Visualizing morphological data

Stefan Hartmann
HHU Düsseldorf

Visualizing (morphological) data

Stefan Hartmann
HHU Düsseldorf

Plan for today

- Basics of data visualization
- Data wrangling in R
- Basics of ggplot2
- A first example graph
- Break
- Case studies and hands-on exercises

Principles of data visualization

Data visualization

Heinrich Heine
Universität Düsseldorf

Why visualize?

- **For yourself**
  - Exploring your data
  - detecting outliers
  - checking assumptions of statistical tests or models (e.g. are the data normally distributed?)
  - etc.
- **For others**
  - Showing your findings in a clear and efficient way
  - Graphs tend to be more reader-friendly than tables...
  - and *much* more reader-friendly than long inline lists!

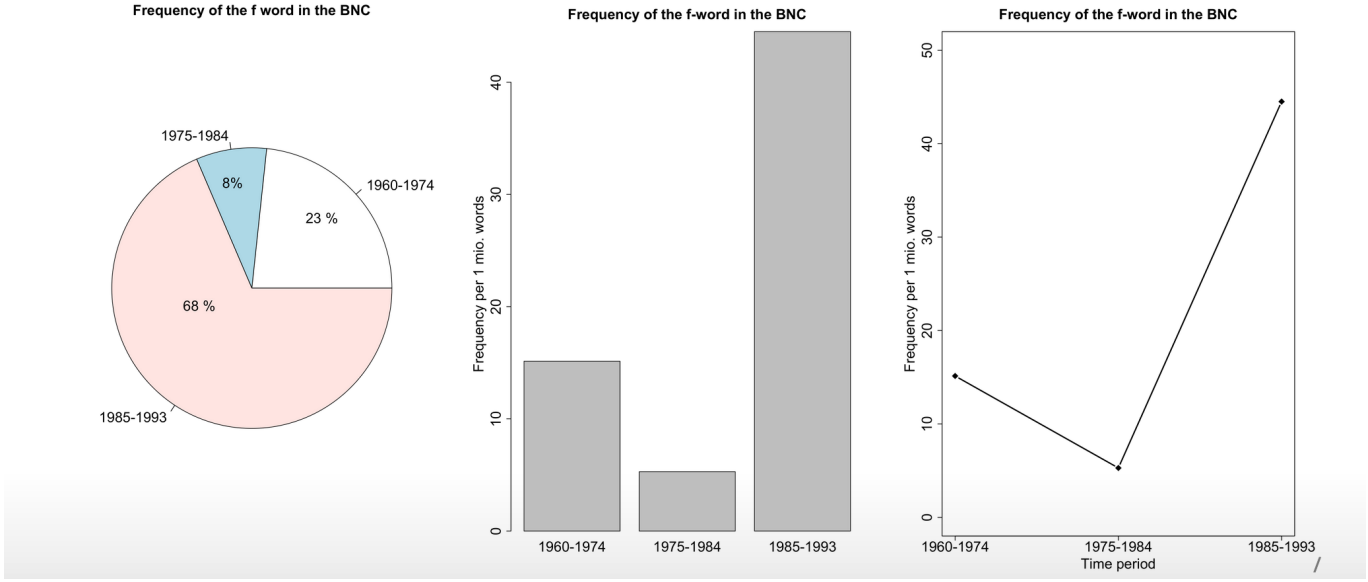
Data visualization

Choosing the "right" plot

- What kind of data are you dealing with?
- What is your research question?

