Financial Analytics - Syllabus

Course Overview

Audience

This course is designed for analysts interested in pursuing a career in financial services with an emphasis on business analytics. Financial services traditionally includes banking, insurance, securities markets, and regulators. However, financial services are also found in organizational treasuries, departments of finance, CFO offices, investor relations, and any project where budgets, valuations, road maps, planning, and process efficiency require future projections of cash flows, growth rates, risk, and return. ## Course Overview

Description

An introduction to methods and tools useful in decision-making in the financial industry, which may from time to time include: macroeconomic event studies, analysis of term structures, equity data analysis, style analysis, credit risk, extreme value analytics, trading analytics, volatility measurement, portfolio management, and enterprise risk management.

Course credits

The successful completion of this course will earn 3 credit-hours.

Pre-requisites

I strongly urge students to complete work in Data Analysis (data management, statistics, optimization, visualization) and Managerial Finance (cash flow, present value, risk and return, portfolios, markets) before attempting Financial Analytics.

Linear algebra, calculus, Boolean logic, numerical methods, optimization, and programming languages are not strictly pre-requisites. However, work in these areas enhances one's numeracy, that is, capability to abstract complex, ill-posed business problems into tractable systematic models of behavior often mathematically expressed.

Throughout the course we will use technically accurate notions of simultaneous equations, optimization of single and multiple variable functions, cumulants (summation and aggregation), probability statements (for quantiles, confidence intervals, and hypothesis testing), linear model formulation and interpretation, random sampling, Monte Carlo simulation and bootstrapping. The course will introduce and review these areas as they emerge in financial analytic topics.

Resources

The main resources for the course are two books:

1. [Teetor] Paul Teetor. 2011. *R Cookbook*. O'Reilley: Sebastopol, CA..
2. [Foote] William G. Foote. 2017. *Financial Engineering Analytics: A Topical Manual Using R*. Manuscript available here.

The live sessions will expand on key aspects of each chapter in Foote and prepare the student for the weekly assignment. R scripts, `RMarkdown` source files, and data sets accompany each week.

The content and `R` implementation in Foote derive from several sources. Among the recommended sources are:

1. [Ruppert and Matteson] David Ruppert and David S. Matteson. 2014. *Statistics and Data Analysis for Financial Engineering, Second Edition*, Springer. The book's website is available here.
2. [Brealey, et al.] Richard Brealey, Stewart Myers, and Franklyn Allen, 2015. *Principles of Corporate Finance*, various editions, McGraw-Hill.
3. [McNeil, et al.] Alexander McNeil, Rudiger Frey, and Paul Embrechts, 2015. *Quantitative Risk Management: concepts, Techniques, and Tools* Princeton, 2014.
4. [Hastie, et al.] James Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, Springer Science & Business Media, 2009. The book's website is available here where you can also download a pdf copy of the book.
5. Numerous articles are available here on various financial modeling topics.

Ruppert and Matteson is a comprehensive text on the use of statistical and operational research techniques in financial engineering. McNeil, et al. is a far more rigorously mathematical treatment of the modern theory and practice of market, credit, and operational risk management. Brealey, et al. is a standard reference and course text in finance that has numerous worked examples and insightful explanations. Hastie, et al. is among the standards in statistical learning and data analytics.

In addition to Teetor's book, here are many `R` books useful for managing implementation of models in this course. Particularly useful `R` books include:

1. Phil Spector. *Data Manipulation with R.*
2. Norman Matloff. *The Art of R Programming: A Tour of Statistical Software Design.*
3. John Taveras. *R for Excel Users* at https://www.rforexcelusers.com/book/.
4. Winston Chang. 2014. *R Graphics Cookbook.* O'Reilley: Sebastopol, CA., with website here.

You will ultimately need all three (and whatever else you can get your hands on) in your professional work. John Taveras' book is an excellent bridge and compendium of Excel and R practices.

Several excellent online statistics resources exist. Among them is https://www.openintro.org/stat/textbook.php?stat_book=isrs developed at Johns Hopkins University.

