



# Data visualization with ggplot2

Stefan Hartmann  
HHU Düsseldorf

- Basics of data visualization
- Basics of ggplot2
- Example graphs for basic plot types
  - Scatterplot
  - Line plot
  - Boxplot, violin plot
  - Beeswarm plot
- Hands-on exercises
- Some (fairly random but perhaps useful) tips and tricks

## Software and packages

- Please make sure that you have R and RStudio installed.
- You also need the following packages / families of packages:

tidyverse

patchwork

scales

ggbeeswarm

- If they are not installed yet, you can install them in R using the `install.packages()` command, e.g.  
`install.packages("tidyverse")`



# Principles of data visualization

## 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!

## Choosing the "right" plot

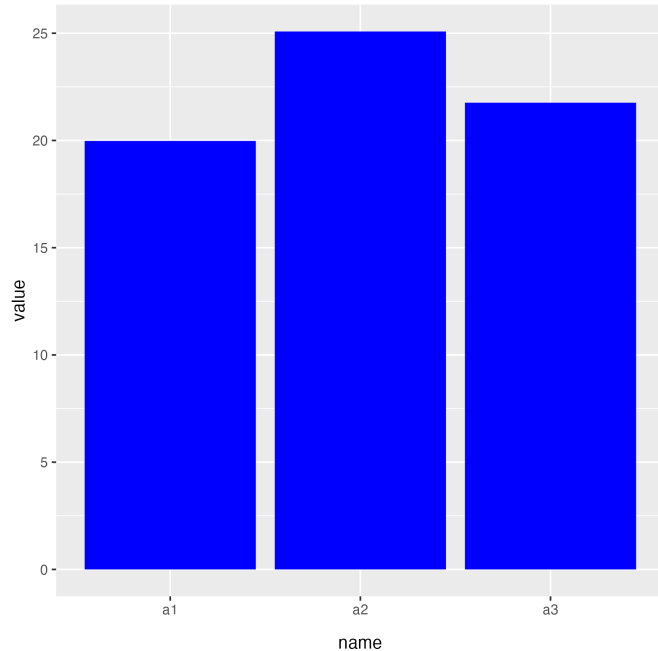
- What kind of data are you dealing with?
- What is your research question?
- What kind of audience are you expecting?

- 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 dispersion measures (e.g. standard deviations or standard errors)

- Show the data
- Avoid distorting the data
- Aim for a good "ink-to-data ratio": Display as much information as possible with as little ink as possible
- Avoid overplotting (e.g. 3-dimensional plots when only 2 dimensions are displayed)
- Use meaningful x and y labels

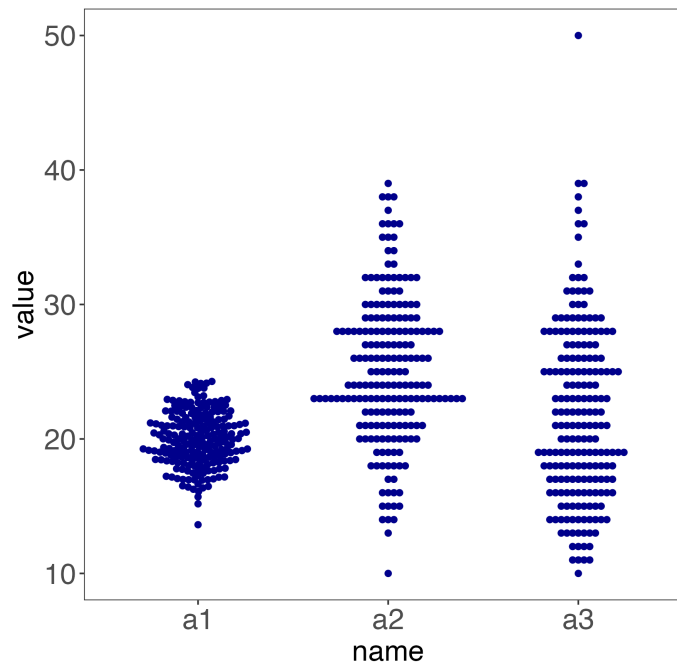
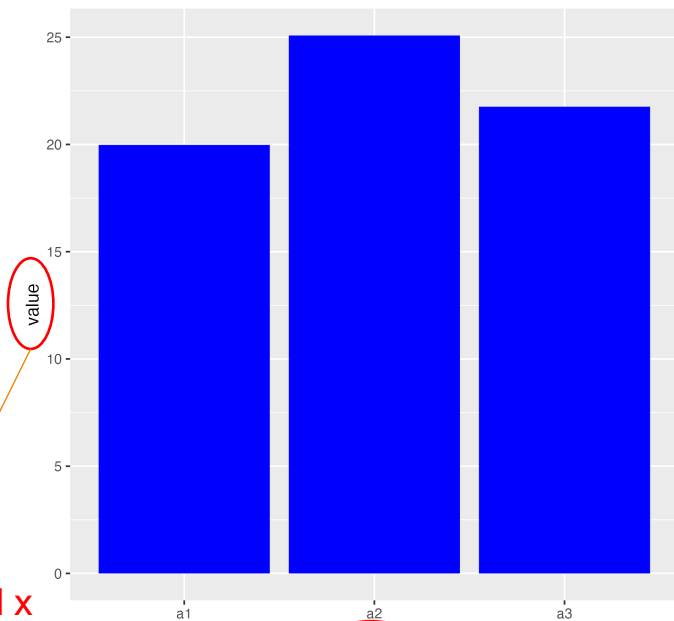


