G53 (08 22524): Risk Analytics

Colin Rowat
Room 220, J.G. Smith Building
c.rowat@bham.ac.uk
www.socscistaff.bham.ac.uk/rowat

2019-10-10

> regulator-approved standard risk models used in financial institutions …under-forecast risk before the announcement and over-forecast risk after the announcement, getting it wrong in all states of the world. Jon Danielsson
>
> You must understand, anyone who was worried was fired a long time ago and is not in this room. (Veteran risk manager to Ragjuram Rajan, spring 2007)
>
> It's a complete tempest in a teapot. Every bank has a major portfolio. In those portfolios, you make investments that you think are wise, that offset your exposures. Obviously, it's a big portfolio. We're a large company and we try to run it. It's sophisticated, well, obviously, a complex thing. But at the end of the day, that's our job, is to invest that portfolio wisely and intelligently to – over a long period of time to earn income and to offset other exposures we have. (Jamie Dimon, or Doug Braunstein?)
>
> The real risk is the one you can't see (Richard Bookstaber)
>
> Tim Geithner said he realised Merrill Lynch's risk culture was not in great shape when John Thain, then chief executive, did not know the name of his chief risk officer – who at the time was sitting next to him. …the driver of bank failure is not insufficient capital but rather a bad "risk culture". (Simon Samuels)

WARNING: Risk Analytics is taught to job market standards. It will be the most demanding module that you take this year. Its average mark is consistently the lowest of all modules in the Department of Economics; about a third of students fail their first sit of the exam. You can perform well on G53 if you prepare for lectures and problems classes in advance, and start early on the project. Students who attend all of the lectures (without reading the material in advance) and classes (without working through the problems in advance), and who successfully complete the term project (but without beginning it in the first term), consistently fail their first sit; if you follow their pattern, you can expect to as well. I tell students this every year; every year, students fail to prepare adequately, and fail. Some, as a result, do not obtain the MSc.

1 Goals

Risk Analytics is designed to train you to perform risk assessment and management, a high demand area of the financial services sector. You will be introduced to:

1. concepts of risks analysis and management, including market, credit and operational risk;
2. the theory of risk analysis, drawing from statistics, finance and computer science;
3. the practice of risk management.

Thus, you are to gain a conceptual framework and proper technical grounding for understanding risk, as well as learning how to actually practice risk analysis and management.

By the end of the module, you should be able to:

1. display a systematic understanding of the main risk measures available, the motivations underlying each, and be able to critically evaluate each of them, and suggest new measures that improve on them;
2. display a similar understanding of the major existing regulatory requirements associated with risk management, the limitations of these requirements, and be in a position to critically comment on drivers of future risk management regulation;
3. be able to assess, develop and apply strategies for managing portfolio risk, with a particular focus on understanding recent failures in risk management.

2 Lectures, classes and office hours

2.1 Lectures

See my.timetables for the lecture schedule. Note that I will be away on 21 and 28 October, so those lectures are cancelled, and replaced by lectures on 15 October and 6 November from 1 – 3pm.

2.2 Problem classes

See my.timetables for the weekly class schedule. Both pencil-and-paper and Python exercises will be set. Giovanni Guglielmi (GXG647@student.bham.ac.uk will lead the classes by asking volunteers to present answers to questions. You must prepare the problems in advance; failing to prepare properly for the classes will lead you to fall behind in the module, placing you at increased risk of failure. Gianni will help *you* find and fix mistakes in *your* solutions; he will *not* solve the problems for you.

Working through the pencil-and-paper problems before the classes is one of the best ways to prepare for the exam; working through the Python exercises will strengthen your skills with one of the most important software packages in quantitative finance. The document from which the coding exercises are drawn specifies MATLAB; you will need to transpose the problems into Python. You will need to draw on Python libraries like NumPy, SciPy and scikit-learn.

lecture	7 Oct	14 Oct	15 Oct	4 Nov	6 Nov	11 Nov	18 Nov	25 Nov	2 Dec	9 Dec
class	17 Oct	24 Oct	31 Oct	7 Nov	14 Nov	21 Nov	28 Nov	5 Dec	12 Dec	

Table 1: Dates of lectures and corresponding classes

You can arrange to see Gianni outside of the classes by e-mail.

2.3 Office hours

During term time, I have office hours on Mondays from 4:00 – 5:00pm and on Tuesdays from 9:00 – 10:00am and 4:00 – 5:00pm. In emergencies, I can arrange by e-mail to see you outside these hours. My office hours from 4pm on Tue 15 Oct through to 28 Oct are cancelled, and replaced by: 11–12 on 29 Oct, 9–10am, 11–1pm and 2–3pm on 30 Oct. I am always happy to correspond by e-mail.

