Group Project Instructions

1 Group Project

The instructions in this document are quite detailed. The reasons are:

1. As the software for your project grows to encompass more than a few isolated lines of code snippets, it will quickly get out of hand if you do not use a solid system for keeping track of files and folders. It is therefore essential to organize your project, and (believe it or not) **you will actually *save* yourself a lot of time if you follow the project structure described in this document, especially Section 1.7**.

2. The detailed instructions allow the project to run smoothly for *all* participants (and not just for *some* who might be tempted to game the system). This way *all* students can profit to the maximum extent from the new knowledge and experience learned in this project.

To disincentivize potential free-riders and to have a fair and consistent evaluation for all students, **any violation of the rules and instructions in this document** (e.g. late handing in of the presentation slides to the TA before the presentation, going overtime when presenting, failing to reply to emails from the instructor or TA within 24 hours, late handing in of the group project, etc.) will have a **negative impact on your evaluation**, except under special circumstances.

1.1 Overview

The group project is on the design of an equity-linked note or capital guarantee fund. The note can be linked with an equity index, a basket of stocks, an exchange rate, commodity, other traded assets or even non-traded events.

The group project consists of two parts:

1. In-class group presentation. Your group presents your ideas and/or preliminary results in class. The instructor and/or the TA will provide feedback and suggestions/comments on your project.

2. Group report. Your group incorporates the feedback received during the presentation and delivers a written group report.

In the group presentation and group report you should discuss the following items:

- Economic motivation for your product: Discuss and explain recent economic developments and trends that are relevant for your product. Based on this, show how investors can benefit from buying your product. In this item, you should mainly focus on economic intuition.

- Design: State how much coupon or dividend you will have, what is the participation ratio, and what the buyers will get at the end of maturity.
- Pricing methodology: Explain how you determine the coupons, dividends, and participation ratio by studying the current value of the embedded options.
- Investment and hedging strategy: Offer your investment advice to the retail investor and hedging advice to the bank that offers the products. Explain how you calculate the Δ if a delta-hedging strategy is used.

In the past, many groups have spent a relatively large amount of time during their presentation on risk analysis. For example, they have discussed market risk, liquidity risk, and counterparty risk. While all of these risks are important, they are fairly general and *essentially the same* for all group projects. In other words, in most cases, there is nothing in the risk analysis that is specific or unique with regard to your group project. As a consequence, I think it is better to remove the part on risk analysis from the presentation and instead spend more time on other parts of your group project that are more economically relevant.

1.2 Group Formation

Each group should consist of **three to four** students. Please form your group at the latest **on the day after the add/drop period** and notify the TA, otherwise the TA will randomly assign group memberships.

Please form your groups as soon as possible, nominate a group leader, and inform the TA. Part of the group leader's job is to be the group's main contact person to the instructor and TA, e.g. for scheduling the group presentations and other things. For this, the group leader's responsibility is to check his email on a daily basis and reply to any emails from the instructor or TA *within 24 hours* to ensure that there are no material delays in the whole organizational process, which may involve coordinating several groups at once. The group leader should give the TA the list of names of your group members at the latest **on the day after the add/drop period**. If you have not joined a group by that date, the TA will randomly assign group memberships and no changes can be made afterwards. No group switching is allowed over the semester. You may contact the TA for the latest group arrangements.

1.3 Duration of Each Presentation

Each group has at most **18 minutes** for their presentation in total (including discussion, which may last for several minutes).

If the announced presentation time might seem to be short, it simulates real life. In real-life situations, you also will have to convince your customers, your boss, or your colleagues that you have a fantastic idea or proposal. And in a real-life situation you will also have to do this in a limited amount of time. So let's take this opportunity to practice your presentation skills under real-life conditions.

1.4 Dates of the Group Presentations

The **last day of class** will definitely have presentations, but depending on the class size after the add/drop period, we might also have presentations on the **second-to-last day of class**. Therefore, you should **email your group's preferred ranking of presentation dates to the TA**. If the class size after the add/drop period is sufficiently small, all students will present on the last day of class. On the other hand, if the class size after the add/drop period requires two presentation dates, we will use your preferred ranking to allocate presentation dates on a **first-come, first-served basis**.

