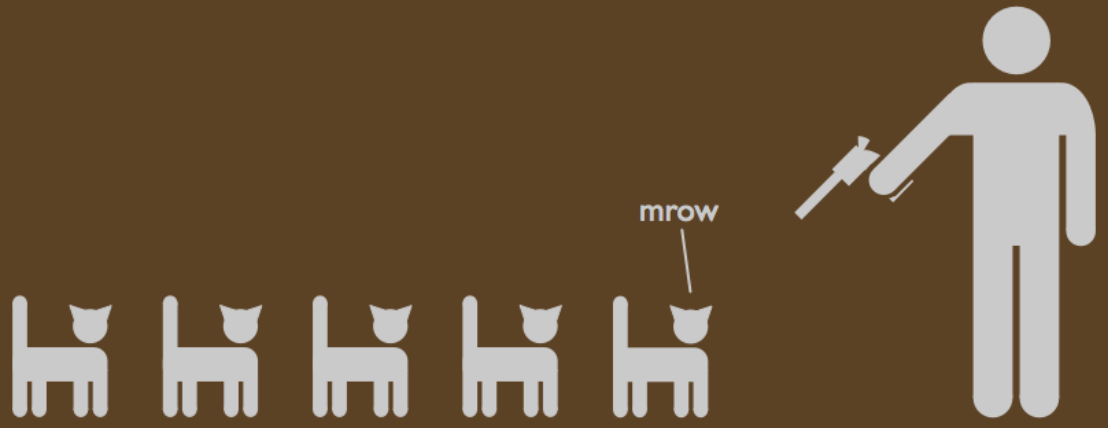
Digital humanities

Vizualizace vědeckých dat

Jindřich Marek

every time you make a powerpoint

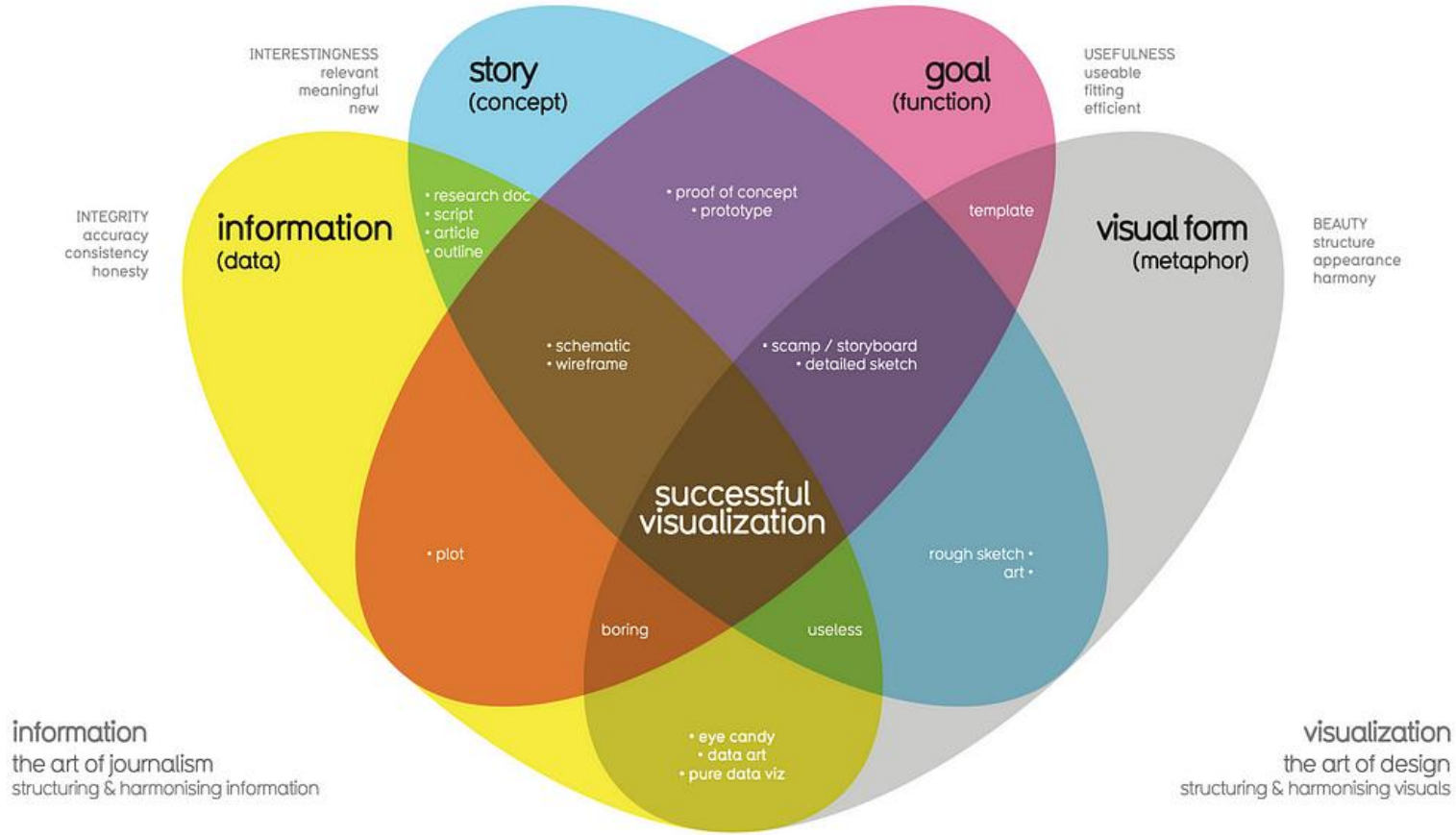


edward tufte kills a kitten

mark goetz

What Makes a Good Visualization?