3 Reading material and resources

The textbook for this module is Meucci (2005).[1] It is available online as an electronic resource via the library; sections of it can be downloaded from https://www.arpm.co, which also contains worked problems (used for problems classes), technical appendices, and MATLAB code. You may wish to join the LinkedIn group based on it.[2] In working through the problems, you should also become familiar with the appendices of Meucci (2005) and the online technical appendices. The primary strength of Meucci (2005) is its emphasis on solid methodological foundations; while these are applicable to modelling any aspect of risk, it devotes less effort to presentation and discussion of any but the simplest models.

The University also has access to the Numerical Algorithm Group's Python library.

Students will also receive an Interactive Brokers (IB) trading account.[3] This is an extremely powerful, industry-standard trading platform. Among other features, it allows users to run their own code on the trading platform via its API.[4]

McNeil, Frey and Embrechts (2015, Chapter 7) will be used for its material on copulas; a digitised version is available online here. McNeil, Frey and Embrechts (2015) is more extensive and often clearer in exposition than Meucci (2005); its webpage contains errata as well as some R code. Resti and Sironi (2007, Part V) will be cited on risk regulation. It is also comprehensive, devoting whole parts to models for each of the main classes of risk – market, credit and operational. The library has copies of both of these texts.[5]

Shin (2010) presents an alternative critique to the usual one, noting that widespread use of VaR can create *endogenous risk* and instability.

Whenever possible, you should read the original journal articles. You can find these either by using Google's www.scholar.google.com, or FindIt@Bham.[6]

If I hand material out in class, I shall also post it on my website. Any important announcements made during a lecture will also be posted there.[7]

4 Assessment

The module is assessed by means of a two hour final exam, accounting for 60% of the overall mark, and a 30% Python-based portfolio risk modelling project and two class tests (at 5% each).

[1] The 2009 edition corrects some misprints in the initial printing.

[2] Meucci teaches a one-week course in August, for which a certificate is available. The University of Birmingham has a partnership agreement with ARPM.

[3] You are free to continue using these accounts until the end of May.

[4] https://interactivebrokers.github.io/tws-api/introduction.html documents the Python API.

[5] Connor, Goldberg and Korajczyk (2010) also seems good, decomposing risk into industry, country and macroeconomic factors; it explicitly treats foreign exchange, credit and liquidity risks.

[6] If you are having IT difficulties, including finding articles online, please seek help either from a classmate or from the University's dedicated IT support as I am unable to provide this sort of support myself.

[7] If you miss a lecture, please catch up by first speaking to a classmate.

4.1 Final exam

The exam will be open book: a copy of Meucci (2005) and a binder of notes (including Meucci's technical appendix, worked problems, photocopied articles and chapters, etc.) should be brought into it. The only electronic devices allowed in the exam are calculators: laptop computers, e-book readers, etc. may not be used. The technical difficulty of the questions will be similar to those in Meucci's worked problems. Preparation for the exam should involve working through those. Before the exam there will be a revision lecture; you are responsible for bringing your questions to the lecture; I will help you answer them. You will be required to answer all questions on the exam.

4.2 Class tests

There will be a class test in the last 30 minutes of the 7 November class. On 5 December, a take-home test will be distributed at the end of the class; you are to submit a scanned copy of your answers to Canvas by 7pm on 6 December.

4.3 Term project

The term project is to use Python to build a trading system that maximises relative wealth against the MSCI World benchmark during its deployment period, subject to the constraint that its expected shortfall (ES) in the 1% tail is less than 1% of its value. After you submit the project, I will deploy your algorithm on my machine, run it, and assess its performance at the end of an hour.

1. Pick a unique 6-digit GICS industry code and five US firms in that industry whose stocks are on IB's student trading platform and Alpha Vantage.
2. By 5pm GMT 28 October, add your industry and firms to the list at

   https://pad.riseup.net/p/rl6GvL7DyTgiHws6fhAR

   I will then randomly match each of you to another student, your 'validator'.
3. Bind the five univariate distributions with a five-dimensional Archimedean copula of your choice.
4. In calculating ES, your time series modelling should use:
   (a) one of the six standard estimation methods mentioned in Daníelsson (2015).
   (b) one of the estimation methods available in scikit-learn.
5. Start developing your code as soon as possible. This is a serious project: students who leave it until next term will do poorly, and risk failing.
6. By noon on 3 February 2020, submit your project as a Jupyter Notebook to your validator.
7. As validator, submit a report (again, as a Jupyter Notebook) on the project sent to you by noon on 10 February 2020.[8]

[8] Check https://github.com/jupyterlab to see if a Commenting/Annotating repository has been created. If so, use it to add comments/annotations to the original .ipynb project. If not, indicate your comments in the .ipynb file by prefixing markdown with your initials.