If the TA has not received your ranking on the **day after the add/drop period**, the TA will randomly determine your presentation date and no changes can be made afterwards. We will try our best to accommodate your date preferences, but we cannot make any guarantees. Please keep in mind that the presentations are an important part of your final grade for this course and that it **is your duty to be present in class on the presentation date assigned to your group**. Please also keep in mind that following University regulations, even if your group is not presenting on a given day (e.g. when other groups present), class attendance will still be recorded as usual.

According to the syllabus and the group project instructions, it is your duty to be present in class on whichever date is allocated to you by the TA. Nonetheless, in case you are facing a special situation where you cannot present on one of these dates, we will try to accommodate this as far as it is feasible. As such, we might or might not make an exception, depending on your particular case, and move your presentation to another date. In case you are facing such a special situation, your group leader should write an email to the instructor and the TA as soon as possible (at the latest two weeks before the first presentation date). Please keep in mind that according to the syllabus, there is no "right" to a special treatment, and that your request for a particular presentation date might be granted or denied.

1.5 Deadline for Handing In the Group Report

The deadline for handing in the group report is at **11:59pm on the fifth working day after the last presentation**. For example, if the last presenation is on a Tuesday and if there is no public holiday in between, then the deadline is on Tuesday of the following week at 11:59pm.

You can still incorporate the feedback received during the presentation and give a final polish to your group report.

1.6 Group Presentation

The order of the presentations (e.g. which group presents first on a given day) will be determined randomly on each presentation day. So please make sure you arrive punctually to class in case your group is randomly chosen to present first.

Every member of your group should present a part of your presentation slides. Every student will be evaluated individually on his part of the presentation.

Keep in mind that the evaluation of the presentation will be about *how* you present, and not so much about *what* you present (e.g. the content of the slides, which will be evaluated separately when you later hand in your whole group project). Of course, this does *not* mean that the content is unimportant. For example, during the presentation I might ask some clarifying questions or I might give you some feedback on how to improve your product, and you are expected to have in-depth knowledge about your part of the presentation. However, it will not be “easier” to get a high mark if you present a very technical part, since, as mentioned above, it matters more *how* you present something as opposed to *what* you present.

There are a few simple rules that can improve anyone’s presentation skills and that will be used to assess your individual presentation:

- Make and keep eye contact with the audience.

- Smile a bit (you have exciting material to share with the class!).
- Talk to the audience, not to the screen or to the monitor.
- Talk slowly and add pauses.
- Don't put your hands in your pockets.
- Don't pace around on the stage. Find a comfortable place to stand and stay there. Usually it is a good idea to stand to the left of the screen from the audience's point of view, but this also depends on the room layout. (This does not mean you are not allowed to walk around; it only means you should avoid walking around continuously, creating a nervous atmosphere.)
- "A word is a word, but a picture paints a thousand words." – Anon. Use graphics, but keep them simple (don't overload the slides, avoid clip art).
- Practice your speech at home. If you have the opportunity, videotape yourself while doing the practice presentation and watch yourself afterwards. (No need to hand in the video; it is just for yourself and you can delete it afterwards.) You will often be surprised when watching yourself on video how *your self-perception differs from the way other people perceive you*. It will often be a revealing experience that will help you to improve your presentation skills significantly before making the in-class presentation.
- Plan your message first, visuals last.

Please keep in mind to share this time fairly between your group members, such that everybody has enough time to present his part. I am mentioning this since in the past it sometimes happened that the first person to present took much more time, and therefore the last person to present in a group ran out of time at the end. (In this case the first speaker will lose points.) Please try to avoid these problems as far as possible by practicing the presentation in advance as a whole group, and by allocating the time among each group member fairly.

It is better for you to finish your presentation a bit early rather than too late. Please take this into account when practicing the presentation. Please also take into account when planning your presentation time that there will be some discussion during the presentation. For example, the instructor might ask clarifying questions or might provide comments and suggestions during the presentation. Based on past experience, students are often surprised how much time this can take, so please add a "margin of safety" in your time plan. As a **rule of thumb**, you should allocate ***at least*** **around 2 minutes of presentation time per slide** (excluding "special" slides such as title slides). In other words, you should limit the number of slides such that you do not have to rush through your presentation. This is, of course, a challenge because it forces you to identify the key topics you want to present (and discard unimportant topics). This is a good exercise because later in your working life you will be confronted with similar challenges.

