



# **DATA VISUALIZATION**

## ggplot2 & Grammar of Graphics

- Credits & References
- Pleas for Data Visualization
- The Grammar of Graphics
  - Basic Layers ([data](#), [aesthetics](#), [geometry](#))
  - Advanced Layers ([statistics](#), [scales](#), [facets](#), [coordinates](#), [themes](#))
- What to plot? Important visualizations for different applications

# CREDITS

---

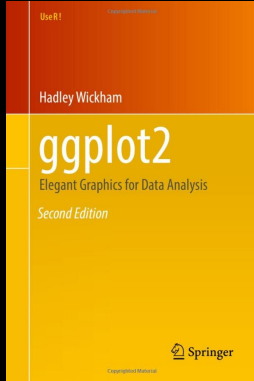
This slide deck is heavily inspired by the workshop “**Plotting anything with ggplot2**” by Tomas Lin Pedersen:

- [Workshop video part 1](#)
- [Workshop video part 2](#)
- [Slides](#)

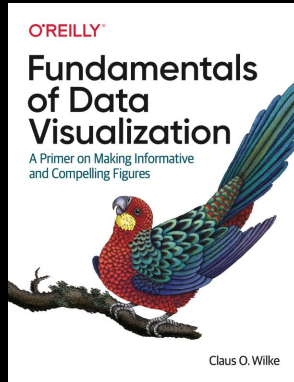


# REFERENCES

---



Wickham, Hadley. ggplot2. Springer Science + Business Media, LLC, 2016. Online verfügbar:  
<https://ggplot2-book.org/>

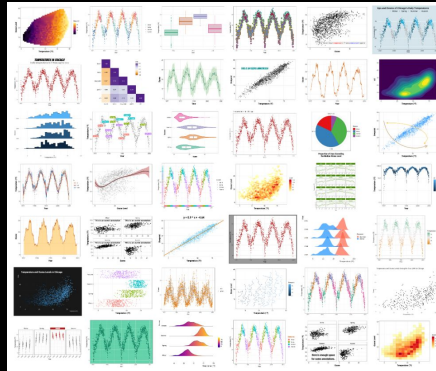


Wilke, C. Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures. First edition, O'Reilly Media, 2019.

Online verfügbar:  
<https://clauswilke.com/dataviz/index.html>

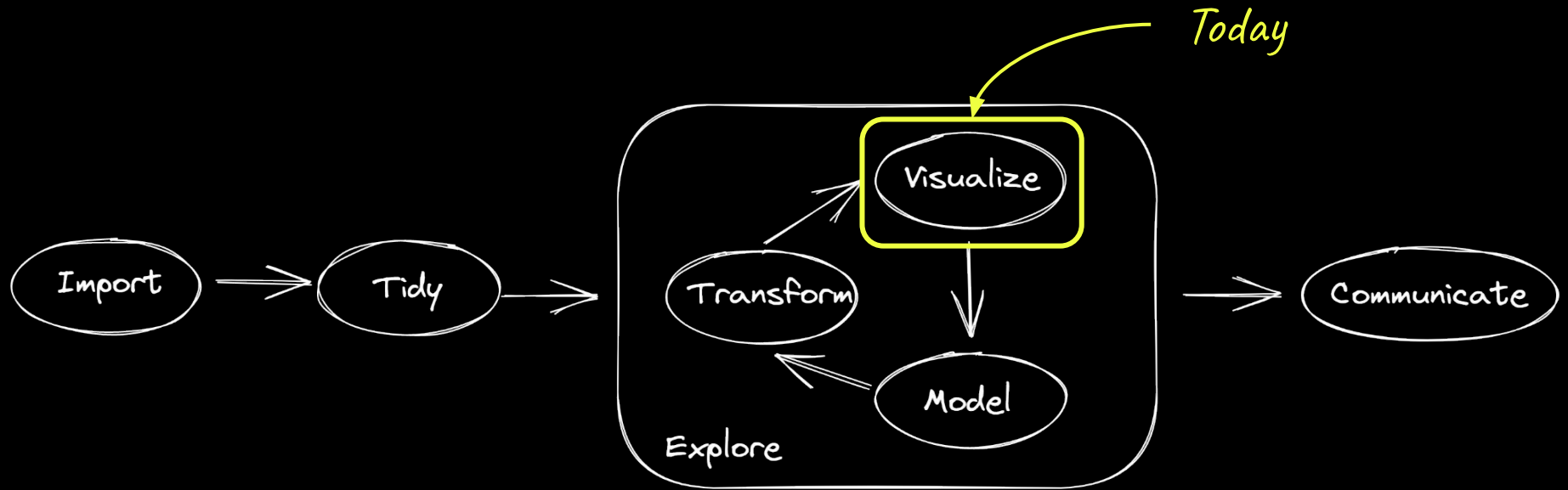
### A `ggplot2` Tutorial for Beautiful Plotting in R

<https://cedricscherer.netlify.app/2019/08/05/a-ggplot2-tutorial-for-beautiful-plotting-in-r/>