`R` and `RStudio`

`R` is a free, open-source programming language for statistical computing. All of our work in this class can and will be done using `R`. You will need regular, reliable access to a computer running an up-to-date version of R. If this is a problem, let the professors know right away.

`RStudio` is a free, open-source R programming environment. It contains a built-in code editor, many features to make working with R easier, and works the same way across different operating systems. Use of RStudio is required for the course, and strongly recommended in general.

There are many online resources for learning about it and working with R and RStudio, in addition to the texts:

- The official introduction to `R` from the CRAN website, *An Introduction to R*, available at https://cran.r-project.org/doc/manuals/R-intro.pdf.

- John Verzani, *simpleR*, at https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf,
- The *Google* `R` *Style Guide*, at https://google.github.io/styleguide/Rguide.xml, offers rules for naming, spacing, etc., which are generally good ideas for reproducability and dissemination of analytical products.
- *Quick-R* at http://www.statmethods.net/. This site is primarily aimed at those who already know a commercial statistics package like SAS, SPSS or Stata, but it's very clear and well-organized, and others may find it useful as well.
- Patrick Burns, *The R Inferno* at http://www.burns-stat.com/pages/Tutor/R_inferno.pdf. "If you are using R and you think you're in hell, this is a map for you." - *Patrick Burns* at http://www.burns-stat.com/documents/books/the-r-inferno/
- Thomas Lumley, *R Fundamentals and Programming Techniques* (large PDF file at http://faculty.washington.edu/tlumley/Rcourse/R-fundamentals.pdf)
- Rstudio, *Building Shiny Applications* at http://shiny.rstudio.com/, with tutorials, galleries, example code, and articles. We will be building basic applications with Shiny, a package that will embed your analytical implementations, allow for user interaction and graphical display.

Assignment Formatting

All assignments must be turned in electronically, through the learning management system, by each student. All assignments will involve writing a combination of code and actual prose. You must submit your assignment in a format which allows for the combination of the two, and the automatic execution of all your code. The easiest way to do this is to use `R Markdown`. `R Markdown` also allows the use of interactive modeling through `Shiny` applications.

Work submitted as Word files, unformatted plain text, etc., are not acceptable at any time during the course. Each assignment will require the submission of at least one `R Markdown` script file and the `html` file that the `R Markdown` script generates. When using data sets, this course will only use `csv` (comma separated variable files generated by Excel or in text files. If the submission uses a `csv` file, that file must also be submitted with the `R Markdown` script and generated `html` output files. The student may also submit a supplemental `R` script file, suitably commented, that represents the `R` code chunks in the `R Markdown` script.

Managing the data base of submitted assignments throughout the course will be aided by standards including file name construction for assignment submission. To this end, every file submitted must have a file name which includes the student's name, course identifier, and clearly indicates the type of assignment (project) and its number (week). Here is the format we will use: `yourName_courseidentifier_Assignment#.ext`, where # is the week number and ext is the file name extension. For example W.G. Foote would submit an `RMarkdown` file with this filename: `wgfoote_SectionNo_Assignment1.Rmd`, where the file extension `Rmd` is the extension that `RStudio` uses for `R Markdown` documents. File extensions `R`, `html`, and `csv` are the other three admissible file types.

Other Matters

Grading

Grades are A-F with A > 95%, A- 90-95%, B+ 85-90%, B 80-85%, B- 75-80%, c+ 70-75%.

Grades for work performed in this course are distributed as follows.

1. Four (4) team projects, equally weighted 15% each, constitute 60% of the final grade. These are "all or nothing" grades. Teams will submit projects; instructors will evaluate the projects for content and technical components. If the evaluation results in less than 100%, the team will have the opportunity through the final week of course to revise and resubmit the project to improve the grade. Whatever grade is achieved by that time will be the final project grade.