## Show the data



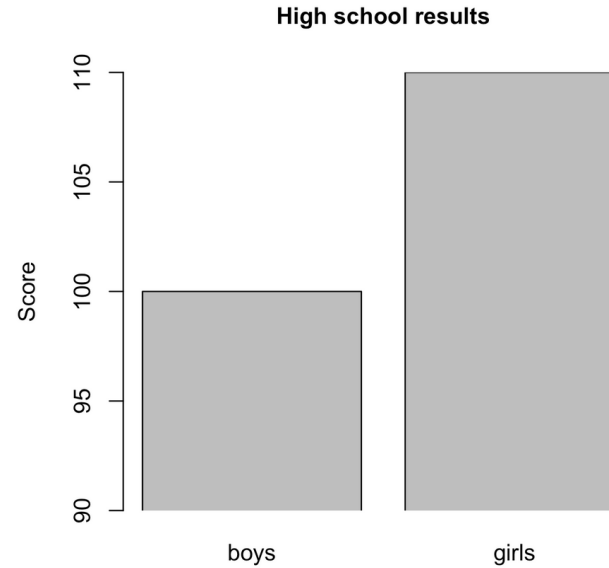
# Best practices

## Show the data

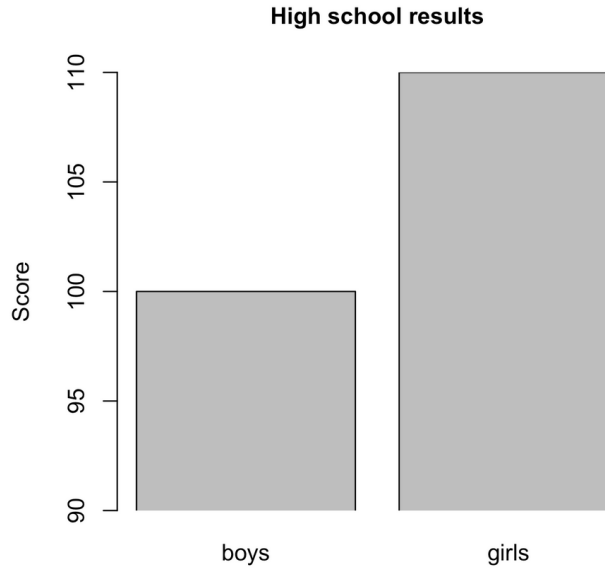
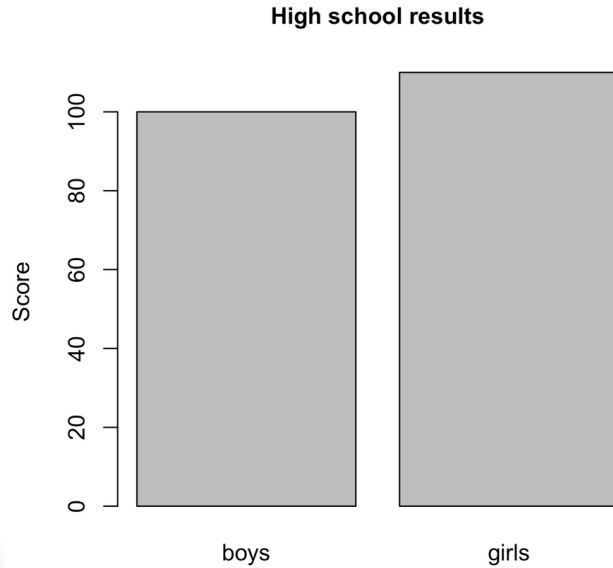