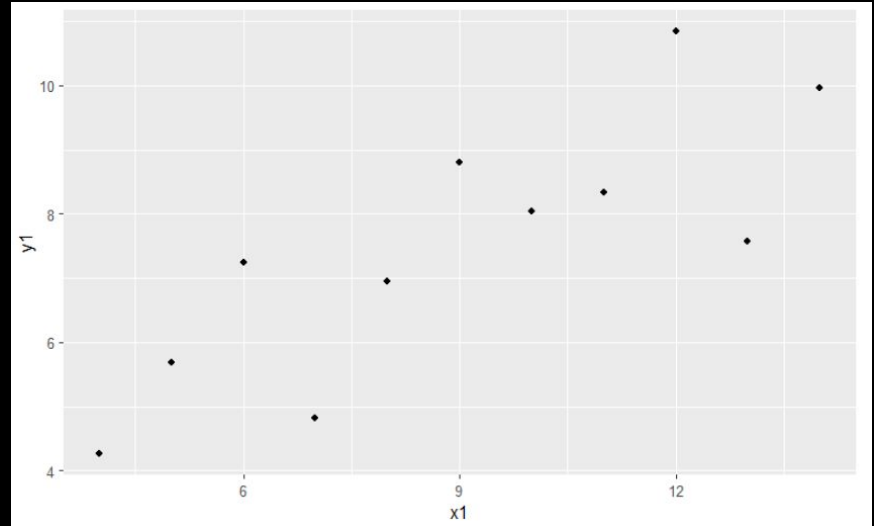
# DATA ANALYTICS PROCESS

## DATA VISUALIZATION



# **PLEAS FOR DATA VISUALIZATION**

- Find two examples [here](#)





# THE GRAMMAR OF GRAPHICS

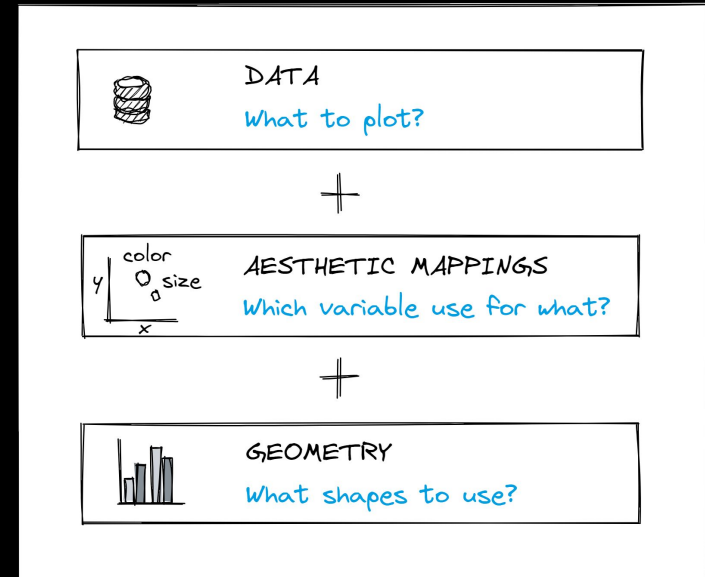
BASIC LAYERS

# THE GRAMMAR OF GRAPHICS

## BASIC LAYERS

Read more: <https://ggplot2-book.org/layers.html>

- In the **Grammar of Graphics**, a visualization consists of a minimum of three layers:
  - **Data**
  - **Mapping** of data to aesthetic elements
  - **Geometric shapes**
- **ggplot2** implements this idea → Visualizations are built as a stack of these layers



