mathbb{E}_{t-1}[\nabla_t] &= \int_{\mathcal{Y}} \frac{\partial \log p(y_t | \mathbf{y}_{1:t-1}; \psi)}{\partial \psi_t} p(y_t | \mathbf{y}_{1:t-1}; \psi) dy \\ &= \int_{\mathcal{Y}} \frac{\partial p(y_t | \mathbf{y}_{1:t-1}; \psi)}{\partial \psi_t} \frac{1}{p(y_t | \mathbf{y}_{1:t-1}; \psi)} p(y_t | \mathbf{y}_{1:t-1}; \psi) dy \\ &= \int_{\mathcal{Y}} \frac{\partial p(y_t | \mathbf{y}_{1:t-1}; \psi)}{\partial \psi_t} dy \\ &= \frac{\partial}{\partial \psi_t} \int_{\mathcal{Y}} p(y_t | \mathbf{y}_{1:t-1}; \psi) dy \\ &= \frac{\partial}{\partial \psi_t} 1 = 0\end{aligned}$$

Provided that the Leibniz integral rule can be applied. It follows that $\{\nabla_{t-s}, s > 0\}$ forms a Martingale Difference Sequence.

Properties of the score

Since $\mathbb{E}_{t-1}[\nabla_t] = 0$, we have $Var(\nabla_t) = \mathbb{E}_{t-1}[\nabla_t^2] = \mathcal{I}_t(\psi_t)$, i.e., the Fisher information matrix $\mathcal{I}_t(\psi_t)$ is the variance of the score.

It follows that the forcing variable $u_t = \mathcal{I}_t^{-1}\nabla_t$ has zero expectation and variance \mathcal{I}_t^{-1}

Unfolding the process of ψ_t we obtain:

$$\psi_t = \frac{\omega}{1-\beta} + \alpha \sum_{s=0}^{\infty} \beta^s u_{t-s-1}$$

such that $\mathbb{E}[\psi_t] = \frac{\omega}{1-\beta}$ and

$$Var(\psi_t) = \alpha^2 \sum_{s=0}^{\infty} \beta^{2s} \mathcal{I}_{t-s-1}^{-1}$$

Since $\mathbb{E}[u_t u_{t-s}] = 0$ for all $s \neq 0$.

Generalizations of the scaling mechanism

The choice of $S_t = \mathcal{I}_t^{-1}$ is somehow arbitrary. More generally, it is possible to write $S_t = \mathcal{I}_t^{-d}$ for $d \in (0, 1/2, 1)$, that is:

- $d = 0$: No scaling $u_t = \nabla_t$
- $d = 1/2$: Inverse square root scaling $u_t = \mathcal{I}_t^{-1/2} \nabla_t$, such that $\text{Var}(u_t) = 1$.
- $d = 1$: Inverse scaling $u_t = \mathcal{I}_t^{-1} \nabla_t$

The parameter d can be selected using: i) likelihood criteria, ii) theory arguments, iii) computational reasons.

For example, \mathcal{I}_t is not always available in closed form. In this case $d = 0$ is a good strategy.

Handling parameter constraints

Most of the times we are in the situation when $\psi_t \in \Psi \subset \Re$. Think for example at ψ_t being the volatility at time t .

In these cases, we often work with: $\psi_t = \lambda(\tilde{\psi}_t)$, where $\tilde{\psi}_t \in \Re$ and $\lambda : \Re \rightarrow \Psi$ is a differentiable mapping function which is measurable with respect to the filtration $\mathcal{F}_{t-1} = \sigma(\mathbf{y}_{1:t-1})$.

We then rewrite the filter as:

$$\begin{aligned}\psi_t &= \lambda(\tilde{\psi}_t) \\ \tilde{\psi}_t &= \omega + \alpha \tilde{u}_{t-1} + \beta \tilde{\psi}_{t-1}\end{aligned}$$

Handling parameter constraints

After model reparametrization, the forcing variable needs to be modified as $\tilde{u}_t = \tilde{S}_t \tilde{\nabla}_t$. Assume $d = 1$ and set $\tilde{S}_t = \tilde{\mathcal{I}}_t^{-1}$. We have that:

$$\begin{aligned}\tilde{\nabla}_t &= \frac{\partial \log p(y_t | \mathbf{y}_{1:t-1}; \tilde{\psi}_t)}{\partial \tilde{\psi}_t} \\ \tilde{\mathcal{I}}_t &= \mathbb{E}_{t-1}[\tilde{\nabla}_t^2]\end{aligned}$$

It is evident that:

$$\begin{aligned}\tilde{\nabla}_t &= \frac{\partial \psi_t}{\partial \tilde{\psi}_t} \nabla_t \\ \tilde{\mathcal{I}}_t &= \frac{\partial \psi_t}{\partial \tilde{\psi}_t}^2 \mathcal{I}_t\end{aligned}$$

Handling parameter constraints

We have that for different choices of d , the jacobian of the mapping function $\frac{\partial \psi_t}{\partial \tilde{\psi}_t}$ affects the updating mechanism in different ways:

- $d = 0$: $\tilde{u}_t = \frac{\partial \psi_t}{\partial \tilde{\psi}_t} \nabla_t$.
- $d = 1/2$: $\tilde{u}_t = u_t$.
- $d = 1$: $\tilde{u}_t = \frac{\partial \psi_t}{\partial \tilde{\psi}_t}^{-1} u_t$

It follows that, handling parameter constraints with GAS models is straightforward and poses very little additional difficulties.

Volatility example under Gaussianity

Assume:

$$Y_t | \mathbf{y}_{t-1} \sim \mathcal{N}(0, \sigma_t^2),$$

The log pdf is proportional to:

$$\log p(y_t | \mathbf{y}_{t-1}; \sigma_t^2) \propto -\frac{1}{2} \left(\log(\sigma_t^2) + \frac{y_t^2}{\sigma_t^2} \right),$$

The score of σ_t^2 is:

$$\nabla_t = -\frac{1}{2\sigma_t^2} \left(\frac{y_t^2}{\sigma_t^2} - 1 \right)$$

The information matrix for σ_t^2 is:

$$\mathcal{I}_t = \frac{1}{2\sigma_t^4}$$

Volatility example under Gaussianity

If we set $d = 1$, in the case of the GAS model with Gaussian innovation and time-varying variance, we find $u_t = y_t^2 - \sigma_t^2$. The corresponding updating equation is:

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha(y_{t-1}^2 - \sigma_{t-1}^2) + \beta\sigma_{t-1}^2 \\ &= \omega + Ay_{t-1}^2 + B\sigma_{t-1}^2,\end{aligned}$$

where $A = \alpha$ and $B = \beta - \alpha$.

That is, a GAS model with Gaussian innovation, no mapping function and $d = 1$ resemble the GARCH(1,1) model. When $d \neq 1$ or we employ an exponential link function such as $\sigma_t = \exp(\tilde{\sigma}_t)$, the resulting model is different.

More of such special cases

- Exponential distribution (ACD and ACI): Engle and Russell (1998) and Russell (1999), respectively.
- Gamma distribution (MEM): Engle (2000) and Engle and Gallo (2006).
- Poisson: Davis et al. (2005).
- Multinomial distribution (ACM): Russell and Engle (2005).
- Binomial distribution: Cox (1958) and Rydberg and Shephard (2003)

Volatility example under Student's t

Assume $Y_t | \mathbf{y}_{1:t-1} \sim \mathcal{T}(\psi_t, \nu)$, where \mathcal{T} represents a Student's t distribution with scale $\psi_t > 0$ and $\nu > 0$ degrees of freedom.

The log density is proportional to:

$$\log p(y_t | \mathbf{y}_{1:t-1}; \psi_t, \nu) \propto -\log \psi_t + \frac{1}{\psi_t} \frac{(\nu + 1)y_t^2}{\nu \psi_t^2 + y_t^2}.$$

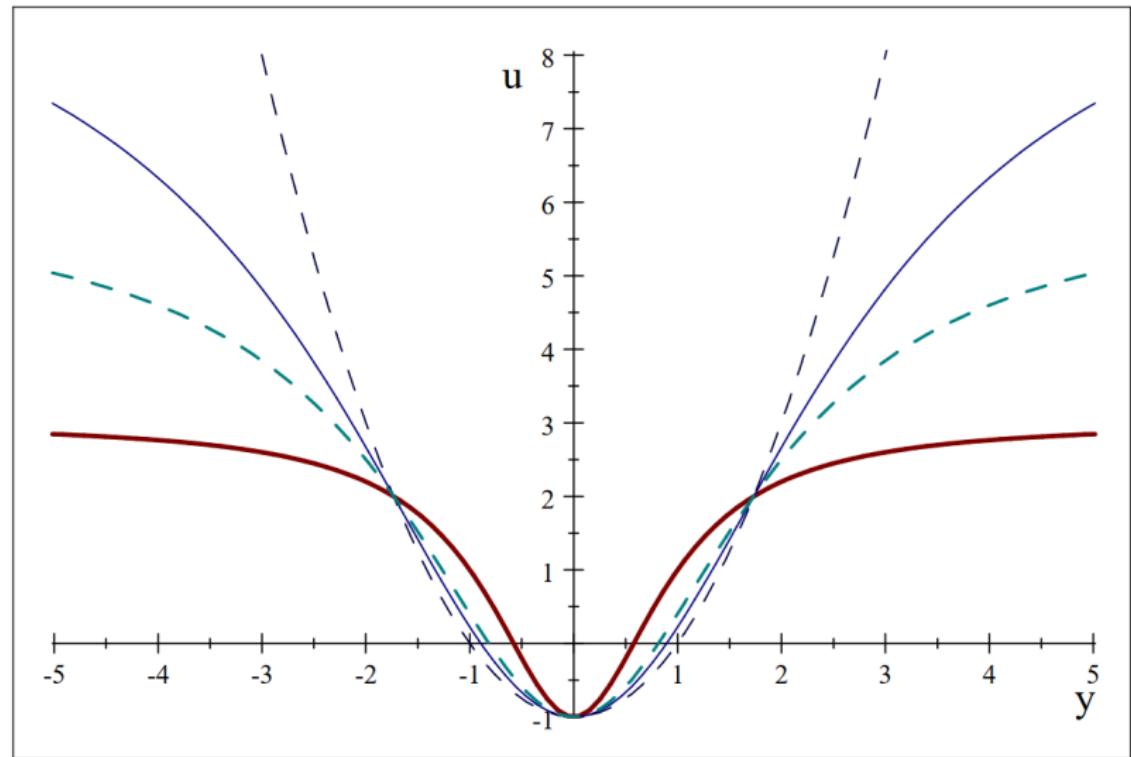
The score with respect to $\tilde{\psi}_t = \log(\psi_t)$ is:

$$\tilde{\nabla}_t = \frac{(\nu + 1)y_t^2}{\nu \psi_t^2 + y_t^2} - 1$$

The Fisher information matrix for $\tilde{\psi}_t$ is:

$$\tilde{\mathcal{I}}_t = \frac{2\nu}{\nu + 3}$$

Volatility example under Student's t



More about the volatility filter for the Student's t model

Taken from Ardia et al. (2017)

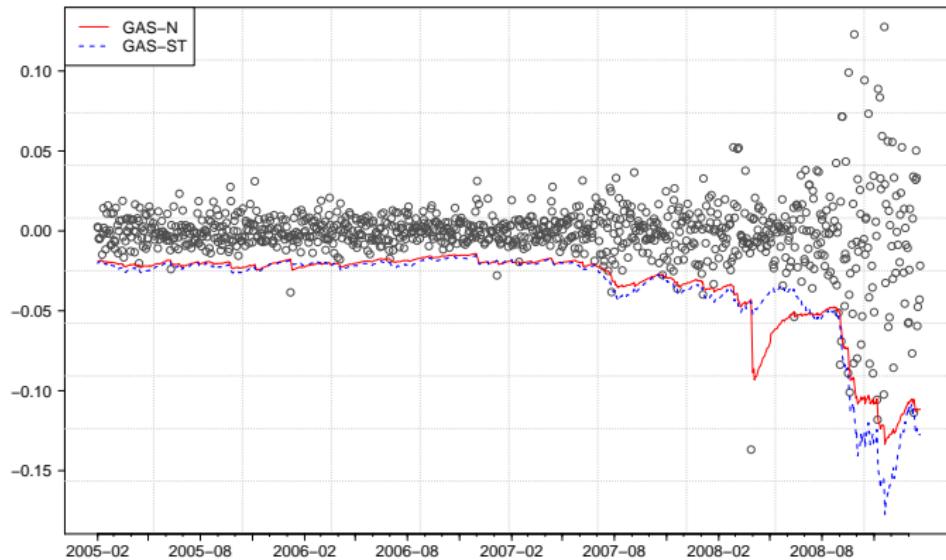


Figure: One-step ahead VaR forecasts for General Electric (GE) at the $\alpha = 1\%$ confidence level for the GAS- \mathcal{N} (solid) and GAS- ST (dotted) models.

More about the volatility filter for the Student's t model

Taken from Harvey (2013)

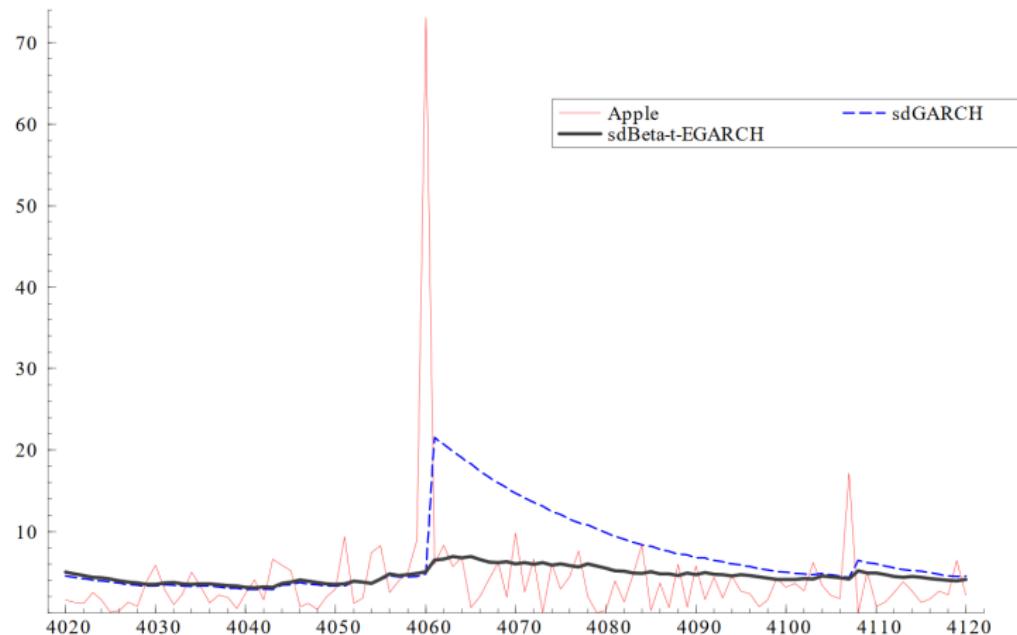


Figure: Absolute Apple returns and estimated volatility for GARCH and GAS with Student's t distribution.

More about the volatility filter for the Student's t model

Taken from www.gasmodel.com

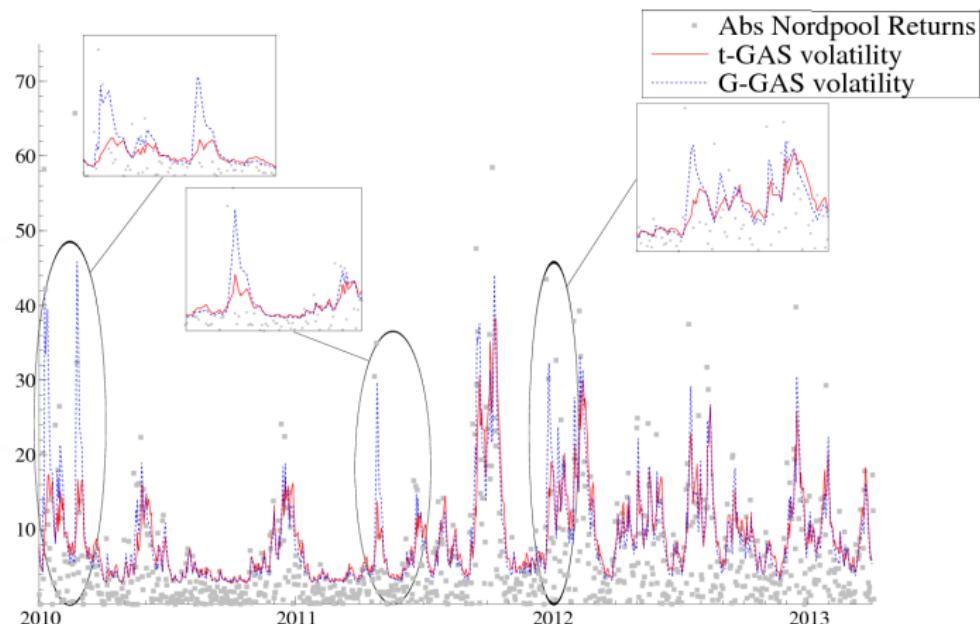


Figure: GAS estimated volatility for Nordpool electricity prices. Gaussian GAS is equivalent to Gaussian GARCH.

Estimation of GAS models

Estimation of GAS models can be done by Maximum Likelihood. For the Student's t volatility model let $\boldsymbol{\theta} = (\omega, \alpha, \beta, \nu)'$. The ML estimator for $\boldsymbol{\theta}$ is simply:

$$\hat{\boldsymbol{\theta}}^{ML} = \arg \max_{\boldsymbol{\theta} \in \Theta} \sum_{t=1}^T \log p(y_t | \mathbf{y}_{1:t-1}; \psi_t(\boldsymbol{\theta}), \nu) \quad (7)$$

We generally impose: $|\beta| < 1$, $\alpha > 0$, $\omega \in \Re$ and $\nu > 2$. However, this may vary depending on the particular application.

Estimation of GAS models can be done in the R computational environment using the **GAS** package!

The **GAS** package for R

The **GAS** package for **R** permits to: i) simulate, ii) estimate, and iii) make predictions using GAS models.

- **GAS** can deal with univariate and multivariate models.
- Mostly written in C++.
- Works in parallel.
- It is available from CRAN and GitHub.

The two papers: Ardia et al. (2016) and Ardia et al. (2017) describe the main functionalities.

GAS models are not only for financial applications!!! Visit www.gasmodel.com.

GAS models are not only for financial applications

Label	Name	Type	Parameters	Scaling Type
norm	Gaussian	univariate	location, scale	Identity, Inv, InvSqrt
snorm	Skew-Gaussian	univariate	location, scale, skewness	Identity
std	Student-t	univariate	location, scale, shape	Identity, Inv, InvSqrt
sstd	Skew-Student-t	univariate	location, scale, skewness, shape	Identity
ast	Asymmetric Student-t with two tail decay parameters	univariate	location, scale, skewness, shape, shape2	Identity, Inv, InvSqrt
ast1	Asymmetric Student-t with one tail decay parameter	univariate	location, scale, skewness, shape	Identity, Inv, InvSqrt
ald	Asymmetric Laplace Distribution	univariate	location, scale, skewness	Identity, Inv, InvSqrt
poi	Poisson	univariate	location	Identity, Inv, InvSqrt
negbin	Negative Binomial	univariate	location, scale	Identity, Inv, InvSqrt
ber	Bernoulli	univariate	location	Identity, Inv, InvSqrt
gamma	Gamma	univariate	scale, shape	Identity, Inv, InvSqrt
exp	Exponential	univariate	location	Identity, Inv, InvSqrt
beta	Beta	univariate	scale, shape	Identity, Inv, InvSqrt
mvnrm	Multivariate Gaussian	multivariate	location, scale, correlation	Identity
mvt	Multivariate Student-t	multivariate	location, scale, correlation, shape	Identity

Table: Statistical distributions for which the R package **GAS** provides the functionality to simulate, estimate and forecast the time-variation in its parameters.

