* Who am I
* Who are you?
* Why are we here today?
  + Come together and share eperiences, solutions and problems
  + Sharing my experiences in teaching R
  + Attempt to find some approaches for your own teaching hurdles
* Course Structure
  + 2,5 hours
  + Input Meike (60 min)
    - The problems
    - Mindset:
      * Always get feedback from the full group
      * Take feedback serious but always keep in mind the context
        + Why caused it? What can be changed about it?
      * Set realistic teaching goals and stick with them – do not engage in a race to the bottom
      * Try different approaches! Do not repeat, what did not work
      * But sometimes minor changes can have a big impact
      * Be very transparent and straightforward
      * Give enough support: extra resurces, extra exercises, give feedback – always keep in mind the struggles
      * Good teaching takes a lot of time, improve your course every time a little bit, be kind to yourself, but also not lazy
      * Do not underestimate how the seminar structure can fundamentally change the mindest of the students -> you are the one who sets the tone, be aware of your power and influences
      * Be transparent when and why you react on something that happened in the background
    - Anwesenheit
      * If not mandatory, create a mandatory exercise at thome that you will check
      * Gives them the freedom to not be in the class in person, but does make sure they work on the content
      * The presence is not mandatory, learning the content is
      * Dropping out of a course, moving into another semester or even failing it is ok
    - General helpful concepts and resources
      * Understanding R as a language
        + Teach the basics!!!
        + Practice is everything
        + Reading code, debugging code, writing code
        + Small exercices vs realistic exercises
      * What are R Basics?
        + Variables
        + Flexible vs hard coding
        + Data types
        + Comments vs Code
        + Missing values
        + Functions
        + Arguments in functions (order, necessary, optional)
        + Help site of a function
        + R packages (understand that packages are writing by many indivuduals that do not follow every guideline!)
        + Dealing with error messages & Debugging Tipps
        + R projects
        + Reading in data sets, writing them out
        + R Scipts, Rmarkdown
        + Loops, if-else
      * Failing is programming
        + focus on helping youself!!!
        + Debugger
        + Debugging tipps
        + External resources
        + Explain the help menu and how to read it (whya the basics are important)
        + How do I deal with an error message?
      * Motivation:
        + Show them in the befinning what crazy stuff they can do after they completed the course successfully -> create test questions, if some students have a lot of pre knowlodge to give them realistic feedback, where they stand
        + Make it usefull for the students -> necessary for projects -> in the curriculum
        + show the potential what can be done in R
        + Use interesting themes/toppings (column names are easily changed)
        + Make it more playful -> lets create Art with ggplot -> the result of the scatterplot is a dinosaur etc
        + Give good feedback!
      * Constructive alignment
        + Make the exams fitting to your teaching goals
        + The exam is set: align your teaching towards it
      * Flipped classroom
        + Concepts
        + What makes it work
        + What makes it fail
      * Learnr package
      * Otter as an example
      * Create your own data sets with simulations
    - ChatGPT a blessing and a course
      * Do not accept solutions that are outsode of what the course teached
      * Pick the exam well, so it becomes uninteresting to cheat
    - Use CHAtGPT yourself
      * Be as efficient as possible and focus on what matters
      * Make your feedback more friendly
      * Come up with exercises, or explanations for code
    - Exams
      * Open book exams (debug code, extend code, explain code, explain error messages, let them use the function help to asnwer a question of something is new to them)
      * Data science contest
  + Workshops (90 min)