also, use  
meaningful x  
and y labels

# Avoid distorting the data

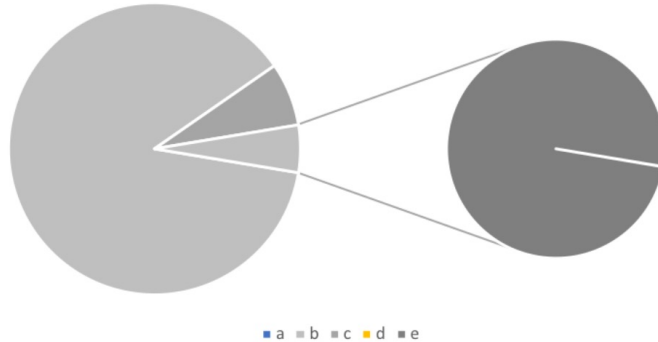


# Avoid distorting the data

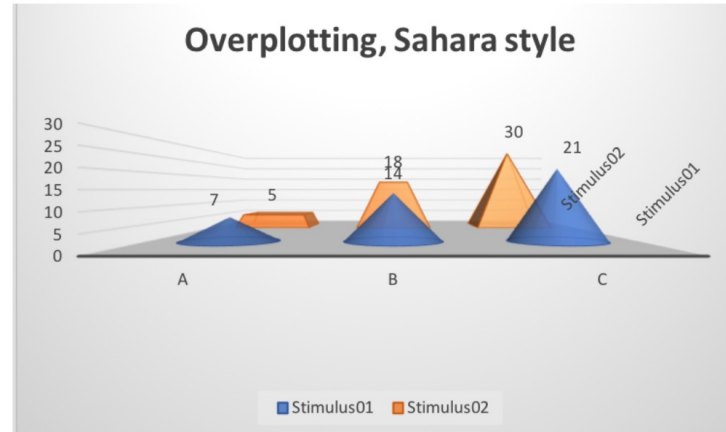


# "Ink-to-data ratio" / Overplotting

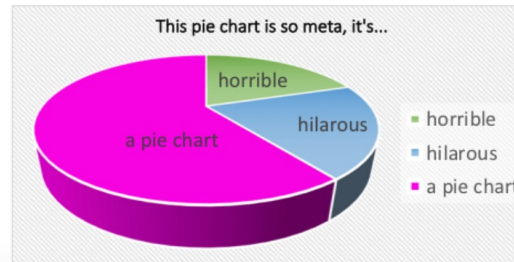
Overplotting, Star Trek style



Overplotting, Sahara style

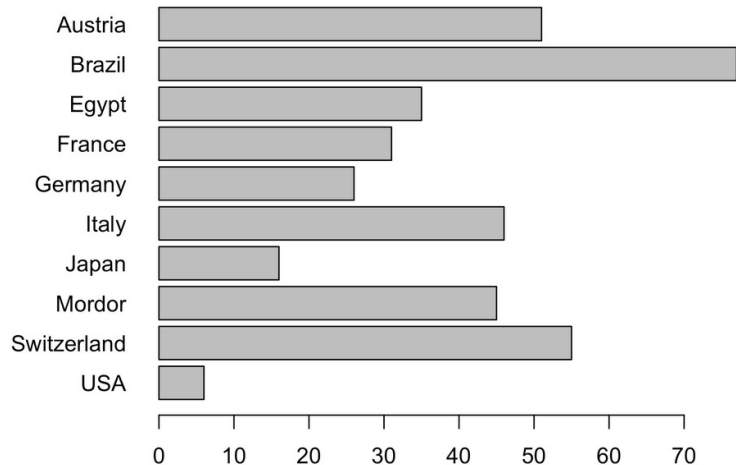


This pie chart is so meta, it's...

