

Effective risk management with R

Pat Burns @ EARL

2014 September



This talk was given 2014 September 16 at the EARL Conference in London.

It started on a low note in that Markus (<http://www.magesblog.com/>) failed to deliver on his promise of a chorus line as a part of my introduction.

Two comments before diving in:

- 1) The topic is **financial** risk management, but all of us continuously balance risk and reward in both our personal and professional lives. So some of this can be put into a broader context.
- 2) There's a problem with this slide: I don't know anything about **effective** risk management.

Ineffective risk management with R

Pat Burns @ EARL

2014 September



I'm much better versed in **ineffective** risk management. But I didn't think the organizers of the conference would be keen on this title.

So I lied.

Why is Risk Hard?

Well what's the problem?

Risk manager job description:

**Don't be surprised
by Surprises**

Risk managers are supposed to foresee surprises.

And there will be surprises in finance.

Hyman Minsky

Hyman Minsky told us why there will be surprises.

Instability hypothesis

Minsky is best known for the instability hypothesis.

When the economy and markets are stable, then there are two reasons to bet big:

- 1) It is safe to bet big then. So why not?
- 2) When prices don't move much, then you have to bet big in order to make much money.

**"But, Yossarian, suppose everyone
felt that way."**

**"Then," said Yossarian, "I'd certainly
be a damned fool to feel any other
way, wouldn't I?"**

– Catch-22

Add in some Yossarian theory.

Instability hypothesis

**Stability implies
Leverage implies
Instability**

Reasons to bet big + Yossarian theory = Instability

When the instability manifests itself in a bad way,
then no one is so keen on betting big any more.

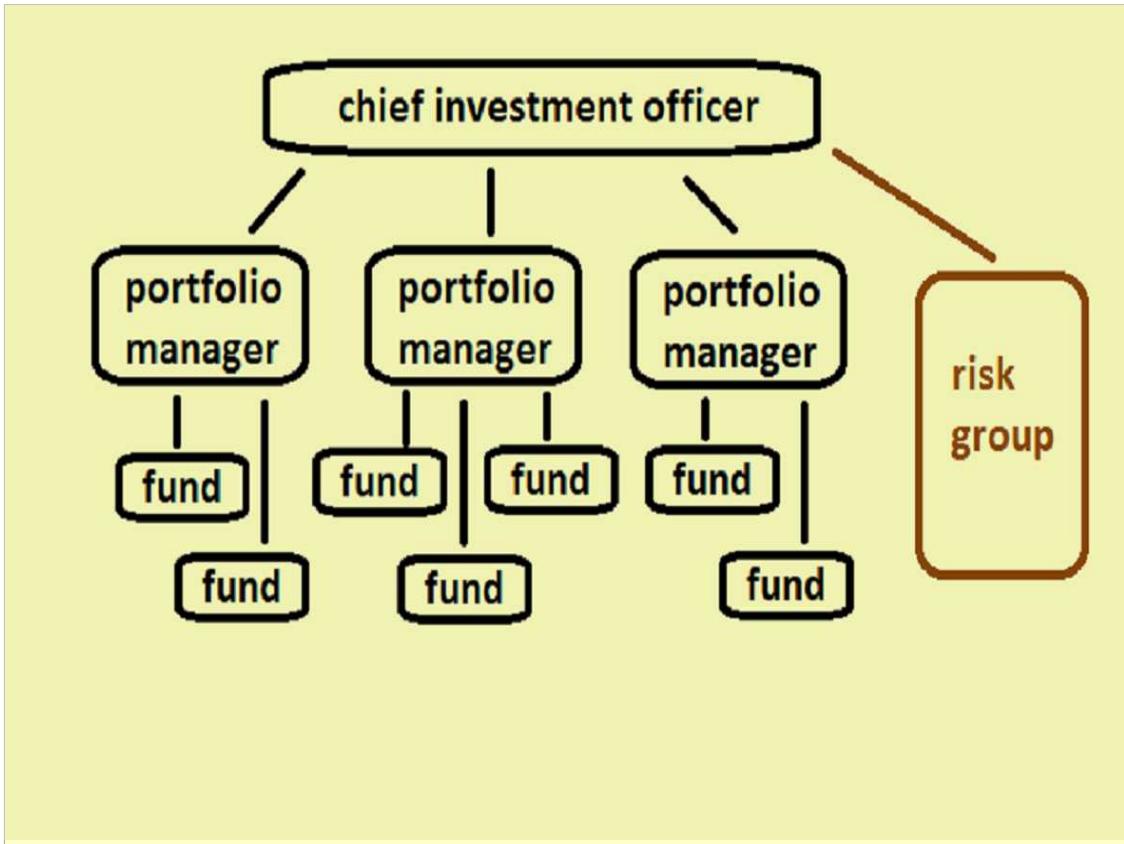
That is called a Minsky moment.

Autumn 2008 was a Minsky moment.



Another way of stating the instability hypothesis is that we have the collective memory of a goldfish.

Photo by voink via stock.xchng



Here is a typical structure of a large fund management company.

There are multiple fund managers. They typically have more than one fund, and they all report to the chief investment officer.

The risk group interacts with the fund managers, but is at the service of the CIO.

There are two types of risk to worry about:

- * the risk of each fund
- * the risk of the collection of funds. If all the funds do a little badly, that could be big, big trouble.



I see such a risk group having three major tasks.



Spot where the rocks are.

Public domain photo from US government



Get the chief investment officer on board their thinking.

Photo by whirlybird via stock.xchng



Encourage the funds to sail in formation.