Please email your presentation slides to the TA **at the latest two hours before the presentation** so that the TA can prepare the projector. This procedure ensures that we can minimize the waiting time between presentations in class. To hedge against technical problems, please also bring your presentations on your own USB flash drive to class. However, to minimize the waiting time during presentations, **you are only allowed to your own USB flash drive if the TA had technical problems uploading your previously handed-in presentations.**

There is no dress code for the presentation. However, it usually doesn't hurt to dress more formally. Although the way you dress will *not* influence your evaluation, most people feel more comfortable and confident if they dress up professionally. And that's exactly what you need when you want to convince other people during your presentation.

1.7 Group Report

Your group report should consist of:

1. Presentation slides in PDF format. You are allowed to change the slides after the presentation (but before the deadline for turning in your group project, see Section 1.5) to incorporate the comments and suggestions received from the instructor. However, please do keep in mind that it should still be possible to hold the presentation in the length of time allocated for your group presentation (see Section 1.3). Again, as a rule of thumb, you should allocate *at least* around 2 minutes of presentation time per slide (excluding "special" slides such as title slides).

2. Source code of computer programs in R (Rcpp is considered part of R, so you're free to use it if you want to). Please save the source code in plain text files with ".R" (or ".cpp" if you're using Rcpp) as the file extension.

   You should write all your programs in R. If needed, you can use Excel or OpenOffice.org for data import/export (e.g. as we have discussed towards the end of the "R Primer" lecture notes), but *all your calculations should be done in R*. Make sure that your computer programs are well-documented and easily readable. This is very important because it shows professionalism. Use sensible names for variables. Annotate your code with comments to explain what is going on. Visually separate code that belongs together with blank lines before and after it. Make sure to use proper indentation. See the "Style Guide" section in the "R Primer" lecture notes for more details.

   You should strive for elegant code. This means you should not only write code that gets the job done, but you should also try to write "beautiful" code. This is important for real-life, since code that is elegant and beautiful will make your future working life projects more easy to understand, more easy to maintain, and more easy to extend.

   In short, your R code should be

   - readable
   - elegant, i.e. not just a quick hack that gets the job done, no matter how ugly the code looks like.
   - maintainable, e.g. the code should be structured in such a way that it is easy to find and fix bugs, and it should be easy to adapt the code e.g. if the input data format changes.
   - extensible, i.e. if (hypothetically) your boss would, six months after your project's end date, request you to extend it to a different underlying, you should be able to extend your code base with as little hassle as possible.
   - modular, e.g. factor out parts of your code that belongs together into a separate function.

   Unless absolutely necessary, do not use any R package for your calculations. Instead, you should be able to do all the calculations yourself (without using any R package) by applying what you have learned in this course. In case you use an R package nonetheless, please provide some justification about why you have to use this package and why you cannot write the program yourself.

There are three exceptions to this rule: First, for downloading data you may use packages, e.g. the ones we have discussed in the "R Primer" lecture notes. (Of course you are also free to get the data from other sources *without* using any R package, e.g. directly from Bloomberg as CSV or Excel files.) The second point is about creating graphics that are data visualizations. (I am not talking about other graphics that are not data visualizations, e.g. diagrams; for these graphics you can use any program you like to create them.) For data visualizations, you may use R packages, e.g. the quantmod package (like we did in the lecture notes "R Primer" and also "Swaps and Volatility Trading") or the ggplot2 package (see https://ggplot2.tidyverse.org). Using these packages is not a requirement, but it may make it easier for you to create graphics and also to make the graphics look more pretty. In any case, please do *not* use any other software (e.g. Excel) for creating data visualizations. Third, you may use "data infrastructure" packages such as xts or data.table, i.e. packages that allow you to manage and transform your data.

Your code in the text file(s) should look like this:

```
x <- 8
y <- x^2
```

Your code in the text file(s) should *not* look like this:

```
> x <- 8
> y <- x^2
```

The reason for this requirement is that your code should run "as is," and it can only do that if you save it in the first format above. You can verify this by copy-pasting the code in the second format above into the R console: when you do this, you will receive an error message.