Examples of distributions

Distribution	Density	Link function
Poisson	$\frac{\lambda_t^{y_t}}{y_t!} e^{-\lambda_t}$	$\lambda_t = \exp(\alpha_t)$
Neg. Binomial	$\frac{\Gamma(k_1+y_t)}{\Gamma(k_1)\Gamma(y_t+1)} \left(\frac{k_1}{k_1+\lambda_t}\right)^{k_1} \left(\frac{\lambda_t}{k_1+\lambda_t}\right)^{y_t}$	$\lambda_t = \exp(\alpha_t)$
Exponential	$\lambda_t e^{-\lambda_t y_t}$	$\lambda_t = \exp(\alpha_t)$
Gamma	$\frac{1}{\Gamma(k_1)\beta_t^{k_1}} y_t^{k_1-1} e^{-y_t/\beta_t}$	$\beta_t = \exp(\alpha_t)$
Weibull	$\frac{k_1}{\beta_t} \left(\frac{y_t}{\beta_t}\right)^{k_1-1} e^{-(y_t/\beta_t)^{k_1}}$	$\beta_t = \exp(\alpha_t)$
Gaussian vol	$\frac{1}{\sqrt{2\pi}\sigma_t} e^{-y_t^2/2\sigma_t^2}$	$\sigma_t^2 = \exp(\alpha_t)$
Student's <i>t</i> vol	$\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{(\nu-2)\pi}\Gamma(\frac{\nu}{2})\sigma_t} \left(1 + \frac{y_t^2}{(\nu-2)\sigma_t^2}\right)^{-\frac{\nu+1}{2}}$	$\sigma_t^2 = \exp(\alpha_t)$
Gaussian copula	$\frac{1}{2\pi\sqrt{1-\rho_t^2}} \exp\left[\frac{-z_{1t}^2+z_{2t}^2-2\rho_t z_{1t} z_{2t}}{2(1-\rho_t^2)}\right] \prod_{i=1}^2 \frac{1}{\sqrt{2\pi}} e^{-z_{it}^2/2}$	$\rho_t = \frac{1-\exp(-\alpha_t)}{1+\exp(-\alpha_t)}$
Student's <i>t</i> copula	$\frac{\Gamma(\frac{\nu+2}{2})\Gamma(\frac{\nu}{2})}{\Gamma(\frac{\nu+1}{2})} \frac{\frac{1}{\sqrt{1-\rho_t^2}} \left[1 + \frac{z_{1t}^2+z_{2t}^2-2\rho_t z_{1t} z_{2t}}{\nu(1-\rho_t^2)}\right]^{-\frac{\nu+2}{2}}}{\prod_{i=1}^2 (1+z_{it}/\nu)^{-\frac{\nu+1}{2}}}$	$\rho_t = \frac{1-\exp(-\alpha_t)}{1+\exp(-\alpha_t)}$

Examples of distributions

Distribution	GAS	ACM	
	$\nabla_t(\theta_t)$	$\mathcal{I}_t(\theta_t)$	s_t
Poisson	$\frac{y_t}{\lambda_t} - 1$	$\frac{1}{\lambda_t}$	y_t
Neg. Binomial	$\frac{y_t}{\lambda_t} - \frac{k_1 + y_t}{k_1 + \lambda_t}$	$\frac{k_1}{\lambda_t(k_1 + \lambda_t)}$	y_t
Exponential	$\frac{1}{\lambda_t} - y_t$	$\frac{1}{\lambda_t^2}$	y_t
Gamma	$\frac{y}{\theta_t^2} - \frac{k_1}{\beta_t}$	$\frac{k}{\beta_t^2}$	y_t / k_1
Weibull	$\frac{k_1}{\beta_t} \left[\left(\frac{y_t}{\beta_t} \right)^{k_1} - 1 \right]$	$\left(\frac{k_1}{\beta_t} \right)^2$	$\frac{y_t}{\Gamma(1+k_1^{-1})}$
Gaussian vol	$\frac{1}{2\sigma_t^2} \left(\frac{y_t^2}{\sigma_t^2} - 1 \right)$	$\frac{1}{2\sigma_t^4}$	y_t^2
Student's t vol	$\frac{1}{2\sigma_t^2} \left(\frac{\omega_t y_t^2}{\sigma_t^2} - 1 \right)$ $\omega_t = \frac{\nu+1}{(\nu-2)+y_t^2/\sigma_t^2}$	$\frac{\nu}{2(\nu+3)\sigma_t^4}$	y_t^2
Gaussian cop	$\frac{(1+\rho^2)(\hat{z}_{1,t} - \rho_t) - \rho_t(\hat{z}_{2,t} - 2)}{(1-\rho^2)^2}$	$\frac{1+\rho_t^2}{(1-\rho_t^2)^2}$	$z_{1,t} z_{2,t}$
Student's t cop	$\frac{(1+\rho^2)(\omega_t \hat{z}_{1,t} - \rho_t) - \rho_t(\omega_t \hat{z}_{2,t} - 2)}{(1-\rho^2)^2}$ $\omega_t = \frac{\nu+2}{\nu + \frac{\hat{z}_{2,t} - 2\rho_t \hat{z}_{1,t}}{1-\rho^2}}$	$\frac{(\nu+2+\nu\rho_t^2)}{(\nu+4)(1-\rho_t^2)^2}$	$z_{1,t} z_{2,t}$

References I

- Ardia, D., Boudt, K., and Catania, L. (2016). Generalized Autoregressive Score Models in R: The **GAS** Package. *Available at SSRN*: <https://ssrn.com/abstract=2871444>.
- Ardia, D., Boudt, K., and Catania, L. (2017). Value-at-Risk Prediction in R with the **GAS** Package. *Available at SSRN*: <https://ssrn.com/abstract=2871444>.
- Bollerslev, T. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *The review of economics and statistics*, pages 542–547.
- Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 215–242.
- Cox, D. R. (1981). Statistical analysis of time series: Some recent developments [with discussion and reply]. *Scandinavian Journal of Statistics*, pages 93–115.
- Creal, D., Koopman, S. J., and Lucas, A. (2013). Generalized autoregressive score models with applications. *Journal of Applied Econometrics*, 28(5):777–795.

References II

- Creal, D. D., Koopman, S. J., and Lucas, A. (2008). A general framework for observation driven time-varying parameter models.
- Davis, R. A., Dunsmuir, W. T., and Streett, S. B. (2005). Maximum likelihood estimation for an observation driven model for poisson counts. *Methodology and Computing in Applied Probability*, 7(2):149–159.
- DeJong, D. N. and Dave, C. (2011). *Structural macroeconomics*. Princeton University Press.
- Durbin, J. and Koopman, S. J. (2012). *Time series analysis by state space methods*, volume 38. OUP Oxford.
- Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007.
- Engle, R. F. (2000). The econometrics of ultra-high-frequency data. *Econometrica*, 68(1):1–22.

References III

- Engle, R. F. and Gallo, G. M. (2006). A multiple indicators model for volatility using intra-daily data. *Journal of Econometrics*, 131(1-2):3–27.
- Engle, R. F. and Russell, J. R. (1998). Autoregressive conditional duration: a new model for irregularly spaced transaction data. *Econometrica*, pages 1127–1162.
- Fokianos, K., Rahbek, A., and Tjostheim, D. (2009). Poisson autoregression. *Journal of the American Statistical Association*, 104(488):1430–1439.
- Frühwirth-Schnatter, S. (2006). *Finite mixture and Markov switching models*. Springer Science & Business Media.
- Hafner, C. M. and Manner, H. (2012). Dynamic stochastic copula models: Estimation, inference and applications. *Journal of Applied Econometrics*, 27(2):269–295.
- Harvey, A. C. (2013). *Dynamic models for volatility and heavy tails: with applications to financial and economic time series*, volume 52. Cambridge University Press.
- Harvey, A. C. and Chakravarty, T. (2008). Beta-t-(e) garch.

References IV

- Masreliez, C. (1975). Approximate non-gaussian filtering with linear state and observation relations. *IEEE Transactions on Automatic Control*, 20(1):107–110.
- Patton, A. J. (2006). Modelling asymmetric exchange rate dependence. *International economic review*, 47(2):527–556.
- Russell, J. R. (1999). Econometric modeling of multivariate irregularly-spaced high-frequency data.
- Russell, J. R. and Engle, R. F. (2005). A discrete-state continuous-time model of financial transactions prices and times: The autoregressive conditional multinomial–autoregressive conditional duration model. *Journal of Business & Economic Statistics*, 23(2):166–180.
- Rydberg, T. H. and Shephard, N. (2003). Dynamics of trade-by-trade price movements: decomposition and models. *Journal of Financial Econometrics*, 1(1):2–25.
- Taylor, S. J. (1986). *Modelling financial time series*. Wiley.
- West, M. and Harrison, W. (1997). *Bayesian forecasting and Dynamic Models*. Springer.

MULTIVARIATE VOLATILITY MODELLING

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Why modelling multivariate returns?

Economics and financial economics present problems whose solutions need the specification and estimation of a multivariate distribution.

- the standard portfolio allocation problem
- the risk management of a portfolio of assets
- pricing of derivative contracts based on a more than one underlying asset (e.g., Quanto options)
- Financial contagion (shocks transmission volatility and returns)

What and How

Stylized facts:

- Volatility clustering
- Dynamic covariances and dynamic correlations

Financial variables have time-dependent second order moments.

Parametric models:

- Multivariate GARCH models
- Multivariate Stochastic volatility models
- Multivariate realized volatility models

General multivariate volatility model

Vector of returns:

$$\mathbf{y}_t = (y_{1t}, \dots, y_{Nt})' \quad (N \times 1)$$

$$\mathbf{y}_t - \boldsymbol{\mu}_t = \boldsymbol{\epsilon}_t = H_t^{1/2} \mathbf{z}_t$$

Let $\{\mathbf{z}_t\}$ be a sequence of $(N \times 1)$ i.i.d. random vector with the following characteristics:

$$E[\mathbf{z}_t] = \mathbf{0}$$

$$E[\mathbf{z}_t \mathbf{z}_t'] = \mathbf{I}_N$$

So

$$\mathbf{z}_t \stackrel{iid}{\sim} G(\mathbf{0}, \mathbf{I}_N),$$

for some distribution G

General multivariate volatility model

$$E_{t-1}(\epsilon_t) = 0$$

$$\begin{aligned} E_{t-1}(\epsilon_t \epsilon'_t) &= \mathbf{H}_t \\ E(\epsilon_t \epsilon'_t) &= \Sigma \end{aligned}$$

$$E_{t-1}[\cdot] = E[\cdot | \mathcal{F}_{t-1}]$$

where \mathbf{H}_t is a matrix ($N \times N$) positive definite and measurable with respect to the information set \mathcal{F}_{t-1} , that is the σ -field generated by the past observations: $\{\epsilon_{t-1}, \epsilon_{t-2}, \dots\}$.

The correlation matrix:

$$\text{Corr}_{t-1}(\epsilon_t) = \mathbf{R}_t = \mathbf{D}_t^{-1/2} \mathbf{H}_t \mathbf{D}_t^{-1/2}$$

$$\mathbf{D}_t = \text{diag}(h_{11,t}, \dots, h_{NN,t})$$

Conditions on \mathbf{H}_t

MVMs provide a parametric structure for the dynamic evolution of \mathbf{H}_t . MVMs must satisfy:

- ① Diagonal elements of \mathbf{H}_t must be strictly positive;
- ② Positive definiteness of \mathbf{H}_t ;
- ③ Stationarity: $E[\mathbf{H}_t]$ exists, finite and constant w.r.t. t .

Ideal Multivariate volatility model

Ideal characteristics of a MVM:

- ① Estimation should be flexible for increasing N
- ② It should allow for covariance spillovers and feedbacks;
- ③ Coefficients should have an economic or financial interpretation

Different approaches

Three approaches for constructing multivariate GARCH models:

- ① direct generalizations of the univariate GARCH model of Bollerslev (1986); (VEC, BEKK)
- ② linear combinations of univariate GARCH models; ((generalized) orthogonal models and latent factor models.)
- ③ nonlinear combinations of univariate GARCH models; (constant and dynamic conditional correlation models, copula-GARCH models)

Targeting: a way to simplify estimation

GARCH(1,1):

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

Long-run variance (if $(\alpha + \beta) < 1$):

$$\sigma^2 = E[\sigma_t^2] = \omega(1 - \alpha - \beta)^{-1}$$

Variance targeting:

$$\sigma_t^2 = \hat{\sigma}^2(1 - \alpha - \beta) + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

$$\hat{\sigma}^2 = T^{-1} \sum_t \hat{\epsilon}_t^2$$

Introduction of targeting transforms the model estimation into a two-step estimation approach:

- ① $\hat{\sigma}^2$
- ② $\hat{\alpha}, \hat{\beta}$

Theory for this two step estimator is presented by Francq et al. (2011).

Different approaches

N assets: N variances + $\frac{1}{2}N(N - 1)$ covariances = $\frac{N}{2}(N + 1)$.

Two alternative approaches:

- Models of \mathbf{H}_t
- Models of \mathbf{D}_t and \mathbf{R}_t

The parametrization of \mathbf{H}_t as a multivariate GARCH, which means as a function of the information set \mathcal{F}_{t-1} , allows each element of \mathbf{H}_t to depend on q lagged of the squares and cross-products of ϵ_t , as well as p lagged values of the elements of \mathbf{H}_t . So the elements of the covariance matrix follow a vector of ARMA process in squares and cross-products of the disturbances.

Notation

Let **vech** denote the vector-half operator, which stacks the lower triangular elements of an $N \times N$ matrix as an $[N(N + 1)/2] \times 1$ vector.

Let \mathbf{A} be (2×2) , then $\text{vech}(\mathbf{A})$

$$\text{vech}(\mathbf{A}) = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{22} \end{bmatrix}$$

Since the conditional covariance matrix \mathbf{H}_t is symmetric, $\text{vech}(\mathbf{H}_t)$, which is of dimension $N(N + 1)/2$, contains all the unique elements in \mathbf{H}_t .

The VEC GARCH

A natural **multivariate extension** of the univariate GARCH(p,q) model is

$$\begin{aligned}\text{vech}(\mathbf{H}_t) &= \mathbf{W} + \sum_{i=1}^q \mathbf{A}_i^* \text{vech}(\boldsymbol{\epsilon}_{t-i} \boldsymbol{\epsilon}'_{t-i}) + \sum_{j=1}^p \mathbf{B}_j^* \text{vech}(\mathbf{H}_{t-j}) \\ &= \mathbf{W} + \mathbf{A}^*(L) \text{vech}(\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}'_t) + \mathbf{B}^*(L) \text{vech}(\mathbf{H}_t)\end{aligned}$$

$$\mathbf{A}^*(L) = \mathbf{A}_1^* L + \dots + \mathbf{A}_q^* L^q$$

$$\mathbf{B}^*(L) = \mathbf{B}_1^* L + \dots + \mathbf{B}_q^* L^p$$

The VEC GARCH: $N = 2$

$$N^* \equiv \frac{N(N+1)}{2}$$

$$\begin{aligned} \mathbf{W} & : [N(N+1)/2] \times 1 \\ \mathbf{A}_i^*, \quad \mathbf{B}_j^* & : [N^* \times N^*] \end{aligned}$$

$N = 2$, Vech-GARCH(1,1):

$$\begin{bmatrix} h_{11,t} \\ h_{21,t} \\ h_{22,t} \end{bmatrix} = \begin{bmatrix} w_1^* \\ w_2^* \\ w_3^* \end{bmatrix} + \begin{bmatrix} a_{11}^* & a_{12}^* & a_{13}^* \\ a_{21}^* & a_{22}^* & a_{23}^* \\ a_{31}^* & a_{32}^* & a_{33}^* \end{bmatrix} \begin{bmatrix} \epsilon_{1,t-1}^2 \\ \epsilon_{1,t-1}\epsilon_{2,t-1} \\ \epsilon_{2,t-1}^2 \end{bmatrix} + \begin{bmatrix} b_{11}^* & b_{12}^* & b_{13}^* \\ b_{21}^* & b_{22}^* & b_{23}^* \\ b_{31}^* & b_{32}^* & b_{33}^* \end{bmatrix} \begin{bmatrix} h_{11,t-1} \\ h_{21,t-1} \\ h_{22,t-1} \end{bmatrix}$$

The VEC GARCH

- This general formulation is termed *vec representation* by Engle and Kroner (1995).
- Earlier references can be found in Kraft and Engle (1982) and Bollerslev et al. (1992).
- The number of parameters is $\frac{N(N+1)}{2} \left[1 + (p+q) \frac{N(N+1)}{2} \right]$.
- Even for low dimensions of N and small values of p and q the number of parameters is very large; for $N = 5$ and $p = q = 1$ the unrestricted version of (1) contains 465 parameters.
- The number of parameters is of order $O(N^4)$: the *curse of dimensionality*.

For any parametrization to be sensible, we require that \mathbf{H}_t be positive definite for all values of ϵ_t in the sample space. Unfortunately, in the *vech* representation this restriction can be difficult to check, let alone impose during estimation.

Reduce the number of parameters: The Diagonal VEC

A natural restriction is the *diagonal representation*, in which each element of the covariance matrix depends only on past values of itself and past values of $\varepsilon_{jt}\varepsilon_{kt}$.

In the diagonal model the \mathbf{A}_i^* and \mathbf{B}_j^* matrices are all taken to be diagonal.

For $N = 2$ and $p = q = 1$, the diagonal model is written as:

$$\begin{bmatrix} h_{11,t} \\ h_{21,t} \\ h_{22,t} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} + \begin{bmatrix} a_{11}^* & 0 & 0 \\ 0 & a_{22}^* & 0 \\ 0 & 0 & a_{33}^* \end{bmatrix} \begin{bmatrix} \varepsilon_{1,t-1}^2 \\ \varepsilon_{1,t-1}\varepsilon_{2,t-1} \\ \varepsilon_{2,t-1}^2 \end{bmatrix} + \begin{bmatrix} b_{11}^* & 0 & 0 \\ 0 & b_{22}^* & 0 \\ 0 & 0 & b_{33}^* \end{bmatrix} \begin{bmatrix} h_{11,t-1} \\ h_{21,t-1} \\ h_{22,t-1} \end{bmatrix}$$

$$h_{ij,t} = w_i^* + a_{ii}^* \varepsilon_{i,t-1} \varepsilon_{j,t-1} + b_{ii}^* h_{ij,t-1}$$

This specification has been used by Bollerslev et al. (1988). Note that your book writes this as

$$\mathbf{H}_t = \boldsymbol{\Omega} + \sum_{i=1}^p A_i \odot (\varepsilon_{t-i} \varepsilon'_{t-i}) + \sum_{j=1}^q B_j \odot \mathbf{H}_{t-j}$$

BEKK model

Engle and Kroner (1995) propose a parametrization that imposes **positive definiteness restrictions**.

Consider the following model

$$\mathbf{H}_t = \mathbf{CC}' + \sum_{i=1}^q \mathbf{A}_i \boldsymbol{\epsilon}_{t-i} \boldsymbol{\epsilon}'_{t-i} \mathbf{A}'_i + \sum_{j=1}^p \mathbf{B}_j \mathbf{H}_{t-j} \mathbf{B}'_j \quad (1)$$

where \mathbf{C} , \mathbf{A}_i and \mathbf{B}_j are $(N \times N)$.

- The intercept matrix is decomposed into \mathbf{CC}' , where \mathbf{C} is a lower triangular matrix.
- Without any further assumption \mathbf{CC}' is positive semidefinite.
- This representation is general, it includes all positive definite diagonal representations and nearly all positive definite *vech* representations.

(The acronym BEKK stands for Baba, Engle, Kraft, and Kroner.)

BEKK model

Consider the simple BEKK(1,1) model:

$$\mathbf{H}_t = \mathbf{C}\mathbf{C}' + \mathbf{A}_1\boldsymbol{\epsilon}_{t-1}'\mathbf{A}_1' + \mathbf{B}_1\mathbf{H}_{t-1}\mathbf{B}_1' \quad (2)$$

BEKK identification (Engle and Kroner (1995))

Suppose that the diagonal elements in \mathbf{C} are restricted to be positive and that a_{11} and b_{11} are also restricted to be positive. Then there exists no other \mathbf{C} , \mathbf{A}_1 , \mathbf{B}_1 in the model (2) that will give an equivalent representation.

BEKK model

The purpose of the restrictions is to eliminate all other observationally equivalent structures.

