Syllabus

April 7, 2021

This syllabus is subject to change at any time.

1 FINANCIAL DATA SCIENCE [M-WIWI-105610]

This Bachelor module (9 ECTS) consists of ONE course only: “Financial Data Science [T-WIWI-111238]”.

2 Content of the Module

This course talks about Robo Advisory, Linear Factor Models, Statistical Arbitrage, Monte Carlo Simulation, Machine Learning and New Developments in Asset Management.

2.1 Robo Advisory

In the section on Robo Advisory we start to wonder how to model preferences of investors. What do investors want and what do they not want. We revisit the concept of ‘expected utility theory’ to tease out essential financial economic concepts such as ‘what is a risk premium?’, ‘What is an insurance premium?’, ‘What are risk-averse, risk-neutral vs. risk-loving preferences?’ In the next step, we apply a 2nd-order approximation on any wealth only dependent expected utility function to arrive at ‘mean-variance preferences’, the work-horse model of robo advisors. We use these specialized preferences to understand the concept of ‘minimum variance frontier’, ‘efficient vs inefficient portfolios’ and ‘diversification’. In the final step, we solve the constrained and unconstrained portfolio optimization problem of a mean-variance investor.

Using Python, students learn how to compute utility functions and marginal utility functions and how to compute risk premiums of risky investments. Students also use Python to solve a strategic asset allocation problem in constrained and non-constrained environments. Said differently, students get exposure to numerical optimization to implement a key cornerstone in robo advisory.

2.2 Linear Factor Models

In the section Linear Factor Models we talk about why if every investor was a robo advisor with mean-variance preferences, the implied risk premiums of any investment, independent of whether
physical or crypto-like, would follow the predictions of the ‘capital asset pricing model’. We use the ‘capital asset pricing model’ to tease out practical insights, to predict the sign and magnitude of risk premiums across asset classes, to decompose investment risks (variance of returns) into diversifiable and non-diversifiable components. Putting up a data science perspective, we learn how to bring and test this nobel prize winning concept with data. We extend this perspective to multifactor linear models and to the ‘arbitrage pricing theory’. We finish this section by discussing whether predictable risk premium movements in asset returns are consistent with the concept of ‘market efficiency’.

Using Python, students program in an object-oriented way expected return predictions from the ‘capital asset pricing model’.

2.3 Statistical Arbitrage

In the section Statistical Arbitrage, students learn how to program statistical arbitrage strategies. Students learn how to use ‘auto-regressive-moving-average models’ to predict the probability density function of future returns. These forecasts allow us an ex-ante view on what can happen and what is the probability of such events.

Using Python, students learn how to determine the optimal number of lags in auto-regressive time-series models using the Akaike and the Bayesian information criterium.

2.4 Monte Carlo Simulation

In the section Monte Carlo Simulation, students learn why and how to conduct monte carlo simulations. Monte Carlo simulations are a simulation-based procedure to produce an ex-ante probability density function of future returns. Students learn basic procedures to simulate random numbers.

Using Python, students simulate the ex-ante probability density function of the uniform distribution, the Gaussian distribution and of an auto-regressive time-series model of known order.

2.5 Machine Learning

In the section of Machine Learning we start with a simple question: How do machines learn unknown parameters of parametric models? Students refresh how ‘least-squares’ methods, also called ‘L2-optimal learning’, is a robust approach to learn unknown parameters based on a training set of data and a (probability density) model. Students learn that unknown parameters of linear (probability density) models can be found analytically, and hence fast, using ‘generalized least squares’ learning methods and training data. We discuss differences and similarities to the ‘ordinary least squares’ approach of learning. The class also shows how to adjust these ‘L2-optimal’ learning methods to account for measurement errors, omitted variables and endogeneity problems.
The second part of the **Machine Learning** section shows several examples of how ‘maximum likelihood optimal learning’ works. Students learn that ‘maximum likelihood optimal’ learning procedures can be applied to linear and non-linear models. The course also derives that ‘maximum likelihood optimal’ solutions for unknown parameters of a linear model with an ex-ante Gaussian density function coincides with the ‘ordinary least squares’ solution.