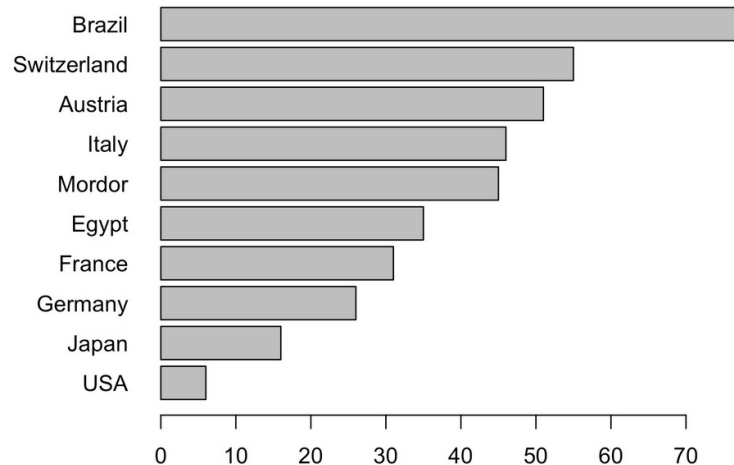


- If there is no natural order to your data, order them by value

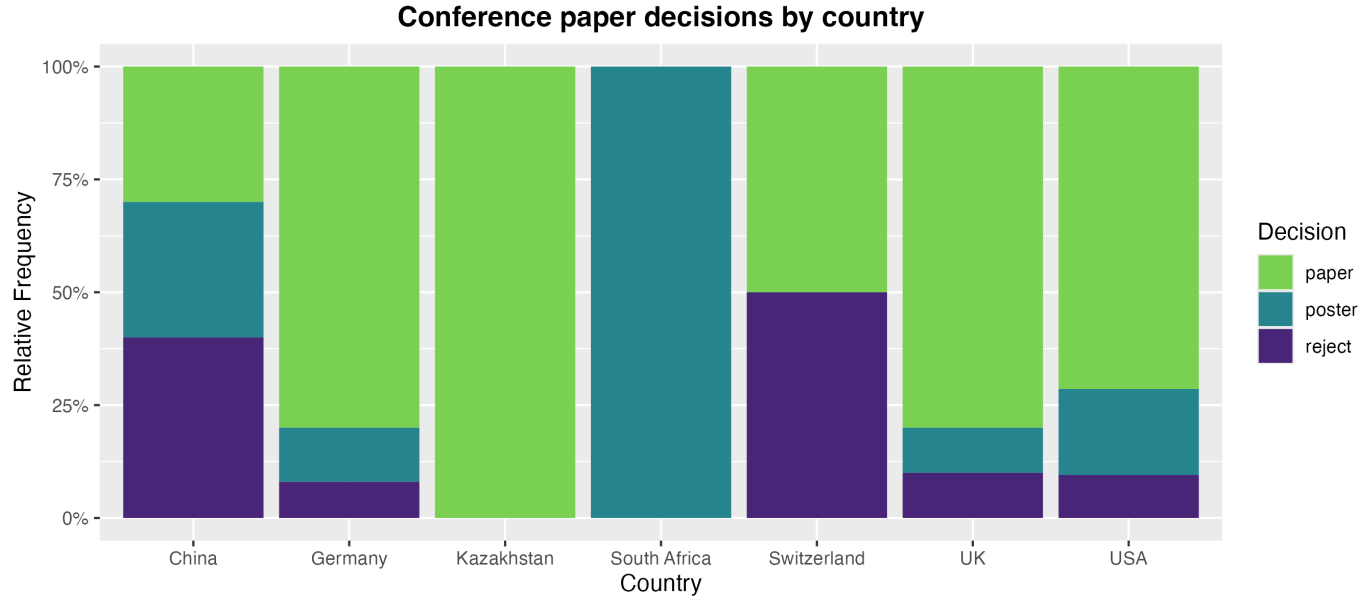
**Some random stuff**



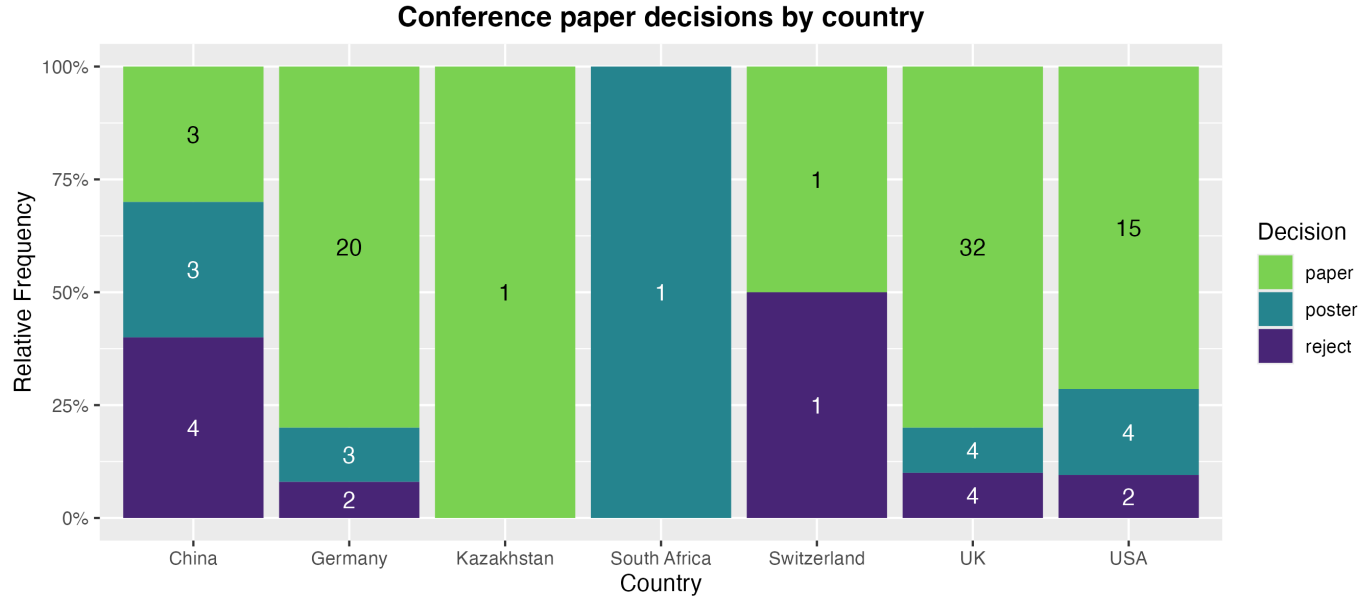
**Some random stuff**



- Display categorical data as frequencies **and** percentages, making sure that the number of observations is included.