3. In case you use empirical data (e.g. time series data of a stock market index) for your project, you should also include the data to assure reproducibility of your results. You should save the data in binary `*.rds` files and your code should read your data into R. For more information see `?readRDS`, for example
`x <- readRDS("mydata.rds")`.

   Even if you have download your data from the internet using for example the quantmod package, you should still save a copy of your data to binary (i.e. `*.rds`) file(s), and your code should be able to import this data into R. The reason is that, depending on your data, the downloading using quantmod can take very long and depending on connection problems might have to be restarted several times. It is therefore better to have a local copy of the data and run the analysis using this local copy. In summary, in case you use a package (e.g. quantmod) that downloads the data from the internet, you should include five things: First, the code that downloads the data from the internet (e.g. using quantmod); second, the code that saves this data locally to a binary file (using `saveRDS`); third, the downloaded data in `*.rds` files; fourth, the code that reads the binary files into R (using `readRDS`); fifth, the code for your financial analysis.

4. `README.txt` file. This is a plain text file that contains instructions for users of your code. (In the case of this project, the instructor or TA are the users.) If you want, you can write this file in Markdown to give it a more professional look.[1] (Using Markdown is not required for the README file, but it is recommended.) In case you decide to use Markdown, please save the file as `README.md` instead of `README.txt`. In any case, no matter whether you use Markdown, you should *not* have more than 80 characters (including space etc.) in a single line. This "eighty column rule" has historical origins, but it still makes sense nowadays because it improves readability.

   The README.txt file shows how to run your code. For example, if you have split up your program into several files, README.txt should state which file is the master script that should be run. Your README.txt file should also describe the structure of your data files (e.g. if they are comma- or tab-separated or are saved in R's binary format, what the variables in each data file are, etc.). Importantly, since it is a plain text file, please keep each line at most 80 characters long. Otherwise it will be difficult to read. Most text editors are able to do this automatically; if you're not sure how to do this using your particular text editor, ask Google. For example, in the Emacs text editor you just hit `M-q`.

5. Summary report. Because of the relatively short duration of your in-class presentation, you might not be able to show all the details of your project on the presentation slides. Instead, you might only be able to get the key points across because of time constraints. To address this problem, you can write a summary report.

   The summary report should be in PDF format and should not exceed *five pages* in total (excluding any potential title page, i.e. the title page does not count towards the five-page limit). The reason for this requirement is that it leads you to focus on the most essential elements of your product. This is like in real-life, where you often have to convince people that your idea/product is good, and in a real-life situation you often have a very short amount of time or very limited space to convince people.

   Usually I discourage the use of an appendix for the group report. But if for some reason you really want to include an appendix, then the appendix will be included in the page count for the five pages. In other words, the group report including the appendix should at most be five pages long.

   You should avoid cramming everything into the five page limit, e.g. by using a very small font size or having very narrow page margins. Instead, you should use a "normal" page layout and remove the *content* that is not essential, thus keeping only the really important and essential parts. Furthermore, in the summary report, your main focus should *not* be to simply repeat what you already have on the slides, but to provide more detailed information and background on your project.

6. A file showing how your group split up the work, i.e. who did what and how the work-sharing was done. The file should include a table with names in the first column, the percentage of work contributed in the second column, and a brief description of the work in the third column. The file should be in PDF format and its content should not exceed half a page.

All files with the exception of source code, data files, and README.txt should be turned in using the PDF format. This ensures that formulae and symbols are displayed consistently across different computer configurations.

[1] See http://daringfireball.net/projects/markdown/syntax.

The first page of your presentation slides and of your summary report should contain your group number. Furthermore the first page should contain a list of all group members including name and university number.

Put your source code into the subdirectory named "code" and your empirical data into the "data" subdirectory (like we have done it for the use case at the end of the "R Primer" lecture notes). The PDF files and the README.txt file should be in the top-level directory. Then put all that stuff into a separate folder (**don't produce a tarbomb!**) and archive and compress your whole project tree as `.tar.gz` format before emailing it to the instructor and TA. You can use, for example, the open-source software 7-Zip for this task (it's up to you which software you use, as long as you produce a `.tar.gz` file). Please make sure that the file really is saved in `.tar.gz` format. I am mentioning this because in the past some students have *wrongly* saved the file in a different format (e.g. `.tar`), and then have "solved" this problem by renaming the filename to `.tar.gz`; this approach is wrong and you should not do this to get full points.