The third part of the **Machine Learning** section presents classical learning methods on how machines learn how much risk, measured as volatility, is inherent in assets. Students also learn how to use these learning approaches to forecast volatility into the future. From a technical perspective, students learn about ‘auto-regressive conditional heteroscedastic’ models and ‘generalized auto-regressive conditional heteroscedasticity’ models. Students also learn how to improve these learning approaches to account for skewness risk.

The fourth part of the **Machine Learning** section addresses how to use the previous three machine learning tools in a ‘multivariate’ context. Technically, the course introduces the ‘vector auto-regressive models’ framework. This approach is also used for tactical ‘what happens if’ analysis.

All four parts of the **Machine Learning** section allow students to use Python to visualize and better understand the concepts. First, students write object-oriented Python code from scratch to apply ‘L2-optimal’ and ‘maximal likelihood optimal’ learning schemes to estimate parameters of a linear time-series model. Second, students use Python to learn whether a time-series is heteroscedastic. Third, students implement from scratch a 2-pass estimation approach to use a ‘L2-optimal’ learning approach to estimate the time-varying expected return and time-varying expected volatility of investments. Fourth, students compare the precision and accuracy of the previously mentioned 2-pass estimation approach to a ‘maximal likelihood optimal’ learning scheme. Last not least, students also code from scratch how to estimate ‘vector auto-regressive models’ and how to implement and conduct ‘what happens if analysis’ using the concept of ‘impulse response analysis’.

### 2.6 New Developments in Asset Management

In the section **New Developments in Asset Management**, we confront mean-variance optimal portfolios with rule-based portfolio strategies. The course relies on a case study that highlights the latter to outperform the former. This surprising feature is used to understand that the latter is a special subclass of the former. This observation helps us understand how better data science helps to build better portfolios. Next, the course moves on to talk about ‘factor investing’, ‘sovereign wealth funds’ and ‘smart beta’ investment strategies. The course then moves on to teach how the static approach of ‘linear factor models’ can be carried over to a dynamic investment setting. Students learn about the ‘intertemporal capital asset pricing model’ that leads optimal portfolios being composed of mean-variance optimal portfolios and a hedge portfolio. Students also learn how to incorporate non-tradeable factors into factor models.

While previously in the course, students estimated expected returns using time-series models, the end of the course introduces a cross-sectional approach for estimating expected returns. This approach of Fama-MacBeth is used to estimate factor premiums.

The course ends with a critical examination of ‘factor anomalies’ and so called ‘factor zoos’.
3 Organisational Aspects

3.1 Flipped Class-Room Concept

This class has an innovative structure that you may not have been exposed to before. In one sentence, passive learning, you do at home (with your study group), while active learning with the professor, we do together on Monday 09:45am.

- Students obtain access to
  - Video lectures
  - Book-like tutorials on video lectures
  - Quizzes and solutions to check basic understanding of video lectures
  - Concept questions, ranging from basic to challenging, without solutions

which they (with or without study group) study at home.

- Students work on weekly Python work sheets at home.

- ‘Prof Cafe’ on MS Teams at Monday 9:45am: professor and students build a study group to solve challenging concept questions

- Python Tutorial on MS Teams at Monday 12:15am where PhD student of professor and students build a study group to help solving the upcoming Python weekly worksheet.

3.2 Workload

The 9 ECTS module translates into a total workload of 270 hours. As the exam takes place in the 14th meeting, the weekly workload is approximately 19 hours. Despite these official hours, I anticipate 10 weekly hours of work that is composed of

- 1 hour video lecture
- 2 hour post preparation of video lecture, quizzes, preparing questions for ‘Prof Cafe’
- 2 hour ‘Prof Cafe’ on MS Teams to work with the professor through questions and challenging questions
- 1 hour Python tutorial
- 4 hour for Python and concept-oriented problem sets.