- Display categorical data as frequencies **and** percentages, making sure that the number of observations is included.







# Basics of ggplot

## What is the Tidyverse?



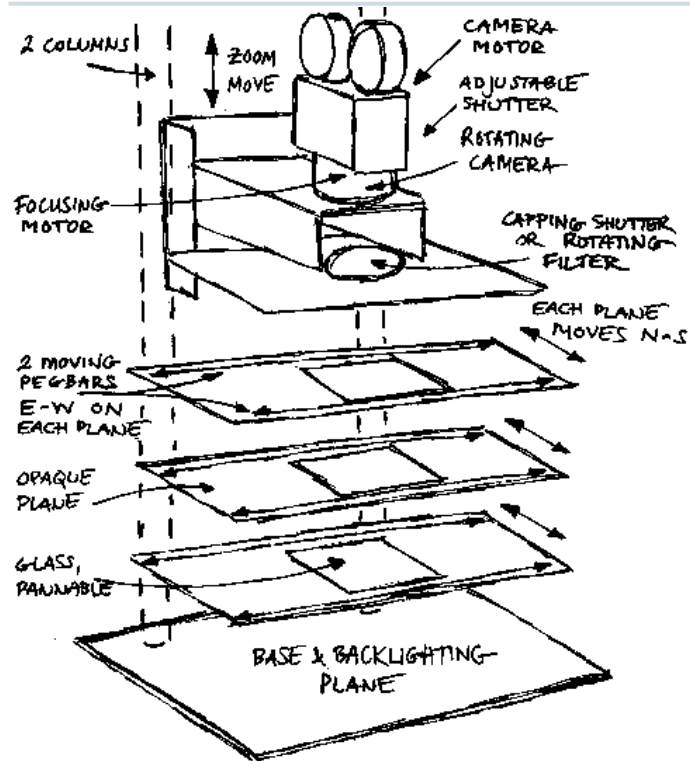
- family of packages developed by RStudio/Posit
- implement an own "dialect" of R
- still fully compatible with base R, but adding more syntax possibilities