Please label the attachment as "`Group_XX.tar.gz`", where XX is the two-digit number of your group (e.g. please use, for example, "`Group_08.tar.gz`" and not "`Group_8.tar.gz`"). Your email subject should be "`Group_Report_of_Group_XX`".

When the instructor or the TA extracts your group report, the software should run "as is." That is, it should not be necessary to tweak your software, e.g. by having to adjust the path to subprograms or to the location of your data. If for some reason there is no way around tweaking your code, you should include detailed instructions in the README.txt file on how to modify your code to make it run.

Before handing in your R code, please close RStudio (without saving the workspace), open it again and run `rm(list=ls())` for good measure, and then run all your code from scratch. This ensures that your code *really* runs without any problems. I mention this because, based on past experience, it sometimes happened that students did not detect some final bugs in their code. The reason was that when writing their program, they still had an old version of a variable defined in the background. The existence of this old variable prevented the bug from occurring. However, when you really start from scratch (by closing RStudio and opening it again, like the instructor would do when evaluating your code), the bug would appear again because this time the old variable is gone. So to ensure that you will not lose unnecessary points in the evaluation, please perform this final consistency check before handing in your code: close RStudio, open it again, and then run all your code from scratch to see whether any unnoticed bugs have crept in.

Please email your group report both to the instructor and the TA. The TA will then send you a brief email confirming the receipt. If you do not receive this confirmation email from the TA, it means your submission was not received and therefore cannot be counted (in the worst case you receive zero points). Therefore, if you have sent out the group report but you did not receive the confirmation email, please contact the TA or the instructor to ensure that your submission did not get lost because of technical problems.

1.8 English Skills

In all the files that you hand in, your English spelling and grammar should be error-free. The group project is a great opportunity to perfect your English skills. Experience shows that even a top student will have a **very** difficult time landing a top job if he has spelling or grammar mistakes in his job market application. Do not let this obstacle stand between you and your dream job.

2 Hints for the Project

This section contains additional hints that have proven to be useful to students in the past.

2.1 Example Group Project

The course website contains an example group project from a previous year. Maybe it is useful to you for inspiration and illustration.

The fact that I make this project available, however, does NOT mean that this group project is absolutely perfect and it does NOT mean that you should try to emulate it as closely as possible. It is just an example for your inspiration (nothing more or less) and it is possible that you get high marks if you end up working on a completely different kind of project.

2.2 Additional R Tools

There are some additional tools in R that we could not cover in detail in this course because of time constraints. They might nonetheless still be useful for your group project if you are willing to invest a little extra time to learn them. There is, however, no requirement to use these tools and it is completely up to you whether you employ some them for your group project.

- Shiny (http://shiny.rstudio.com) is a web application framework for R that makes it easy to turn your analyses into interactive web applications. You can run it directly on your computer to create something interactive in R, e.g. for interactive demonstration during your presentation. If you are looking for inspiration, you can view the gallery on the Shiny website (http://shiny.rstudio.com/gallery), or another gallery at http://www.showmeshiny.com.

  Please note that if you are planning to use Shiny during your group presentation, please ensure that you remove all technical obstacles *before* your presentation, e.g. by doing a little trial run in the classroom. I am mentioning this since in the past some students were not well-prepared and faced problems getting their Shiny application to work in the classroom during their presentation.

- Optimization: If you need numerical optimization to structure your products, you are most likely looking for `?optimize` for general-purpose *one*-dimensional optimization or `?optim` for general-purpose *multi*-dimensional optimization. Please review the relevant parts of the "R Primer" lecture notes for further details, and take a look at the examples at the bottom of the R help pages, e.g. `?optimize` or `?optim`. If that does not satisfy your needs and you need more, please take a look at the relevant task view here: http://cran.r-project.org/web/views/Optimization.html.

- ggplot2 (http://ggplot2.org): If you want to produce beautiful graphics in R, look no further!

2.3 Numerically Calculating Deltas

Suppose you use Monte Carlo methods to calculate your option prices. Then it is still possible to calculate the Δ of your option numerically in case you need it for delta-hedging. This statement holds even if you have complicated options that have path-dependencies.

