

# Visualisation I

07.04.2022, Data Science (SpSe 2022): T6

**Prof. Dr. Claudius Gräbner-Radkowitzch**

**Europa-University Flensburg, Department of Pluralist Economics**

[www.claudius-graebner.com](http://www.claudius-graebner.com) | [@ClaudiusGraebner](https://twitter.com/ClaudiusGraebner) | [claudius@claudius-graebner.com](mailto:claudius@claudius-graebner.com)

# Prologue:

# Prologue

## Feedback and exercises

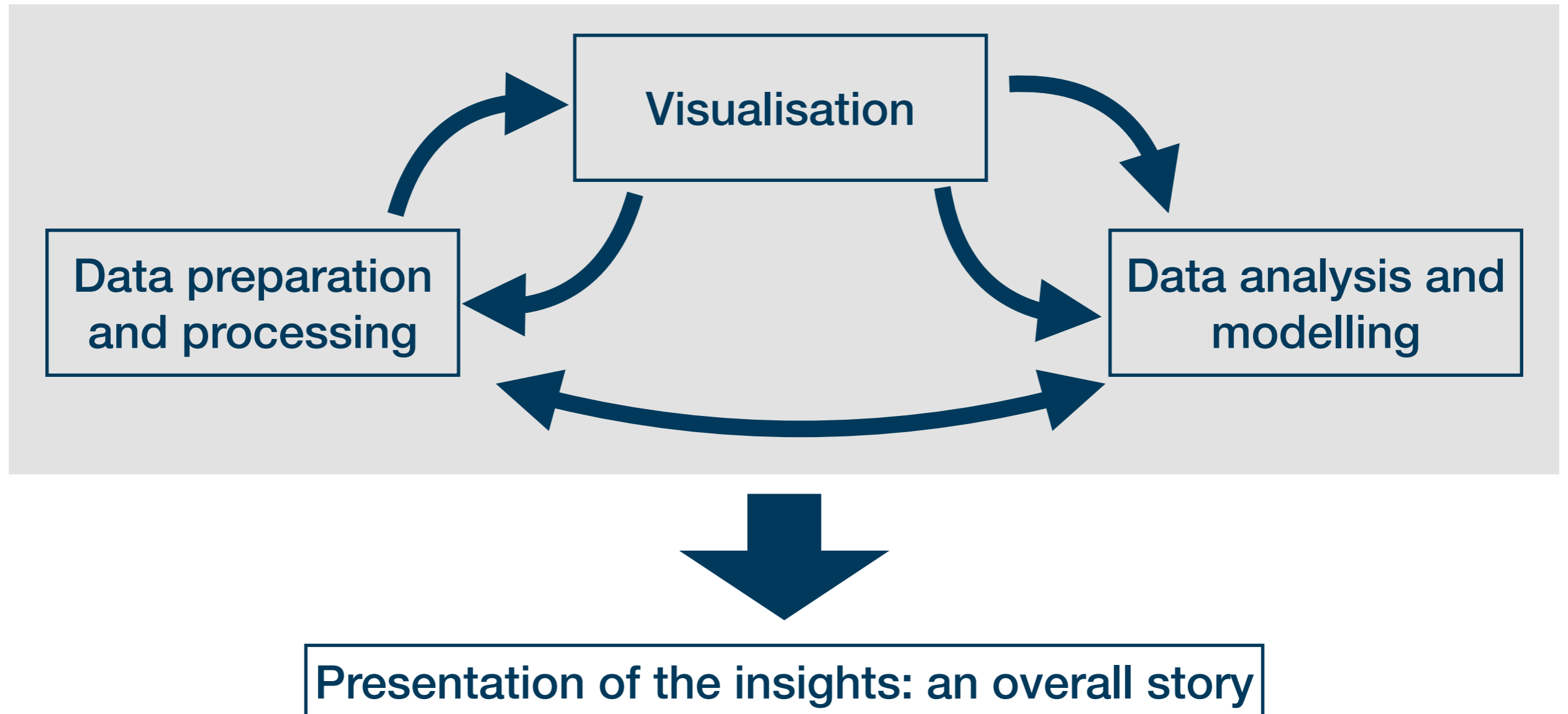
- XX of you filled out the feedback survey. Main take-aways:
  - TBA
- What were the main problems with the exercises?