2. Personal participation in the course, worth 15% of the final grade is measured by completing 6 asynchronous modules, each contributing 1% and totaling 6% of the final grade, and attending 9 live sessions (dropping lowest of the 10 live sessions), each contributing 1% and totaling 9% of the final grade. Participation includes attendance, completion of asynchchronous exercises, and contributions in the form of questions and provisional answers during the live sessions, especially in reports from break-out meetings during live sessions.

3. One (1) team project that represents the accumulated skills and capabilities learned throughout the course with sections that have primary authorships assigned to each team member constitutes 25% of the final grade. The grade for this project, out of 100%, will effectively determine the final grade for the course.

Students, in teams of two to four, will be given the same score for a completed team project. Individual final grades will be based on team, participation, and individual authorship of sections in the final project.

Grading will be based, as appropriate to the exercise and student response, 50% on technical and 50% on business fulfillment of requirements expressed or implied in questions and issues raised in the projects.

Grades for all assignments will follow this general rubric:

- **Words:** The written content responds to technical and business issues thoroughly. By thoroughtly in meant that all relevant questions have been exhausted in the sense that there are no further questions, given assumptions and scope, to be faced in the analysis. The text is laid out cleanly, with clear divisions and transitions between sections and sub-sections. The writing itself is well-organized, free of grammatical and other mechanical errors, divided into complete sentences, logically grouped into paragraphs and sections, and easy to follow from the presumed level of knowledge and assumptions of the analysis.

- **Numbers:** All numerical results or summaries are reported to suitable precision, and with appropriate measures of uncertainty attached when applicable.

- **Pictures:** All figures and tables shown are relevant to the argument for ultimate conclusions. Figures and tables are easy to read, with informative captions, titles, axis labels and legends, and are placed near the relevant pieces of text.

- **Code:** The code is formatted and organized so that it is easy for others to read and understand. It is indented, commented, and uses meaningful names. It only includes computations which are actually needed to answer the analytical questions, and avoids redundancy. Code borrowed from the notes, from books, or from resources found online is explicitly acknowledged and sourced in the comments. Functions or procedures not directly taken from the notes have accompanying tests which check whether the code does what it is supposed to. All code runs, and the `R Markdown` file `knits` to `pdf_document` output, or other output agreed with the instructor.

- **Modeling:** Model specifications are described clearly and in appropriate detail. There are clear explanations of how estimating the model helps to answer the analytical questions, and rationales for all modeling choices. If multiple models are compared, they are all clearly described, along with the rationale for considering multiple models, and the reasons for selecting one model over another, or for using multiple models simultaneously.

- **Inference:** The actual estimation and simulation of model parameters or estimated functions is technically correct. All calculations based on estimates are clearly explained, and also technically correct. All estimates or derived quantities are accompanied with appropriate measures of uncertainty.

- **Conclusions:** The substantive, analytical questions are all answered as precisely as the data and the model allow. The chain of reasoning from estimation results about the model, or derived quantities, to substantive conclusions is both clear and convincing. Contingent answers (for example, "if X, then Y, but if A, then B, else C") are likewise described as warranted by the model and data. If uncertainties

in the data and model mean the answers to some questions must be imprecise, this too is reflected in the conclusions.

- **Sources:** All sources used, whether in conversation, print, online, or otherwise, are listed and acknowledged where they used in code, words, pictures, and any other components of the analysis.

Course Specific Policies

Students are expected to behave in a professional and courteous manner at all times when interacting with all members of the course learning community. Respect for others is demonstrated through attendance, meaningful participation, and punctuality. Every effort should be made to be present for the entirety of each session especially since weekly assignments will be made conditional on content in live sessions.

All projects must be completed and submitted by the due dates and times set out. This will allow the entire class to review and revise submissions in a timely fashion. Submissions to the 2SU learning management system are based on eastern time. Late submissions will result in student inability to accumulate the knowledge needed to advance to the next week's coverage of course topics. Late submission will also delay necessary instructor feedback to the student in a timely fashion. As the course continues to layer on more skills and capabilities, a late submission with inaccurate or incorrect implementations of financial applications will only deprecate the student's ability to successfully complete future assignments.