For example, as relates to the term $\mathbf{A}_1 \epsilon_{t-1} \epsilon'_{t-1} \mathbf{A}'_1$ the only other observationally equivalent structure is obtained by replacing \mathbf{A}_1 by $-\mathbf{A}_1$. The restriction that a_{11} (b_{11}) be positive could be replaced with the condition that a_{ij} (b_{ij}) be positive for a given i and j , as this condition is also sufficient to eliminate $-\mathbf{A}_1$ from the set of admissible structures.

BEKK model, $N = 2$

MGARCH(1,1)-BEKK, $N = 2$:

$$\mathbf{H}_t = \mathbf{CC}' + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \varepsilon_{1t-1}^2 & \varepsilon_{1t-1}\varepsilon_{2t-1} \\ \varepsilon_{2t-1}\varepsilon_{1t-1} & \varepsilon_{2t-1}^2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}' + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} h_{11t-1} & h_{12t-1} \\ h_{21t-1} & h_{22t-1} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}'$$

Positive definiteness of H_t

BEKK-GARCH(p,q) model (Engle and Kroner (1995)):

Sufficient condition for positive definiteness of H_t

If $\mathbf{H}_0, \mathbf{H}_{-1}, \dots, \mathbf{H}_{-p+1}$ are all positive definite, then the BEKK parametrization yields a positive definite \mathbf{H}_t for all possible values of ε_t if \mathbf{C} is a full rank matrix or if any $\mathbf{B}_j j = 1, \dots, p$ is a full rank matrix.

BEKK: Three cases

① BEKK

$$\mathbf{H}_t = \Sigma + \mathbf{A}_1 (\epsilon_{t-1} \epsilon'_{t-1}) \mathbf{A}'_1 + \mathbf{B}_1 (\mathbf{H}_{t-1}) \mathbf{B}'_1$$

Number of parameters is $\frac{N(N+1)}{2} + 2N^2 = O(N^2) + O(N^2) = O(N^2)$.

② Diagonal BEKK

$$\mathbf{H}_t = \Sigma + \mathbf{A}_1 (\epsilon_{t-1} \epsilon'_{t-1}) \mathbf{A}'_1 + \mathbf{B}_1 (\mathbf{H}_{t-1}) \mathbf{B}'_1$$

where \mathbf{A}_1 and \mathbf{B}_1 are diagonal. Number of parameters is
 $\frac{N(N+1)}{2} + 2N = O(N^2) + O(N) = O(N^2)$.

③ Scalar BEKK

$$\mathbf{H}_t = \Sigma + \alpha (\epsilon_{t-1} \epsilon'_{t-1}) + \beta (\mathbf{H}_{t-1})$$

for $\alpha + \beta < 1$. Number of parameters is

$$\frac{N(N+1)}{2} + 2 = O(N^2) + O(1) = O(N^2).$$

BEKK: from $O(N^2)$ to $O(N)$ and $O(1)$

The same targeting approach used for univariate GARCH can be used with the BEKK model (assuming that a stationary solution exists).

For example, we can set

$$\Sigma = \left(I - \sum_{i=1}^p \mathbf{A}_i - \sum_{j=1}^q \mathbf{B}_j \right) \hat{\mathbf{S}},$$

where $\hat{\mathbf{S}} = \frac{1}{T} \sum_{t=1}^T \epsilon_t \epsilon_t'$. This is discussed by Engle and Mezrich (1996).

The estimation procedure then reduces to:

- 1) Estimate the empirical covariance of ϵ_t , $\hat{\mathbf{S}}$.
- 2) Estimate the other parameters of the model fixing $\mathbf{S} = \hat{\mathbf{S}}$

In this way, the second estimation step (which is a numerical optimization) is of order

- 1) $O(N^2)$ for the unrestricted BEKK.
- 2) $O(N)$ for the diagonal BEKK.
- 3) $O(1)$ for the scalar BEKK.

Some theory for this estimation procedure is discussed in Francq et al. (2016).

Estimation by ML

Let θ be the vector that contains all the parameters of the model. The log-likelihood function for θ given a sequence of T observations $\{\epsilon_1, \dots, \epsilon_T\}$ obtained under the assumption of conditional multivariate normality is:

$$\log L_T(\theta; \epsilon_1, \dots, \epsilon_T) = -\frac{1}{2} \left[TN \log(2\pi) + \sum_{t=1}^T \left(\log |\mathbf{H}_t| + \epsilon_t' \mathbf{H}_t^{-1} \epsilon_t \right) \right]$$

- The assumption of conditional normality can be quite restrictive.
- The symmetry imposed under normality is difficult to justify, and the tails of even conditional distributions often seem fatter than that of normal distribution.

If the interest is only on the variances, the estimator can be regarded as QML.

Direct Modelling of Correlations

- These models are based on a decomposition of the \mathbf{H}_t .
- The conditional var-cov matrix is expressed as

$$\mathbf{H}_t = \mathbf{D}_t^{1/2} \mathbf{R}_t \mathbf{D}_t^{1/2}$$

where \mathbf{R}_t is possibly time-varying and \mathbf{D}_t is a diagonal matrix containing the conditional variances.

- Conditional correlations and variances are separately modeled.

Constant Conditional Correlations (CCC)

Bollerslev et al. (1990)'s Constant Conditional Correlations model:

The time-varying conditional covariances are parameterized to be proportional to the product of the corresponding conditional standard deviations. The model assumptions are:

$$E_{t-1} [\epsilon_t \epsilon_t'] = \mathbf{H}_t$$

$$\{\mathbf{H}_t\}_{ii} = h_{it} \quad i = 1, \dots, N$$

$$\{\mathbf{H}_t\}_{ij} = h_{ijt} = \rho_{ij} h_{it}^{1/2} h_{jt}^{1/2} \quad i \neq j \quad i, j = 1, \dots, N$$

$$\mathbf{D}_t = \text{diag} \{h_{1t}, \dots, h_{Nt}\}$$

Constant Conditional Correlations (CCC)

The conditional covariance matrix can be written as:

$$\mathbf{H}_t = \mathbf{D}_t^{1/2} \mathbf{R} \mathbf{D}_t^{1/2}$$

$$\mathbf{H}_t = \begin{bmatrix} h_{1t}^{1/2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & h_{Nt}^{1/2} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} & \dots & \rho_{1N} \\ \rho_{21} & 1 & \dots & \vdots \\ \vdots & \vdots & \dots & \rho_{N-1N} \\ \rho_{N1} & \dots & \rho_{NN-1} & 1 \end{bmatrix} \begin{bmatrix} h_{1t}^{1/2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & h_{Nt}^{1/2} \end{bmatrix}$$

When $N = 2$

$$\begin{aligned} \mathbf{H}_t &= \begin{bmatrix} h_{1t}^{1/2} & 0 \\ 0 & h_{2t}^{1/2} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} \\ \rho_{21} & 1 \end{bmatrix} \begin{bmatrix} h_{1t}^{1/2} & 0 \\ 0 & h_{2t}^{1/2} \end{bmatrix} \\ &= \begin{bmatrix} h_{1t} & \rho_{12} h_{1t}^{1/2} h_{2t}^{1/2} \\ \rho_{12} h_{1t}^{1/2} h_{2t}^{1/2} & h_{2t} \end{bmatrix}. \end{aligned}$$

Constant Conditional Correlations (CCC)

- If the conditional variances along the diagonal in the \mathbf{D}_t matrices are all positive, and the conditional correlation matrix \mathbf{R} is positive definite, then the sequence of conditional covariance matrices $\{\mathbf{H}_t\}$ is guaranteed to be positive definite a.s. for all t .
- Furthermore the inverse of \mathbf{H}_t is given by

$$\mathbf{H}_t^{-1} = \mathbf{D}_t^{-1/2} \mathbf{R}^{-1} \mathbf{D}_t^{-1/2}.$$

When calculating the log-likelihood function only one matrix inversion is required for each evaluation.

- CCC is generally estimated in two steps:
 - ① conditional variances are estimated employing the marginal likelihoods
 - ② \mathbf{R} is estimated using the sample estimator of standardized residuals
 $\hat{\eta}_t = \hat{\mathbf{D}}_t^{-1/2} \mathbf{y}_t$ (assuming $\mu_t = \mathbf{0}$).

Constant Conditional Correlations (CCC)

- The CCC solves the *curse of dimensionality* problem of MGARCH models
- The number of parameters is $O(N^2)$ but these are not jointly estimated. The two-step estimation procedure impacts on the computational issues.
- Asymptotic properties of QMLE estimators studied in Ling and McAleer (2003).

Dynamic Conditional Correlations (DCC)

The CCC has two main limitations:

- ① No spillover neither feedback effects across conditional variances
- ② Correlations are static

The evolution of CCC is the Dynamic Conditional Correlation (DCC) Model of Engle (2002). The DCC is an extension of the Bollerslev's CCC Model.

Dynamic Conditional Correlations (DCC)

The conditional correlation between two random variables, X_t and Y_t is defined as:

$$\rho_{YX,t} = \frac{\text{Cov}_{t-1}(X_t Y_t)}{\sqrt{E_{t-1}(X_t - \mu_{X,t})^2 E_{t-1}(Y_t - \mu_{Y,t})^2}}$$

Assets returns conditional distribution:

$$\mathbf{y}_t | \mathcal{F}_{t-1} \sim N(\mathbf{0}, \mathbf{H}_t)$$

$$\mathbf{H}_t = \mathbf{D}_t^{1/2} \mathbf{R}_t \mathbf{D}_t^{1/2}.$$

$$\mathbf{D}_t = \text{diag}(\text{Var}_{t-1}(y_{1t}), \dots, \text{Var}_{t-1}(y_{Nt}))$$

where the $\text{Var}_{t-1}(y_{it})$, $i = 1, \dots, N$ are modeled as univariate GARCH processes.

Dynamic Conditional Correlations (DCC)

The standardized returns are:

$$\boldsymbol{\eta}_t = \mathbf{D}_t^{-1/2} \mathbf{y}_t$$

$$E_{t-1}(\boldsymbol{\eta}_t \boldsymbol{\eta}'_t) = \mathbf{D}_t^{-1/2} \mathbf{H}_t \mathbf{D}_t^{-1/2} = \mathbf{R}_t = \{\rho_{ij,t}\}$$

To ensure that \mathbf{R}_t is positive defined we can employ the following transformation

$$\mathbf{R}_t = \tilde{\mathbf{Q}}_t^{-1/2} \mathbf{Q}_t \tilde{\mathbf{Q}}_t^{-1/2},$$

where $\tilde{\mathbf{Q}}_t$ is a diagonal matrix with typical elements $\tilde{q}_{ii,t} = q_{ii,t}$. This implies that
The conditional correlation are

$$\rho_{ij,t} = \frac{q_{ij,t}}{\sqrt{\tilde{q}_{ii,t} \tilde{q}_{jj,t}}}.$$

Where $q_{ij,t}$ are assumed to follow a GARCH(1,1) model

$$q_{ij,t} = \bar{q}_{ij}(1 - \alpha - \beta) + \alpha \eta_{i,t-1} \eta_{j,t-1} + \beta q_{ij,t-1} \quad (3)$$

The term \bar{q}_{ij} is not the unconditional correlation ($\bar{q}_{ij} \neq \rho_{ij}$) between η_{it} and η_{jt} ; the unconditional correlation between η_{it} and η_{jt} has no closed form.

Wrong targeting!

- Engle (2002) assumes that $\bar{\rho}_{ij} \simeq \bar{q}_{ij}$, and writes

$$q_{ij,t} = \bar{\rho}_{ij}(1 - \alpha - \beta) + \alpha\eta_{i,t-1}\eta_{j,t-1} + \beta q_{ij,t-1} \quad (4)$$

- Aielli (2013) and Engle et al. (2007) suggest to modify the standard DCC in order to correct the asymptotic bias which is due to the fact that $\frac{1}{T} \sum_t \epsilon_t \epsilon_t'$ does not converge to $\bar{\mathbf{Q}}$.
- It is known though that the impact of this is very small, see Engle and Sheppard (2001).

DCC has a number of additional issues which are largely ignored in empirical applications, see Caporin and McAleer (2013).

Positive definiteness

The matrix \mathbf{Q}_t is positive definite for all t as long as it is a weighted average of positive definite matrices and positive semidefinite matrices.

To ensure p.-d-ness of \mathbf{Q}_t we must impose $\alpha + \beta < 1$ In matrix form:

$$\mathbf{Q}_t = \bar{\mathbf{Q}}(1 - \alpha - \beta) + \alpha(\boldsymbol{\eta}_{t-1}\boldsymbol{\eta}'_{t-1}) + \beta\mathbf{Q}_{t-1}$$

where $\bar{\mathbf{Q}}$ is positive definite and set to the unconditional covariance matrix of $\boldsymbol{\eta}_t$.

The log-likelihood function can be written as:

$$\begin{aligned}
 \log L_T &= -\frac{1}{2} \sum_{t=1}^T (N \log(2\pi) + \log |\mathbf{H}_t| + \mathbf{y}'_t \mathbf{H}_t^{-1} \mathbf{y}_t) \\
 &= -\frac{1}{2} \sum_{t=1}^T (N \log(2\pi) + \log |\mathbf{D}_t^{1/2} \mathbf{R}_t \mathbf{D}_t^{1/2}| + \mathbf{y}'_t \mathbf{D}_t^{-1/2} \mathbf{R}_t^{-1} \mathbf{D}_t^{-1/2} \mathbf{y}_t) \\
 &= -\frac{1}{2} \sum_{t=1}^T (N \log(2\pi) + \log |\mathbf{D}_t| + \log |\mathbf{R}_t| + \boldsymbol{\eta}'_t \mathbf{R}_t^{-1} \boldsymbol{\eta}_t)
 \end{aligned}$$

Adding and subtracting $\mathbf{y}'_t \mathbf{D}_t^{-1/2} \mathbf{D}_t^{-1/2} \mathbf{y}_t = \boldsymbol{\eta}'_t \boldsymbol{\eta}_t$

$$\begin{aligned}
 \log L_T &= -\frac{1}{2} \sum_{t=1}^T (N \log(2\pi) + \log |\mathbf{D}_t| + \mathbf{y}'_t \mathbf{D}_t^{-1/2} \mathbf{D}_t^{-1/2} \mathbf{y}_t \\
 &\quad - \boldsymbol{\eta}'_t \boldsymbol{\eta}_t + \log |\mathbf{R}_t| + \boldsymbol{\eta}'_t \mathbf{R}_t^{-1} \boldsymbol{\eta}_t) \\
 &= -\frac{1}{2} \sum_{t=1}^T (N \log(2\pi) + \log |\mathbf{D}_t| + \mathbf{y}'_t \mathbf{D}_t^{-1} \mathbf{y}_t \\
 &\quad - \frac{1}{2} \sum_{t=1}^T (\boldsymbol{\eta}'_t \mathbf{R}_t^{-1} \boldsymbol{\eta}_t - \boldsymbol{\eta}'_t \boldsymbol{\eta}_t + \log |\mathbf{R}_t|)
 \end{aligned}$$

Likelihood decomposition

Volatility component:

$$\mathcal{L}_V(\theta) \equiv \log L_{V,T}(\theta) = -\frac{1}{2} \sum_{t=1}^T (N \log(2\pi) + \log |\mathbf{D}_t| + \mathbf{y}'_t \mathbf{D}_t^{-1} \mathbf{y}_t)$$

Correlation component:

$$\mathcal{L}_C(\theta, \phi) \equiv \log L_{C,T}(\theta, \phi) = -\frac{1}{2} \sum_{t=1}^T (\eta'_t \mathbf{R}_t^{-1} \eta_t - \eta'_t \eta_t + \log |\mathbf{R}_t|)$$

θ denotes the parameters in \mathbf{D}_t and ϕ the parameters in \mathbf{R}_t .

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_V(\theta) + \mathcal{L}_C(\theta, \phi)$$

$$\mathcal{L}_V(\theta) = -\frac{1}{2} \sum_{t=1}^T \sum_{i=1}^N \left(\log(2\pi) + \log(h_{i,t}) + \frac{r_{i,t}^2}{h_{i,t}} \right).$$

The likelihood is apparently the sum of individual GARCH likelihoods, which will be jointly maximized by separately maximizing each term.

Two-step procedure:

1

$$\hat{\theta} = \arg \max \{ \mathcal{L}_V(\theta) \}$$

2

$$\max_{\phi} \{ \mathcal{L}_C(\hat{\theta}, \phi) \}.$$

Under regularity conditions, consistency of the first step will ensure consistency of the second step. The maximum of the second step will be a function of the first step parameter estimates. If the first step is consistent then the second step will be too as long as the function is continuous in a neighborhood of the true parameters.

Time-varying correlation of Tse and Tsui (2002)

Tse and Tsui (2002) proposes a specification similar to the DCC. In the time-varying correlation (TVC) model of Tse and Tsui (2002) the correlation matrix evolves as an ARMA process

$$\mathbf{R}_t = (1 - \alpha - \beta)\mathbf{R} + \alpha\boldsymbol{\Psi}_{t-1} + \beta\mathbf{R}_{t-1},$$

where \mathbf{R} is a $N \times N$ constant matrix as in the DCC specification, and the driving force $\boldsymbol{\Psi}_t$ is a $N \times N$ matrix with general element:

$$\psi_{ij,t} = \frac{\sum_{h=0}^{m-1} \eta_{i,t-h} \eta_{j,t-h}}{\sqrt{\left(\sum_{h=0}^{m-2} \eta_{i,t-h}^2\right) \left(\sum_{h=0}^{m-2} \eta_{j,t-h}^2\right)}}.$$

Note that $m \geq 1$ is an integer that must be selected by the econometrician. Usually $m = 1$ is used. \mathbf{R}_t is positive definite if $\alpha \geq 0$, $\beta \leq 1$ with $\alpha + \beta \leq 1$, and \mathbf{R} and \mathbf{R}_1 are positive definite.

References I

- Aielli, G. P. (2013). Dynamic conditional correlation: on properties and estimation. *Journal of Business & Economic Statistics*, 31(3):282–299.
- Bollerslev, T., Chou, R. Y., and Kroner, K. F. (1992). Arch modeling in finance: A review of the theory and empirical evidence. *Journal of econometrics*, 52(1-2):5–59.
- Bollerslev, T., Engle, R. F., and Wooldridge, J. M. (1988). A capital asset pricing model with time-varying covariances. *Journal of political Economy*, 96(1):116–131.
- Bollerslev, T. et al. (1990). Modelling the coherence in short-run nominal exchange rates: a multivariate generalized arch model. *Review of Economics and statistics*, 72(3):498–505.
- Caporin, M. and McAleer, M. (2013). Ten things you should know about the dynamic conditional correlation representation. *Econometrics*, 1(1):115–126.

References II

- Engle, R. and Mezrich, J. (1996). Garch for groups: A round-up of recent developments in garch techniques for estimating correlation. *RISK-LONDON-RISK MAGAZINE LIMITED-*, 9:36–40.
- Engle, R. F. and Kroner, K. F. (1995). Multivariate simultaneous generalized arch. *Econometric theory*, 11(1):122–150.
- Engle, R. F., Shephard, N., and Sheppard, K. (2007). Fitting and testing vast dimensional time-varying covariance models.
- Engle, R. F. and Sheppard, K. (2001). Theoretical and empirical properties of dynamic conditional correlation multivariate garch. Technical report, National Bureau of Economic Research.
- Francq, C., Horvath, L., and Zakoïan, J.-M. (2011). Merits and drawbacks of variance targeting in garch models. *Journal of Financial Econometrics*, 9(4):619–656.
- Francq, C., Horvath, L., and Zakoian, J.-M. (2016). Variance targeting estimation of multivariate garch models. *Journal of Financial Econometrics*, 14(2):353–382.

References III

- Kraft, D. F. and Engle, R. F. (1982). Autoregressive conditional heteroskedasticity in multiple time series. *UCSD manuscript*.
- Ling, S. and McAleer, M. (2003). Asymptotic theory for a vector arma-garch model. *Econometric theory*, 19(2):280–310.
- Tse, Y. K. and Tsui, A. K. C. (2002). A multivariate generalized autoregressive conditional heteroscedasticity model with time-varying correlations. *Journal of Business & Economic Statistics*, 20(3):351–362.