Photo by mishie via stock.xchng

Spot rocks: computation, statistics

CIO on board: graphics

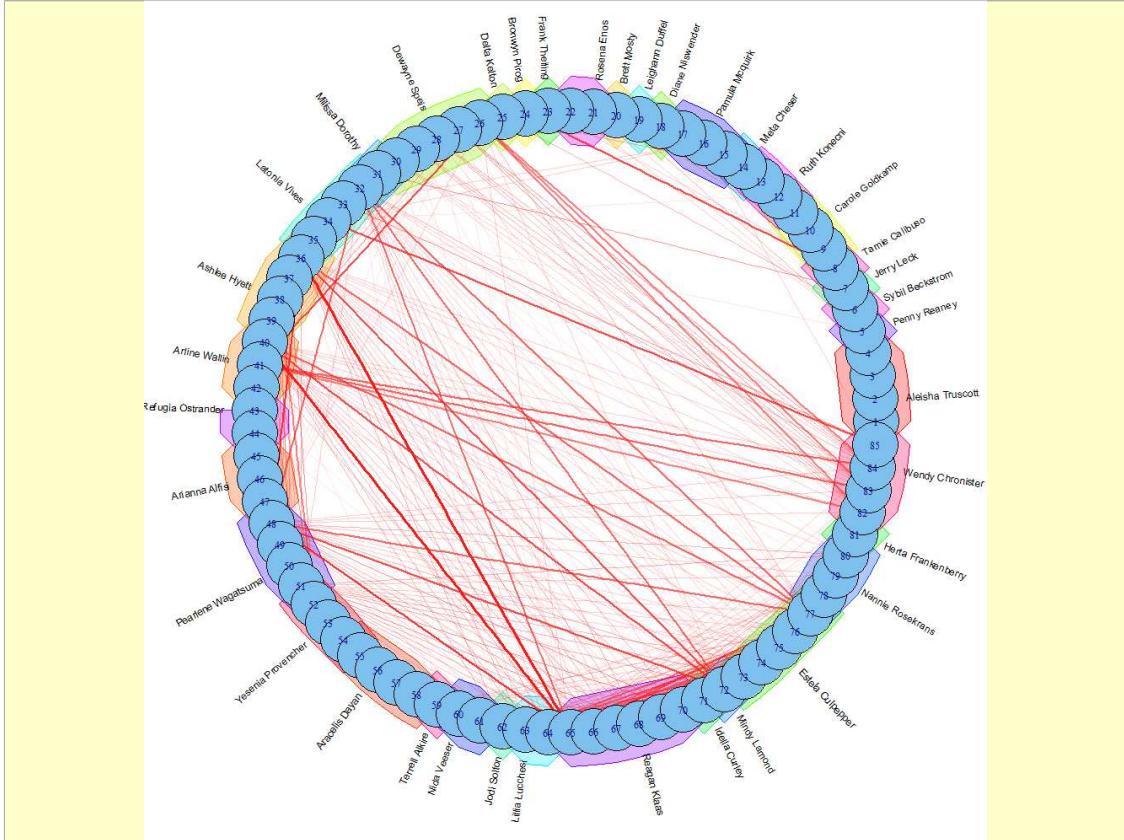
**Funds in formation: graphics,
automation**

What is in blue is what we need from a computing environment to achieve those tasks:

- * numerical computation
- * statistics
- * graphics
- * automation

That looks a whole lot like R to me.

I'll focus on graphics.



This is a graph in multiple senses (the R sense, and the mathematical sense).

Each blue circle is a fund. Funds are grouped by fund manager. A line between two funds represents the weight that is common to the two funds. The darker and thicker a line, the higher the common weight.

If you are looking at the names of the fund managers to try to figure out which fund management company this is, you will be frustrated – the names are randomly generated.

```
# get names from:  
# http://www.quietaffiliate.com/free-first-name-and-last-  
name-databases-csv-and-sql  
  
first_names <- drop(as.matrix(read.table(  
"C:\\\\Users\\\\burnspa\\\\Downloads\\\\CSV_Database_of_First_  
Names.csv",  
header=TRUE)))  
  
last_names <- drop(as.matrix(read.table(  
"C:\\\\Users\\\\burnspa\\\\Downloads\\\\CSV_Database_of_Last_  
Names.csv",  
header=TRUE)))  
  
nn <- length(unique(primCa$PManager))  
  
anony <- paste(sample(first_names, nn),  
sample(last_names, nn))
```

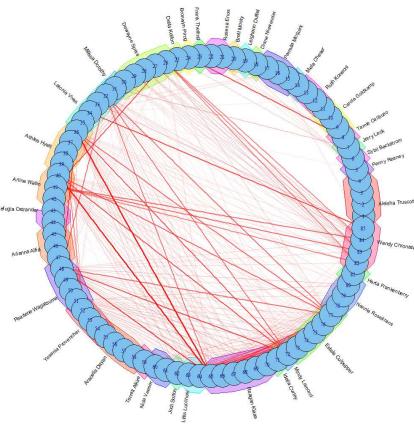
A good graph for a commercial environment is:

- **pretty**
- **informative**

What do we want for a presentation graph in a commercial environment?

We want aesthetically pleasing so that people will look at it in the first place. And keep looking.

We want it to be informative, or else what's the point?