## 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.



# Layers



# Main Components of the Grammar of Graphics

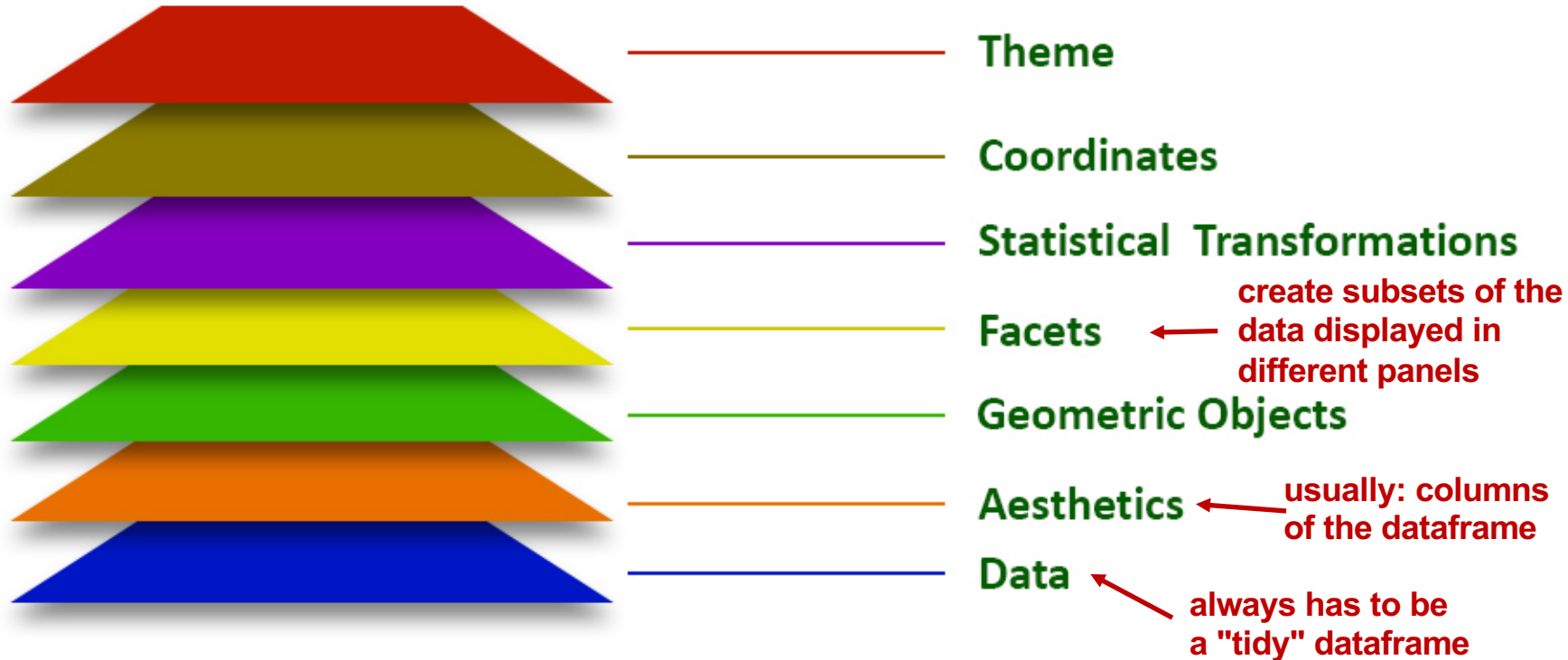


← create subsets of the data displayed in different panels

← usually: columns of the dataframe

← always has to be a "tidy" dataframe

# 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.



## Plot types

From sources across the web

Histogram



Overcoming the monster



Rags to Riches



Rebirth



Scatter plot



Tragedy



Voyage and Return



Comedy



Heatmap



Line plot



Quest



Bar chart



Box plot



Contour



Pie chart



Geographic plots



Area chart



Bar graph



Bubble chart



Conclusion



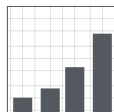
Donut chart



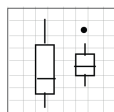
# Different Geoms (Plot Type) in ggplot2