explicit (implicit)



information
the art of journalism
structuring & harmonising information

visualization
the art of design
structuring & harmonising visuals

David McCandless
InformationisBeautiful.net

taken from new book
Knowledge is Beautiful

find out more
bit.ly/KIB_Books

Remove

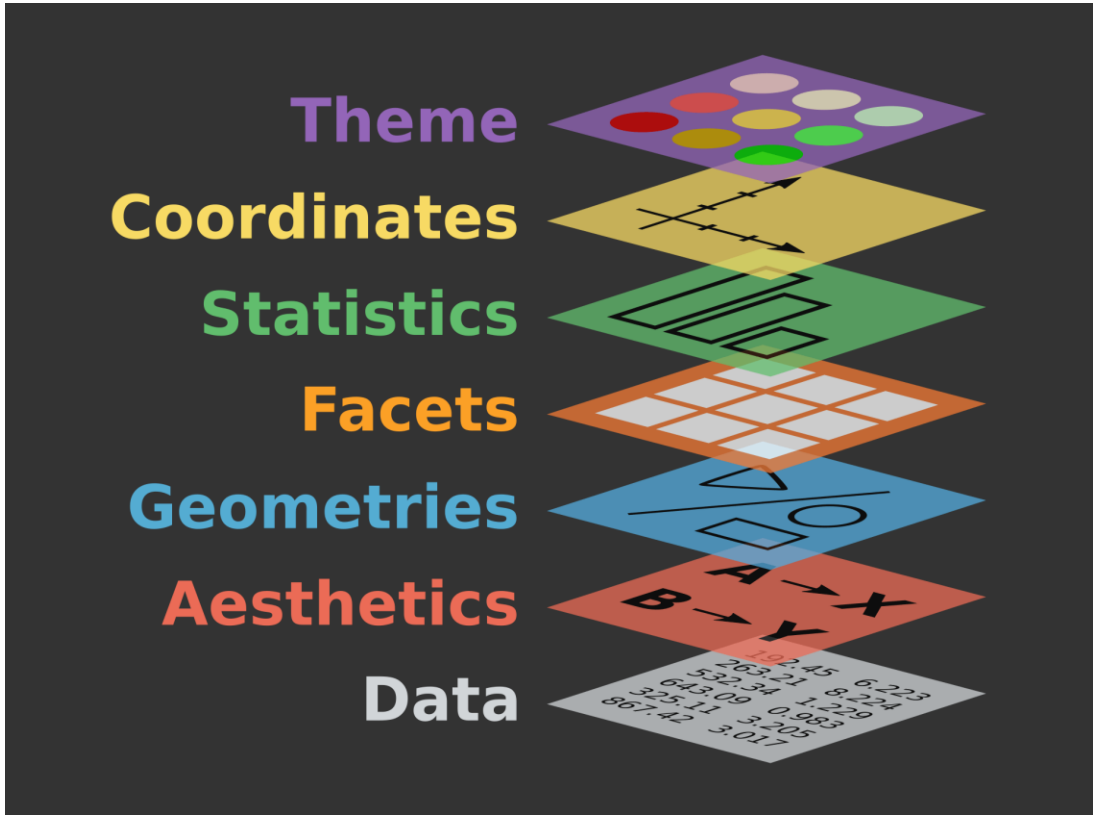
to improve

(the **data-ink** ratio)

Vizualizace: vizuální zobrazení dat

- intuitivní (více/méně, časová osa, …)
- přístupná (relevantní, srozumitelná)
- přesná (ale případně i shrnující)
- jednoduchá (abstrakce)
- elegantní (neodvádějící pozornost)

Grammar of graphics



- https://cfss.uchicago.edu/notes/grammar-of-graphics/
- implementace: ggplot2
  - https://r-graph-gallery.com/ggplot2-package.html

ggplot2

`ggplot2` is a `R` package dedicated to data visualization. It can greatly improve the quality and aesthetics of your graphics, and will make you much more efficient in creating them.

`ggplot2` allows to build almost any type of chart. The R graph gallery focuses on it so almost every section there starts with `ggplot2` examples.

This page is dedicated to general `ggplot2` tips that you can apply to any chart, like customizing a title, adding annotation, or using faceting.

If you love `ggplot2`, you will love my productive r workflow project where I show how it interacts with Quarto, Git and Github! ❤️

A WORLD OF GEOM

`ggplot2` builds charts through layers using `geom_` functions. Here is a list of the different available geoms. Click one to see an example using it.