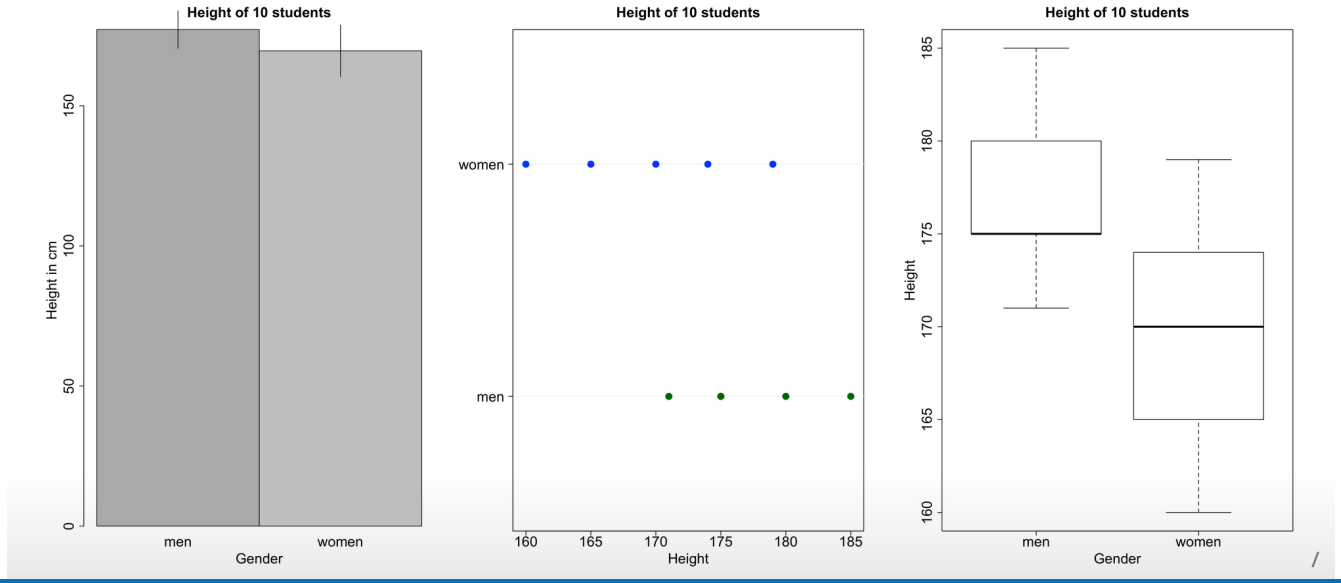
Data visualization

- What kind of variable are we dealing with here?
- Which visualization seems most appropriate to you?



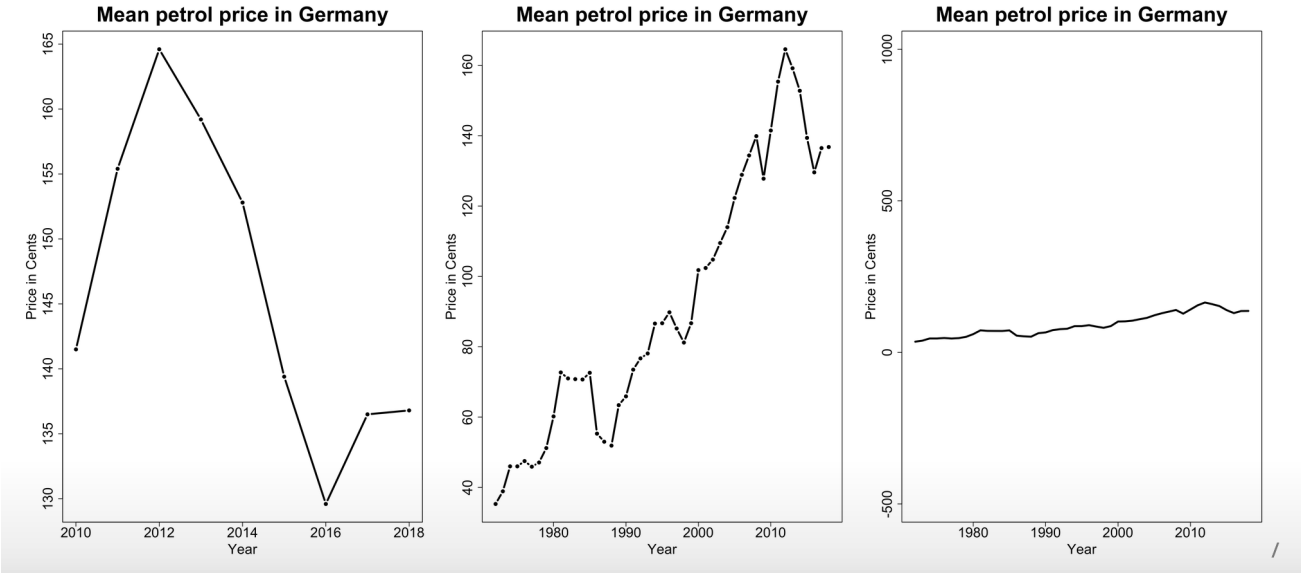
Data visualization

- What kind of variable are we dealing with here?
- Which visualization seems most appropriate to you?