# THE GRAMMAR OF GRAPHICS

## EXAMPLE FOR BASIC LAYERS

---

`ggplot(covid) +` *What is the data?*

`aes(x = date, y = new_cases_smoothed_per_million) +`

`geom_line()` *Which geometric shape represents our data?*

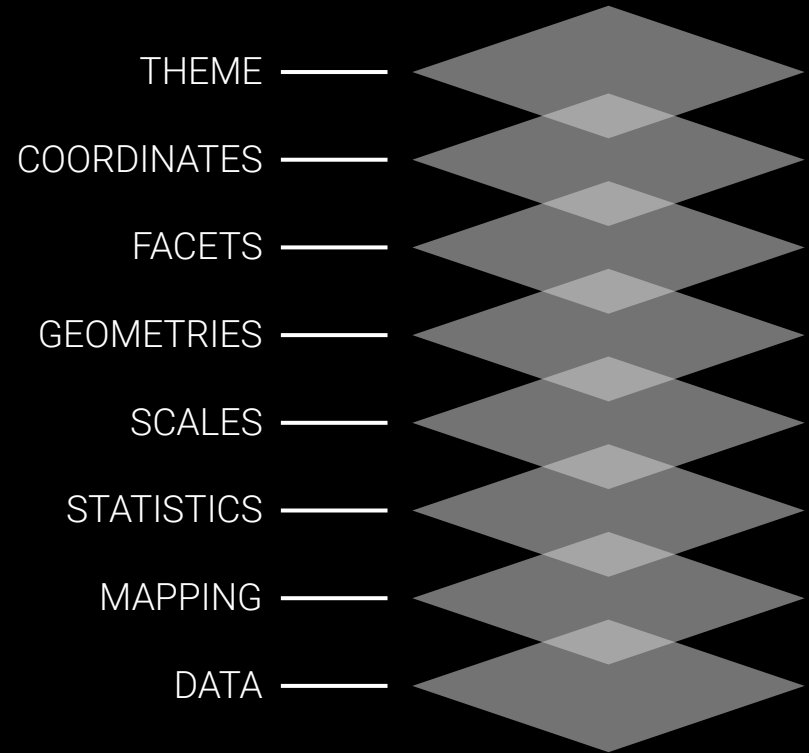
*How to map the data to aesthetics?*

# THE GRAMMAR OF GRAPHICS

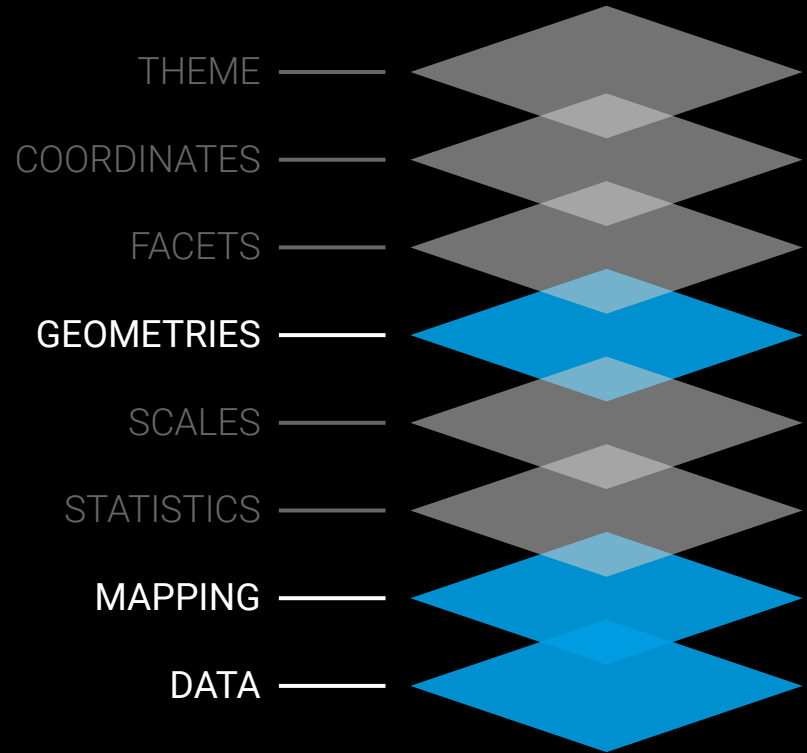
ALL LAYERS

*Any  
data  
visualization*

=



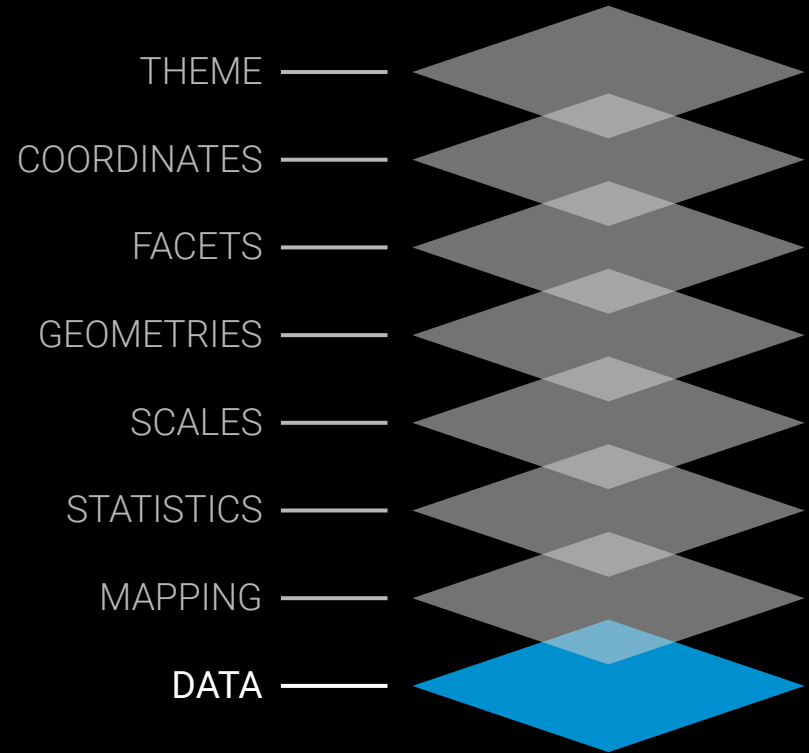
*Only these 3 are needed!  
Everything else has a  
default!*