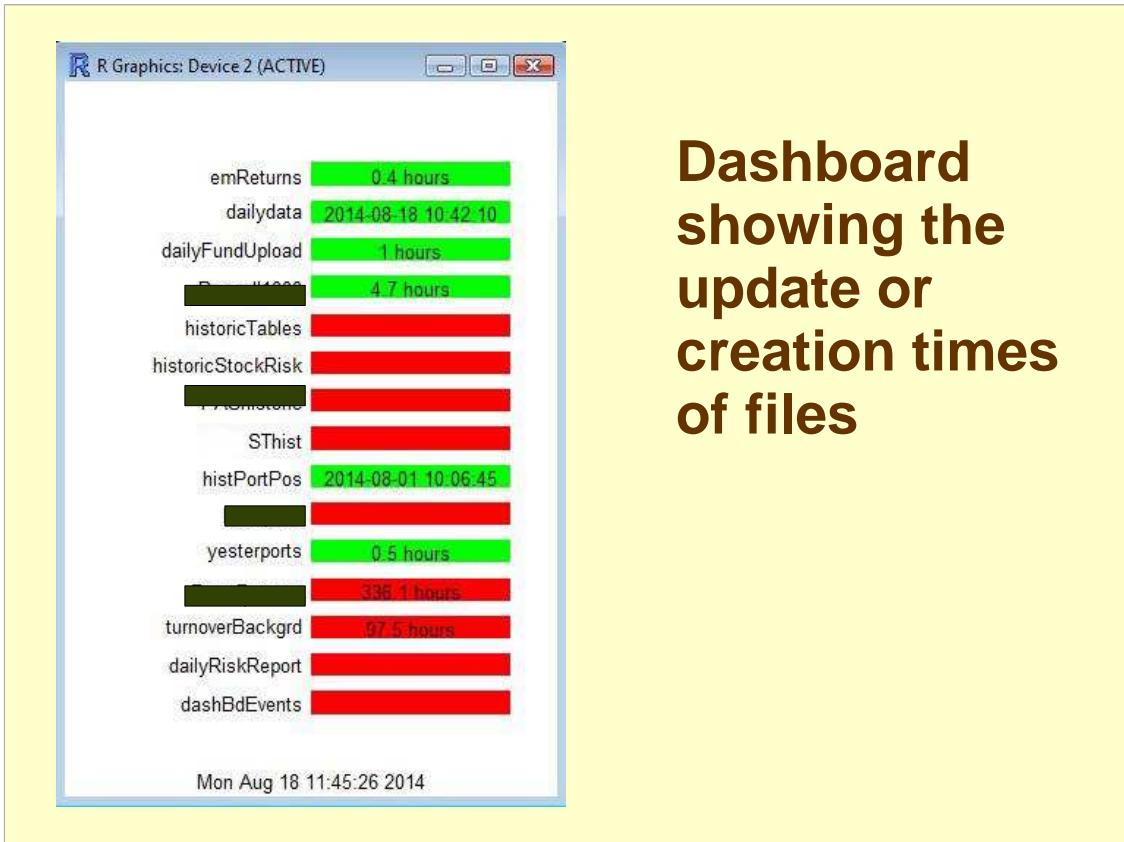


igraph package

This is common weights between funds grouped by portfolio manager

This plot is ever so pretty, but – for this application – useful information is not popping out at us.

Currently I see this style of plot as an answer in search of a question.



Dashboard showing the update or creation times of files

Here is a completely different sort of graph. Rather than presentation, it is for use within the risk group and is in aid of automation.

Most of the bars start the day red. As files are created and updated, more of the bars turn green.

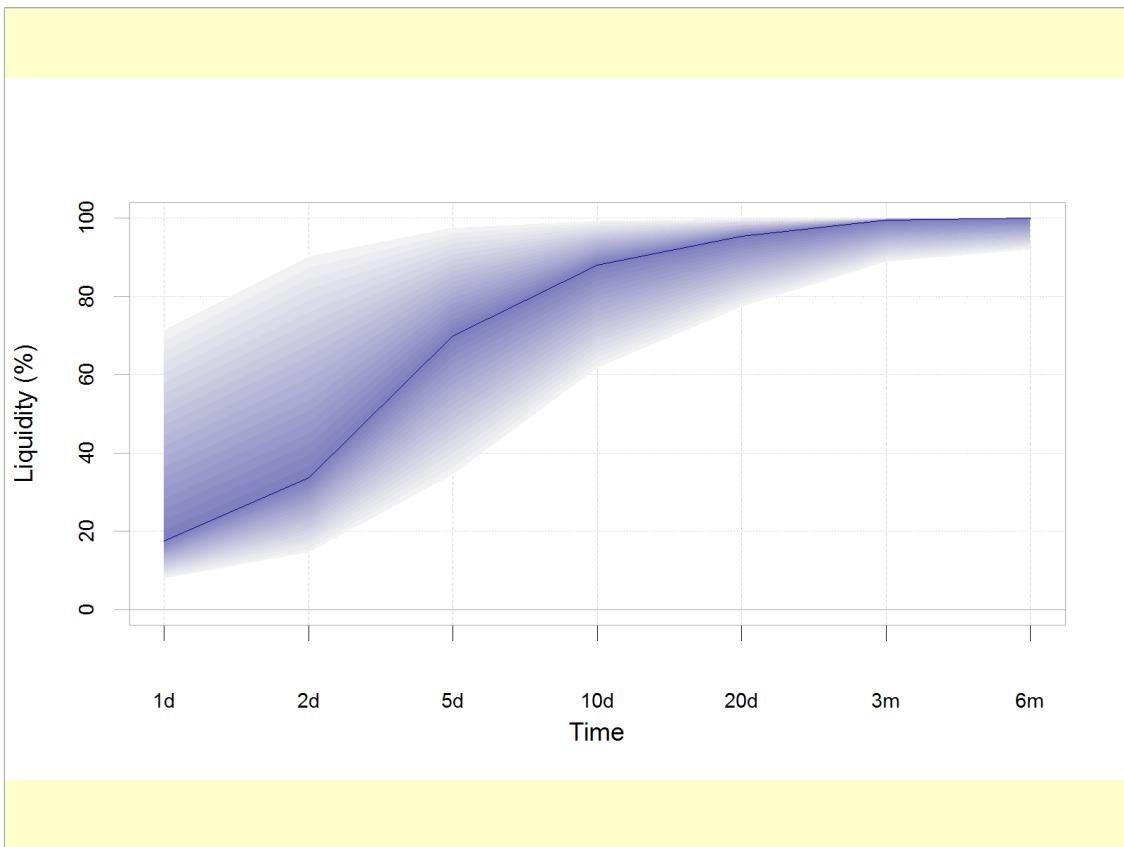
Once all the bars are green, there is hope that the process ran successfully.



The function that does this is just a repeat loop that:

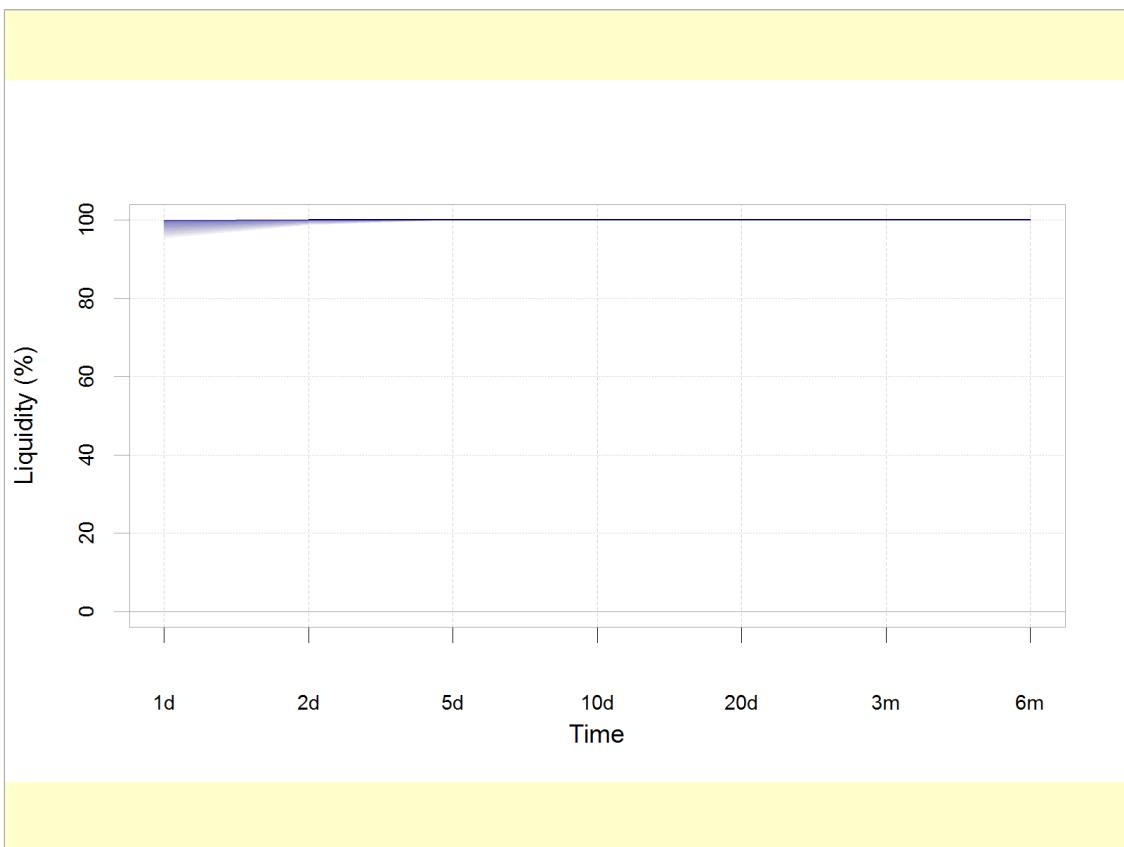
- * gets the necessary information
- * does the plotting
- * goes to sleep for a minute

It only exits if all the bars turn green, or if it is killed off.



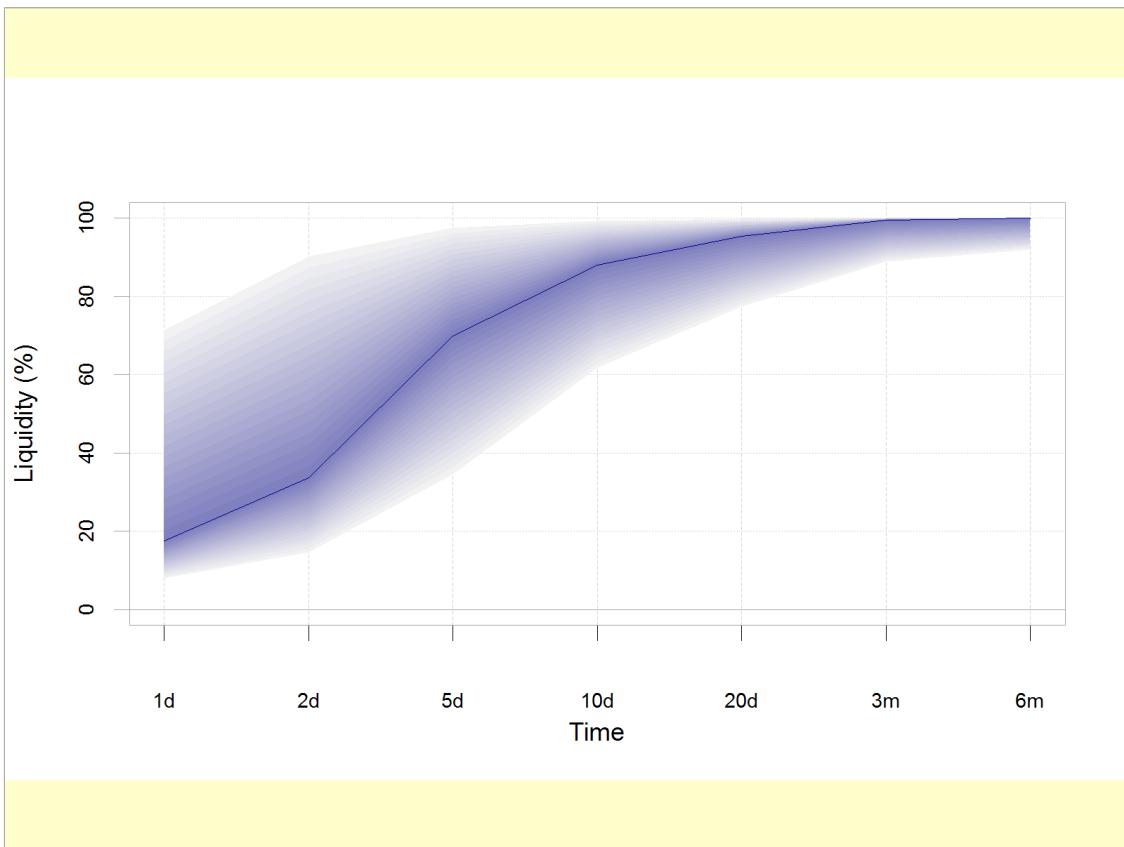
Now we are back to a presentation graph.

This one asks the question: for a particular equity fund how much of it can be sold off in specific lengths of time without getting killed by market impact?



All else being equal, this picture is less risky than the previous one.

But it is hard to believe that all else is equal. One way to have a fund that is easy to liquidate is if there is no money in it.