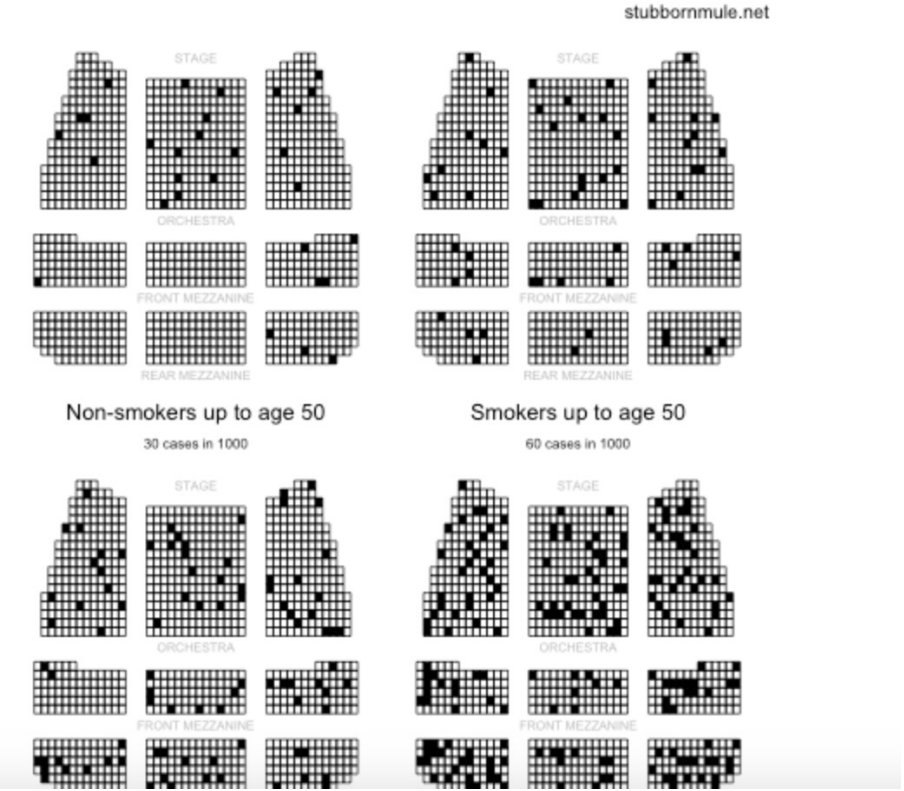
Data visualization

- What kind of variable are we dealing with here?
- Which visualization seems most appropriate to you?



Alternative visualizations

- from http://www.stubbornmule.net/2010/10/visualizing-smoking-risk/
- "Risk Characterization Theatre" from Rifkin & Bouwer (2007)



The plot as a metaphor

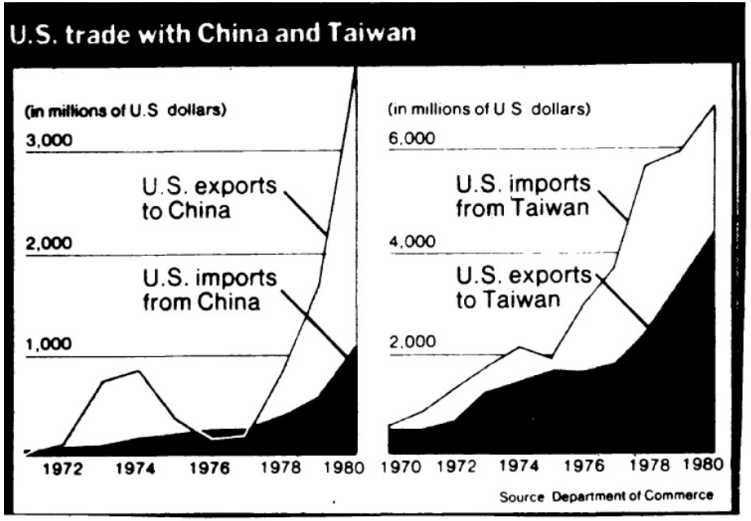
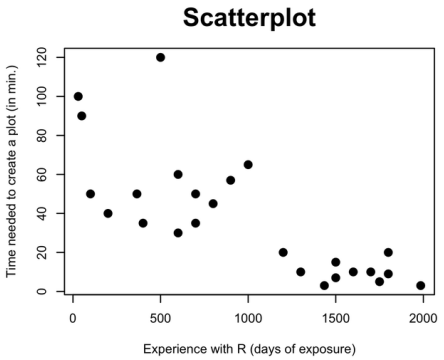
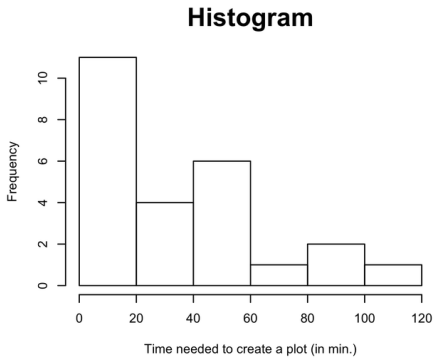


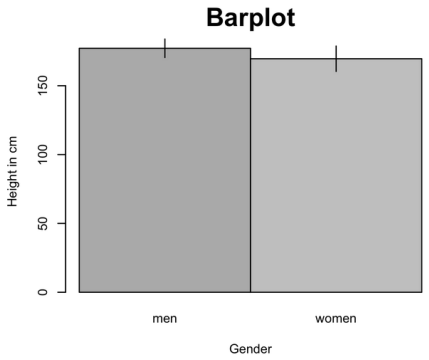
Figure 7. Reversing the metaphor in mid-graph while changing scales on both axes (© *June 14, 1981,* The New York Times).

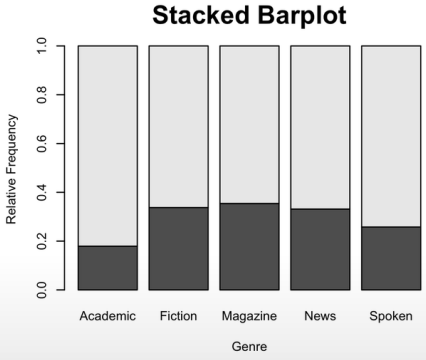
"The essence of a graphic display is that a set of numbers having both magnitudes and an order are represented by an appropriate visual metaphor - the magnitude and order of the metaphorical representation match the numbers." (Wainer 1984: 139)