The remaining 100+ hours can be used by students for more specific exam preparation and/or additional work time per week.
3.3 Grade

The module-wide grading is a “Prüfungsleistung anderer Art” that consists of 2 parts and that add up to a maximal of 100 points. Part 1 is worth a maximum of 30 points and consists of 8 equal weighted weekly worksheets that students work throughout the week and submit prior to the beginning of the ‘Prof Cafe’ on MS Teams. The first worksheet for accumulating points is due in week 5, the last one is due in week 12. The other 6 worksheets are due in week 6, 7, 8, 9, 10, 11. Part 2 is worth a maximum of 70 points and consists of a 2 hour exam that takes place in week 14 during the regular ‘Prof Cafe’ meetup on MS Teams. The grading scheme will be related, not yet, perfectly the same, than the well known AIFB grading scheme.

3.4 Exam Dates and Points Achievable

- Python worksheet #1: Monday 9:45am, week 5, 3.75 pts
- Python worksheet #2: Monday 9:45am, week 6, 3.75 pts
- Python worksheet #3: Monday 9:45am, week 7, 3.75 pts
- Python worksheet #4: Monday 9:45am, week 8, 3.75 pts
- Python worksheet #5: Monday 9:45am, week 9, 3.75 pts
- Python worksheet #6: Monday 9:45am, week 10, 3.75 pts
- Python worksheet #7: Monday 9:45am, week 11, 3.75 pts
- Python worksheet #8: Monday 9:45am, week 12, 3.75 pts
- 2h exam, Monday 9:45am, week 14, 70 pts

3.5 Re-take Dates for “Prüfungsleistung Anderer Art”

- 2h exam: Friday, September 24th 2021, 9:45am - 11:45am
- 8 Python work sheets: handed out September 24th 2021, 12:00 am and need to be submitted back by Friday October 8th 2021, 12:00am.

Re-take exam is ONLY for students who (i) registered for 3.4 yet (ii) failed to pass it due to insufficient points.
3.6 Important Dates

- **Register for course:** asap, yet, at latest 1 day prior to the start of week 5
- **De-register for course:** at latest 1 day prior to the start of week 5
- Submission of weekly Python work sheets: Monday 9:45am in the week it is due
- 2h exam: Monday of week 14, during ‘Prof Cafe’ meetup on MS Teams on 9:45am - 11:45am.

3.7 ‘Prof Cafe’, ‘Python Tutorial’ and Submission of Python Work Sheets

- Monday, 9:45am: ‘Prof Cafe’ on MS Teams:
  
  https://teams.microsoft.com/l/channel/19%3a8fa84a4d3e0e40bea7924b5d19444ad2%40thread.tacv2/General?gro abde-404b-8886-e71ab0a886&tenantId=4f5eac75-46fd-43f8-8d24-62bebd9771c5

- Monday, 12:15pm: ‘Python Tutorial’ on MS Teams:
  
  https://teams.microsoft.com/l/channel/19%3a8fa84a4d3e0e40bea7924b5d19444ad2%40thread.tacv2/General?gro abde-404b-8886-e71ab0a886&tenantId=4f5eac75-46fd-43f8-8d24-62bebd9771c5

- Class content is shared via Ilias:
  
  https://iliasteststudium.kit.edu/goto.php?target=crs%5F1477543&client_id=produktiv

- Submission of Python work sheet solutions: 1 re-submission prior to the deadline is allowed. Submission via Praktomat.

3.8 Policies for Participating at ‘Prof Cafe’ and ‘Python Tutorial’

- **Professor and Tutor**
  - turn on camera and mic right from the start
  - join meetups well prepared
  - willing to work with participants to maximize learning
  - active, Socratic learning environment
  - meetups may be recorded according to KIT policies. If published, videos will be an-
    nonymized (voice and picture)

- **Participants (registered students)**
- **turn on camera and mic** right from the start
- ensure your **name** is visible on MS Teams (name tag or signature)
- **join prepared**, i.e. have completed quizzes, video lectures
- **active** participation in Scoractic learning environment (solving questions of others, raise questions, share experience and insights)