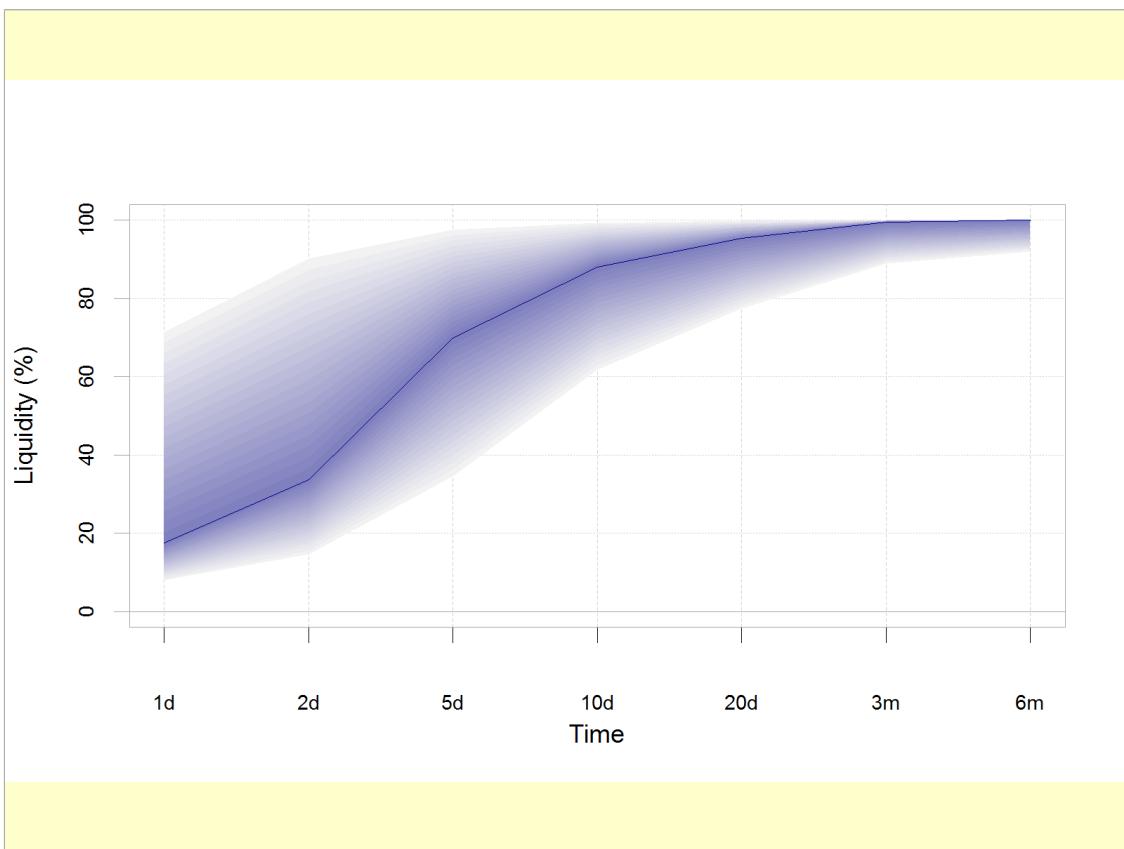


What are we looking at here?

The dark blue line is some specific fraction of current average daily volume. The extent of the coloring is essentially a 100% confidence interval for liquidity.

I think this passes the two criteria for graphics:
people react positively to how it looks, and what is perceived really is the information that is trying to be conveyed.

In the same session Richard Saldanha talked about a function with similar aims.



This is called a fan plot.

The trick to doing it in R is to plot a bunch of overlapping polygons of different colors. Plot the biggest polygon (in this case with the lightest color) first, then iterate until you get to the smallest polygon. Then add the central line.

R functions used include: `polygon`, `rgb`, `lines`

Effective risk management with R

Pat Burns @ EARL

2014 September



Remember when I said that I couldn't give a talk to this title?

I lied.

The trick is to re-interpret the meaning. If it is the risk of using R – meaning we are talking about bugs – then I can talk about that. I'm a pro at bugs.

Sentences from



I wish I'd written

A **bug** is a **test** you
forgot to **write**

I'm ever so jealous that I didn't think of this definition.

A good test suite is the most efficient way to get to quality code.

Packages for testing in R:

testthat

RUnit

There are two packages in R that help you to write test suites. Their functionality is very similar. I prefer testthat by a whisker.

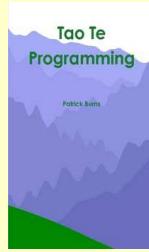
Almost surely either of them will be better than neither.

Commonality

Now we have two ways of thinking about risk: are we managing code or managing money?

Is there similarity between these?

Sentences from



I did write

A **bug** is an **assumption**
that is **violated**

As we've seen financial risk managers have very little power. But we aren't completely powerless.

Dissuade people from believing things they are really ignorant about

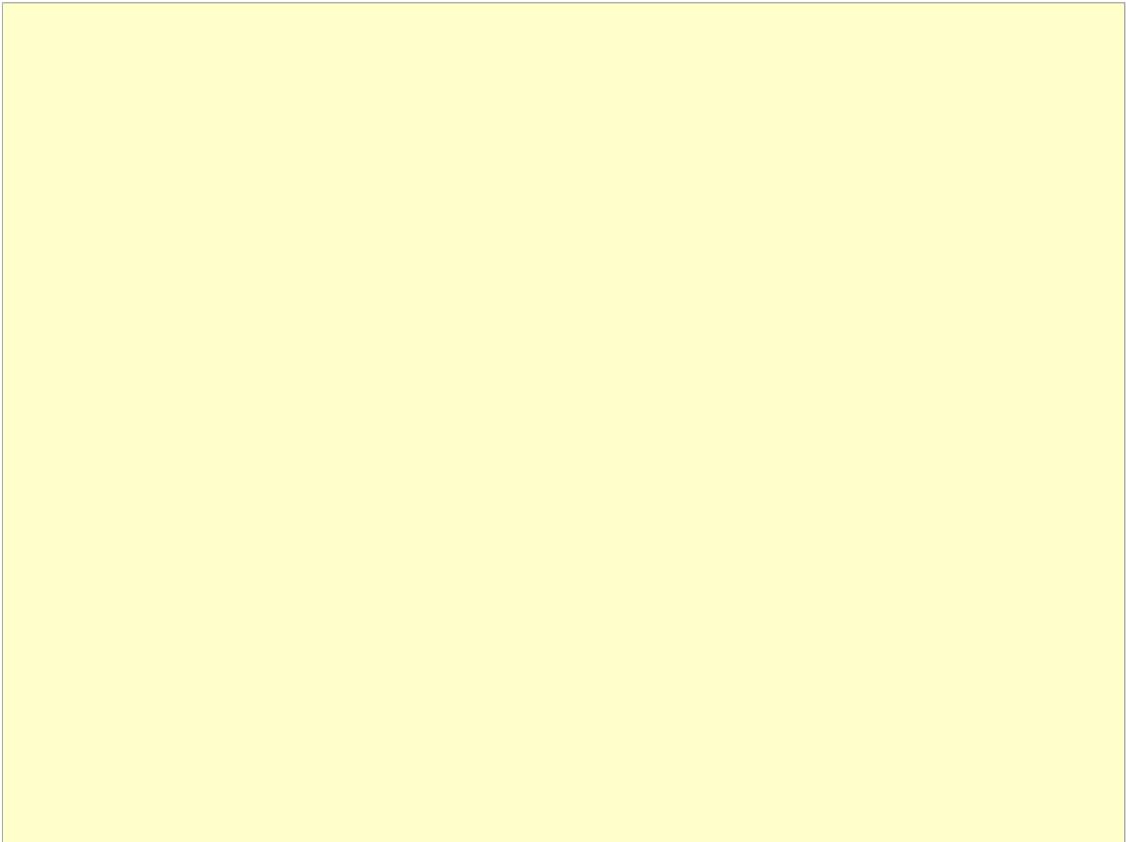
One thing we can – and should – do is help people be less ignorant of their ignorance.