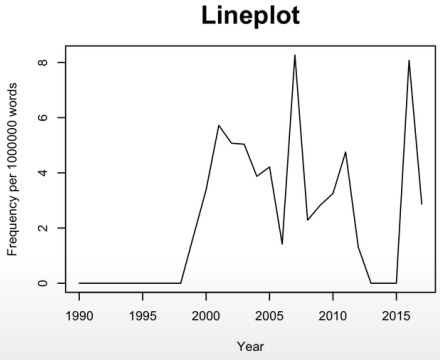
Plot types

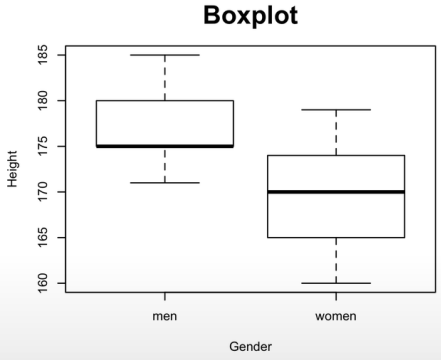

Scatterplot
Time needed to create a plot (in min.)
0
20
40
60
80
100
120
0
500
1000
1500
2000
Experience with R (days of exposure)


Histogram
Frequency
0
2
4
6
8
10
0
20
40
60
80
100
120
Time needed to create a plot (in min.)


Barplot
Height in cm
0
50
100
150
men
women
Gender


Stacked Barplot
Relative Frequency
0.0
0.2
0.4
0.6
0.8
1.0
Academic
Fiction
Magazine
News
Spoken
Genre


Lineplot
Frequency per 1000000 words
0
2
4
6
8
1990
1995
2000
2005
2010
2015
Year


Boxplot
Height
160
165
170
175
180
185
men
women
Gender

Best practice for reporting & displaying data

Heinrich Heine
Universität Düsseldorf

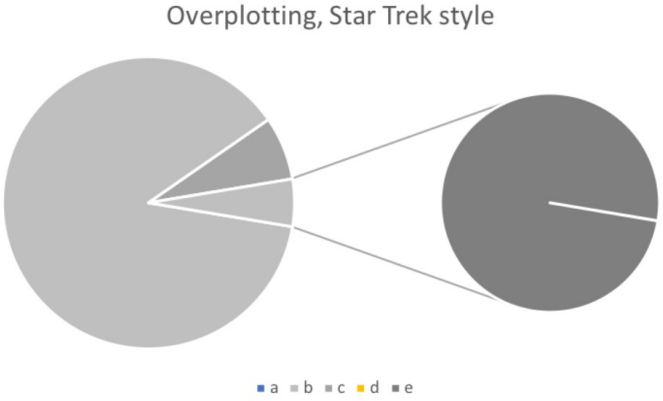
- Most importantly: **Know your data!**
- When reporting percentages, also report the denominator (i.e. the size of your sample)
- Example: "50% of academics are alcoholics" - it makes a difference whether your sample size is 2 or 2,000.
- When reporting comparisons of absolute frequencies, double-check if your samples are comparable.
- Example: "255 women agree that cats are adorable, but only 5 men." - it makes a difference whether your sample consists of 300 women and 300 men or of 300 women and 10 men.
- When reporting means, also report standard deviations.

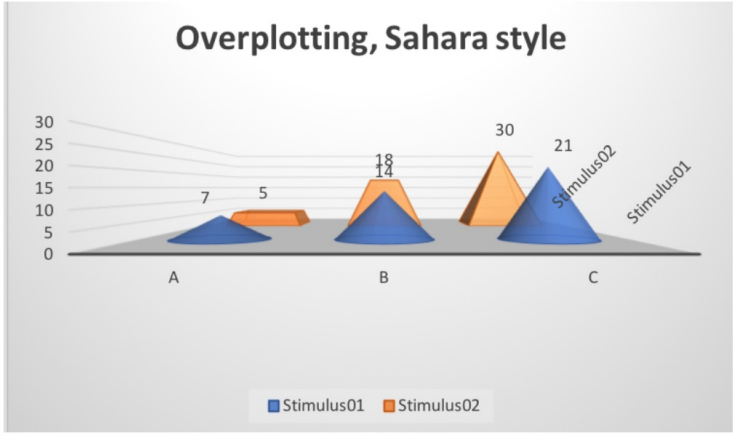
Best practices

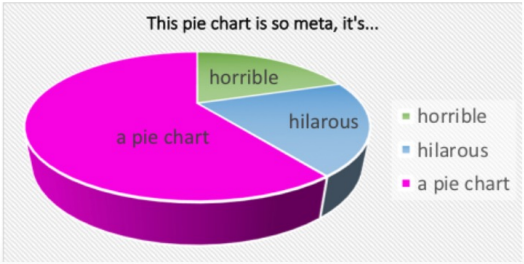
Heinrich Heine
Universität Düsseldorf

- Show the data
- Avoid distorting the data
- Keep "Ink-to-data ratio" as low as possible
- Use meaningful x and y labels
- Avoid overplotting (e.g. 3-dimensional plots when only 2 dimensions are displayed)