# THE DATA LAYER

---

- **Data** must be provided as a data frame (**tibble**)
- Contains only
  - necessary variables
  - relevant rows and
  - the right level of aggregation
  - pre-computed statistics
- Toolset for data transformation (**dplyr**)

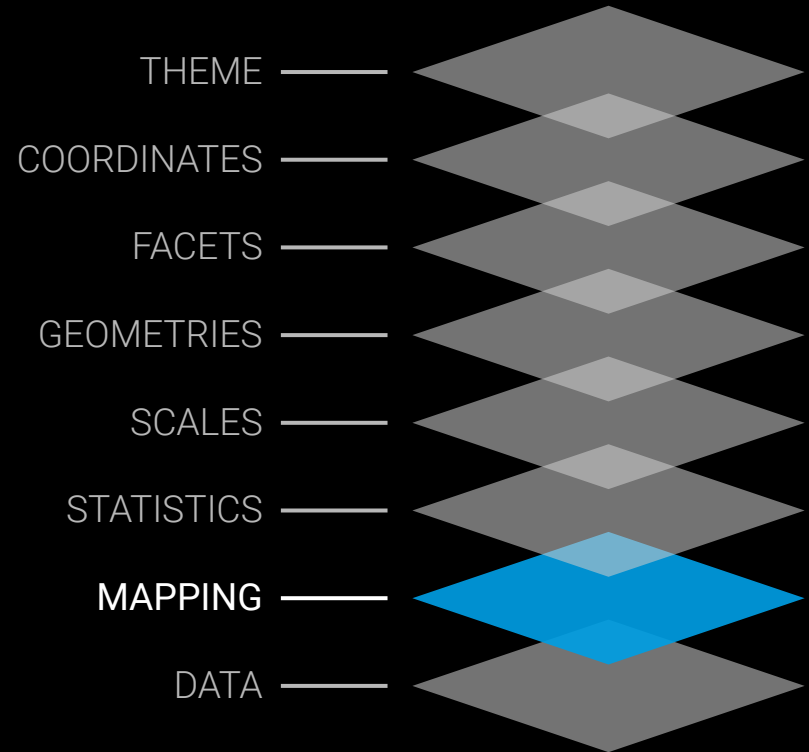


# THE MAPPING LAYER

---

Check out: <https://ggplot2tutor.com/aesthetics>

- The **aesthetics mapping** (**aes**) links variables in the data to graphics properties
- Most important: What should be shown on **x and y-axis**?
- More mappings:
  - Line color & style
  - Fill color
  - Point size & shape
  - Alpha
  - ...

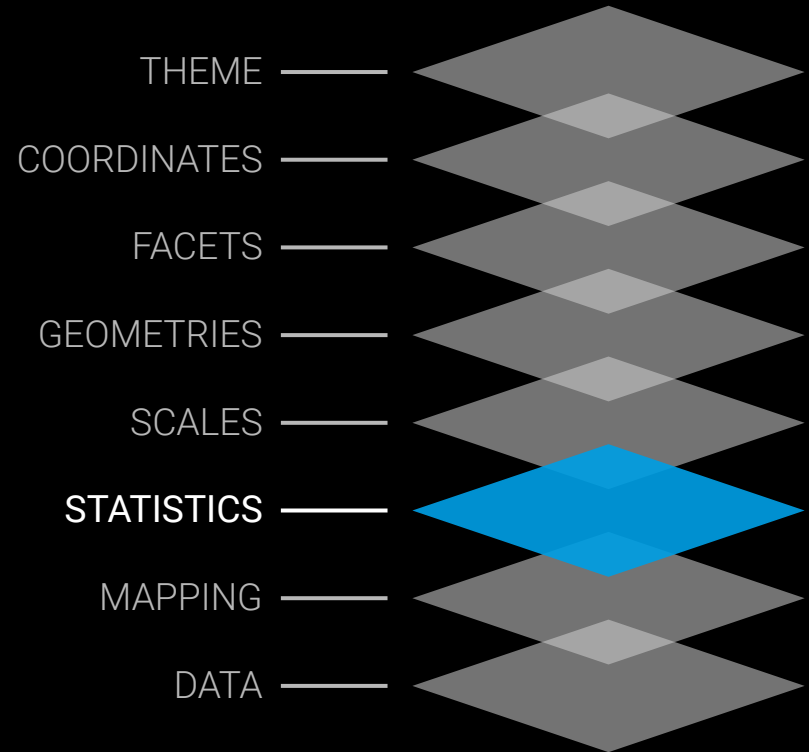




# THE STATISTICS LAYER

---

- If not pre-computed, **statistics** can be calculated by the visualization
- All geometries are assigned a default statistic (and vice versa)
- Example statistics:
  - **identity** → The value provided as is
  - **count** → Count rows
  - **bin** → Bin continuous variables
  - **density** → Estimate density
  - [Many more...](#)



```
tweets %>%
```

```
ggplot() +
```

```
stat_count(aes(x = screen_name))
```