COPULA METHODS IN FINANCIAL ECONOMETRICS

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Main references: Patton (2006a), Patton (2006b), Jondeau and Rockinger (2006), Chollete et al. (2009), Hafner and Manner (2012), Oh and Patton (2016), Patton (2012), Jondeau et al. (2007, Chapter 6.3)

Introduction

- Often we are not interested in modeling a single random variable;
- Joint probability of several random variables, e.g. returns on several assets;
- Multivariate distributions describe such a joint probability;
- Most of the empirical multivariate distributions deviate from the multivariate Gaussian distribution;
- The copula functions are a statistical tool for a simple treatment of the multivariate distribution of a random vector;

Distribution functions: Univariate

Let us first revise distribution functions.

Definition

A univariate distribution function F is a function from $\mathbb{R} = [-\infty, \infty]$ to $[0, 1]$ such that:

- ① F is weakly increasing
- ② $F(-\infty) = 0$ and $F(\infty) = 1$

Distribution functions: Bivariate

Definition

A bivariate distribution function F is a function from $\mathbb{R} = [-\infty, \infty]^2$ to $[0, 1]$ such that:

- ① F is 2-increasing
- ② $F(-\infty, y) = F(x, -\infty) = 0$, and $F(\infty, \infty) = 1$

Hence, F has marginal distributions denoted by

$$F_1(x) = F(x, \infty) \quad \text{and} \quad F_2(y) = F(\infty, y).$$

Multivariate Normal Distribution

We say that the k -dimensional random vector, \mathbf{x} has a multivariate normal distribution with mean vector μ and covariance matrix $\boldsymbol{\Sigma}$, if its probability density function is

$$f(\mathbf{x}) = \frac{1}{2\pi^{k/2}|\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \mu)' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mu) \right\}$$

where $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$. Hence $\mathbf{x} \sim N_k(\mu, \boldsymbol{\Sigma})$.

Multivariate t-Student Distribution

We say that the k -dimensional random vector, \mathbf{x} has a multivariate Student's t -distribution if

$$\mathbf{x} = \boldsymbol{\mu} + \sqrt{\frac{\nu}{W}} \mathbf{Z}$$

where W is chi-squared distributed random variable with ν degrees of freedom and \mathbf{Z} is $N_k(0, \boldsymbol{\Lambda})$. Hence $\mathbf{x} \sim t_k(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu)$.

Note that the covariance matrix of \mathbf{x} is

$$\boldsymbol{\Sigma} = \frac{\nu}{\nu - 2} \boldsymbol{\Lambda}$$

The parameters of a multivariate t distribution can be estimated by ML.

Elliptical Densities

Both MND and MSTD belong to the elliptical family. A multivariate density is elliptical if it can be expressed as

$$f(x) = kg \left\{ (x - \mu)' \Lambda^{-1} (x - \mu) \right\}$$

where g is a non-negative function and k is a normalising constant such that $\int_{\mathcal{R}^k} g(x) dx = 1/k$. As a consequence, the contour of f are concentric ellipses, such that for any $c > 0$,

$$\mathcal{E}(c) = \left\{ x : (x - \mu)' \Lambda^{-1} (x - \mu)' = c \right\}$$

is an ellipse centered at μ .

ML estimation

- In most financial problems, a set of unknown parameters θ , must be estimated. θ typically contains:
 - Parameters governing the conditional mean (ARMA terms, Seasonal components, Jumps)
 - Parameters governing the conditional variance, (GARCH effects)
- The parameters governing the shape of the error distribution (Tails, Asymmetry) are contained in the vector, η .
- The full set of unknown parameters is $\zeta = (\theta', \eta')'$.
- The ML estimator, $\hat{\zeta}_{ML}$, is obtained maximizing

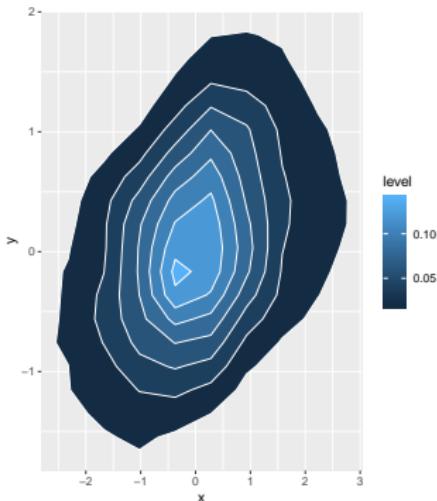
$$\mathcal{L}(\zeta) = \sum_{t=1}^T \ell_t(\zeta)$$

- The optimization wrt θ and η must generally be done in one step.
- Exception is the elliptical family (two-steps estimator). QML for θ under Gaussianity, and then η estimated given $\hat{\theta}$.

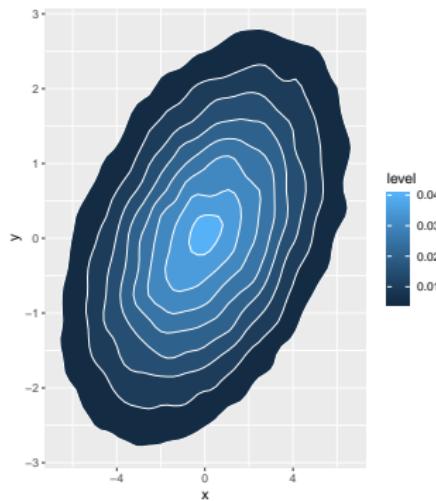
Is this the end of the story?

- Multivariate normal and t distributions (and their skewed extensions) are the only multivariate options available?
- What if the marginals are not Gaussian?
- A multivariate extension of many marginals does not exist.
- What about the dependence between the variables?
- Need for more flexibility!

Contour plot IBM and BAC



(a) Empirical distribution



(b) Estimated Gaussian distribution

Figure: Comparison between the empirical and estimated bivariate distribution between IBM and BAC. Percentage logarithmic returns are computed over the period 2005 – 2018.

Contour plot IBM and BAC

Looking at the scatter plot between the returns of IBM and BAC:

- The joint density does not look Gaussian
- The scatter shows extreme positive and negative dependence (right and left tails)
- Asymmetric dependence (right tail different from the left tail)

How do we fit this joint density with a flexible distribution?

- Fitting a MV Normal with TV Covariance matrix, e.g. multivariate GARCH?
 - Good to capture time-varying correlation and unconditional non-gaussianity
 - What about conditional non-gaussianity?
 - What about extreme dependence?

We need something else...

Copula models: overview

The ability of econometric models to account for the negative consequences for the overall financial system of extreme events strongly relies on their flexibility to feature the highly nonlinear and asymmetric dependence structures of financial returns.

Over the years, the correlation coefficient has emerged as the most natural measure of dependence. However, despite its widespread use, the correlation fails to capture the important tails behaviour of the joint probability distribution, see, e.g., Embrechts et al. (1999, 2002). Hence, modelling the tail dependence and the asymmetric dependence between pairs of assets have been becoming increasingly more important in nowadays financial markets.

Furthermore, the linear correlation coefficient as measure of dependence is usually associated with the assumption of elliptically contoured distributions.

Copula models: why?

The copula methodology overcomes this limitation allowing also for some flexibility in modelling non-linear dependencies, see, e.g., Joe (2015).

Another advantage of using copulas is the decoupling of marginal distributions from the dependence structure. In many cases, the marginal distributions do not depend on the dependence parameter, such that one can first estimate the parameters of the marginal distributions and then in a second step the copula parameters.

The time-varying feature of the dependence behaviour of stock returns, recognised for example by Engle (2002) and Tse and Tsui (2002a), however, motivates the consideration of dynamic copula models where the dependence parameters evolves smoothly as a function of past assets co-movements.

Copula models: breaks in the dependence structure

The dynamic conditional copula theory has been first developed by Patton (2006a), although the problem of modelling the joint co-movement of stock returns was already present in Bollerslev et al. (1988) and Engle et al. (1990), among others.

Furthermore, occasionally, we observe breaks into the dependence structure, which are more evident during crises periods and other infrequent events, as documented, for example, by Pelletier (2006a).

As regards dependence breaks, Markov switching (MS) models have been proven to effectively capture non-smooth evolutions of the volatility and correlations dynamics. Jondeau and Rockinger (2006), Rodriguez (2007), and Chollete et al. (2009) for example, have firstly adopted MS copula with static regime-dependent parameters to analyse financial contagion.

Copula definition

A d -dimensional copula is a distribution function on $[0, 1]^n$ with standard uniform marginal distributions. Hence C is a mapping of the form $C : [0, 1]^d \rightarrow [0, 1]$, i.e. a mapping of the unit hypercube into the unit interval.

The following three properties must hold:

- $C(u_1, \dots, u_n)$ is increasing in each component u_i .
- $C(1, \dots, 1, u_i, 1, \dots, 1) = u_i$ for all $i \in \{1, \dots, n\}$, $u_i \in [0, 1]$.
- For all $(a_1, \dots, a_n), (b_1, \dots, b_n) \in [0, 1]^n$ with $a_i \leq b_i$ we have:

$$\sum_{i_1=1}^2 \cdots \sum_{i_n=1}^2 (-1)^{\sum_{l=1}^n i_l} C(u_{1,i_1}, \dots, u_{n,i_n}) \geq 0,$$

where $u_{l1} = a_j$ and $u_{l2} = b_j$ for all $l \in \{1, \dots, n\}$.

Sklar (1959)'s theorem

Every multivariate cumulative distribution function:

$$H(y_1, \dots, y_n) = P(Y_1 \leq y_1, Y_n \leq y_n),$$

of a random vector (Y_1, \dots, Y_n) , can be expressed in terms of its marginals $F_i(y_i) = P(Y_i \leq y_i)$ and a copula C :

$$H(y_1, \dots, y_n) = C(F_1(y_1), \dots, F_n(y_n)).$$

Provided that a density exists, it is given by:

$$h(y_1, \dots, y_n) = c(F_1(y_1), \dots, F_n(y_n)) \prod_{i=1}^n f_i(y_i),$$

where $c(\cdot)$ is the density of the copula.

Hence we note that $C : [0, 1]^n \rightarrow [0, 1]$, that is: $C(F_1(y_1), \dots, F_n(y_n))$ defines a n -dimensional cumulative distribution function.

One Definition and one Proposition

Definition 5.4 of McNeil et al. (2015)

If the random variable \mathbf{X} has joint distribution function F with continuous marginal distributions F_1, \dots, F_n , then the copula of F is the distribution function C of $(F_1(X_1), \dots, F_n(X_n))$.

Proposition 5.6 of McNeil et al. (2015)

Let (X_1, \dots, X_n) be a random vector with continuous margins and copula C and let T_1, \dots, T_n be strictly increasing functions. Then $(T_1(X_1), \dots, T_n(X_n))$ also has copula C .

For a proof see (McNeil et al., 2015, p. 188)

This proposition is useful when we build *implicit* copulas.

Types of Copula: independence

The independence copula is given by:

$$C(u_1, \dots, u_n) = \prod_{i=1}^n u_i,$$

where $u_i = F_i(y_i)$.

It is clear from Sklar's Theorem, that random variables with continuous distributions are independent if and only if their dependence structure is given by the independence copula.

Types of Copula: implicit copulas

If $\mathbf{Y} \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a Gaussian random vector, then its copula is a so-called Gaussian copula. In order to build the Gaussian copula consider the transformations:

$$T_i(Y_i) = \frac{Y_i - \mu_i}{\sigma_i}, \quad \text{for } i = 1, \dots, n.$$

This is a series of strictly increasing transformations and thus proposition 5.6 of McNeil et al. (2015) can be applied. It follows that $\mathbf{X} \sim \mathcal{N}_n(\mathbf{0}, \mathbf{R})$, where $\mathbf{X} = (X_1, \dots, X_n)'$, with $X_i = T_i(Y_i)$ and $\mathbf{R} = \text{cor}(\mathbf{Y})$, has the same copula of \mathbf{Y} .

By definition 5.4 of McNeil et al. (2015), the copula of \mathbf{X} is given by:

$$\begin{aligned} C^{Ga}(\mathbf{u}; \mathbf{R}) &= P(\Phi(X_1) \leq u_1, \dots, \Phi(X_n) \leq u_n) \\ &= \Phi(\Phi(u_1)^{-1}, \dots, \Phi(u_n)^{-1}; \mathbf{R}), \end{aligned}$$

where $\Phi(\cdot)$ is the distribution function of a standardized univariate Gaussian distribution and $\Phi(\cdot)$ denotes the joint distribution function of \mathbf{X} .

Types of Copula: implicit copulas

We can build implicit copulas starting from any joint distribution function with continuous marginals.

For example, the n -dimensional t copula takes the form:

$$C^t(\mathbf{u}; \nu, \mathbf{R}) = \mathbf{t}(t(u_1; \nu)^{-1}, \dots, t(u_n; \nu)^{-1}; \nu, \mathbf{R}),$$

where $t(\cdot)$ is the distribution function of a standard univariate t distribution, $\mathbf{t}(\cdot; \nu, \mathbf{R})$ is the joint distribution function of the vector $\mathbf{X} \sim \mathcal{T}_n(\nu, \mathbf{0}, \mathbf{R})$ and \mathbf{R} is a correlation matrix.

- Implicit copulas do not usually have a closed form formulation.
- Implicit copulas constructed starting from elliptical distributions (such as the Gaussian and Student's t) are called *elliptical*.

Types of Copula: explicit copulas

While the Gaussian and t copulas are copulas implied by well-known multivariate distribution functions and do not themselves have simple closed forms, we can write down a number of copulas which do have simple closed forms.

Gumbel copula

$$C^{Gu}(\mathbf{u}; \theta) = \exp \left[- \left(\sum_{i=1}^n (-\ln u_i)^\theta \right)^{1/\theta} \right],$$

for $1 \leq \theta < \infty$.

Clayton copula

$$C^{Cl}(\mathbf{u}; \theta) = \left(\sum_{i=1}^n u_i^{-\theta} - 1 \right)^{-1/\theta},$$

for $0 < \theta < \infty$.

The likelihood

Assume to observe T realizations of $\mathbf{Y}_t = (Y_{1,t}, \dots, Y_{n,t})'$. Assume \mathbf{Y}_t is *iid* distributed with density:

$$h(\mathbf{y}_t, \boldsymbol{\theta}) = c(F_1(y_1), \dots, F_n(y_n), \boldsymbol{\theta}_c) \prod_{i=1}^n f_i(y_i, \boldsymbol{\theta}_i),$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}'_c, \boldsymbol{\theta}'_1, \dots, \boldsymbol{\theta}'_n)'$ and $\boldsymbol{\theta}_c$ are copula specific parameters and $\boldsymbol{\theta}_i$ are the parameters of the i -th marginal distribution.

The log likelihood of observing the sequence $\mathbf{Y}_1, \dots, \mathbf{Y}_T$ is:

$$\begin{aligned}\mathcal{L}_{\mathbf{Y}}(\boldsymbol{\theta}) &= \sum_{t=1}^T \log h(\mathbf{y}_t, \boldsymbol{\theta}) \\ &= \sum_{t=1}^T \sum_{i=1}^n \log f_i(y_{i,t}, \boldsymbol{\theta}_i) + \sum_{t=1}^T \log c(F_1(y_1, \boldsymbol{\theta}_1), \dots, F_n(y_n, \boldsymbol{\theta}_n), \boldsymbol{\theta}_c)\end{aligned}$$

Multi stage Maximum Likelihood estimator

One of the main appealing characteristics of the copula framework, with respect to standard distributions, relies on its ability to model the marginals' dynamics separately from the joint dependence structure, see, e.g., Nelsen (2006).

Marginals and dependence separability has some additional advantages even from the econometric point of view, since it permits to employ a two-step procedure to estimate the parameters. This two-step procedure is known as Inference Function for margins (IFM) and is usually referred to Godambe (1960) and McLeish and Small (1988).

Patton (2006a) has recently extended the theory for this two step estimation procedure to the case of conditional copulas: $C(F_1(y_1), \dots, F_n(y_n) | \mathcal{F}_{t-1})$, for some filtration \mathcal{F}_t .

Multi stage Maximum Likelihood estimator (MSML)

- Estimate θ_i by:

$$\hat{\theta}_i = \arg \max_{\theta_i \in \Theta_i \subseteq \mathbb{R}^{d_i}} \sum_{t=1}^T \log(f_i(y_{i,t}, \theta_i))$$

for $i = 1, \dots, n$.

- Estimate θ_c by:

$$\hat{\theta}_c = \arg \max_{\theta_c \in \Theta_c \subseteq \mathbb{R}^{dc}} \sum_{t=1}^T \log(c(F_1(y_{1,t}, \hat{\theta}_1), \dots, F_n(y_{n,t}, \hat{\theta}_n), \theta_c))$$

Clearly the MSML estimator is asymptotically less efficient than one-stage full MLE (unless $Y_i \perp\!\!\!\perp Y_j$ for all $i \neq j$).

Alternative estimation strategies

- Semiparametric estimation: Employ a nonparametric model for the marginal distributions and a parametric model for the copula. See Genest et al. (1995), Shih and Louis (1995), Chen and Fan (2006a), and Chen and Fan (2006b).
- Nonparametric: Fully nonparametric estimation of the copula. See Genest and Rivest (1993), Genest et al. (2011), Genest et al. (2009), Fermanian et al. (2004), Scaillet and Fermanian (2002), Ibragimov (2009), and Sancetta and Satchell (2004).
- Other estimation procedures are available such as: Method of Moments, Generalized Method of Moments, Simulation methods etc., see Smith (2011) for a review.

Visualizing the dependence between two Random Variables

A nice tool to visualize the dependence between two random variables Y_1 and Y_2 is provided by the so called “quantile dependence” defined as:

$$\begin{aligned}\lambda_L(q) &= P(U_1 \leq q, U_2 \leq q)/q, && \text{for } q \in (0, 0.5] \\ \lambda_U(q) &= P(U_1 > q, U_2 > q)/(1 - q), && \text{for } q \in [0.5, 1)\end{aligned}$$

Note that $\lambda_L(0.5) = \lambda_U(0.5)$ by construction since $U_i = F_i(Y_i) \sim \mathcal{U}(0, 1)$ for $i = 1, 2$.

The quantities $\lambda_L(q)$ and $\lambda_U(q)$ can be computed for different values of q and represented graphically.

SP100 and SP600

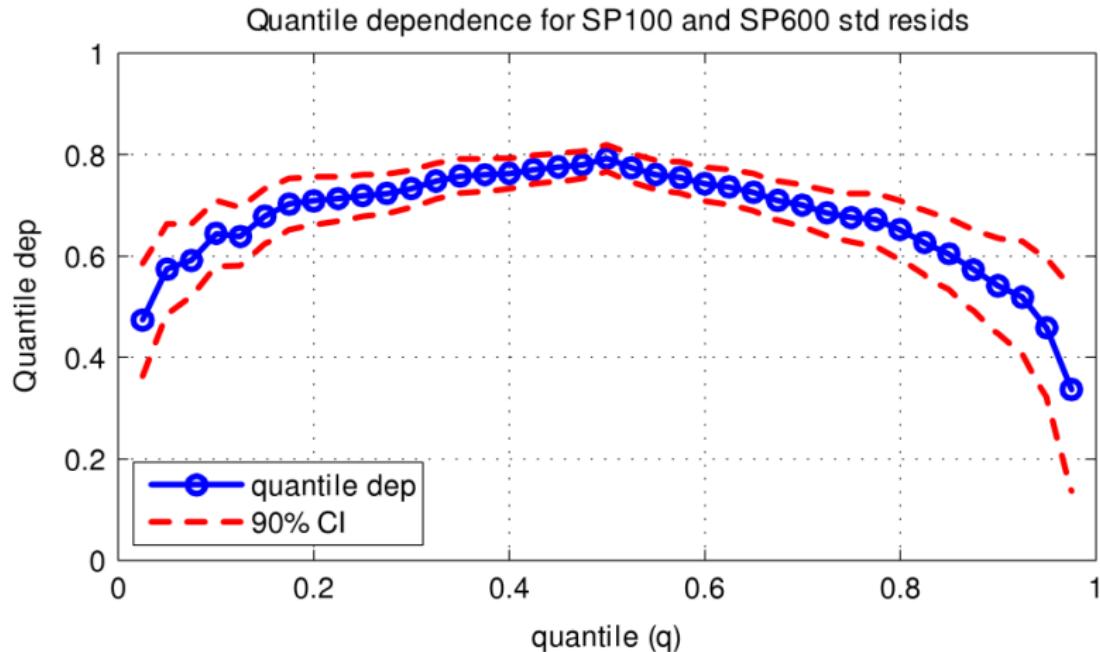


Figure: From Patton (2012): estimated quantile dependence between the standardized residuals for the S&P 100 index and the S&P 600 index along with 90% bootstrap confidence intervals.