Beware of overplotting

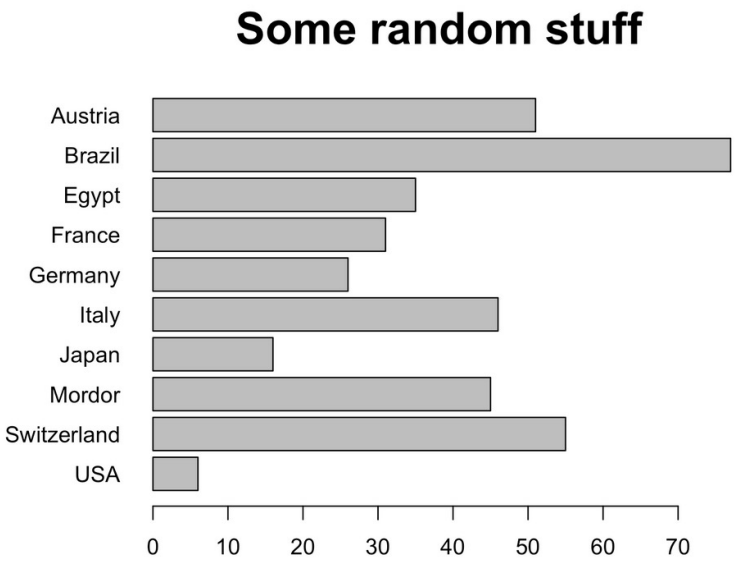


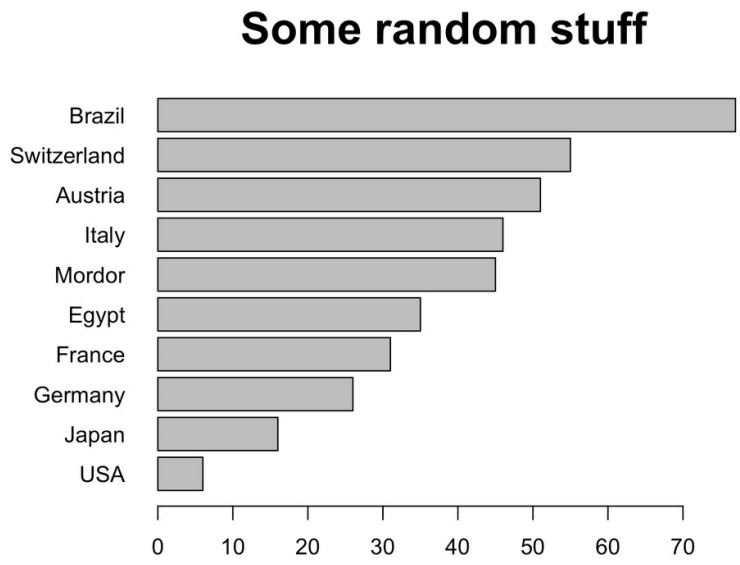




Further tips

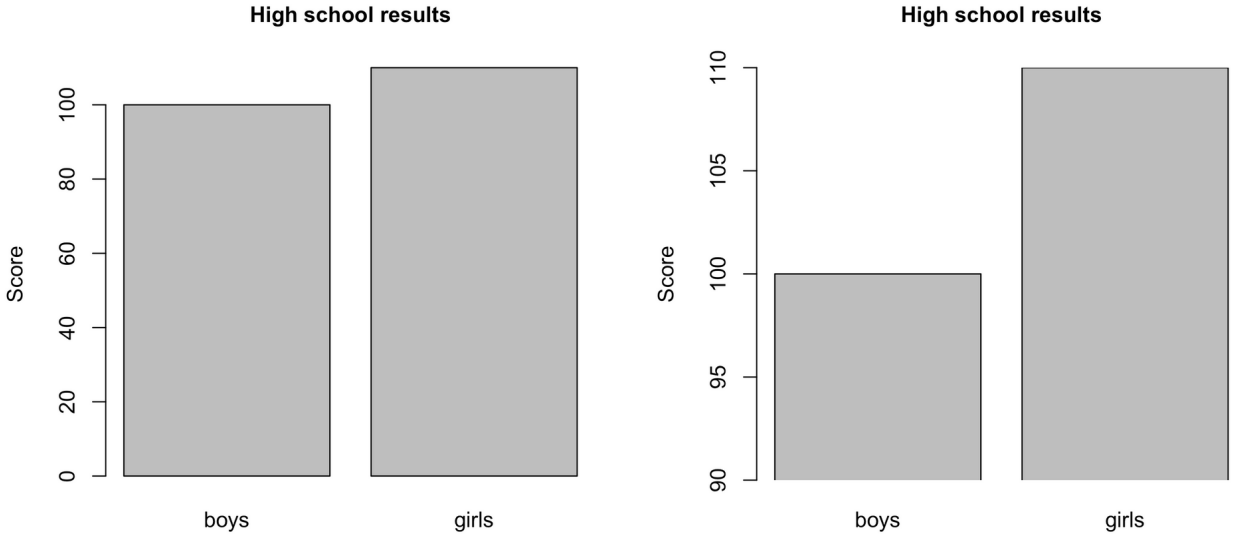
- If there is no natural order to your data, order them by value





Further tips

- Don't cut the y axis unless there are good conceptual reasons to do so.