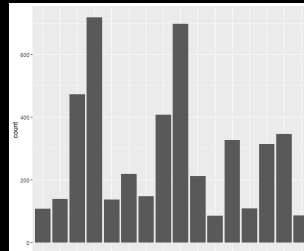
*The count statistics uses  
bars per default*

```
tweets %>%
```

```
ggplot() +
```

```
geom_bar(aes(x = screen_name))
```

*The bar geometry uses the  
count statistic per default*

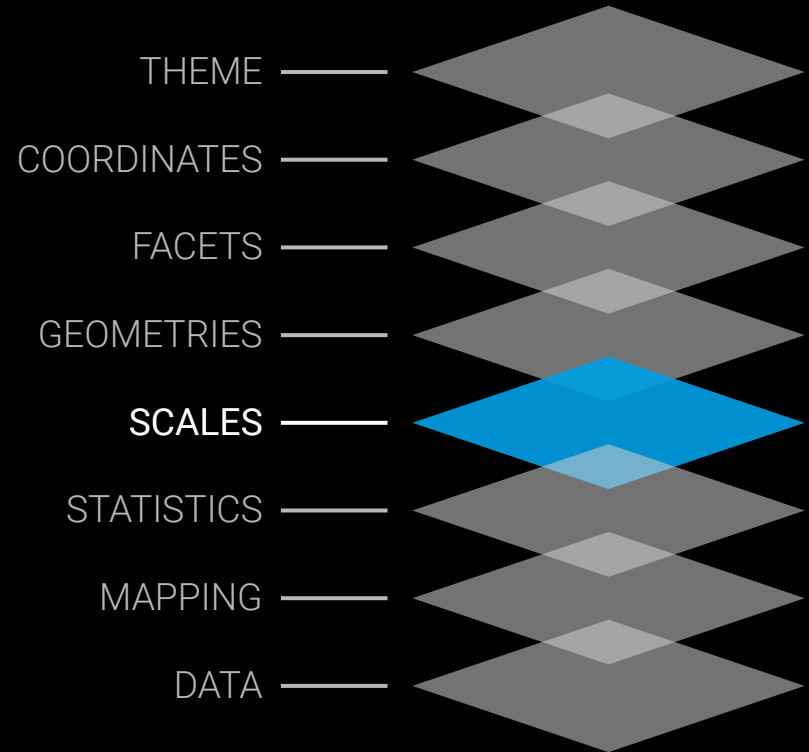


# THE SCALES LAYER

---

Check out: <https://ggplot2tutor.com/scales>

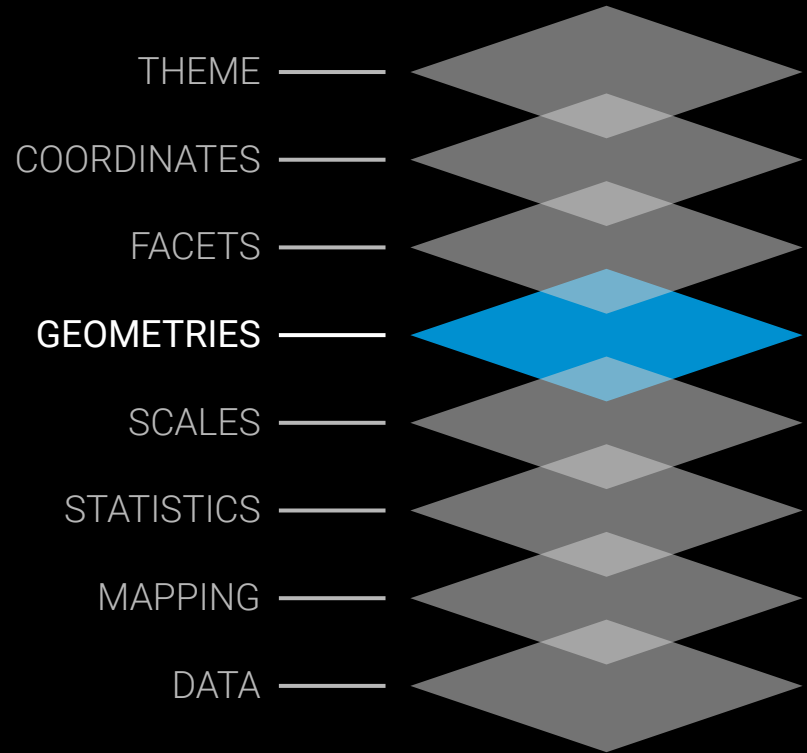
- All aesthetics mappings have a **scale** attached
- A **scale** maps values in the data to the x and y-axis, colors or sizes for shapes
- All scale functions follow the same naming scheme:
  - `scale_<aes>_<type>()`
- We use scales mainly for:
  - Color palettes
  - Axis labeling (breaks, formatting)