"Because he's got flies in his eyes," Orr explained with exaggerated patience. "How can he see he's got flies in his eyes if he's got flies in his eyes?"

– Catch-22

Risk management is debugging the investment process

I'm led to this metaphor that I think **might** be useful.

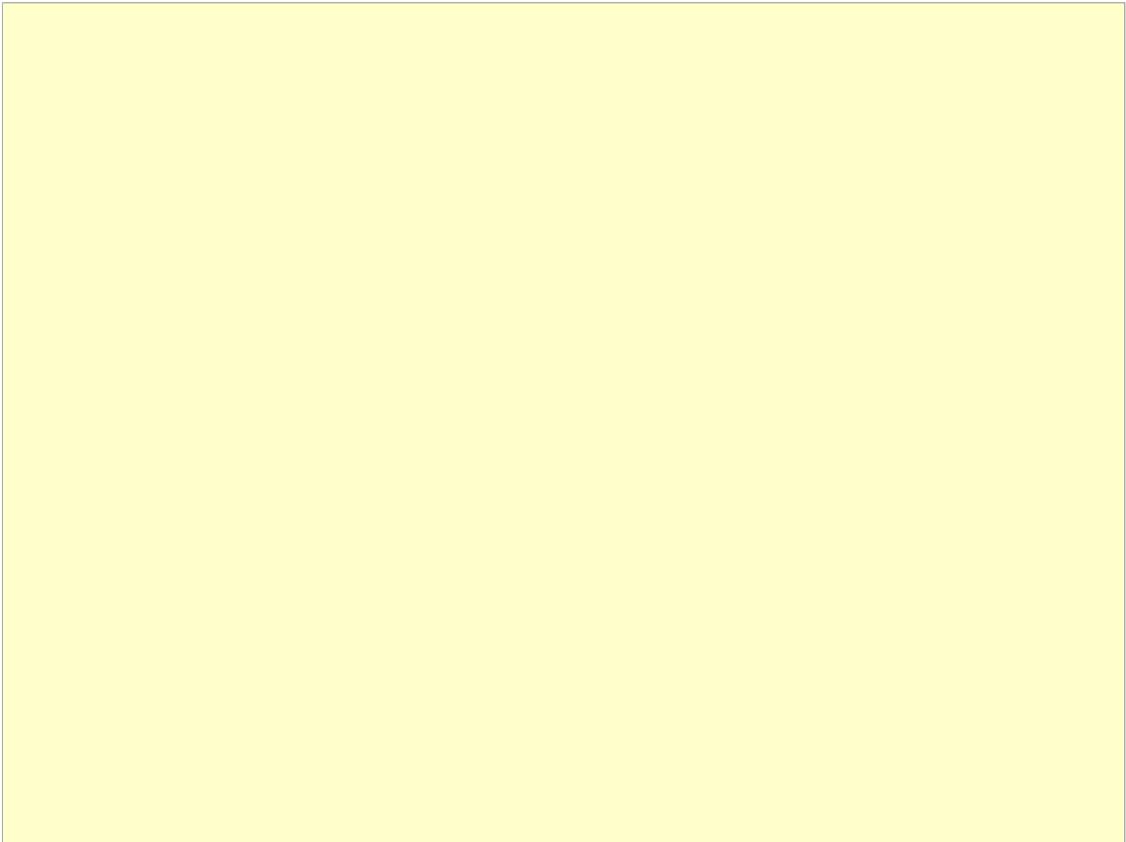


I purposely went through the talk fast with the firm belief that questions would be much more interesting than whatever other drivel I came up with.

The first question was: “How do you measure risk?”

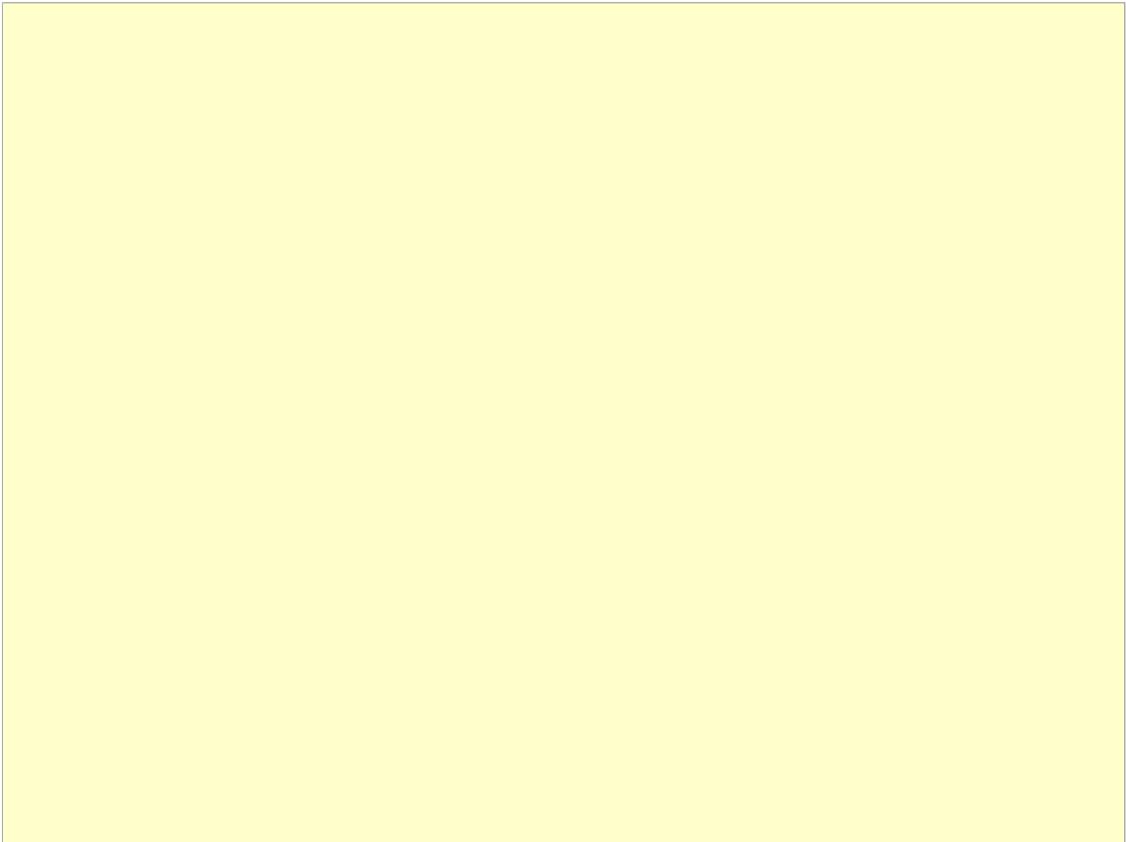
Thus I had the opportunity to immediately regret my strategy. Clearly the I-would-answer-you-but-we-don't-have-time trick wasn't going to work.