# Goals for today

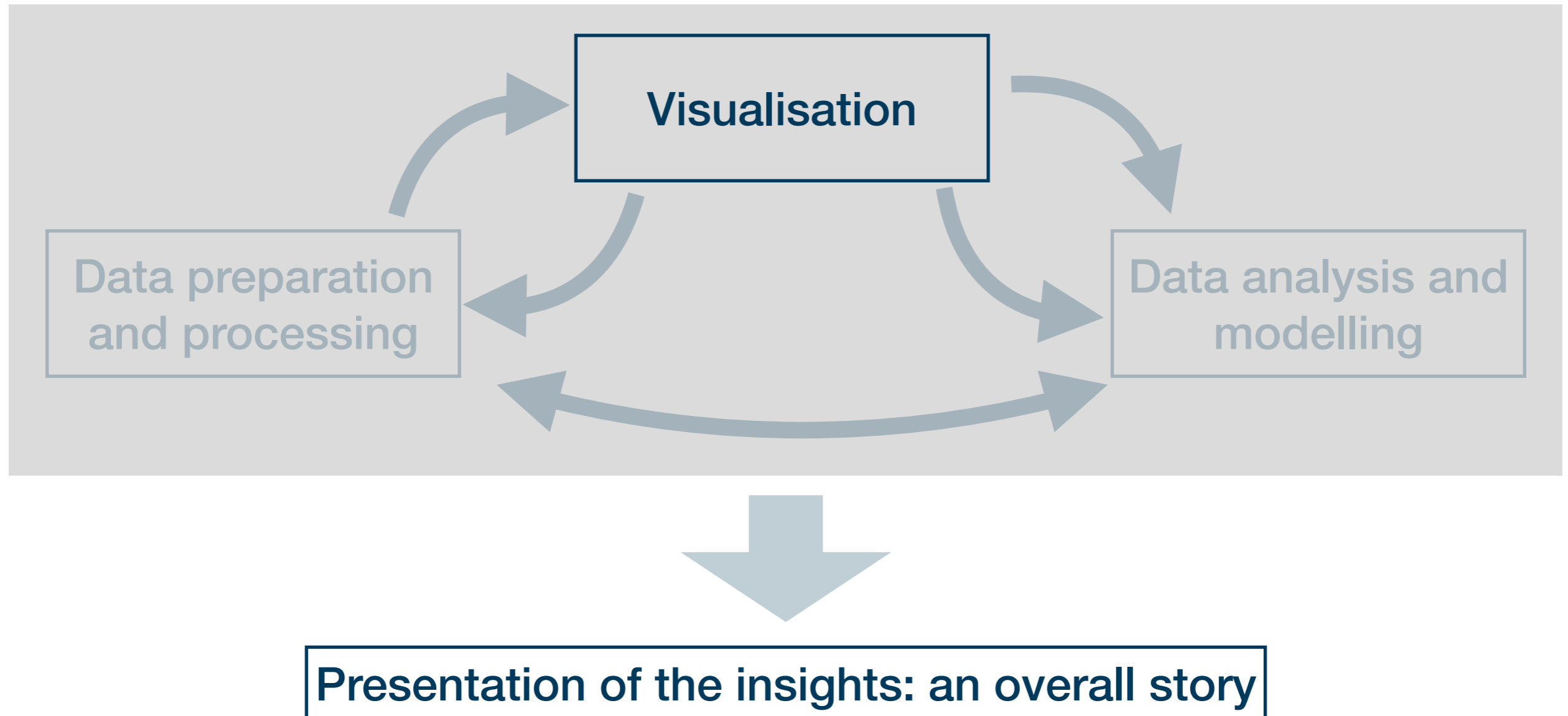
- I. Understand how plots are created layer-wise via the `ggplot2` package
- II. Learn how to map variables in data frames to visual aspects of a plot
- III. Figure out how you can re-use code across different visualisation tasks