Let $\Pi(t,S_t;\mathscr{X})$ denote the price of the derivative at time t when the stock price is S_t. Remember that when at time t you know that the stock price is $S_t = s$ (for you standing at time t it is nonrandom because you can observe the stock price at time t), then Δ is defined as

$$\Delta := \frac{\partial \Pi}{\partial s}(t,s;\mathscr{X}).$$

If you do not know the functional form of $\partial\Pi/\partial s$, there is still a trick that you can use to calculate its value approximately. **All you have to know is some values of the function Π. In particular, you do not have to know its derivative at all.** This makes it useful for Monte Carlo methods because with those methods you can (more or less) easily calculate the option values, taking various current stock prices as given.

There are several alternative versions of this trick, and they work as follows. Suppose that $h > 0$ is a small number. The first alternative is to use a so-called *forward difference*:

$$\Delta \approx \frac{\Pi(t,s+h;\mathscr{X}) - \Pi(t,s;\mathscr{X})}{h}.$$

The second alternative is to use a so-called *backward difference*:

$$\Delta \approx \frac{\Pi(t,s;\mathscr{X}) - \Pi(t,s-h;\mathscr{X})}{h}.$$

The third alternative is to use the so-called *central difference*:

$$\Delta \approx \frac{\Pi(t,s+h/2;\mathscr{X}) - \Pi(t,s-h/2;\mathscr{X})}{h}.$$

The central difference is usually the **best method** of the three because usually it is most accurate. All three methods require you to calculate two prices of the derivative with Monte Carlo methods, so they are relatively easy to implement.

2.4 Structured Products that Depend on the Return of the Underlying

Suppose you have a structured product and that some of its payoff has the functional form

$$\max\left(\frac{S_T - S_0}{S_0}, 0\right),$$

where S_t is the price of the underlying at time t. This means that the payoff depends on the return of the underlying. We saw in the lecture notes that this payoff is equal to

$$\max\left(\frac{S_T}{S_0} - 1, 0\right)$$

and therefore can be priced using the Black-Scholes formula for a European call option applied to $Z_t := \frac{S_t}{S_0}$ and strike price equal to one.

There is, however, usually no option traded with this type of payoff. How can you as the issuer then hedge your structured product? You could of course use delta-hedging, but we know that this is not 100% precise. As a better solution, you could also make use of the following identity:

$$\max\left(\frac{S_T}{S_0} - 1, 0\right) = \frac{1}{S_0}\max(S_T - S_0, 0) = \frac{1}{S_0}\max(S_T - K, 0),$$

where $K := S_0$. This means you just have to find an at-the-money call option at time 0 and adjust the fraction of your portfolio wealth invested into this option by $1/S_0$.

2.5 Delta-Hedging

You do not *have* to use delta-hedging. If there are some derivatives readily available on markets to trade, you can use them for your project and do not have to "replicate" them using delta-hedging. In case there are derivatives available that you could trade and you still use delta-hedging, you should explain the reasons why you use delta-hedging instead of trading the derivative. For example, you could argue that there are some economic frictions in trading the derivative and that delta-hedging avoids dealing with those frictions. Or you have some institutional reasons why you use delta-hedging. But as a **general rule**, you should only use delta-hedging if there are serious reasons why you cannot hedge in a more straightforward way.

2.6 When to Use Monte Carlo Simulation

You should only use numerical methods to price your product if you do not have an analytical pricing formula available. For example, if your product contains a European call option, you should not use Monte Carlo simulation to price the call option. Instead, you should use the Black-Scholes formula. I am mentioning this because usually numerical methods (especially Monte Carlo simulation, as we have discussed in the lecture notes) are not as exact as analytical formulae: numerical methods often only represent an approximation to the correct option price, given a certain model for the dynamics of the underlying.

Furthermore, students sometimes want to include into their group report some results from a Monte Carlo simulation as a "consistency check," although they have also done the pricing using some analytical formula (e.g. the Black-Scholes formula). It is fine if you want to run this consistency check just for yourself in case you are not sure whether you applied the analytical formula in a correct way. But do *not* include both a Monte Carlo simulation *and* the analytical pricing in the group report or your presentation. The reason for this requirement is that the Monte Carlo simulation is superfluous (assuming that you have not done any mistakes when using the analytical pricing method).