Academic Integrity

Our Academic Integrity Policy holds students accountable for the integrity of the work they submit. Students should be familiar with the policy and know that it is their responsibility to learn about course-specific expectations, as well as about overall governing policy.

Our policy governs appropriate citation and use of sources, the integrity of work submitted in exams and assignments, and the veracity of signatures on attendance sheets and other verification of participation in class activities. The policy also prohibits students from submitting the same written work in more than one class without receiving written authorization in advance from both instructors.

In this course, all sources, whether verbal, online, in print, or other, must be cited following prevailing business and academic requirements and practice.

Disability-Related Accommodations

If you believe that you need accommodations for a disability, please contact us as soon as possible. Since accommodation may require early planning and services might not be available retroactively, please contact us as soon as possible.

Weekly Schedule

Weeks by topics for Summer 2019. For specific details on each topic, weekly work flow, and projects see the **Topics** section of this site.

Topic 1: R in Finance

Week 1: R Warm-Ups for Finance.

- `R` computations, data structures, financial, probability, and statistics calculations, visualization. Documentation with `R Markdown`.

Week 2: More R Warm-Ups.

- Functions, loops, control bootstrapping, simulation, and more visualization.

Topic 2. Macrofinancial Analysis.

Week 3: Descriptive Analytics

- Data from FRED, Yahoo, and other sources.

Week 4: Inferential Analytics

- Empirical characteristics of economic and financial time series. Boostrapping confidence intervals.
- Team project 1 due

Topic 3: Market Risk.

Week 5: Risk Measures

- Quantile (i.e., Value at Risk) and coherent (i.e., Expected Shortfall) risk measures.
- Final project one page synopsis due

Week 6: Portfolio Loss and Capital Requirements

- Historical and General Pareto Distribution estimation and simulation for portfolio capital measurement.
- Team project 2 due

Topic 4: Portfolio Analytics

Week 7: Cash and Risky Asset

- Risk tolerances, thresholds and the allocation of collateral and risky asset.
- Final project data sets exploration due

Week 8: Tangency Portfolio

- Minimum variance, tangency, and optimal allocation with multiple risky assets and quadratic programming.
- Team project 3 due

Topic 5: Enterprise Risk Analytics

Week 9: Enteprise Risk Simulation

- Specifying and simulating enterprise risk scenarios as a portfolio with quantile and coherent risk measures.
- Final project initial draft due

Week 10: Build the Financial Application

- Build an interactive financial web application using all of the finance and `R` platform tools
- Team project 4 due

Week 11: Final Project

- Team and individaul final project due with portfolio analytics
- All team project updates due

Data

All of the data in the topics comes from R packages or from the zipped file you can download from here.

Here is a video that runs through the following data download work flow.

1. Set up a working directory on your computer. Typically this is located in the user's documents directory. In this working directory you will save *.Rmd files.
2. Within the working directory, set up a data directory called `data`. This is a subdirectory of your working directory.
3. Download all of the data into the `/data` directory on your computer. You will be accessing this data, for the most part `CSV` (comma separated values) files using, as an example, the following code.

Be sure to set up your files in a directory, then set the working directory as the source file location. In that way your call to `/data/metals.csv` (for example) will connect. Otherwise you will get the following error:

```
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'data/metals.csv': No such file or directory
```

The **fix** is to properly set your working directory. In Rstudio on the tool bar go to **Session > Set Working Directory > Source File Location**. If you have set up the `data/` folder with the data as a sub-direcctory of the directory which is your source file location, `read.csv()` will be able to connect to your data set.

R Documentation

Here is a set of directions, glossary, examples, and general advice for building an `R` infrastructure for financial analytics.

Here is a tutorial produced by the RStudio team that helps you install `R`, `RStudio` and some useful packages.

It is expected that R, RStudio, and LaTeX will be installed on your computers in time for the first live session. If you are having difficulty with installations, please post your issue(s) on the course section's wall. Others (including your instructors) may have had similar experiences and may have worked a solution already.