# THE GEOMETRY LAYER

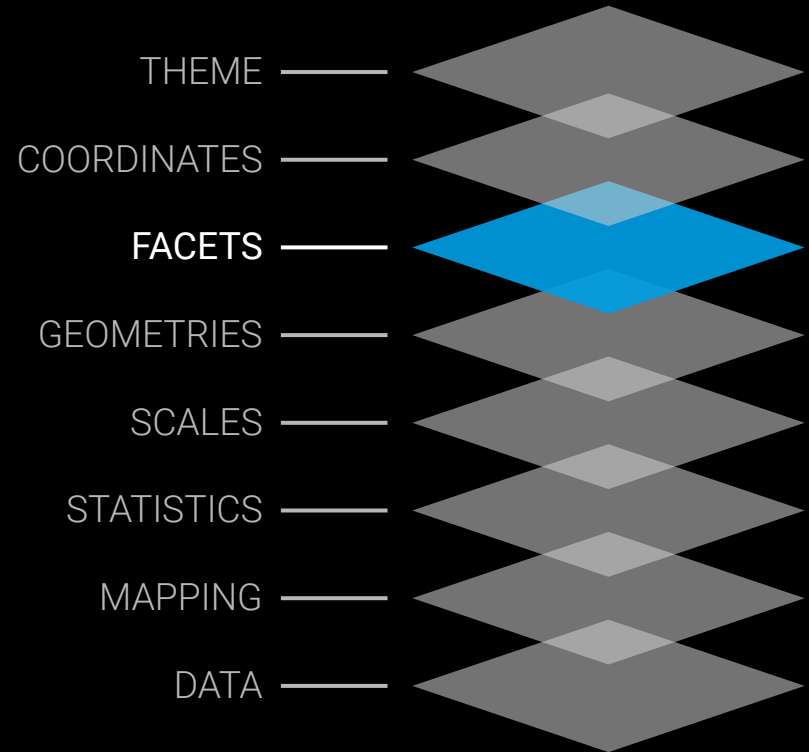
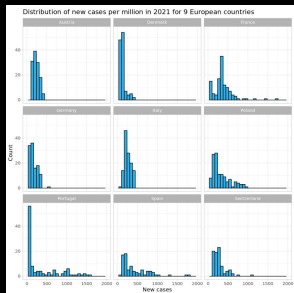
---

- The geometry is central to how the plot visualizes data
- Depending on the geometry, different aesthetics **can or must** be mapped
- We can add more than one geometry to a plot
- `geom_<type>()`



# THE FACETS LAYER

- Create small panels with the **same visualizations for different data**
- Panel logic determined by variable in the data
- Good to **avoid overplotting** and maintain readability!
- `facet_wrap()` vs `facet_grid()`

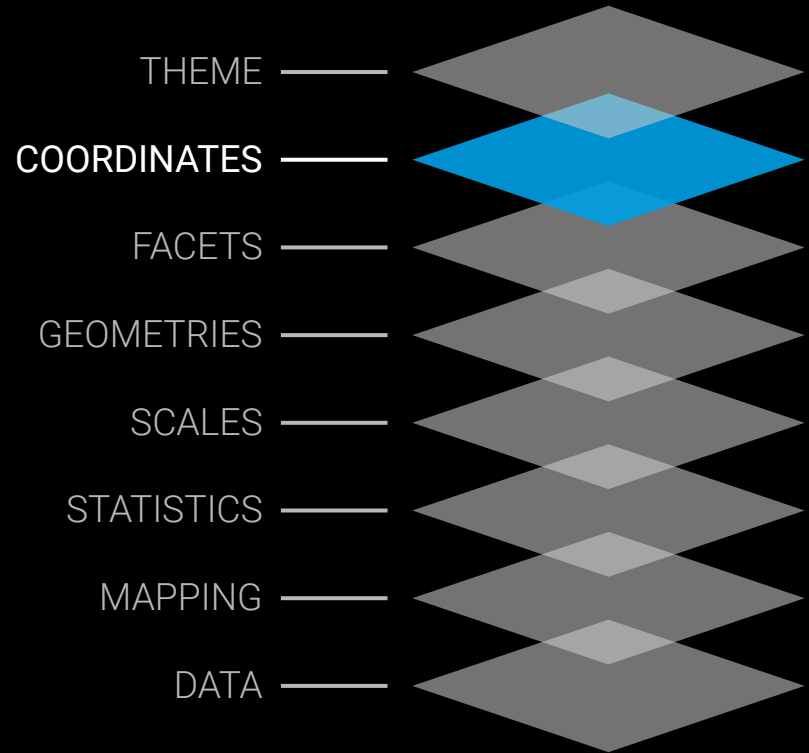


# THE COORDINATES LAYER

---

- Specify the coordinate system underlying the visualization:
  - Cartesian (default)
  - Polar
- Allows for **changing axis limits** (just like scales)
- `coord_flip()` is useful to quickly flip x and y