8. By 5pm on 17 February 2020, submit the project via Canvas.

Your submission should have the following structure:

1. six page PowerPoint presentation to your clients, as follows:
   (a) title slide.
   (b) executive summary: a brief overview, including your understanding of your goals, your algorithm, how it works, how you tested it, how it performed, and why. Write this last, once you know.
   (c) algorithm and market: an intuition introduction to the algorithm and market you're trading in.
   (d) customisation/fine tuning: intuitively, what you did to tune your algorithm to work on your chosen market.
   (e) performance: how has your algorithm done when tested? Why?
   (f) conclusions: *specific* ways you can improve your platform or code.
2. a ZIP file containing a Jupyter Notebook technical document, including the .ipynb file and any supporting files. This must — first — allow me to run your algorithm, and — second — document your code in enough detail for me to understand what you have done (including how you've responded to your validator's report). You do not need to provide full derivations; references to standard texts are fine.
3. a ZIP file containing the validator's Jupyter Notebook report.

To pass, your code must – on my machine — run, open TWS, login to your IB account, close out all positions, reset your account balance to US$1mn, and trade for an hour.[9] You should use port #7497 and set the client ID to your student ID. All this should be fully automated: I should not need to manually log in or manually install packages.[10]

Beyond this, the project's mark will reflect its performance, ambition, the quality of its execution, and the clarity of its exposition.

Here are some more specific *do*'s and *don't*s on the project:

- *don't* ignore the module content to use techniques with which you are more familiar;
- *don't* go over the page limit, or otherwise ignore these instructions, even if you have something important to say;
- *do* work together in teams to develop code;
- *do* go beyond the standard Meucci approach if you can find a better solution;
- *do* re-use existing code, as long as you properly document this;
- *don't* tell clients about IID invariants;
- *don't* implement the wrong mandate, even if it's easier for you;
- *don't* blindly copy the code of the first student to get something running;

[9] Last year, we had some problems with TWS; if they return, you can use Alpha Vantage instead. This does not work as well: data are lagged; outside US trading hours, it does not execute submitted trades and may not update prices.

[10] I am running Python 3.6.7 in Anaconda 4.7.12 under Windows 7.

- *do* provide clear intuitions throughout;
- *do* label graphs properly, including axes;
- *do* describe your data sources;
- *don't* waste space in your technical document explaining generic matters (e.g. why we need to identify invariants).
- *don't* make vacuous suggestions for improving your algorithm (e.g. used longer time series, tested on more data, etc.)
- *do* write new code if it doesn't exist, and upload it to GitHub

To install Juypter Notebooks on your personal machine follow the instructions here, noting the strong recommendation to do so via Anaconda.

The IB platform will contain all the historical data required for the project. The IB Python API is designed to work via the command line, so occasionally runs into problems in a Notebook. (This stackoverflow thread documents a problem with argparse.) One workaround is to individually alter elements of the API. Another is to use a framework like IB insync to meld the API into non-command line environments like Notebooks.

Other financial libraries that can be used in Python include Alphalens, QuantLib, and Pyfolio.

Students can also use the Bloomberg terminals available in the Alan Walters building. (Familiarity with Bloomberg terminals would look good on a CV.) The University also subscribes to Thomson Reuters' Datastream and ThomsonONE.

References

Connor, Gregory, Lisa R. Goldberg and Robert A. Korajczyk (2010). *Portfolio Risk Analysis*. Princeton University Press.

Daníelsson, Jón (18 January 2015). *What the Swiss FX shock says about risk models*. Tech. rep. URL: `http://voxeu.org/article/what-swiss-fx-shock-says-about-risk-models`.

McNeil, Alexander J., Rüdiger Frey and Paul Embrechts (2015). *Quantitative Risk Management: Concepts, Techniques, and Tools*. revised. Princeton Series in Finance. Princeton University Press.

Meucci, Attilio (2005). *Risk and Asset Allocation*. Springer Finance. Springer.

Resti, Andrea and Andrea Sironi (2007). *Risk Management and Shareholders' Value in Banking*. Wiley Finance. John Wiley & Sons.

Shin, Hyun Song (2010). *Risk and liquidity*. Clarendon Lectures in Finance. Oxford: Oxford University Press.