# Basics of visualization

# The role of visualisation in data science



# The role of visualisation in data science



# About visualisations

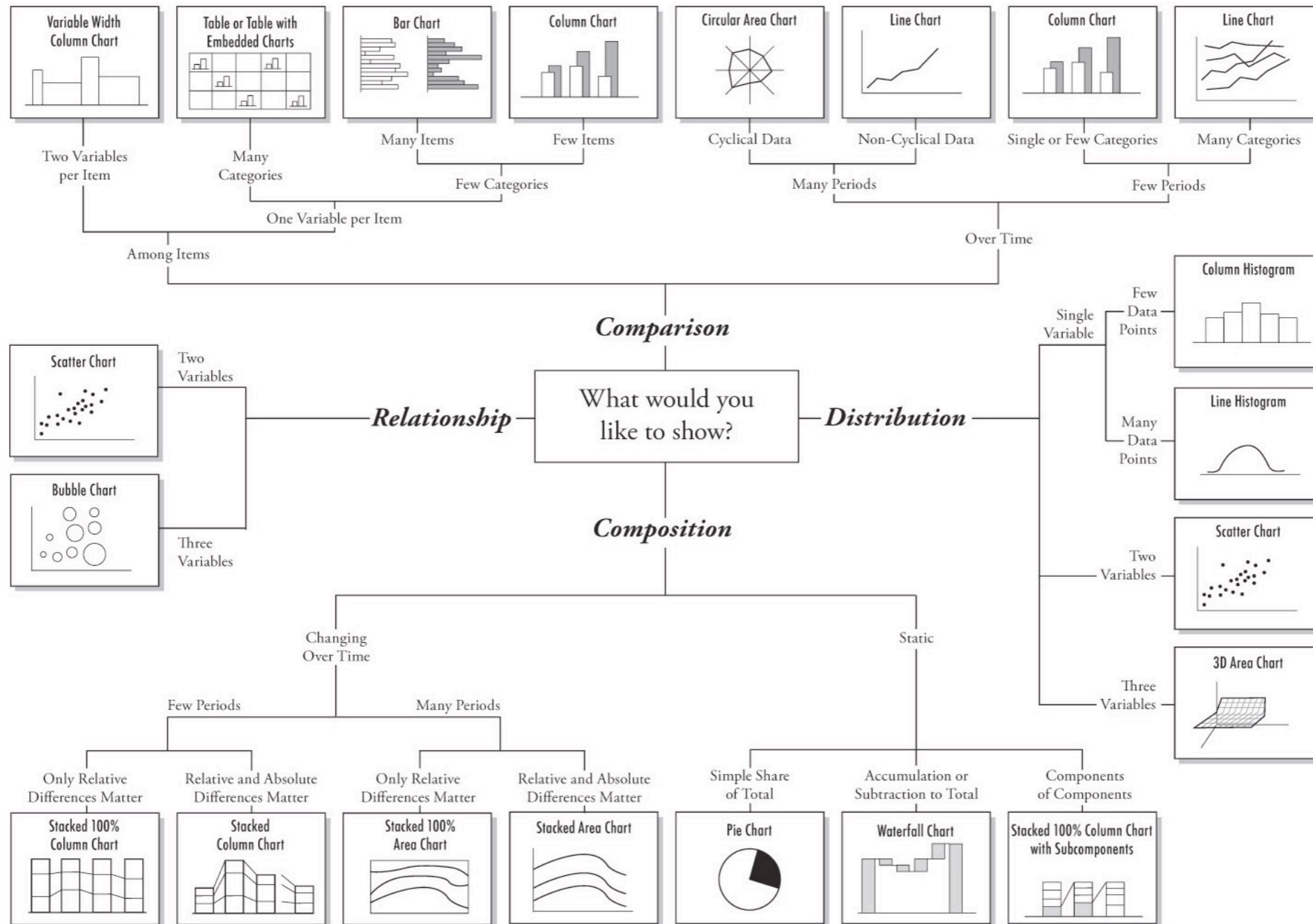
- Visualisations can be used for many purposes
  - Exploratory data analysis → understand your data → prepare/refine models
  - Communication → inform others about your results
  - Manipulation → convince others or recognise others convincing you
- Here we will learn about how to create visualisations using the package ggplot2
- An easy-to-read, widely-used and powerful visualisation engine
- Many great extensions, e.g. for animated GIFs, control charts, and many more...