*We will rarely use this layer!*

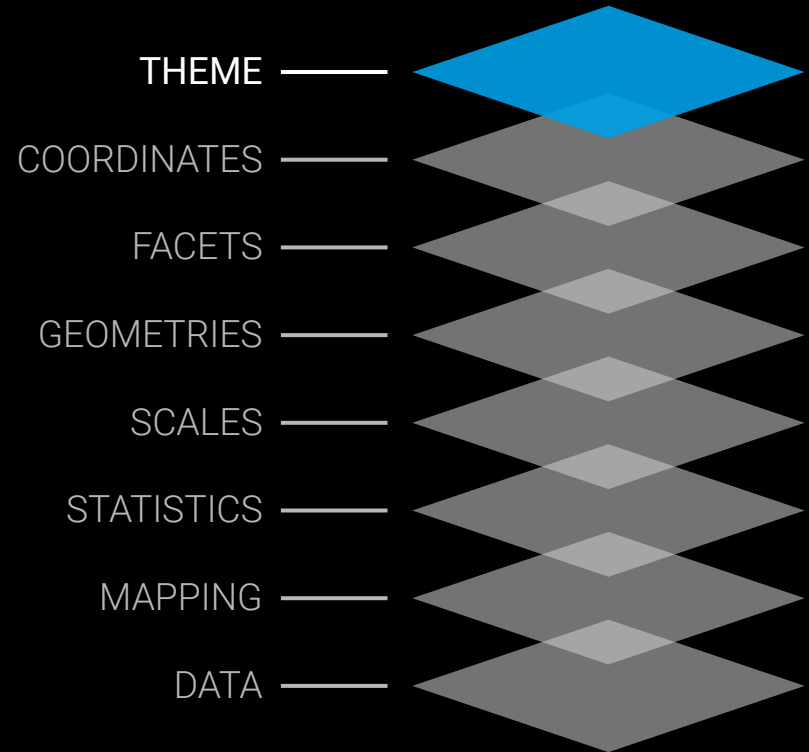


# THE THEME LAYER

---

Check out: <https://ggplot2tutor.com/theme>

- Style the plot
  - Background colors
  - Fonts (axis, titles)
  - Legends
  - ...
- There are predefined themes for us to use:
  - `theme_bw()`
  - `theme_light()`
  - `theme_dark()`



**WHAT TO PLOT?**

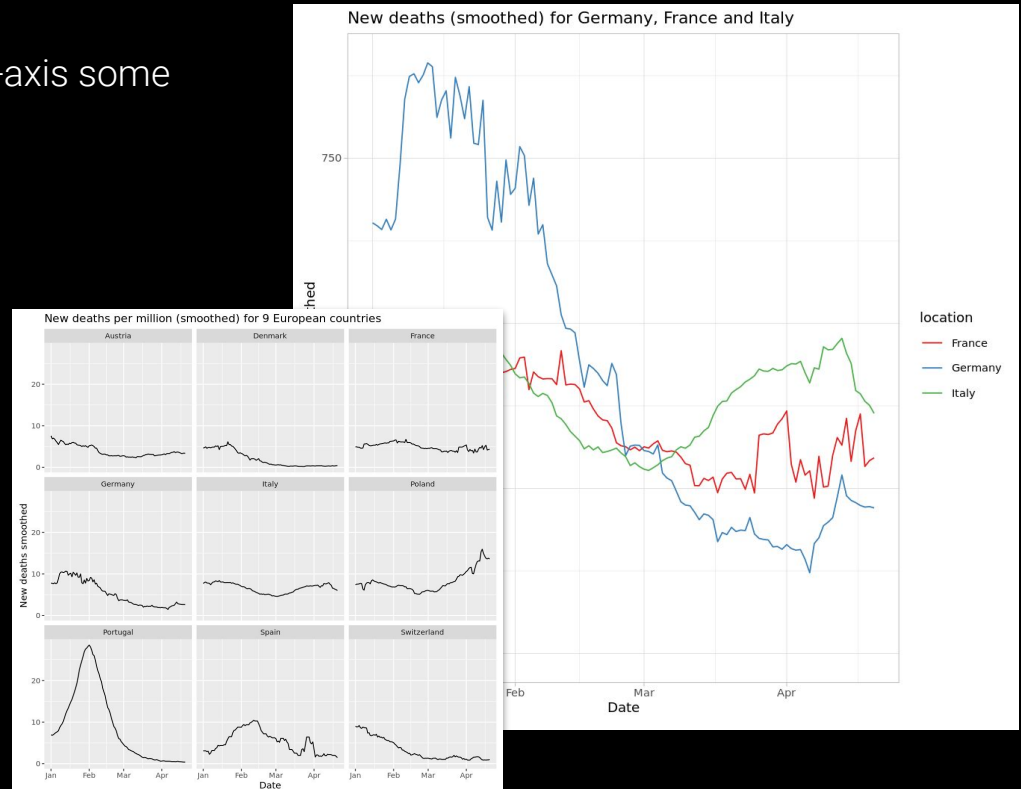


# WHAT TO PLOT?

## TRENDS & DEVELOPMENTS

Read more: <https://clauswilke.com/dataviz/time-series.html>

- x-axis displays the time (usually), the y-axis some value over time:
  - Line chart
  - Area Chart
  - One vs. multiple series
  - Facets
- Example: Covid19