Data wrangling in R

Types of data

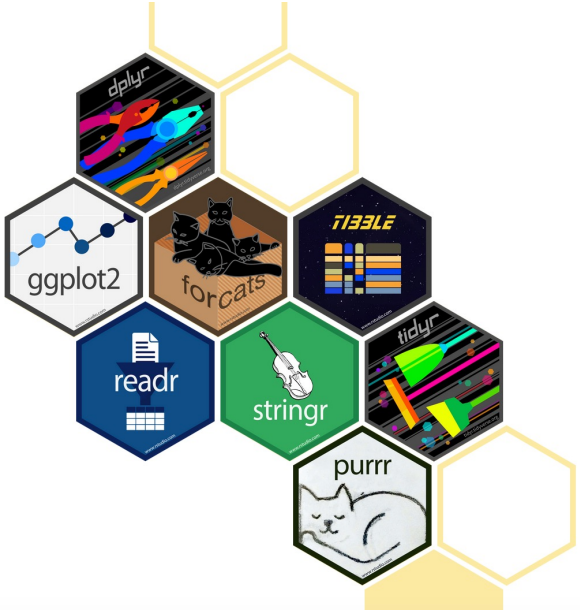
Types of data in R

- **character**: "a", "swc"
- **double**: 2, 15.5
- **integer**: 2L (the L tells R R to store this as an integer)
- **logical**: TRUE, FALSE
- **complex**: 1+4i (complex numbers with real and imaginary parts)

Base R and Tidyverse

Heinrich Heine
Universität Düsseldorf

What is the Tidyverse?



- family of packages developed by RStudio/Posit
- implement an own "dialect" of R
- still fully compatible with base R, but adding more syntax possibilities

Base R and Tidyverse

Heinrich Heine
Universität Düsseldorf

Pros and cons of the Tidyverse

- Pro:
  - syntax is arguably more intuitive and cleaner once you get used to it (especially "piping"!)
  - offers really neet functions for data wrangling
  - ggplot2 offers great possibilities for visualization
  - it is very widely used and many replies in forums rely on it

- Cons:
  - it is under very active development, some functions become deprecated or change their names → problems for reproducibility
  - can be a bit patronizing (although this can sometimes be a good thing)
  - some things are ridiculously counterintuitive

Data wrangling and preprocessing

Heinrich Heine
Universität Düsseldorf

Steps

- Structuring
- Cleaning
- Validating
- Handling missing data (→ imputation)
- Merging data from different sources
- Transforming variables (can also be seen as already part of the statictical analysis*)

*In fact, all these steps are, strictly speaking, part of the analysis!

Data wrangling

"Tidy data"

- Use "tidy data": One variable per column
- One observation per row

Hi there this is my cool fancy Excel spreadsheet with the results of my psycholinguistic experiment!!!				
Subject ID	190808			
	Stimulus 1	response time	980	
		answer	yes	
	Stimulus 2	response time	1080	
			no	
		metadata	age	28
			gender	female

Subject ID	Stimulus	Response	Answer	Age	gender
190808	1	980	yes	28	female
190808	2	1080	no	28	female
809653	1	830	no	45	male
809653	2	420	yes	45	male
207436	1	320	yes	25	male
207436	2	954	no	25	male
185848	1	430	yes	30	female
185848	2	850	no	30	female
947379	1	530	yes	84	female
947379	2	1045	no	84	female
374957	1	890	yes	18	female
374957	2	1150	no	18	female

"Long" vs. "wide" format

Heinrich Heine
Universität Düsseldorf

Subject ID	Time_Stim	Time_Stimulus	Answer_S	Answer_S	Age	gender
190808	980	1080	yes	no	28	female
809653	830	420	no	yes	45	male
207436	320	954	yes	no	25	male
185848	430	850	yes	no	30	female
947379	530	1045	yes	no	84	female
374957	890	1150	yes	no	18	female

wide

Subject ID	Stimulus	Response	Answer	Age	gender
190808	1	980	yes	28	female
190808	2	1080	no	28	female
809653	1	830	no	45	male
809653	2	420	yes	45	male
207436	1	320	yes	25	male
207436	2	954	no	25	male
185848	1	430	yes	30	female
185848	2	850	no	30	female
947379	1	530	yes	84	female
947379	2	1045	no	84	female
374957	1	890	yes	18	female
374957	2	1150	no	18	female

long

"Long" vs. "wide" format

Heinrich Heine
Universität Düsseldorf

From long to wide format

- If necessary, data can be converted from "long" to "wide" format using the pivot_wider() and pivot_longer() functions from the tidyverse family of packages.

- (They may not be 100% intuitive but with a bit of trail and error they work really well!)