# What kind of plot do you want?

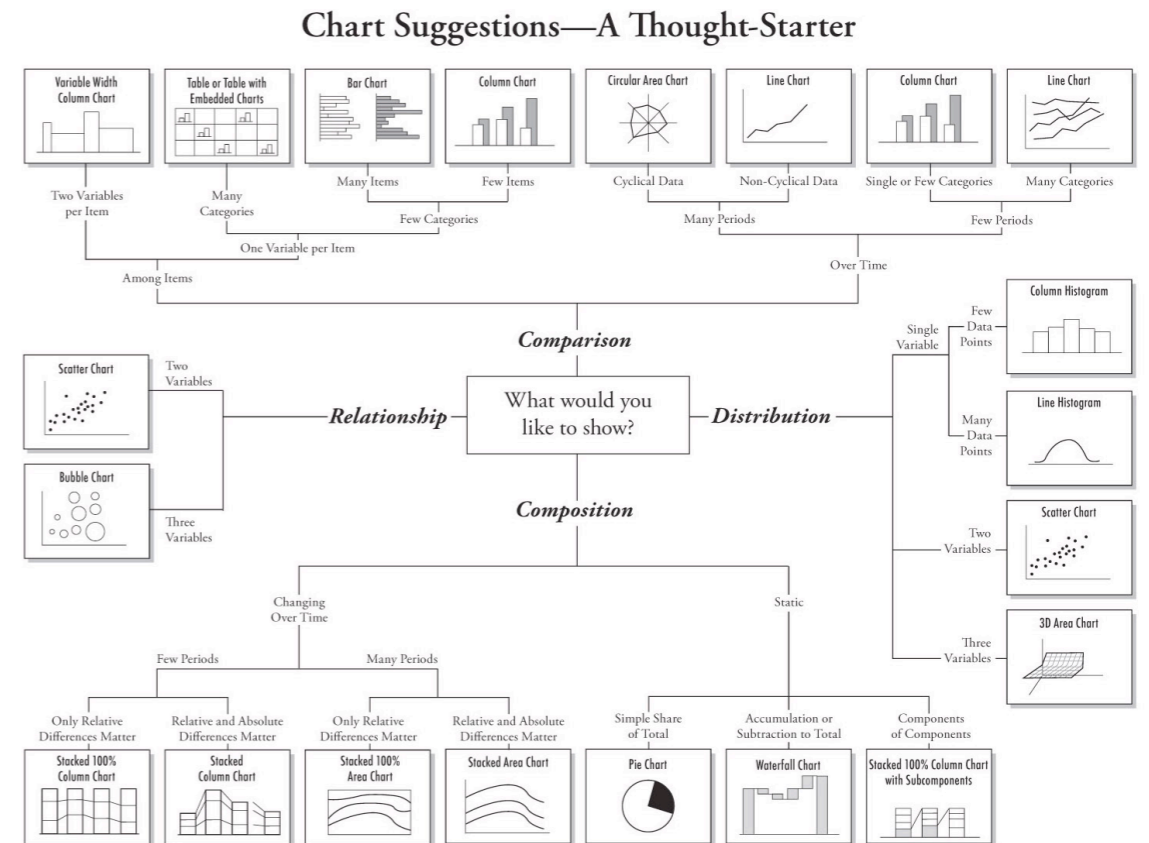
## Chart Suggestions—A Thought-Starter



www.ExtremePresentation.com  
© 2009 A. Abela — a.v.abela@gmail.com

# What kind of plot do you want?

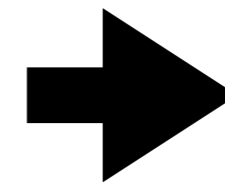
- Visualisation always involves prior thinking and theory
- The great thing about `ggplot2` is that the syntax is the same for all graphs
- Once the basic workflow is mastered, it's not difficult to create any of these



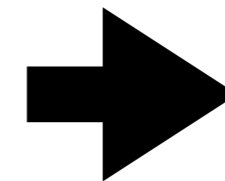
www.ExtremePresentation.com  
© 2009 A. Abela — a.v.abela@gmail.com

- During our lecture we focus on 2 examples:

- The scatterplot/bubble chart from session 1
- A line chart



Readings provide first generalisation



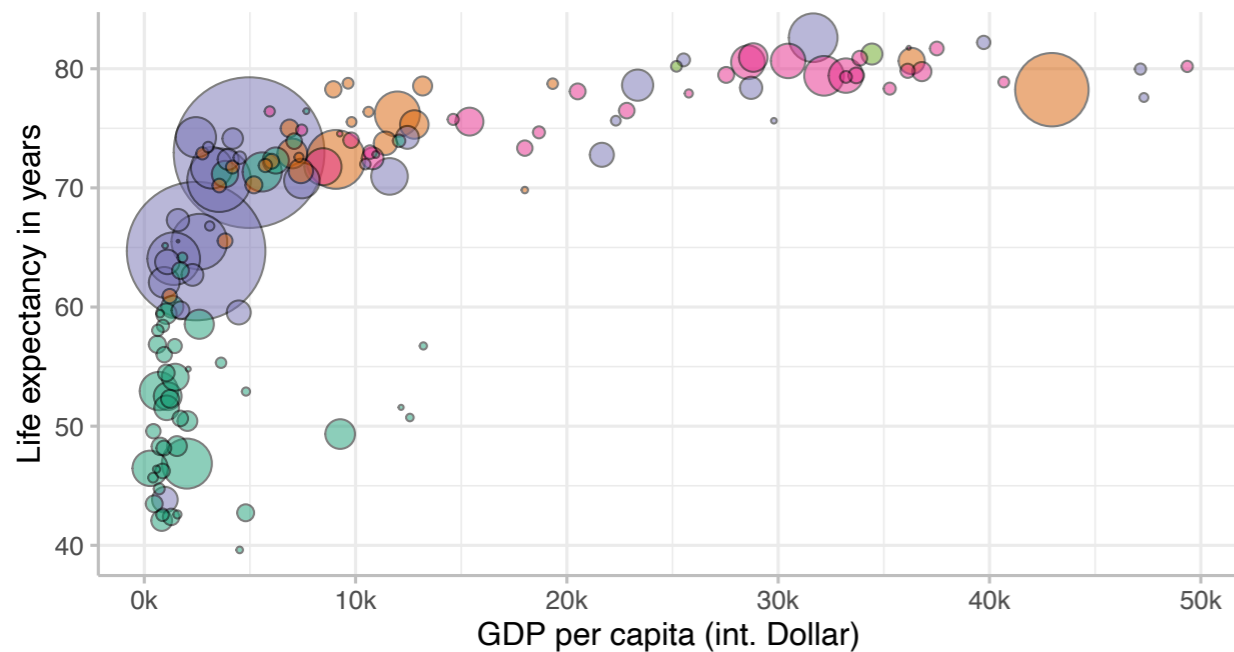
Visual fine-tuning in the second visualisation lecture