SP100 and SP600: estimated copulas

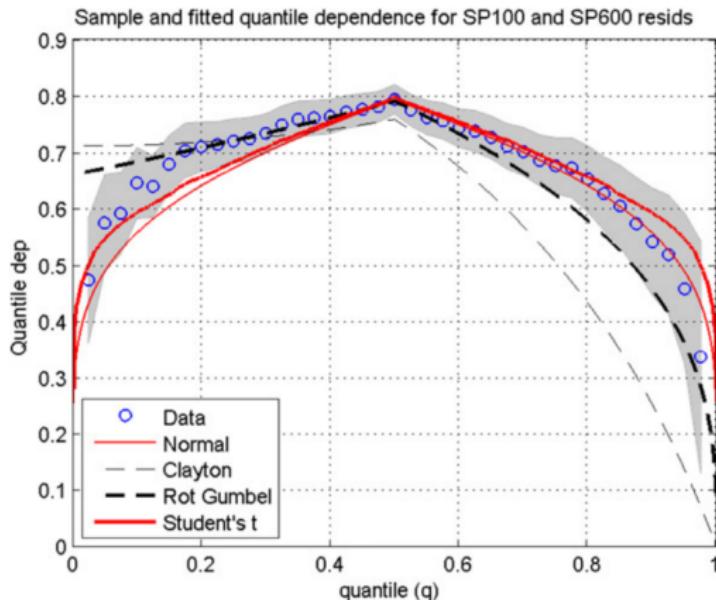


Figure: From Patton (2012): sample quantile dependence between the standardized residuals for the S&P 100 index and the S&P 600 index and 90% bootstrap confidence intervals (shaded), as well as the quantile dependence implied by four copula models.

Others measure of association

Other measures of associations between two random variables X and Y are:

- The Spearman's rank correlation coefficient defined as:

$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)},$$

where $d_i = \text{rank}(x_i) - \text{rank}(y_i)$

- The Kendall's τ coefficient defined as:

$$\tau = \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j \neq i} \text{sgn}(x_i - x_j) \text{sgn}(y_i - y_j)$$

SP100 and SP600: rolling rank correlation

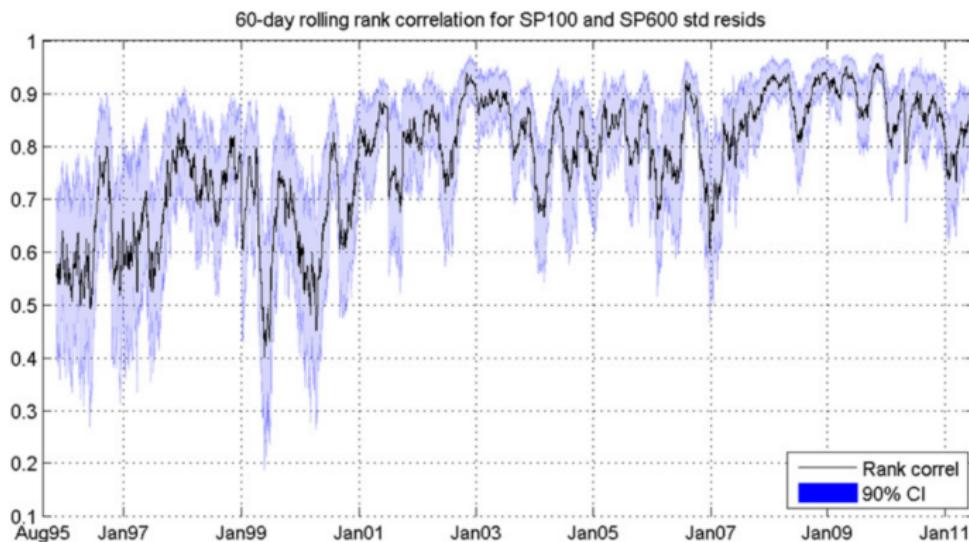


Figure: From Patton (2012): rank correlation between the standardized residuals for the S&P 100 index and the S&P 600 index over a 60-day moving window, along with 90% bootstrap confidence intervals.

The Patton (2006a) model

Patton (2006a,b) initially proposed a time-varying copula model for two random variables $Y_{1,t}$ and $Y_{2,t}$.

As we saw before, the marginals specification can be freely specified such as ARMA-GARCH.

After estimation of the marginals model a careful goodness of fit analysis has to be performed before moving to the analysis of the dependence structure. Indeed, a bad fit of the marginal distributions can heavily affect the estimation of the copula specification. See Patton (2012) for several goodness of fit procedures.

The Patton (2006a,b) model

Let $\rho_t \in (-1, 1)$ be the correlation coefficient of a bivariate elliptical random variable. ρ_t is assumed to follow:

$$\text{Normal : } \rho_t = \Lambda \left(\omega + \beta \rho_{t-1} + \alpha \frac{1}{M} \sum_{j=1}^M \Phi^{-1}(u_{1,t-j}) \Phi^{-1}(u_{2,t-j}) \right)$$

$$\text{Student's } t : \quad \rho_t = \Lambda \left(\omega + \beta \rho_{t-1} + \alpha \frac{1}{M} \sum_{j=1}^M T^{-1}(u_{1,t-j}) T^{-1}(u_{2,t-j}) \right),$$

where $\Lambda(x) = (1 - e^{-x}) / (1 + e^{-x})$ is the modified logistic function and Φ^{-1} and T^{-1} are the inverse cdf of a Gaussian and Student's t standardized random variables, respectively. M is a tuning parameter that can be set equal to 10 for daily observations.

The Patton (2006a,b) model

Let $\theta_t \in [a, b]$ be the dependence parameter of a bivariate archimedian copula. θ_t is assumed to follow:

$$\text{Archimedian : } \theta_t = \tilde{\Lambda}_{[a,b]} \left(\omega + \beta \theta_{t-1} + \alpha \frac{1}{M} \sum_{j=1}^M u_{1,t-j} u_{2,t-j} \right)$$

where $\tilde{\Lambda}_{[a,b]} = a + (b - a) / (1 + e^{-x})$ is the modified logistic function in $[a, b]$. M is a tuning parameter that can be set equal to 10 for daily observations.

The forcing variable is of the form:

- $D^{-1}(u_{1,t-s})D^{-1}(u_{2,t-s})$ for elliptical copulas.
- $u_{1,t-s}u_{2,t-s}$.

The choice of the forcing variable

Does this choice of the forcing variable make sense? It is intuitively appealing and builds on our understanding of covariances: if the transformed marginals have the same sign, the correlation should increase. The reverse holds if the transformed marginals are of opposite sign.

Consider the Gaussian case in the situation when $\Phi^{-1}(u_{1,t}) = 1$ and $\Phi^{-1}(u_{2,t}) = 1$. In this case (assume $M = 1$ for simplicity) the forcing variable is $\Phi^{-1}(u_{1,t})\Phi^{-1}(u_{2,t}) = 1$. Alternatively, consider the case $\Phi^{-1}(u_{1,t}) = 0.25$ and $\Phi^{-1}(u_{2,t-j}) = 4$, also in this case $\Phi^{-1}(u_{1,t})\Phi^{-1}(u_{2,t}) = 1$. So the update of ρ will be the same regardless of which of the two scenarios we observe.

A score driven approach

What happens if we assume a GAS model for the ρ_t ? Creal et al. (2013) derive the score of a bivariate Gaussian copula as:

$$\nabla_t = \Phi^{-1}(u_{1,t})\Phi^{-1}(u_{2,t}) - \rho_t - \rho_t \frac{(\Phi^{-1}(u_{1,t})^2 + \Phi^{-1}(u_{2,t})^2 - 2)}{(1 + \rho_t^2)}.$$

The most important part is given by the numerator of the last term $(\Phi^{-1}(u_{1,t})^2 + \Phi^{-1}(u_{2,t})^2 - 2)$.

Notice that this term is a Martingale Difference sequence and allows the filter to distinguish between the two scenarios of before. Furthermore, the filter will also account for the current amount of correlation ρ_t . If $\rho_t = 0$, the update is the same as of that of Patton (2006a).

GAS vs Patton (2006a,b)'s model

	$10^3 \omega$	A_1	$\ln(B_1/1 - B_1)$	B_1	Log-likelihood
<i>German mark (euro)–US \$, Japanese yen–US \$</i>					
GAS	6.11 (2.48)	0.058 (0.009)	5.30 (0.37)	0.995 (0.990, 0.998)	1218.16
Patton	-1.60 (0.85)	0.036 (0.003)	4.27 (0.10)	0.986 (0.983, 0.989)	1191.51
<i>German mark (euro)–US \$, British pound–US \$</i>					
GAS	12.55 (3.55)	0.082 (0.008)	4.97 (0.23)	0.993 (0.988, 0.996)	2218.82
Patton	-0.97 (0.84)	0.025 (0.002)	4.71 (0.11)	0.991 (0.989, 0.993)	2090.42

Figure: From Creal et al. (2013): Parameter estimates for the GAS and Patton models. The data are the marginal AR-GARCH transforms of log exchange rates for the German mark-US dollar and Japanese yen-US dollar (left panel) and for the German mark-US dollar and British pound-US dollar (right panel), January 1986–August 2008. The asymptotic confidence interval is in parentheses for B_1 ; otherwise the standard error is in parentheses.

The copula–GARCH model of Jondeau and Rockinger (2006)

Jondeau and Rockinger (2006) propose three dynamic copula models which they refer to as “copula–GARCH” models. These are:

- 1) The “semiparametric” approach.
- 2) The “Time Varying Copula” approach.
- 3) The “Markov–Switching” approach.

All models are only defined in the bivariate case of a Student's t copula with time-varying correlation parameter ρ_t .

Jondeau and Rockinger (2006): The “semiparametric” approach

Somehow inspired by Gouriéroux and Monfort (1992), The “semiparametric” copula–GARCH model of Jondeau and Rockinger (2006) assumes that:

$$\rho_t = \sum_{j=1}^{16} d_j \mathbb{1} \left\{ \left(\Phi^{-1}(u_{1,t}), \Phi^{-1}(u_{2,t}) \right) \in \mathcal{A}_j \right\},$$

where \mathcal{A}_j is the j -th element of the unit square grid and $d_j \in [-1, 1]$. To each parameter d_j , an area \mathcal{A}_j is associated. This model. .

Some features of this model are:

- It has 16 parameters.
- The choice of \mathcal{A}_j for $j = 1, \dots, 16$ is arbitrary.
- Is not able to capture the persistence in ρ_t .

The “semiparametric” approach

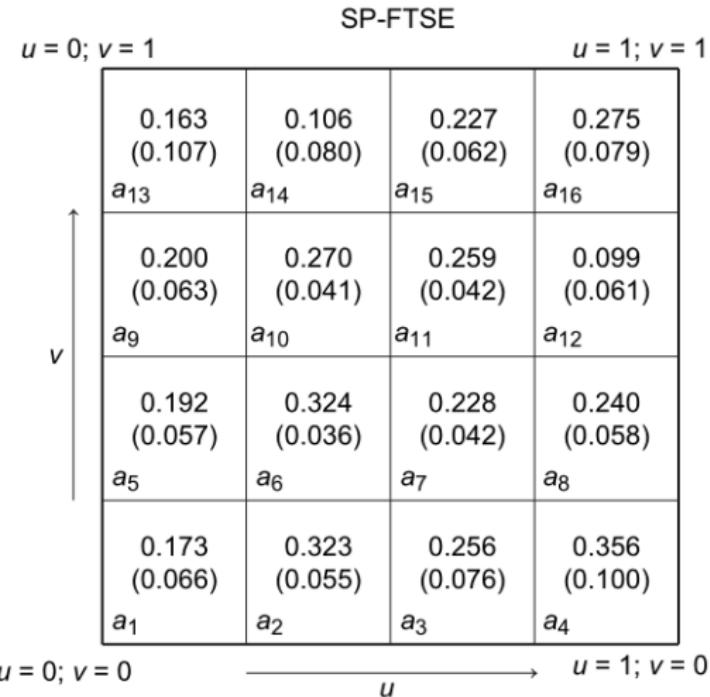


Figure: From Jondeau and Rockinger (2006): The unit square with estimates of parameters d_j s and their standard errors, for the SP-FTSE returns.

Jondeau and Rockinger (2006): The “Time Varying Copula” approach

Somehow inspired by the DCC specification of Tse and Tsui (2002b), the “Time Varying Copula” (TVC) model of Jondeau and Rockinger (2006) assumes:

$$\rho_t = (1 - \alpha - \beta)\rho + \alpha\xi_{t-1} + \beta\rho_{t-1},$$

where:

$$\xi_t = \frac{\sum_{i=0}^{m-1} z_{1,t} z_{2,t}}{\sqrt{\sum_{i=0}^{m-1} z_{1,t}^2 \sum_{i=0}^{m-1} z_{2,t}^2}},$$

where $z_{i,t}$ is the residual of series $i = 1, 2$ at time t . Marginal models in Jondeau and Rockinger (2006) belong to the ARMA–GARCH family.

The “Markov–Switching” approach.

Somehow inspired by Ramchand and Susmel (1998), Chesnay and Jondeau (2001), Ang and Bekaert (2002), and Pelletier (2006b). The Markov–Switching copula model of Jondeau and Rockinger (2006) assumes that:

$$\rho_t = \rho_0 S_t + \rho_1 (1 - S_t),$$

where $S_t \in (0, 1)$ denotes the unobserved regime of the system at time t . S_t is assumed to follow a two- state Markov process, with transition probability matrix given by:

$$\begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix},$$

with:

$$\begin{aligned} p &= P(S_t = 0 | S_{t-1} = 0) \\ q &= P(S_t = 1 | S_{t-1} = 1). \end{aligned}$$

Estimation by direct maximization of the likelihood can be easily implemented.

Comparison between TVC and MS

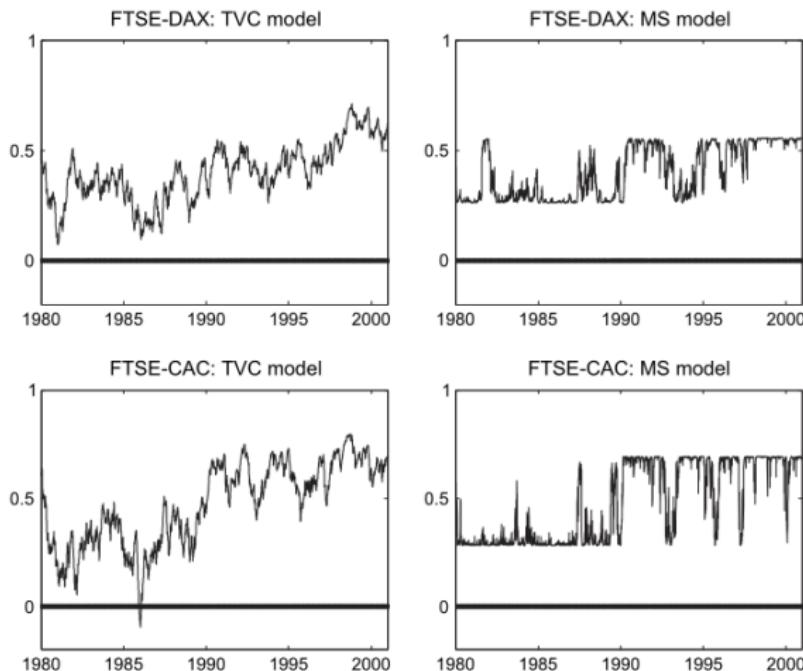


Figure: From Jondeau and Rockinger (2006): The evolution of the parameter r_t obtained with the TVC model and with the Markov-switching model, for the FTSE-DAX and the FTSE-CAC pairs, respectively.

Comparison between TVC and MS

	SP-FTSE	SP-DAX	SP-CAC	FTSE-DAX	FTSE-CAC	DAX-CAC
Constant Copula						
ℓL	139.052	227.500	191.420	433.927	667.453	592.095
AIC	-0.060	-0.099	-0.083	-0.189	-0.291	-0.258
SIC	-0.057	-0.096	-0.080	-0.186	-0.288	-0.255
TVC						
ℓL	140.101	239.233	201.314	499.231	826.917	724.092
AIC	-0.060	-0.103	-0.086	-0.217	-0.360	-0.315
SIC	-0.054	-0.097	-0.081	-0.211	-0.354	-0.309
MS						
ℓL	140.291	249.223	204.163	488.740	829.692	699.747
AIC	-0.060	-0.107	-0.087	-0.212	-0.361	-0.304
SIC	-0.054	-0.102	-0.080	-0.206	-0.356	-0.297

Figure: From Jondeau and Rockinger (2006): Log-likelihood, AIC and SIC information criteria for the Constant, TVC Markov-switching copula models.

References I

- Ang, A. and Bekaert, G. (2002). International asset allocation with regime shifts. *The Review of Financial Studies*, 15(4):1137–1187.
- Bollerslev, T., Engle, R. F., and Wooldridge, J. M. (1988). A capital asset pricing model with time-varying covariances. *Journal of Political Economy*, 96(1):116–131.
- Chen, X. and Fan, Y. (2006a). Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification. *Journal of Econometrics*, 135(1-2):125–154.
- Chen, X. and Fan, Y. (2006b). Estimation of copula-based semiparametric time series models. *Journal of Econometrics*, 130(2):307–335.
- Chesnay, F. and Jondeau, E. (2001). Does correlation between stock returns really increase during turbulent periods? *Economic Notes*, 30(1):53–80.
- Chollete, L., Heinen, A., and Valdesogo, A. (2009). Modeling International Financial Returns with a Multivariate Regime-Switching Copula. *Journal of Financial Econometrics*, 7(4):437–480.

References II

- Creal, D., Koopman, S. J., and Lucas, A. (2013). Generalized autoregressive score models with applications. *Journal of Applied Econometrics*, 28(5):777–795.
- Embrechts, P., McNeil, A., and Straumann, D. (1999). Correlation: pitfalls and alternatives. *RISK-LONDON-RISK MAGAZINE LIMITED*, 12:69–71.
- Embrechts, P., McNeil, A. J., and Straumann, D. (2002). Correlation and dependence in risk management: properties and pitfalls. In *Risk management: Value-at-Risk and beyond (Cambridge, 1998)*, pages 176–223. Cambridge Univ. Press, Cambridge.
- Engle, R. (2002). Dynamic Conditional Correlation: a simple class of multivariate Generalized Autoregressive Conditional Heteroskedasticity models. *Journal of Business & Economic Statistics*, 20(3):339–350.
- Engle, R. F., Ng, V. K., and Rothschild, M. (1990). Asset pricing with a factor-arch covariance structure: Empirical estimates for treasury bills. *Journal of Econometrics*, 45(1):213 – 237.
- Fermanian, J.-D., Radulovic, D., Wegkamp, M., et al. (2004). Weak convergence of empirical copula processes. *Bernoulli*, 10(5):847–860.

References III

- Genest, C., Ghoudi, K., and Rivest, L.-P. (1995). A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3):543–552.
- Genest, C., Nešlehová, J., and Ziegel, J. (2011). Inference in multivariate archimedean copula models. *Test*, 20(2):223.
- Genest, C. and Rivest, L.-P. (1993). Statistical inference procedures for bivariate archimedean copulas. *Journal of the American statistical Association*, 88(423):1034–1043.
- Genest, C., Segers, J., et al. (2009). Rank-based inference for bivariate extreme-value copulas. *The Annals of Statistics*, 37(5B):2990–3022.
- Godambe, V. P. (1960). An optimum property of regular maximum likelihood estimation. *Ann. Math. Statist.*, 31(4):1208–1211.
- Gourioux, C. and Monfort, A. (1992). Qualitative threshold arch models. *Journal of econometrics*, 52(1-2):159–199.