I stammered about for a while, mumbling about tracking error, benchmarks and peer groups. Finally I stopped myself and said, “We have some ideas about how to measure it, but really we don't know.”



Richard Saldanha offered his belief that random portfolios are one of the best ways to understand risk. That is the sort of comment that I might have prompted someone to say (but I didn't).

See <http://www.portfolioprobe.com> for more on random portfolios.

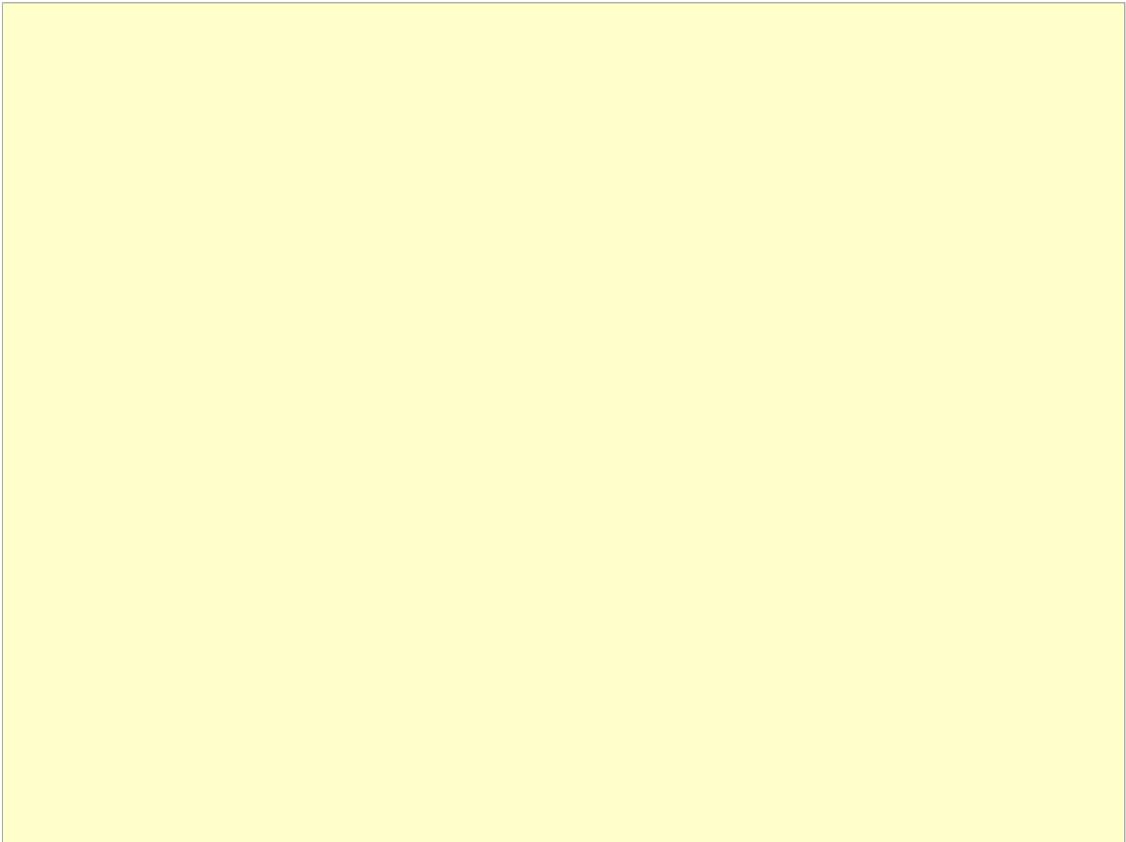


Another topic that was brought up was the risk of using Excel, with the London Whale being the specific case mentioned.

I said that the probability of a bug in R is probably just about the same as a bug in a spreadsheet when they are both first written.