Dealing with dataframes

Reading in dataframes

- Commands for reading in spreadsheets:

CSV:	readr::read_csv()
TSV etc.:	readr::read_delim(delim="\t", ...)
Excel:	readxl::read_xlsx()
plain text:	readr::read_lines()
large files:	vroom::vroom or datatable::data.table()

Dealing with dataframes

Merging multiple dataframes

- In many cases we want to **combine** dataframes, e.g. because one dataframe contains metadata pertaining to the other dataframe
- Example: We have one file with corpus data, with one column specifying the author, and one file with birth and death dates of each author crawled from the German National Library and/or Wikipedia
- We can combine the dataframes using dplyr's *join()* commands.

Dealing with dataframes

Heinrich Heine
Universität Düsseldorf

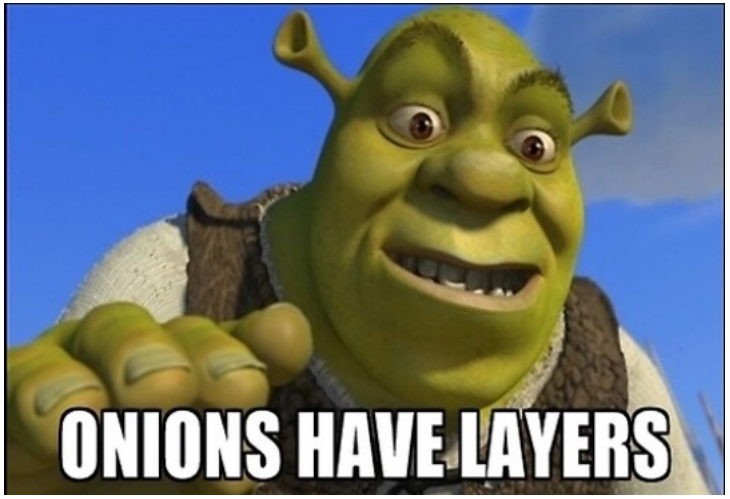
Hands-on task

- ...

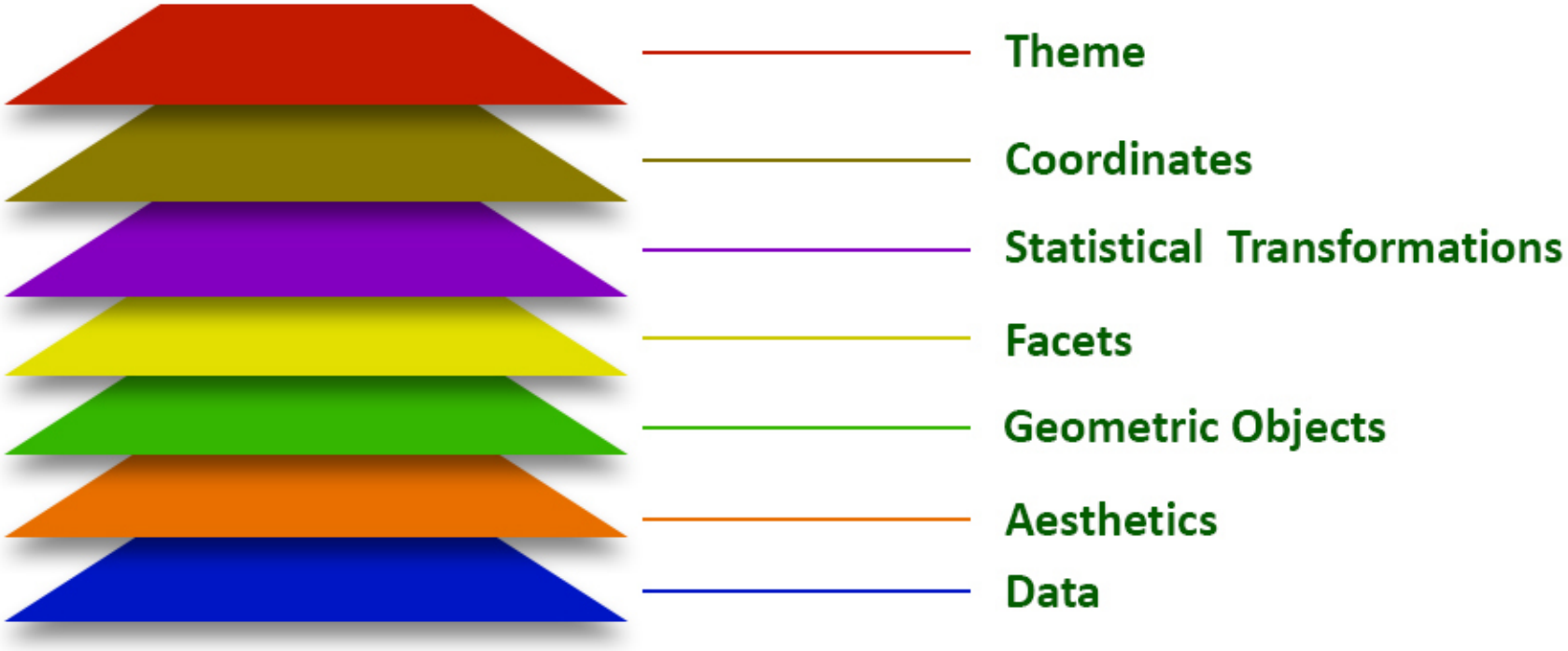
Basics of ggplot

The syntax of ggplot

- A ggplot consists of three components
  - the **data,**
  - a set of **aesthetic mappings**,
  - at least one **layer** (usually created with the geom function) describing how to render each observation.