References IV

- Hafner, C. M. and Manner, H. (2012). Dynamic stochastic copula models: estimation, inference and applications. *Journal of Applied Econometrics*, 27(2):269–295.
- Ibragimov, R. (2009). Copula-based characterizations for higher order markov processes. *Econometric Theory*, 25(3):819–846.
- Joe, H. (2015). *Dependence modeling with copulas*, volume 134 of *Monographs on Statistics and Applied Probability*. CRC Press, Boca Raton, FL.
- Jondeau, E., Poon, S.-H., and Rockinger, M. (2007). *Financial modeling under non-Gaussian distributions*. Springer Science & Business Media.
- Jondeau, E. and Rockinger, M. (2006). The Copula-GARCH model of conditional dependencies: An international stock market application. *Journal of International Money and Finance*, 25(5):827 – 853.
- McLeish, D. L. and Small, C. G. (1988). *The theory and applications of statistical inference functions*, volume 44 of *Lecture Notes in Statistics*. Springer-Verlag, New York.

References V

- McNeil, A. J., Frey, R., and Embrechts, P. (2015). *Quantitative Risk Management: Concepts, Techniques and Tools: Concepts, Techniques and Tools*. Princeton university press.
- Nelsen, R. B. (2006). *An introduction to copulas*. Springer Series in Statistics. Springer, New York, second edition.
- Oh, D. H. and Patton, A. J. (2016). Time-varying systemic risk: Evidence from adynamic copula model of cds spreads. *Journal of Business & Economic Statistics*, 0(ja):1–47.
- Patton, A. J. (2006a). Estimation of multivariate models for time series of possibly different lengths. *Journal of Applied Econometrics*, 21(2):147–173.
- Patton, A. J. (2006b). Modelling asymmetric exchange rate dependence. *Internat. Econom. Rev.*, 47(2):527–556.
- Patton, A. J. (2012). A review of copula models for economic time series. *Journal of Multivariate Analysis*, 110:4 – 18. Special Issue on Copula Modeling and Dependence.

References VI

- Pelletier, D. (2006a). Regime switching for dynamic correlations. *Journal of Econometrics*, 131(2):445 – 473.
- Pelletier, D. (2006b). Regime switching for dynamic correlations. *Journal of econometrics*, 131(1-2):445–473.
- Ramchand, L. and Susmel, R. (1998). Volatility and cross correlation across major stock markets. *Journal of Empirical Finance*, 5(4):397–416.
- Rodriguez, J. C. (2007). Measuring financial contagion: A copula approach. *Journal of Empirical Finance*, 14(3):401 – 423.
- Sancetta, A. and Satchell, S. (2004). The bernstein copula and its applications to modeling and approximations of multivariate distributions. *Econometric theory*, 20(3):535–562.
- Scaillet, O. and Fermanian, J.-D. (2002). Nonparametric estimation of copulas for time series.
- Shih, J. H. and Louis, T. A. (1995). Inferences on the association parameter in copula models for bivariate survival data. *Biometrics*, pages 1384–1399.

References VII

- Sklar, M. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8:229–231.
- Smith, M. S. (2011). Bayesian approaches to copula modelling. *arXiv preprint arXiv:1112.4204*.
- Tse, Y. K. and Tsui, A. K. C. (2002a). A multivariate Generalized Autoregressive Conditional Heteroscedasticity model with time-varying correlations. *Journal of Business & Economic Statistics*, 20(3):351–362.
- Tse, Y. K. and Tsui, A. K. C. (2002b). A multivariate generalized autoregressive conditional heteroscedasticity model with time-varying correlations. *Journal of Business & Economic Statistics*, 20(3):351–362.

DEPENDENCE MEASURES AND EXTREME VALUE THEORY

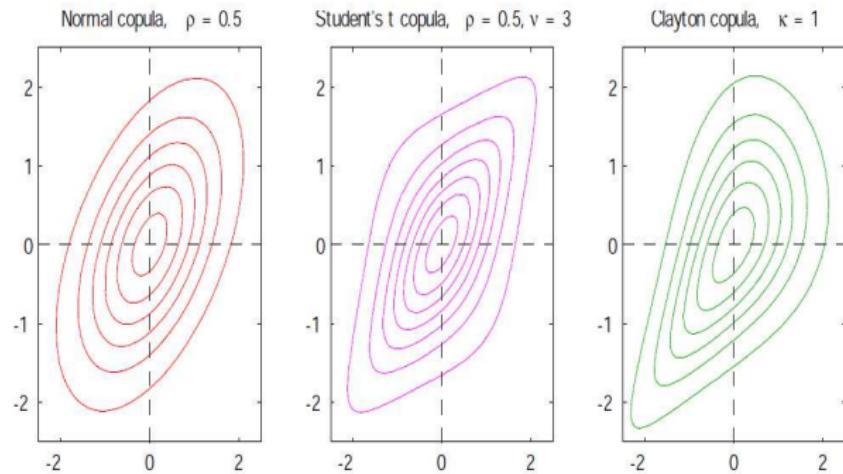
Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Examples

A variety of bivariate distributions with Normal margins

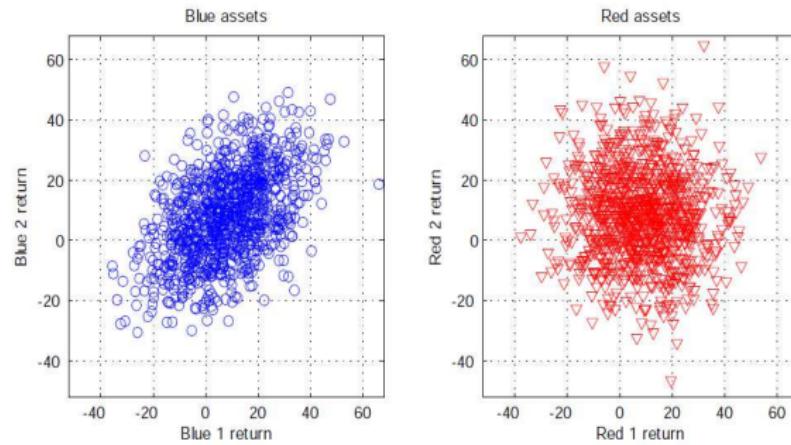


Why do we care so much about joint distribution

- One important economic application where the form of dependence matters is portfolio decision making.
- Consider the following illustration:
 - Two pairs of assets.
 - All assets are individually $N(8\%, 15\%)^2$
 - We will vary their dependence structure (copula) and consider the outcome.

Example 1

Which pair of assets would you prefer?



Example 1

Consider an equally weighted portfolio:

Blue Assets ($\rho = 0.5$):

$$E\left[\frac{1}{2}X + \frac{1}{2}Y\right] = 8\%$$

$$Var\left[\frac{1}{2}X + \frac{1}{2}Y\right] \approx 13^2\%$$

Red Assets ($\rho = 0$):

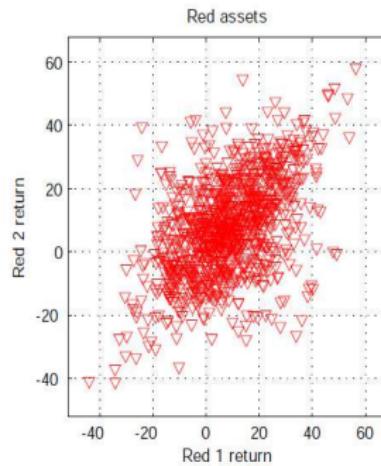
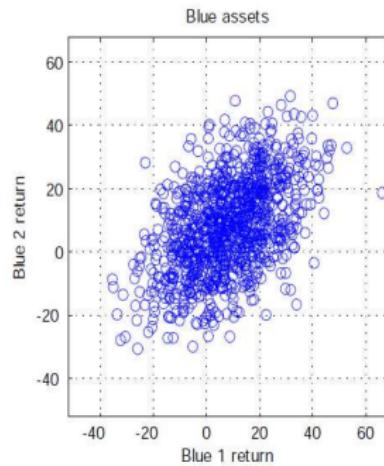
$$E\left[\frac{1}{2}X + \frac{1}{2}Y\right] = 8\%$$

$$Var\left[\frac{1}{2}X + \frac{1}{2}Y\right] \approx 10.5^2\%$$

We should prefer the red assets.

Example 2

Both have $N(8, 15^2)$ margins and $\rho = 0.5$. Which pair of assets would you prefer?



Example 2

All assets have the same marginal distributions, and correlation of both pairs is 0.5.

- So mean-variance comparisons of portfolios will not distinguish between the two pairs.
- But we could see that the red pair (which had a Student's t copula) had more joint crashes and booms than the blue pair (Normal copula).
- This leads to more kurtosis in a portfolio of the red assets.

Summary

These simple examples show that pairs of variables with the same marginal distributions and the same degree of (linear) correlation can be consistent with many different bivariate distributions.

- These joint distributions differ, broadly, in:
 - ➊ The degree to which large events are correlated (tail dependence).
 - ➋ The degree to which negative events have different correlation to positive events (asymmetric dependence)
- In portfolio applications, we know that risk-averse investors have clear preferences over these dependence structures
- To study these dependence structures we need more flexible models of dependence, and richer measures of dependence.

Pearson's Correlation

- The most widely-used measure of dependence is Pearson's linear correlation:

$$\text{Corr}[Y, Z] = \frac{\text{Cov}[Y, Z]}{\sqrt{V[Y]V[Z]}}$$

- This simple measure contains a lot of information, but it suffers from some limitations.

Limitations of Pearson's correlation

- Many different dependence structures (copulas) can have identical linear correlation coefficients (as we saw in the portfolio illustration).
- For example, when combined with $N(0, 1)$ margins, the following copulas all lead to linear correlation of 0.5:
 - Normal with $\rho = 0.5$
 - Student's with $\rho = 0.5$
 - Clayton with $\theta = 1$
 - Gumbel with $\theta = 1.5$

Limitations of Pearson's correlation

- The range of linear correlation is affected by the marginal distributions.
- The actual range of possible values for linear correlation is narrower, and depends on the marginal distributions.
- If the margins are identical (up to affine transformations) then the full range is possible $[-1, 1]$
- If the margins are different then the actual range of possible values can be much narrower.

An Extreme Example

Perfectly dependent variables can have linear correlation arbitrarily close to zero (rare, but possible).

- As in Embrechts, et al. (2002), consider two perfectly dependent variables:

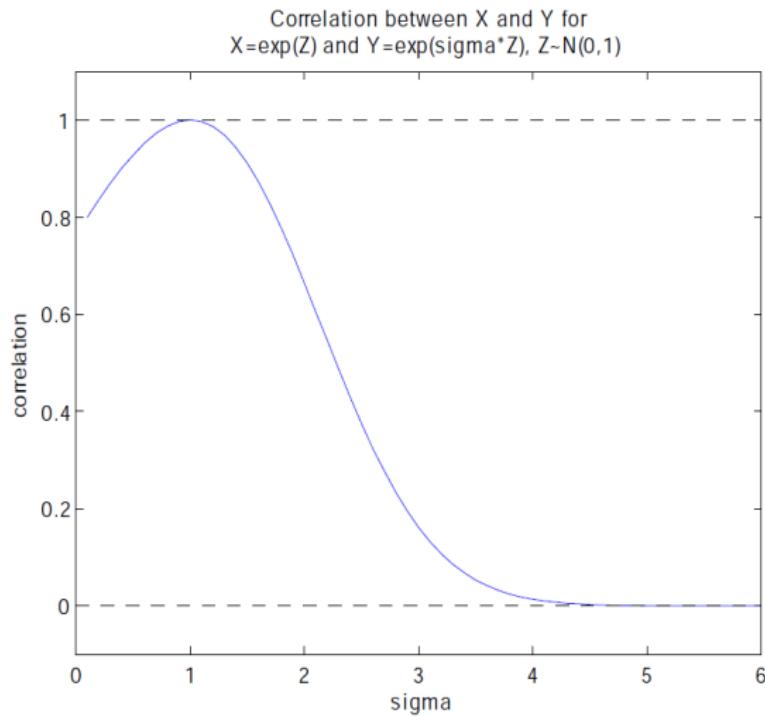
$$\begin{aligned}Z &\sim N(0, 1) \\X &= \exp(Z) \\Y &= \exp(\sigma Z)\end{aligned}$$

then it is possible to show that

$$\text{Corr}[X, Y] = \frac{\exp(\sigma) - 1}{\sqrt{\exp(1) - 1} \sqrt{\exp(\sigma^2) - 1}} \rightarrow 0$$

as $\sigma \rightarrow \infty$.

Limitations of linear correlation



Alternative Measures of Dependence: Spearman's Rank Correlation

Rank correlation is simply the linear correlation of the ranks of the variables.

- The smallest observation has a rank of 1, second-smallest a rank of 2, etc.
- Let $R^X = \text{Rank of } X \text{ in } (X_1, \dots, X_T)$;
- For example, if $X = [5, 6, 1, 8, 3]$, the rank is

$$R^X = [3, 4, 1, 5, 2]$$

then, Spearman's rank correlation between X and Y is

$$\rho^{X,Y} = \text{corr}(R^X, R^Y)$$

Properties of Spearman's Rank Correlation

- The ranks of observations in a sample are unaffected by strictly increasing transformations, and so rank correlation is unaffected by such transformation.
- Consider the PIT, $U_i = F_i(X_i)$, then we have that :

$$\varsigma^{X,Y} = \text{corr}(R^X, R^Y) = \text{corr}(U_x, V_x)$$

- Thus rank correlation is purely a function of the copula of (X,Y).

Kendall's tau

- Kendall's tau is another widely-used measure of dependence. It is based on the proportion of *concordant* pairs of observations in a sample.
- Let x_t, y_t for $t = 1, \dots, T$ be a sample of observations. There are $\frac{T!}{2(T-2)!}$ distinct pairs of observations (x_i, y_i) and (x_j, y_j) .
- A pair of observations is concordant if $(x_i - x_j)(y_i - y_j) > 0$, else the pair is discordant
- Let c and d denote the number of concordant and discordant pairs of observations. Then

$$\tau = \frac{c - d}{c + d}$$

- The Kendall's tau is also function of the copula only

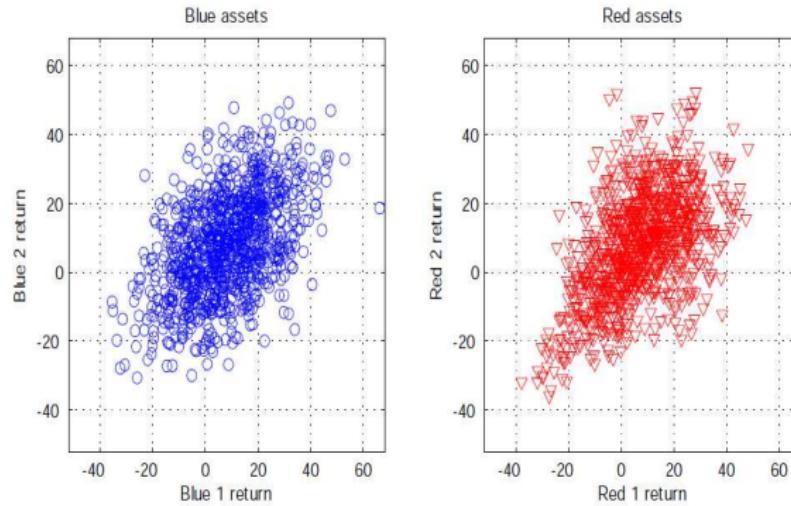
$$\tau = 4E[C(U_x, U_y)] - 1$$

Kendall's tau and Copulas

- The Kendall's tau of the Gaussian Copula is given by $\frac{2}{\pi} \arcsin(\rho)$
- The Kendall's tau of the t-Copula is given by $\frac{2}{\pi} \arcsin(\rho)$
- The Kendall's tau (τ) coefficient for the Gumbel copula is equal to $1 - \frac{1}{\theta}$.
- The Kendall's tau of the Clayton copula is equal to $\frac{\theta}{\theta+2}$.

Example 3

Both have $N(8, 15^2)$ margins and $\rho = 0.5$. Which pair of assets would you prefer?



Example 3

All assets have the same marginal distributions, and correlation of both pairs is 0.5.

- Again, mean-variance comparisons of portfolios will not help here
- But we could see that the red pair (Clayton copula) had more joint crashes and fewer joint booms than the blue pair (Normal copula)
- This leads to more skewness and kurtosis in the portfolio of the red assets.

Asymptotic dependence

- The concepts of asymptotic dependence and independence, i.e. degree of association of tail events
- Important to remove the influence of marginal aspects (no effects on the asymptotic dependence).
- Given a pair of RV, X, Y , the analysis is carried out on $U \equiv F_X(X)$ and $V \equiv F_Y(Y)$.
- Specifically, we consider the behavior of $P(V > u | U > u)$.
- In case of perfect dependence, then $P(V > u | U > u) = 1$.
- In case of perfect independence, then $P(V > u | U > u) = P(V > u)$.

Quantile Dependence and Copulae

- The measure of quantile dependence are functions that are defined as

$$\begin{aligned}\lambda^L(q) &= P(U_x < q | U_y < q) = \frac{P(U_x < q, U_y < q)}{P(U_y < q)} \\ &= \frac{C(q, q)}{q}\end{aligned}$$

$$\begin{aligned}\lambda^U(q) &= P(U_x > q | U_y > q) = \frac{P(U_x > q, U_y > q)}{P(U_y > q)} \\ &= \frac{1 - 2q + C(q, q)}{1 - q}\end{aligned}$$

- The function $\lambda^L(q)$ can be interpreted as the probability that one variable lies in its lower q tail given that the other variable lies in its lower q tail.
- Different copulas imply different quantile dependences.

Tail Dependence

- Pushing the quantiles to the limit...
- The measure of negative tail dependence is defined as

$$\lambda^L = \lim_{q \rightarrow 0} P(U_x < q | U_y < q) = \lim_{q \rightarrow 0} \frac{C(q, q)}{q}$$

$$\lambda^U = \lim_{q \rightarrow 1} P(U_x > q | U_y > q) = \lim_{q \rightarrow 1} \frac{1 - 2q + C(q, q)}{1 - q}$$

Tail Dependence and Copulae

- The Gaussian copula does not generate tail dependence for any $|\rho| < 1$.
- The t-copula has symmetric tail dependence if $\nu < \infty$

$$\lambda^U = \lambda^L = 2\mathcal{T}_{\nu+1} \left(-\sqrt{\nu+1} \sqrt{\frac{1-\rho}{1+\rho}} \right),$$

where $\mathcal{T}_{\nu+1}(\cdot)$ is the cumulative distribution function of a univariate Student's t random variable with $\nu + 1$ degrees of freedom.

- The degree of upper tail dependence for the Gumbel copula is equal to $2 - 2^{\frac{1}{\theta}}$, see Joe (1997).
- The Clayton copula implies a degree of tail dependence equal to $2^{(-1/\theta)}$.

Extreme Value Theory

- Modeling the entire distribution of the returns is useful but possibly complicated
- For some applications it is not needed to look at the entire distribution, e.g. VaR or expected shortfall.
- We can use specific tools created for the description of the **tails**.
- **Extreme Value Theory** is a set of statistical tools for the analysis of extreme realizations, both in univariate and in the multivariate context.
- The focus is on the so-called *tail index*.

Univariate Tail Estimation

Two approaches:

- ① *Extrema approach*: distribution of the maxima/minima
 - ② *Tail approach*: exceedances over a given threshold.
- The extrema approach leads to the definition of a generalized extreme value distribution, gev.
 - Given a sequence (X_1, \dots, X_T) of random variables, we define

$$M_T \equiv -\min(-X_1, \dots, -X_T) = \max(X_1, \dots, X_T)$$

- If (X_1, \dots, X_T) are *i.i.d.*, then

$$P(M_T \leq x) = [F_X(x)]^T$$

Univariate Tail Estimation

Theorem

Let X_t be a sequence of i.i.d. random variables. If there exists a location parameter $\mu_T \in \mathcal{R}$, and a scale parameter $\psi_T > 0$, and some non-degenerate cdf, H such that the limit distribution of the standardized extremes, $Y_T = \frac{M_T - \mu_T}{\psi_T}$ converges to H ,

$$\lim_{T \rightarrow \infty} P\left(\frac{M_T - \mu_T}{\psi_T} \leq y\right) = H(y), \quad \forall y \in \mathcal{R}$$

then $F_X(\cdot)$ is said to belong to the domain of attraction of H , which is a standard extreme value distribution.