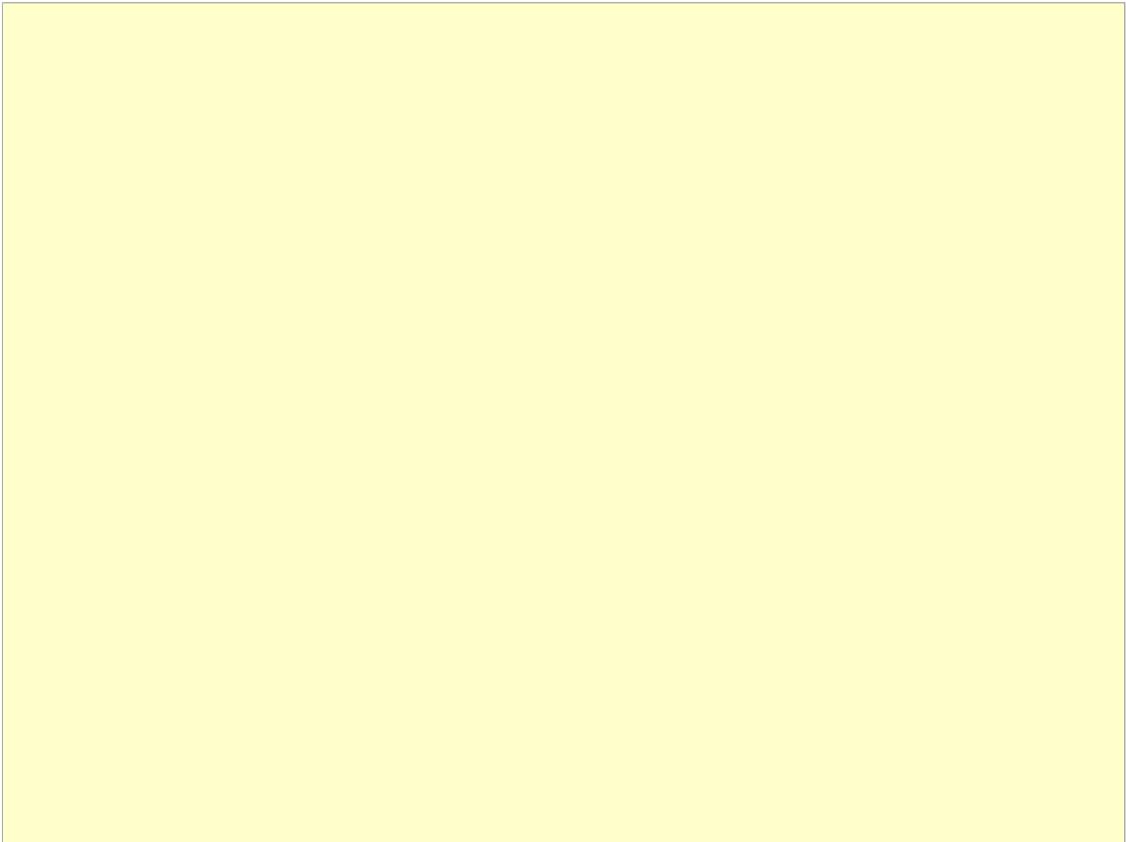
The difference is that in a language like R the bugs are very likely to decrease as time goes on, while the bugs in spreadsheets are much more in a steady state.

I should have also mentioned that test suites are feasible in languages like R, but a valid test suite of a spreadsheet is hard to imagine.



A statistic was offered stating the remarkably small fraction of packages on CRAN that depend on `testthat` or `Runit` (and hence presumably are a reasonable proxy for the packages with a formal test suite). I don't remember the number, but 10% is what pops into my head.

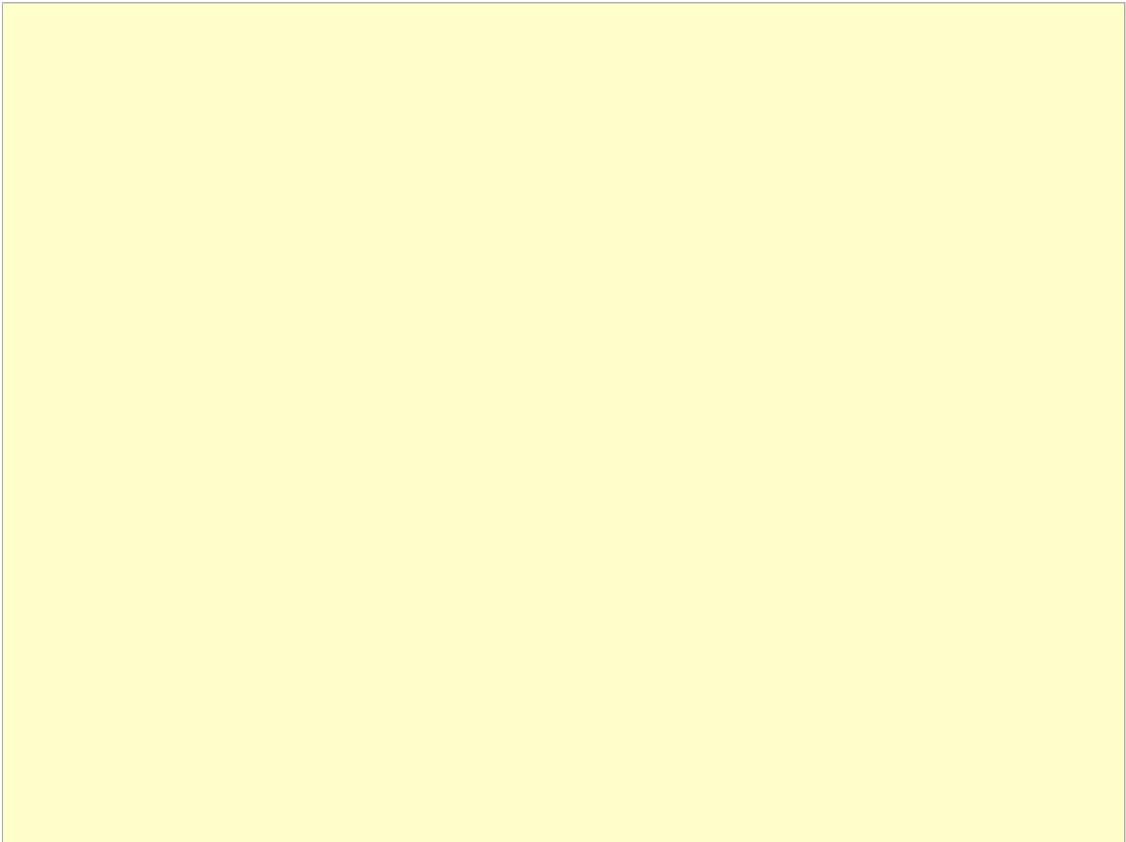
The examples being used as tests was brought up. I didn't comment, but actually I object to that. Yes, examples should work. But examples should be written to inform the uninitiated, and tests should be written to exercise the functionality. They are entirely different tasks. Trying to do both at once will do neither well.



What I did comment on was the difference between a CRAN package and a package written for a company (the mythical risk group in the talk, perhaps). I think CRAN packages should all have thorough test suites and thorough documentation. (Though there are advantages of not trying to formalize that.)

When working for a company the objective is to improve the profitability of the company as much as possible. There is a trade-off between spending time writing test suites and using that time elsewhere. Packages doing critical things should be well-tested, other packages need not be, and probably should not be.

The question is: what's the best use of your time?



The final question was from Tal Galili: “How do I know if the money in my pension is being managed well?”

The I-would-answer-you-but-we-don't-have-time trick did apply here.

I had the facetious reply: “Are you sure you want to know?”

Later he caught me privately when the time trick wouldn't work. My answer was that even when we have all the data we have a very hard time telling if a fund manager is making good decisions. And an individual is not going to be able to get much of the data.