# The practical workflow

# Where we want to go:

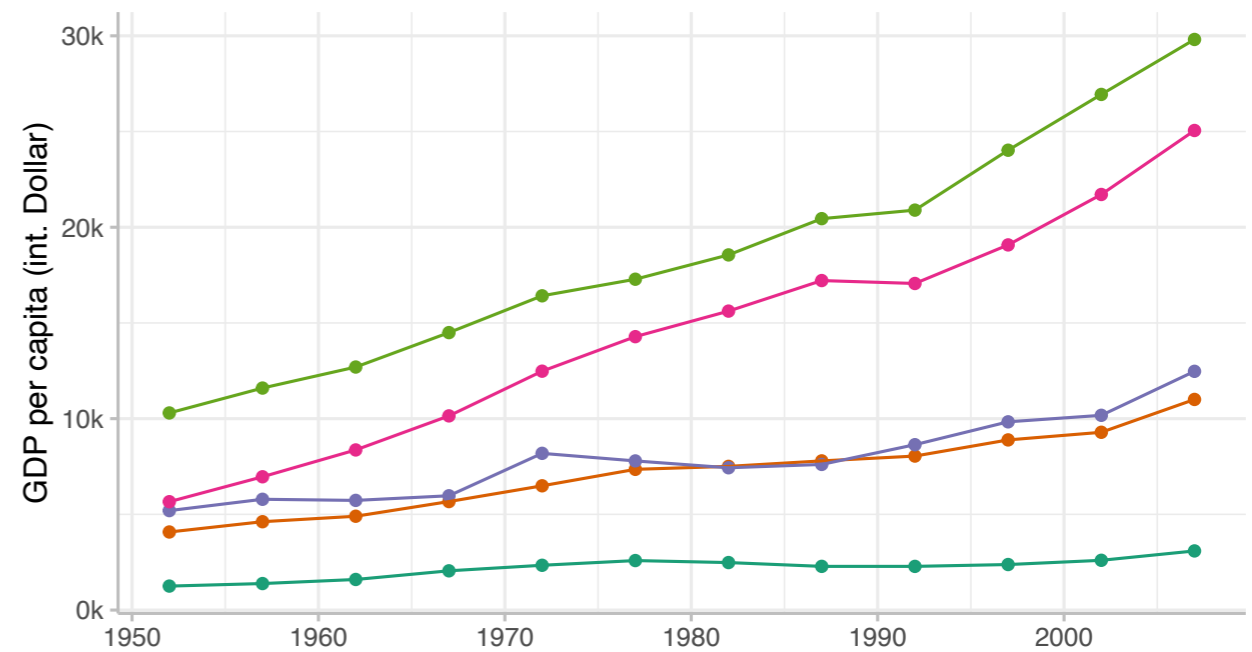
- Today we want to get started with visualisations and produce the following two plots:

Life expectancy and income per capita



Note: size of bubbles represents population. Data: Gapminder

The divergence of income per capita



Note: country data averaged over continents. Data: Gapminder

- We will see that the mechanics are very similar for different plots
  - Based on the readings you will be able to make even much more plots already now!

# The general idea

- Every plot in `ggplot2` is generated in two major steps
  - You describe the plot in all its details via a list ← This is where all the work gets done 😊
  - You call the list and R renders the plot for you ← This is where errors become apparent 😡
- To create the list-like description, `ggplot2` offers you a ton of helper functions
- You always start with an empty plot, then add layers above this empty plot, adjust details and that's it!
- Lets illustrate this using a subset of the gapminder data set only containing data for the year 2017
  - Readymade available to you via the DataScienceExercises package as `DataScienceExercises::gdplifexp2007`

# Developing a ggplot - the general workflow

- Since we are working on the graph development interactively, see my **lecture notes** for documentation purposes

# Summary & outlook

# Summary

- Visualisations serve many purposes, including the exploration of your data and the communication of your results
- We learned how to visualise data stored in data frames via `ggplot2`
- While there are many different plot variants, their syntax is very similar

`ggplot()` +

`<GEOM_FUNCTION>(`

`data = <DATA>`

`mapping = aes(<MAPPINGS>),`

`stat = <STAT>,`

`position = <POSITION>`

`) +`

`<COORDINATE_FUNCTION> +`

`<FACET_FUNCTION> +`

`<THEME ADJUSTMENTS>`

The geometric forms used to represent the data (points, lines, shades,...)

The data to be visualized

The mapping of the variables in data to the plot aesthetics (x/y-axis, size, form,...)

May be set as defaults within `ggplot()`, or separately for each geom

Adjustment to look, labels, etc.



# Summary

- Code for different plots differs mainly by the aesthetic mappings and the geoms used → allows you to re-use a lot of coding heuristics
  - We produced two beautiful plots: a bubble plot and a line graph
- The readings introduce you to other types of plots, which you can easily make more beautiful using the techniques you learned today
- A great way to learn how to plot is to replicate examples from the internet, and adjust them to your own data:



## The R Graph Gallery



# Outlook

- Next week: how to organise an R project and import data into R
  - This allows us in a next step to transform it into a format we can use for visualisation (and, later, modelling)
  - Up till now we only worked with readymade data sets but in reality, the data you get will be messy → learn to produce data as you have already learned use will open many doors

## Tasks until next session:

1. Fill in the **quick feedback survey** on Moodle
2. Do the **readings** posted on the course page → they generalise what you have learned to new plot types
3. Have a look at my lecture notes and redo the plot creation of today
4. Do the **exercises** provided on the course page and **discuss problems** and difficulties via the Moodle forum