## Two Variables (X,Y)

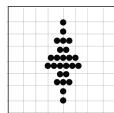
- Discrete X, continuous Y
- Visualise distribution of Y with respect to X



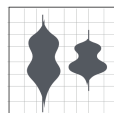
**geom\_col()**  
- heights of bars represent values



**geom\_boxplot()**  
- summarise distribution using median, hinges and whiskers

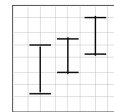


**geom\_jitter()**  
- adds jitter to prevent overplotting

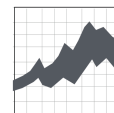


**geom\_violin()**  
- mirrored density plot (smoothed distribution)

## Visualising Errors and Uncertainties



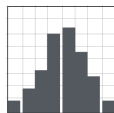
**geom\_errorbar()**  
- uncertainty in continuous Y against discrete X



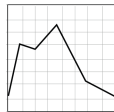
**geom\_ribbon()**  
- uncertainty in continuous Y against continuous X

## One Variable (X)

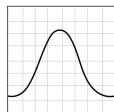
- Continuous X
- Visualise distribution of X



**geom\_histogram()**  
- divide X into bins and count no. observation



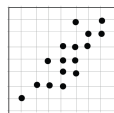
**geom\_freqpoly()**  
- display counts with lines  
- able to overlay multiple distributions



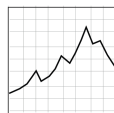
**geom\_density()**  
- smoothed version of the histogram

## Two Variables (X,Y)

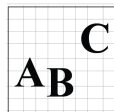
- Continuous X, continuous Y
- Visualise relationship between X and Y



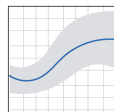
**geom\_point()**  
- scatterplot of X vs Y



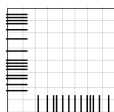
**geom\_line()**  
- connect data points, ordered by X  
- alt: geom\_path()



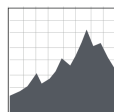
**geom\_text()**  
- labelling data points



**geom\_smooth()**  
- add smoothed curve  
- helps to see trends



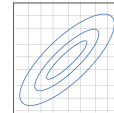
**geom\_rug()**  
- supplement 2D plot with 1D distribution along X and Y



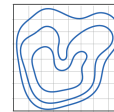
**geom\_area()**  
- can be stacked to see cumulative contribution

## 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

## Which plot types for which purpose?

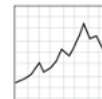
### ■ Scatterplots

- show / explore correlations between two variables
- metric data on both the x- and the y-axis



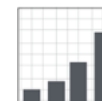
### ■ Line plots

- Lineplots are useful for showing e.g. change over time
- Count variable on y-axis, (at least) ordinal variable on x-axis



### ■ Barplots

- useful to show counts of categorical variables (e.g. number of men vs. number of women in parliament)...
- summary statistics (usually: means) of metric variables across different categories (e.g. mean height of humans vs. Klingons)



## Beeswarm plots

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

- e.g. *(gg)plotly* package
- and *shinyplots* ([shinyplots.io](https://shinyplots.io))

# Some tips and tricks

---

- Cheat sheets
- Code snippets
- Google/Stackoverflow is your friend 😊



Hands-on  
examples...

## Fake dataset: conference acceptance by country

- Plot the number of accepted abstracts by country as a stacked barplot showing percentages and absolute frequencies.
- (Fake data are generated in the code)



## Task 2

### Authentic dataset: Nettle (1999)

- hypothesis: linguistic diversity is correlated with climate factors
  - fertile environments → less reason to travel → less language contact → less linguistic diversity, and vice versa
- measured ecolocial risk using a country's Mean Growing Season (MGS) → how many months per year can you grow crops in the country?