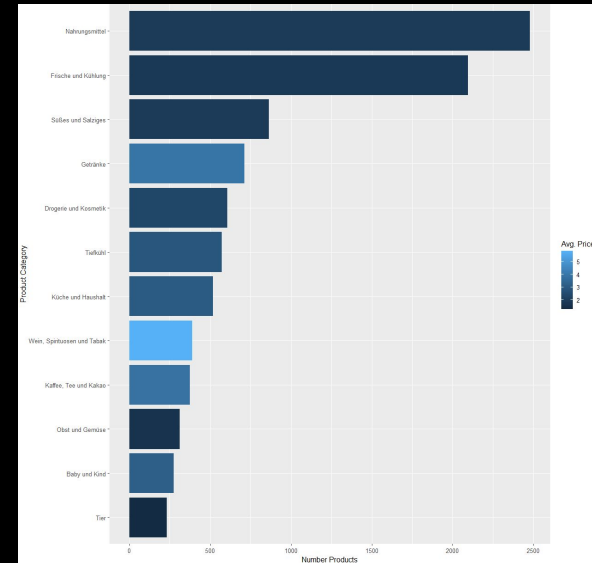
# WHAT TO PLOT?

## AMOUNTS & PROPORTIONS

Read more:

<https://clauswilke.com/dataviz/visualizing-proportions.html>

- A geometry's size (height, width, area) represents values in the data for easy **comparison**:
  - Bar Chart
    - next to each other
    - stacked
  - Pie chart



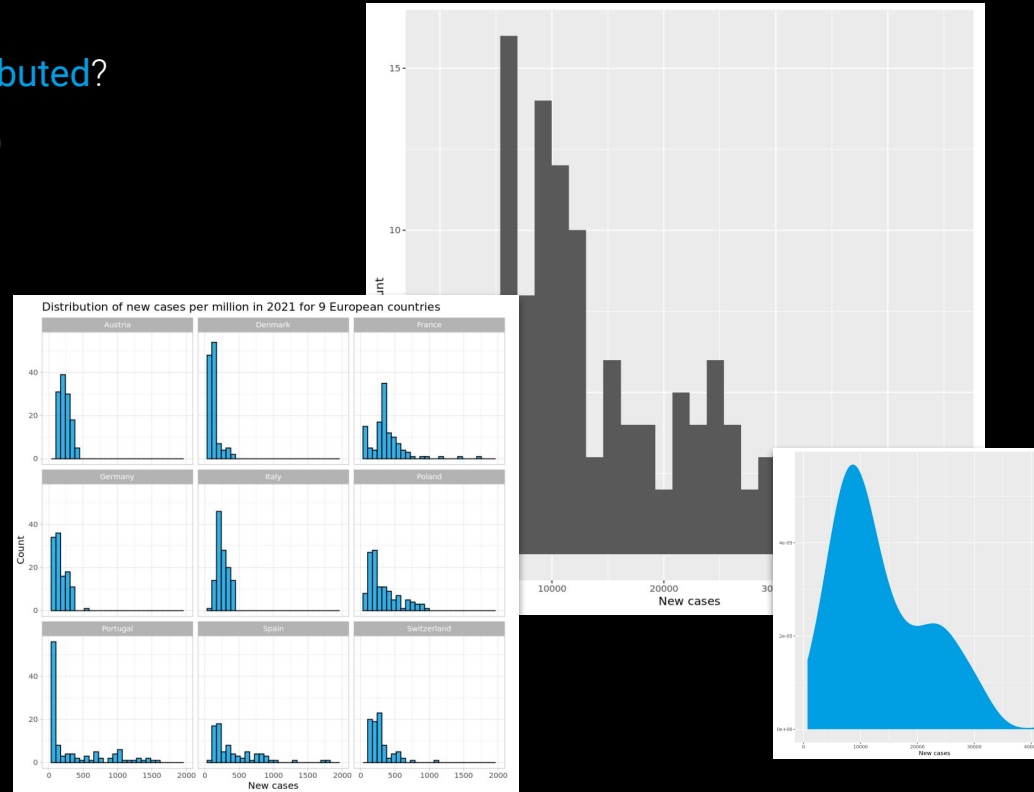
# WHAT TO PLOT?

## DISTRIBUTIONS

Read more:

<https://clauswilke.com/dataviz/histograms-density-plots.html>

- How are observations of a variable **distributed**?
  - Histogram (one vs. multiple series)
  - Density plot
  - Ridgeline Plots
  - Box plots



# WHAT TO PLOT?

## ASSOCIATIONS

Read more:  
<https://clauswilke.com/dataviz/visualizing-associations.html>

- What **associations** between variables can we find in the data?
  - Point diagram (*scatter plot*)
  - Trendlines
  - Heat maps