The gev encompasses the standard extreme value distributions,

$$H_\xi(y) = \begin{cases} \exp(-(1 + \xi y)^{-1/\xi}), & \text{if } \xi \neq 0, 1 + \xi y > 0 \\ \exp(-\exp(-y)), & \text{if } \xi = 0 \end{cases}$$

where ξ is called tail index. If $\xi > 0$, we have the Frechet distribution, if $\xi = 0$ we have the Gumbel and if $\xi < 0$ we have the Weibull.

Quantile Plot

- The quantile plot is a nice graphical instrument to understand the limit distribution of the maxima of the returns.
- Consider $\tau = T/N$ sub-samples of size N over T periods.
- For each period compute $m_i = \max(x_{(i-1)N+1}, \dots, x_{iN})$.
- Order the maxima as, $\tilde{m}_1, \dots, \tilde{m}_\tau$.
- For each \tilde{m}_i , compute the theoretical quantiles,
 $H_{\xi, \mu, \psi}^{-1}\left(\frac{1}{\tau+1}\right), \dots, H_{\xi, \mu, \psi}^{-1}\left(\frac{\tau}{\tau+1}\right)$.
- Assume the distribution is Gumbel, then

$$H_{\xi, \mu, \psi}^{-1}\left(\frac{i}{\tau+1}\right) = -\log\left(-\log\left(\frac{i}{\tau+1}\right)\right)$$

- Plot $\tilde{m}_1, \dots, \tilde{m}_\tau$ wrt $H_{\xi, \mu, \psi}^{-1}\left(\frac{1}{\tau+1}\right), \dots, H_{\xi, \mu, \psi}^{-1}\left(\frac{\tau}{\tau+1}\right)$ on a 2-D plot.
- If the plot is concave, we have a Frechet distribution, if convex the limit distribution is Weibull.

ML estimation of the gev

- Consider an *i.i.d.* sample, m_i for $i = 1, \dots, \tau$.
- The log-likelihood function wrt $\theta = (\mu, \psi, \xi)$ is

$$L_\tau(\theta|m_i) = \sum_{i=1}^{\tau} \ell(\theta)$$

where

$$\ell_i(\theta) = -\log(\psi) - \left(\frac{1}{\xi} + 1 \right) \log \left(1 + \xi \frac{m_i - \mu}{\psi} \right) - \left(1 + \xi \frac{m_i - \mu}{\psi} \right)^{-\frac{1}{\xi}}$$

- If $\xi = 0$,

$$\ell_i(\theta) = -\log(\psi) - \frac{m_i - \mu}{\psi} - \exp\left(-\frac{m_i - \mu}{\psi}\right)$$

- The ML estimator has standard asymptotic for $\xi > -1/2$.

Tail approach

- The tail approach is based on modeling the tails of a distribution.
- We consider an iid sample, (X_1, \dots, X_T) , with distribution $F_X(\cdot)$.
- Let u be a fixed real number, the threshold, then

$$F_u(y) = P(X_t - u \leq y | X_t > u) = \frac{F_X(y+u) - F_X(u)}{1 - F_X(u)}$$

is the excess distribution function.

- The function

$$e(u) = E(X_t - u | X_t > u)$$

is called mean-excess function.

- The excess distribution function can be approximated by the Generalized Pareto distribution (gdp)

$$F_u(y) \approx G_{\xi, \psi}(y) = \begin{cases} 1 - \left(1 + \frac{\xi}{\psi}y\right)^{-1/\xi}, & \text{if } \xi \neq 0 \\ 1 - \exp\left(-\frac{y}{\psi}\right), & \text{if } \xi = 0 \end{cases} \quad (1)$$

- The tail index of the gev is the same as the one of the gpd.

Hill estimator

- An estimate of ξ can be obtained also by a semi-parametric method.
- The Hill estimator is

$$\hat{\xi}_{(q,T)}^H = \frac{1}{q} \sum_{j=1}^q \log \left(\frac{x_{T-j+1,T}}{x_{T-q,T}} \right) \quad 1 \leq q < T$$

- Under the assumption that the distribution of X_t belongs to the Frechet, then

$$\sqrt{q} (\hat{\xi} - \xi) \rightarrow N(0, \xi^2) \tag{2}$$

- Bootstrap techniques for the selection of the optimal q have been proposed.
- Alternatively, an ML estimation can be performed (parametric setup).

RISK MANAGEMENT

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Need for risk management

- There are many different types of risk
 - *Market Risk*: Due to changes in prices;
 - *Credit Risk*: Counter party does not meet contractual obligations;
 - *Liquidity Risk*: Extra cost of liquidating a position;
- It is crucial to develop tools to deal with these sources of risk.
- **Basel Committee on Bank Supervision** (BCBS) imposes capital requirements to cover these risk.
- In 1996 agreement on market risk, and introduction of the *Value-at-risk* (VaR).
- Nowadays, VaR and *Expected Shortfall* (ES) are widely used because they can be applied to all securities.

VaR

- The value at risk is defined with respect to:
 - Time horizon: τ
 - Confidence level: $1 - \alpha$
- The VaR is a bound such that the loss over the horizon is less than this bound with probability equal to the confidence coefficient.
- In other words, the VaR is the minimum potential loss that the portfolio can suffer in the $\alpha\%$ worst cases, over the period τ .
- Statistically speaking, the VaR is a quantile of the return(or loss) distribution.
- For example, if the horizon is one week, and with $\alpha = 1\%$ the VaR is \$5 millions, this means that there is a 1% chance of a loss exceeding \$5 millions over the next week.

VaR

Given $\mathcal{L}_{t+\tau}$ the loss over the holding period τ , the $VaR(\alpha)$ at time t is the α -th upper quantile of $\mathcal{L}_{t+\tau}$. For continuous, loss distribution, the $VaR(\alpha)$ solves

$$Pr(\mathcal{L}_{t+\tau} \geq VaR_t(\alpha)) = \alpha$$

or alternatively

$$Pr(\mathcal{R}_{t+\tau} < VaR_t(\alpha)) = \alpha$$

where $\mathcal{R} = -\mathcal{L}$ is the revenue, defined as

$$\mathcal{R}_{t+\tau} = \frac{\Delta W_{t+\tau}}{W_t} \tag{1}$$

where W_t is the value of the portfolio. Hence, the VaR is

$$VaR_{\alpha,t+\tau|t} = F_{t+\tau|t}^{-1}(\alpha) \tag{2}$$

where $F_{t+\tau|t}(x) = Pr(R_{t+\tau} \leq x | \mathcal{F}_t)$.

VaR: Properties

Artzner et. al (1997,1999) list the properties that any risk measure, $\rho(\cdot)$ should have to be *coherent*:

- Monotonicity: if $X \leq Y$, then $\rho(X) \geq \rho(Y)$
- Homogeneity: if $\kappa \geq 0$ then $\rho(\kappa X) = \kappa \rho(X)$
- Translation Invariance: if F is a risk-free asset with return r_f , then $\rho(X + F) = \rho(X) - r_f$
- Sub-additivity: $\rho(X + Y) \leq \rho(X) + \rho(Y)$

The VaR is not a coherent risk measure because it does not respect sub-additivity.
As a consequence it discourages diversification.

Note that here X and Y are loss distributions (negative of the returns)

Example

- Company sells a bond for \$1000 with $\tau = 1$ year and rate 5%
- If the bank defaults, the entire \$1000 is lost. The probability of default is $p = 4\%$.
- The loss function is (use the normal distribution to make the loss function continuous)

$$\mathcal{L} = (1 - p) \cdot N(-50, 1) + p \cdot N(1000, 1)$$

- Suppose another independent company selling a bond with the same loss function
- Suppose P_1 , made by 2 bonds of company 1, its loss function is

$$\mathcal{L} = 0.04 \cdot \Phi(x; 2000, 4) + 0.96 \cdot \Phi(x; -100, 4)$$

- Suppose P_2 , made by 1 bond of each company, its loss function is

$$\mathcal{L} = 0.04^2 \cdot \Phi(x; 2000, 2) + 2 \cdot (0.96) \cdot (0.04) \cdot \Phi(x; 950, 2) + 0.96^2 \cdot \Phi(x; -100, 2)$$

- $VaR(0.05)_{P_1} = -95.38$ and $VaR(0.05)_{P_2} = 949.53$. This seems to tell us that Portfolio 1 is much less risky than portfolio 2. Is it true?
- Result depends on the choice of α .

Expected Shortfall

- The VaR is not informative on the amount of the loss over the threshold.
- Basak and Shapiro (2001): the VaR disregards the risk of extreme large losses, i.e. large losses behind the confidence level.
- The VaR is not sub-additive.
- A newer risk measure is the *Expected Shortfall*, or ES.
- ES is defined as the expected loss given that the loss exceeds the VaR.
Formally

$$ES(\alpha) = \frac{\int_{\alpha}^1 VaR(u) du}{1 - \alpha}$$

which is the average of $VaR(u)$ over all u that are less or equal to α

$$ES(\alpha) = E[\mathcal{L} | \mathcal{L} \geq VaR(\alpha)]$$

- The ES can also be used for portfolio allocation.

VaR with Gaussian returns and constant parameters

- Suppose that the return on a stock is normally distributed with yearly mean μ and variance σ^2 . Suppose that we purchase \$ 100,000 of that stock, what is the VaR for $\tau = 1$ year?
- The distribution of our position is Gaussian with mean $\mu_L = 100,000 \times \mu$ and standard deviation $\sigma_L = \sigma \times 100,000$.
- Therefore, the VaR is

$$\widehat{VaR}_t = \mu_L + \sigma_L z_\alpha$$

where z_α is the α -th quantile of a normal distribution.

Modeling portfolio returns

- For the computation of the VaR and ES we need:
 - the probability, α
 - the horizon of the investment, τ
 - the value of the portfolio at t , W_t
 - the *cdf* of the portfolio return
- *Choosing a model*: for the conditional density of returns.
- Note of caution: *Aggregation*. We need the distribution over the period τ !

Estimating the VaR

Estimates of VaR based on historic data on prices. Assumptions:

- Stationarity of returns
- Independence of returns (must be relaxed)

Two general approaches:

- Non-parametric
- Parametric

Non-parametric estimation of VaR under independence

- Suppose that we want a confidence coefficient $1 - \alpha$ for the risk measures.
- We estimate the α -th quantile of the return distribution;
- This is estimated as the α quantile of the historical sample distribution of returns,
- The VaR estimated non parametrically is

$$\widehat{VaR}_t^{np}(\alpha) = -W_t \times \widehat{q}(\alpha)$$

where $W_t(\omega_t)$ is the size of the current position and $\widehat{q}(\cdot)$ is the estimated quantile.

- The estimate of the ES is

$$\widehat{ES}_t^{np}(\alpha) = -W_t \times \frac{\sum_{i=1}^T \tilde{r}_i}{M}$$

where $\tilde{r}_i = r_i \times I(r_i < \widehat{q}(\alpha))$, and $M = \sum_{i=1}^T I(r_i < \widehat{q}(\alpha))$.

Historical simulations in practice

- Simplest and fastest way of computing VaR and ES
- Choose window size, N .
- Consider the $T - N + 1$ overlapping sub-samples $\{r_1, \dots, r_N\}, \dots, \{r_{T-N+1}, \dots, r_T\}$.
- Each sub-sample is used to approximate the cdf of the data.
- For the VaR, sort each sub-sample for generic time t , as $\{\tilde{r}_{t-N+1}, \dots, \tilde{r}_t\}$.
- Choose the $\lfloor \alpha N \rfloor$ -th order statistic, then the VaR is

$$\widehat{VaR}_t^{hs}(\alpha) = -W_t \times \tilde{r}_{\alpha N, t}$$

- The expected shortfall is

$$\widehat{ES}_t^{hs}(\alpha) = -W_t \times \frac{1}{\lfloor \alpha N \rfloor} \sum_{i=1}^{\lfloor \alpha N \rfloor} \tilde{r}_{i,t}$$

- VaR and ES obtained with this method do not vary often enough.

Estimating VaR and ES in a time-varying framework

- The assumption of independence of daily returns is too restrictive in practice.
- Daily returns display a small degree of autocorrelation
- But a great amount of volatility clustering

The dynamics in the volatility can be modelled as

- Historical simulation approach (estimates based on rolling windows)
- Semi-parametric (EVT)
- Parametric approach (ARMA-GARCH, JP-Morgan)

Filtered Historical Simulations

- FHR refers to an hybrid mechanism.
- It relies on a simple resampling scheme,
- The term *filtered* refers to the fact that the quantiles are based the set of shocks, $\hat{z}_t = r_t / \hat{\sigma}_{t|t-1}$, which are returns filtered by the GARCH model.
- The percentile, $\hat{q}_z(\alpha)$, is calculated from the set of historical shocks, $\{\hat{z}_1, \hat{z}_2, \dots\}$.
- The VaR is

$$\widehat{VaR}_{t+1|t}^{fhs}(\alpha) = -W_t \times \hat{\sigma}_{t+1|t} \cdot \hat{q}_z(\alpha)$$

- The Expected Shortfall for the one-day horizon can be calculated as

$$\widehat{ES}_{t+1|t}^{fhs}(\alpha) = -W_t \times \hat{\sigma}_{t+1|t} \frac{1}{[\alpha N]} \sum_{i=1}^{[\alpha N]} \tilde{z}_{i,t}$$

GARCH-EVT model

- Estimate a GARCH model on the return series, r_t , for $t = 1, \dots, T$,
- Compute the standardized residuals $\hat{z}_t = r_t / \hat{\sigma}_{t|t-1}$,
- Estimate by ML the parameters, ξ, ψ , of the generalized Pareto distribution based on the N_u exceedances below a threshold u .
- Given ξ and ψ , the α quantile of \hat{z}_t is given by inverting the cdf of the exceedances, i.e.

$$\hat{q}_z(\alpha) = \begin{cases} u + \frac{-\hat{\psi}}{\hat{\xi}} \left(\left(\frac{T}{N_u} \alpha \right)^{-\hat{\xi}} - 1 \right), & \text{if } \xi \neq 0 \\ u + \hat{\psi} \log \left(\frac{T}{N_u} \alpha \right), & \text{if } \xi = 0 \end{cases}$$

- So the VaR is

$$\widehat{VaR}_{t+1|t}^{evt}(\alpha) = -W_t(\omega_t) \times \hat{\sigma}_{t+1|t} \hat{q}_z(\alpha)$$

- ...and the ES is

$$\widehat{ES}_{t+1|t}^{evt}(\alpha) = -W_t(\omega_t) \left(\frac{\hat{\sigma}_{t+1|t} \hat{q}_z(\alpha)}{1 - \hat{\xi}} - \frac{\hat{\psi} - \hat{\xi}u}{1 - \hat{\xi}} \right)$$

Parametric estimation of VaR under independence

- Parametric estimation is based on assumptions on the distribution of returns
- For example we can assume returns (or the standardized returns) to be Gaussian or Student's t distributed.
- Let $F(r|\theta)$ be a family of distributions used to model the return distribution and suppose $\hat{\theta}$ is an estimate of θ , then the VaR is

$$\widehat{VaR}_t^{par}(\alpha) = -W_t(\omega_t) \times F^{-1}(\alpha|\hat{\theta})$$

- $F(r|\theta)$ gives a full description of the probability of the returns for any α .

Parametric estimation of ES under independence

- The estimate of the ES is

$$\widehat{ES}_t^{par}(\alpha) = -\frac{W_t(\omega_t)}{\alpha} \times \int_{-\infty}^{F^{-1}(\alpha|\widehat{\theta})} x \cdot f(x|\widehat{\theta}) dx$$

- Computing this integral may be complicated for non-standard CDFs.
- If returns are Student's t distributed with mean, μ , scale λ and ν degrees of freedom, then

$$\widehat{ES}_t^{t-stud}(\alpha) = W_t(\omega_t) \times \left\{ -\mu + \lambda \cdot \left(\frac{f_\nu[F_\nu^{-1}(\alpha)]}{\alpha} \left[\frac{\nu + [F_\nu^{-1}(\alpha)]^2}{\alpha} \right] \right) \right\}$$

- Under Gaussianity

$$\widehat{ES}_t^{norm}(\alpha) = W_t(\omega_t) \times \left\{ -\mu + \sigma \cdot \left(\frac{\phi([\Phi^{-1}(\alpha)])}{\alpha} \right) \right\}$$

where $\phi(\cdot)$ is the density of the standard Gaussian distribution and $\Phi^{-1}(\alpha)$ is the inverse cdf (quantile function) of the standard Gaussian distribution.

Estimating VaR: time-varying volatility

- Assume that $\tau = 1$ and we have T returns that we need to estimate VaR and ES for next period $T + 1$.
- Let $\hat{\mu}_{t+1|t}$ and $\hat{\sigma}_{t+1|t}$ the conditional mean and volatility of tomorrow's return
- Under Gaussianity, the $VaR_t(\alpha)$ is

$$\widehat{VaR}_{t+1|t}(\alpha) = -W_t(\omega_t) \times \left\{ \hat{\mu}_{t+1|t} + \hat{\sigma}_{t+1|t} \Phi_\alpha^{-1} \right\}$$

- Under t-Student's distribution

$$\widehat{VaR}_{t+1|t}(\alpha) = -W_t(\omega_t) \times \left\{ \hat{\mu}_{t+1|t} + \hat{\lambda}_{t+1|t} q_\alpha(\hat{\nu}) \right\}$$

where $\hat{\lambda}_{t+1|t} = \sqrt{(\hat{\nu} - 2)/\hat{\nu}} \cdot \hat{\sigma}_{t+1|t}$.

Estimating VaR: Riskmetrics approach

- Very simple way to make variance a time-varying process (assume $\mu_{t|t-1} = 0$)
- The law of motion of the variance is

$$\hat{\sigma}_{t+1|t}^2 = \delta \cdot \hat{\sigma}_{t|t-1}^2 + (1 - \delta) \cdot r_t^2 \quad (3)$$

where $0 < \delta < 1$ and the usual choice is $\delta = 0.94, 0.96$.

- The initialization is $\sigma_{2|1}^2 = r_1^2$.
- Under Gaussianity, the $VaR_{t+1|t}(\alpha)$, with $\alpha = 1\%$ is

$$\widehat{VaR}_{t+1|t}^{JP}(\alpha) = -W_t(\omega_t) \times -2.326 \times \hat{\sigma}_{t+1|t}$$

- The $ES_t(\alpha)$ at $\alpha = 1\%$ is

$$\widehat{ES}_{t+1|t}^{JP}(\alpha) = W_t(\omega_t) \times \frac{\phi(-2.3226)}{0.01} \times \hat{\sigma}_{t+1|t}$$

- If RV is available, then the law of motion of volatility can be replaced with

$$\hat{\sigma}_{t+1|t}^2 = \delta \cdot \hat{\sigma}_{t|t-1}^2 + (1 - \delta) \cdot RV_t \quad (4)$$

Multi step ahead VaR

Assume that volatility is estimated by a GARCH process and returns are assumed to be conditionally Gaussian. Also assume $W_t(\omega_t) = 1$ and $\mu = 0$. We saw that

$$\widehat{VaR}_{t+1|t}(\alpha) = -\widehat{\sigma}_{t+1|t}\Phi_\alpha^{-1},$$

what about $\widehat{VaR}_{t+h|t}(\alpha)$ for $h > 1$? We have closed form predictions for the h -step ahead volatility $\widehat{\sigma}_{t+h|t}$ but we don't know the distribution of $Y_{t+h}|\mathcal{F}_t$. In the literature the following approximation has been proposed:

$$\widehat{VaR}_{t+h|t}(\alpha) \approx -\widehat{\sigma}_{t+h|t}\Phi_\alpha^{-1},$$

however, since we know that $Y_{t+h}|\mathcal{F}_t$ has tails fatter than a Gaussian random variable, this approximation can severely underestimate the true $VaR_{t+h|t}(\alpha)$. A solution is to use Monte Carlo simulation: i) simulate from $Y_{t+h}|\mathcal{F}_t$ and, ii) estimate $\widehat{VaR}_{t+h|t}(\alpha)$ as the empirical quantile of the simulated draws.