Main Components of the Grammar of Graphics



Creating a ggplot

- A ggplot is built **layer by layer**
- We start out with the **data** and the **aesthetic mappings**
- Basic syntax:

```
p <- ggplot(data, aes(x = ..., y = ..., group = ...))
```

- We specify the **geometric objects** to plot, e.g.

```
p <- p + geom_line() # lineplot
```

- Optional: We customize the **scales** (position, color, size) and/or change the **theme** of the plot

```
p <- p + scale_color_grey() + theme_minimal()
```

A basic ggplot

- **First step: generating fake data**

Try to create a dataframe with two columns x and y, with x containing the numbers from 1 to 100 and y 100 normally-distributed random numbers (`rnorm(100)`).

- **Second step: visualizing the data**

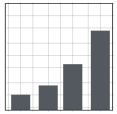
Plot x against y using base R first and then using ggplot.

- **Third step: customizing the plot**

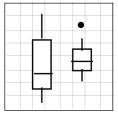
Play around with different scale configurations and themes.

Different Geoms (Plot Type) in ggplot2

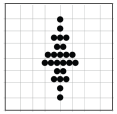
Two Variables (X,Y)
- Discrete X, continuous Y
- Visualise distribution of Y with respect to X



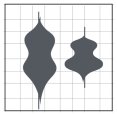
geom_col()
- heights of bars represent values



geom_boxplot()
- summarise distribution using median, hinges and whiskers

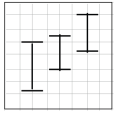


geom_jitter()
- adds jitter to prevent overplotting

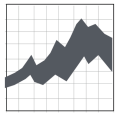


geom_violin()
- mirrored density plot (smoothed distribution)

Visualising Errors and Uncertainties

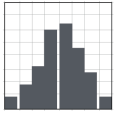


geom_errorbar()
- uncertainty in continuous Y against discrete X

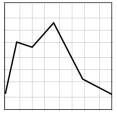


geom_ribbon()
- uncertainty in continuous Y against continuous X

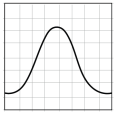
One Variable (X)
- Continuous X
- Visualise distribution of X



geom_histogram()
- divide X into bins and count no. observation

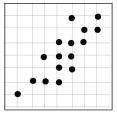


geom_freqpoly()
- display counts with lines
- able to overlay multiple distributions

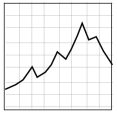


geom_density()
- smoothed version of the histogram

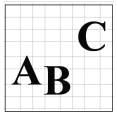
Two Variables (X,Y)
- Continuous X, continuous Y
- Visualise relationship between X and Y



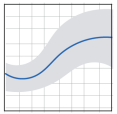
geom_point()
- scatterplot of X vs Y



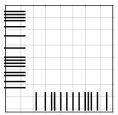
geom_line()
- connect data points, ordered by X
- alt: geom_path()



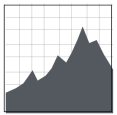
geom_text()
- labelling data points



geom_smooth()
- add smoothed curve
- helps to see trends



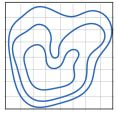
geom_rug()
- supplement 2D plot with 1D distribution along X and Y



geom_area()
- can be stacked to see cumulative contribution

Contour Plots
- Representing a third dimension using contours

geom_density2d()
- contour represents 2D density of data points



geom_contour()
- contour represents z-axis value / height

Heinrich Heine Universität Düsseldorf

More plot types

Beeswarm plots

- Packages *beeswarm* and *ggbeeswarm*
- can be combined with violin or boxplots

Interactive plots

Heinrich Heine
Universität Düsseldorf

- e.g. (*gg*)*plotly* packge
- and *shinyplots* (shinyplots.io)

Some tips and tricks

Heinrich Heine
Universität Düsseldorf

- Cheat sheets
- Code snippets

Hands-on examples...

Task 1a

ung-nominalizations

- The file *ungbaby.csv* contains a concordance of *ung*-nominalizations from a subset of the German Text Archive
- The file *dtababy_author_texttype.csv* contains metadata about the authors and text types.

Find out how the two files can be merged in such a way that the metadata about author and texttype are added to each row, based on the document ID.

Task 1b

ung-nominalizations – Morphological productivity

- Use the resulting dataframe to calculate the **potential productivity** (e.g. Baayen 2009) of the pattern for each text type and for each decade.
- To do so, use the tidyverse *summarise()* command.

Task 1c

ung-nominalizations – Visualizing productivity

- Now **visualize** the productivity development. Select the plot time that seems most appropriate to you and plot one panel for each of the three text types.