Multi Step ahead VaR: Gaussian approximation

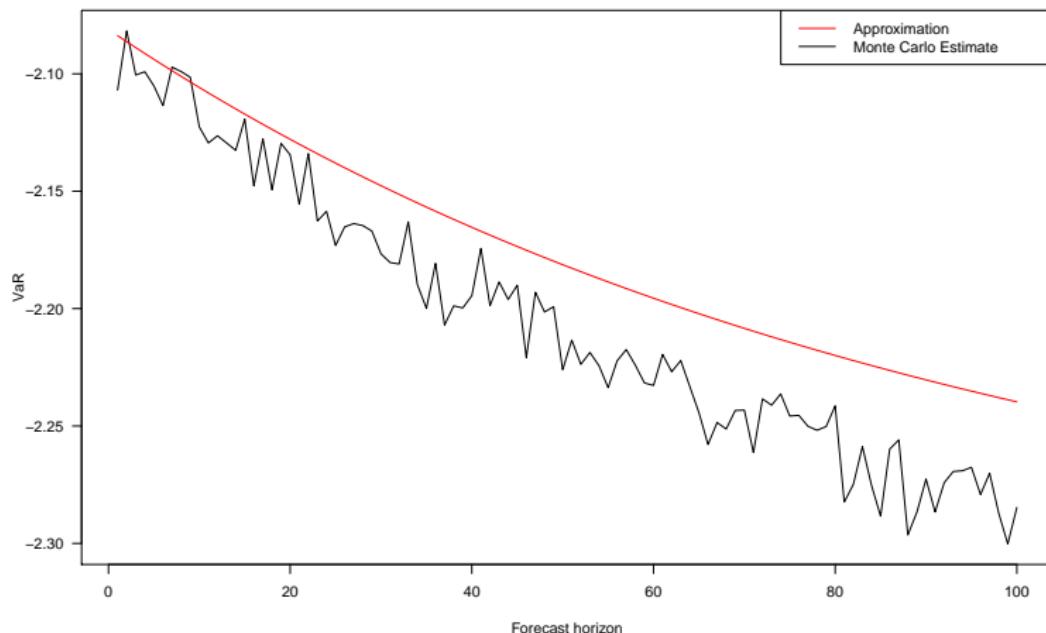


Figure: Comparison between the Gaussian approximated VaR and it's Monte Carlo estimate at $\alpha = 1\%$. See the script Lecture13.R

Back-testing VaR

- How can we compare alternative predictions of the VaR based on different econometric methods for the estimation?
- Several testing methods are generally employed
 - ① Kupiec test - unconditional coverage
 - ② Christoffersen test - conditional coverage
- Loss functions

Unconditional Coverage test

- Under the null hypothesis that the model is *correct*, the number of exceptions follows a binomial distribution.
- We denote $I_t(\alpha)$ the hit variable associated to the ex-post observation of a $VaR(\alpha)$ exception at time t.
- The unconditional probability of a violation must be equal to the coverage rate

$$P(I_t(\alpha) = 1) = E(I_t(\alpha)) = \alpha$$

- Each variable $I_t(\alpha)$ has a Bernoulli distribution with probability α .

BIS Traffic Light System



Zone	Number of exceptions	Increase in scaling factor
Green Zone	0	0,00
	1	0,00
	2	0,00
	3	0,00
	4	0,00
Yellow Zone	5	0,40
	6	0,50
	7	0,65
	8	0,75
	9	0,85
Red Zone	10 or more	1,00

Note: VaR(1%, 1 day), 250 daily observations

Kupiec test

- Kupiec (1995) test attempts to determine whether the observed frequency of exceptions is consistent with the frequency of expected exceptions according to the VaR model at α level.
- The test is a LR test

$$LR_{UC} = -2 \log \left[(1 - \alpha)^{T-H} \alpha^H \right] + 2 \log \left[(1 - H/T)^{T-H} (H/T)^H \right] \rightarrow \chi^2(1)$$

where $H = \sum_{t=1}^T I_t(\alpha)$ denotes the total number of exceedances.

Christoffersen test

- Problem with Kupiec test: Clustering in the exceedances.
- Need to test also for independence of violations.
- Christoffersen (1998) assumes that the violation process $I_t(\alpha)$ can be represented as a Markov chain with two states:

$$\Pi = \begin{bmatrix} 1 - \pi_{01} & \pi_{01} \\ 1 - \pi_{11} & \pi_{11} \end{bmatrix}$$

where $\pi_{ij} = Pr[I_t(\alpha) = j | I_{t-1}(\alpha) = i]$, i.e. the probability of an i on day $t-1$ being followed by a j on day t .

- Under independence, $\pi_{01} = \pi_{11}$.
- The idea behind is that clustered violations represent a signal of risk model misspecification

Christoffersen test

- The corresponding LR statistic for independence is defined by:

$$\begin{aligned} LR_{CC} &= -2 \log \left[(1 - \alpha)^{T-H} \alpha^H \right] \\ &\quad + 2 \log \left[(1 - \pi_{01})^{T_0 - T_{01}} \pi_{01}^{T_{01}} (1 - \pi_{11})^{T_1 - T_{11}} \pi_{11}^{T_{11}} \right] \rightarrow \chi^2(2) \end{aligned}$$

where T_{ij} denotes the number of observations with j followed by i , while $\hat{\pi}_{01} = T_{01}/T_0$ and $\hat{\pi}_{11} = T_{11}/T_1$.

Controlling for the magnitude of the violations

- Lopez (1998) proposes to evaluate the performance of different VaR forecasts based on loss functions
- The loss functions that reflect the concerns of financial institutions for large deviations

$$\text{Loss}_t^f = \begin{cases} 1 + (\text{VaR}_t(\alpha) - R_t)^2 & \text{if } R_t < -\text{VaR}_t(\alpha) \\ 0 & \text{if } R_t > -\text{VaR}_t(\alpha) \end{cases}$$

- Another widely used loss function is the quantile loss function:

$$Q_t = (R_t - \text{VaR}_t(\alpha))(\alpha - \mathbb{1}(R_t < \text{VaR}_t(\alpha)))$$

- From the point of view of the individual firm, the VaR is an opportunity cost, so they want to minimize it

$$\text{Loss}_t^i = \text{VaR}_t$$

SYSTEMIC RISK EVALUATION

Leopoldo Catania

Aarhus University and CReATES

leopoldo.catania@econ.au.dk

Based on: Girardi and Ergün (2013), Mainik and Schaanning (2014), and
Adrian and Brunnermeier (2016)

Introduction: Systemic Risk

- In times of financial crisis, losses spread across financial institutions, threatening the financial system as a whole.
- The spreading of distress gives rise to systemic risk: the risk that the capacity of the entire financial system is impaired, with potentially adverse consequences for the real economy.
- Spillovers across institutions can occur directly due to direct contractual links and heightened counterparty credit risk or indirectly through price effects and liquidity spirals.
- Systemic risk measures gauge the increase in tail comovement that can arise due to the spreading of financial distress across institutions.

Systemic Risk

- There is not a unique definition of Systemic Risk. However, most of the techniques used to evaluate systemic risk focus on the behaviour of a random variable, X , conditionally on a distress event affecting another random variable Y . Of course, the reaction of X to Y depends on the dependence between the two random variables, and, in general, from their joint distribution (X, Y) .
- Many of the systemic risk indicators available nowadays are somehow inspired by the work of Adrian and Brunnermeier (2016) (first working paper in 2008).
- Unfortunately, the systemic risk measure proposed by Adrian and Brunnermeier (2011) and used by Adrian and Brunnermeier (2016) has an important pitfall as detailed in Mainik and Schaanning (2014). Thus, we will focus on the measure introduced by Girardi and Ergün (2013).

CoVaR=

- Consider a random variable X with distribution $F_X(x)$, and another random variable Y , with distribution $F_Y(y)$. The joint distribution is indicated by $F_{X,Y}(x,y) = P(X \leq x, Y \leq y)$.
- The CoVaR measure introduced by Adrian and Brunnermeier (2011) is implicitly defined by:

$$P(X \leq \text{CoVaR}_{X|Y}^=(\alpha, \beta) | Y = \text{VaR}_Y(\beta)) = \alpha,$$

where we see that it is the α -quantile of the conditional distribution $X | Y = \text{VaR}_Y(\beta)$.

- $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ can be also defined as:

$$\text{CoVaR}_{X|Y}^=(\alpha, \beta) = F_{X|Y=\text{VaR}_Y(\beta)}^{-1}(\alpha)$$

$$\Delta CoVaR_{X|Y}^=(\alpha, \beta)$$

- CoVaR itself is not a measure of systemic risk. The measure of systemic risk is defined as:

$$\Delta CoVaR_{X|Y}^=(\alpha, \beta) = CoVaR_{X|Y}^=(\alpha, \beta) - CoVaR_{X|Y}^=(\alpha, 0.5),$$

that is the difference between the CoVaR conditionally on a distress event affecting Y , i.e. $Y = VaR_Y(\beta)$, $CoVaR_{X|Y}^=(\alpha, \beta)$ and the CoVaR conditionally on a “normal” (or benchmark) scenario $Y = VaR_Y(0.5)$.

- Note that $VaR_Y(0.5) = F_Y^{-1}(0.5)$ is equal to the median of Y .

$CoVaR_{X|Y}^=(\alpha, \beta)$ in a Gaussian world

- Assume $(X, Y) \sim N_2(\mu, \Sigma)$, where $\mu = (\mu_X, \mu_Y)$ and $\Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix}$, and $\sigma_{XY} = \sigma_X \sigma_Y \rho$.
- In this case we know that $X|Y = y$ is Gaussian with mean $\mu_{X|Y} = \mu_X + \rho \frac{\sigma_Y}{\sigma_X} (x - \mu_X)$ and variance $\sigma_{X|Y}^2 = \sigma_Y^2 (1 - \rho^2)^2$.
- A bit of algebra shows that in this case $CoVaR_{X|Y}^=(\alpha, \beta) = \mu_X + \sigma_X (\rho \Phi^{-1}(\beta) + \Phi^{-1}(\alpha) \sqrt{1 - \rho^2})$.
- Note that $CoVaR_{X|Y}^=(\alpha, \beta)$ does not depend from μ_Y and σ_Y and if $\rho = 0$ we have $CoVaR_{X|Y}^=(\alpha, \beta) = VaR_X(\alpha)$.

$\Delta CoVaR_{X|Y}^=(\alpha, \beta)$ in a Gaussian world

- Exploiting the definition of $\Delta CoVaR_{X|Y}^=(\alpha, \beta)$ we obtain

$$\Delta CoVaR_{X|Y}^=(\alpha, \beta) = \sigma_X \rho \Phi^{-1}(\beta)$$

Which does not depend on α .

- It is also evident that $\Delta CoVaR_{X|Y}^=(\alpha, \beta)$ in the Gaussian case does not bring any additional information than the correlation ρ .
- Similar results can be derived in the Student's t case. Mainik and Schaanning (2014) provides general results in the general elliptical case.

An important pitfall of $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$

- Mainik and Schaanning (2014) show that $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ is not an increasing function of the correlation between X and Y .
- This is counterintuitive since we expect the measure to increase for a higher level of association.
- It follows that $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ is not monotone in ρ .

Monotonicity of $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$

- To show this, consider the derivative of $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ with respect to ρ in the Gaussian:

$$\frac{\partial \text{CoVaR}_{X|Y}^=(\alpha, \beta)}{\partial \rho} = \sigma_X \left(\Phi^{-1}(\beta) - \frac{\rho \Phi^{-1}(\alpha)}{\sqrt{1 - \rho^2}} \right),$$

- We see that this function is positive if $\Phi^{-1}(\beta)\sqrt{1 - \rho^2} > \rho\Phi^{-1}(\alpha)$ and negative if $\Phi^{-1}(\beta)\sqrt{1 - \rho^2} < \rho\Phi^{-1}(\alpha)$.
- Furthermore, besides the degenerate case $\alpha = \beta = 1/2$, there are four cases depending on the sign of $\Phi^{-1}(\alpha)$ and $\Phi^{-1}(\beta)$:
 - If $\alpha \geq 1/2$ and $\beta \geq 1/2$, then $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ is increasing in ρ for $\rho < \rho_0$ and decreasing for $\rho > \rho_0$, where $\rho_0 = \frac{|\Phi^{-1}(\beta)|}{\sqrt{(\Phi^{-1}(\alpha))^2 + (\Phi^{-1}(\beta))^2}}$.
 - If $\alpha < 1/2$ and $\beta \geq 1/2$, then $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ is increasing in ρ for $\rho > -\rho_0$ and decreasing for $\rho < -\rho_0$.
 - If $\alpha \geq 1/2$ and $\beta < 1/2$, then $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ is increasing in ρ for $\rho < -\rho_0$ and decreasing for $\rho > -\rho_0$.
 - If $\alpha < 1/2$ and $\beta < 1/2$, then $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ is increasing in ρ for $\rho > \rho_0$ and decreasing for $\rho < \rho_0$.

Monotonicity of $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$

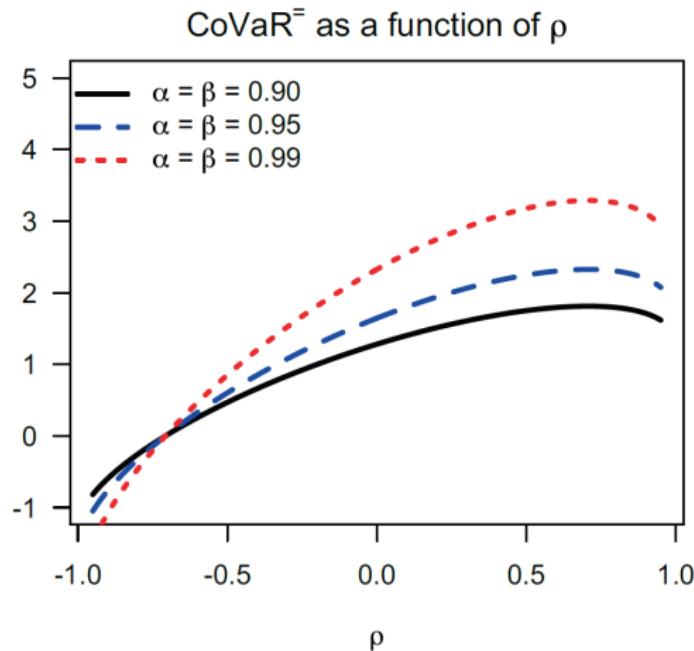


Figure: From Mainik and Schaanning (2014): $\text{CoVaR}_{X|Y}^=(\alpha, \beta)$ for a bivariate Gaussian distribution with correlation ρ .

Another CoVaR

- To solve the non monotonicity issue of $CoVaR_{X|Y}^=(\alpha, \beta)$, Girardi and Ergün (2013) have introduced another measure defined as:

$$P(X \leq CoVaR_{X|Y}(\alpha, \beta) | Y \leq VaR_Y(\beta)) = \alpha,$$

where we see that it is the α -quantile of the conditional distribution $X|Y \leq VaR_Y(\beta)$.

- $CoVaR_{X|Y}(\alpha, \beta)$ can be also defined as:

$$CoVaR_{X|Y}(\alpha, \beta) = F_{X|Y \leq VaR_Y(\beta)}^{-1}(\alpha)$$

CoVaR with inequality conditioning event

Unfortunately, we generally have no closed form for $\text{CoVaR}_{X|Y}(\alpha, \beta)$. However, we note that:

$$\begin{aligned} P(X \leq \text{CoVaR}_{X|Y}(\alpha, \beta) | Y \leq \text{VaR}_Y(\beta)) &= \frac{P(X \leq \text{CoVaR}_{X|Y}(\alpha, \beta), Y \leq \text{VaR}_Y(\beta))}{P(Y \leq \text{VaR}_Y(\beta))} \\ &= \frac{P(X \leq \text{CoVaR}_{X|Y}(\alpha, \beta), Y \leq \text{VaR}_Y(\beta))}{\beta}, \end{aligned}$$

That is: $\text{CoVaR}_{X|Y}(\alpha, \beta)$ is found as the solution of:

$$P(X \leq \text{CoVaR}_{X|Y}(\alpha, \beta), Y \leq \text{VaR}_Y(\beta)) = \alpha\beta$$

or

$$F_{X,Y}(\text{CoVaR}_{X|Y}(\alpha, \beta), \text{VaR}_Y(\beta)) = \alpha\beta$$

This equality can be solved using a rootfinder like the bisection method (see the `uniroot()` function in R).

Monotonicity of $CoVaR_{X|Y}(\alpha, \beta)$

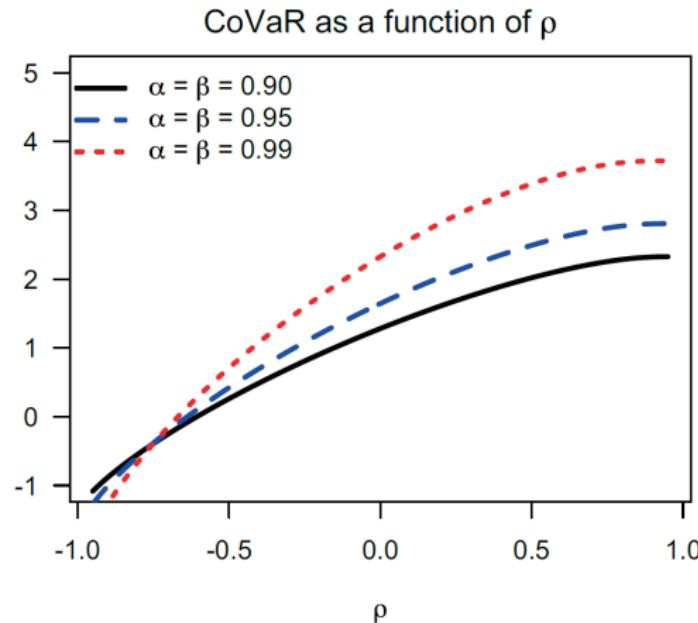


Figure: From Mainik and Schaanning (2014): $CoVaR_{X|Y}(\alpha, \beta)$ for a bivariate Gaussian distribution with correlation ρ .

$$\Delta CoVaR_{X|Y}(\alpha, \beta)$$

Girardi and Ergün (2013) also proposed another definition of $\Delta CoVaR_{X|Y}(\alpha, \beta)$ which is given by:

$$\Delta CoVaR_{X|Y}(\alpha, \beta) = 100 \times \frac{CoVaR_{X|Y}(\alpha, \beta) - CoVaR_{X|b(Y)}(\alpha)}{CoVaR_{X|b(Y)}(\alpha)},$$

where $CoVaR_{X|b(Y)}(\alpha, \beta)$ is implicitly defined as:

$$P(X \leq CoVaR_{X|b(Y)}(\alpha) | \mu_Y - \sigma_Y \leq Y \leq \mu_Y + \sigma_Y) = \alpha,$$

i.e., the benchmark state is $\mu_Y - \sigma_Y \leq Y \leq \mu_Y + \sigma_Y$.

Evaluating systemic risk

- It is important to see that we only need the joint distribution of the pair of Random Variables X and Y to evaluate the CoVaR.
- Of course, if the joint distribution is time-varying, we can compute CoVaR at each point in time. Furthermore, CoVaR can be predicted.
- Girardi and Ergün (2013) have estimated CoVaR assuming a bivariate DCC model for different pairs of assets.
- Their results show that during the sample period June 2000 to February 2008, depository institutions were the largest contributors to systemic risk, followed by broker-dealers, insurance companies, and non-depository institutions.

References I

- Adrian, T. and Brunnermeier, M. K. (2011). CoVaR. *NBER Working Paper No. 17454*.
- Adrian, T. and Brunnermeier, M. K. (2016). Covar. *American Economic Review*, 106(7):1705–41.
- Girardi, G. and Ergün, A. (2013). Systemic risk measurement: Multivariate GARCH estimation of CoVaR. *Journal of Banking & Finance*, 37(8):3169 – 3180.
- Mainik, G. and Schaanning, E. (2014). On dependence consistency of covar and some other systemic risk measures. *Statistics & Risk Modeling*, 31(1):49–77.