`geom_bar` `geom_bin` `geom_boxplot` `geom_density` `geom_error` `geom_hex` `geom_hist` `geom_hline` `geom_jitter` `geom_label` `geom_line` `geom_point` `geom_polygon` `geom_rect` `geom_ribbon` `geom_rug` `geom_segment` `geom_smooth` `geom_text` `geom_tile` `geom_violin` `geom_vline`

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +                                  required
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
 stat = <STAT>, position = <POSITION>) +                  Not required, sensible defaults supplied
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

ggplot(data = mpg, **aes(**x = cty, y = hwy**))** Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5**)** Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

from Data to Viz

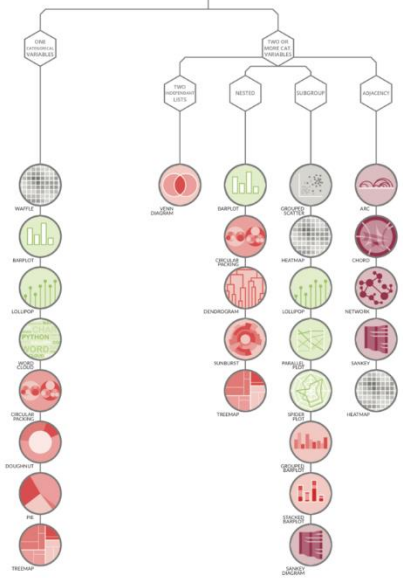
'From Data to Viz' is a classification of chart types based on input data format. It will help you find the perfect chart in three simple steps :

1. Identify what type of data you have.
2. Go to the corresponding decision tree and follow it down to a set of possible charts.
3. Choose the chart from the set that will suit your data and your needs best.

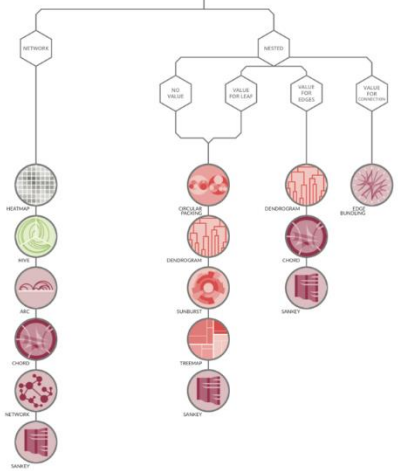
Dataviz is a world with endless possibilities and this project does not claim to be exhaustive. However it should provide you with a good starting point. For an interactive version and much more, visit:

data-to-viz.com

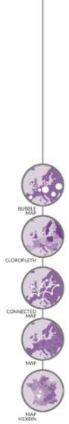
CATEGORIC



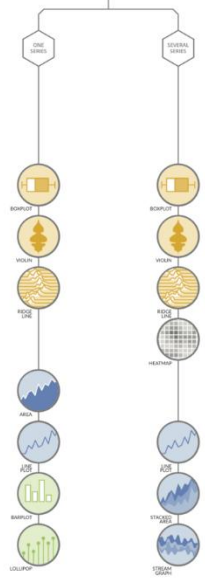
RELATIONAL



MAP



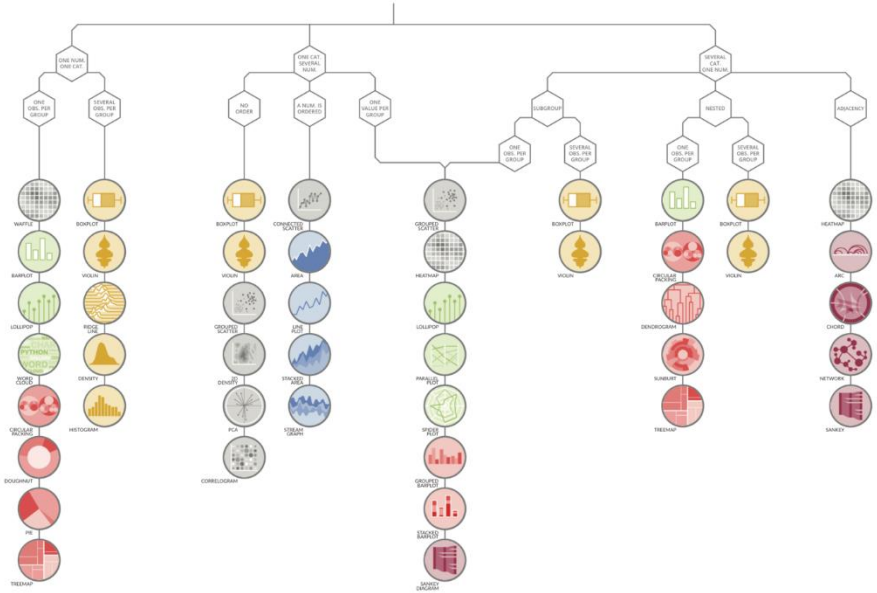
TIME SERIES



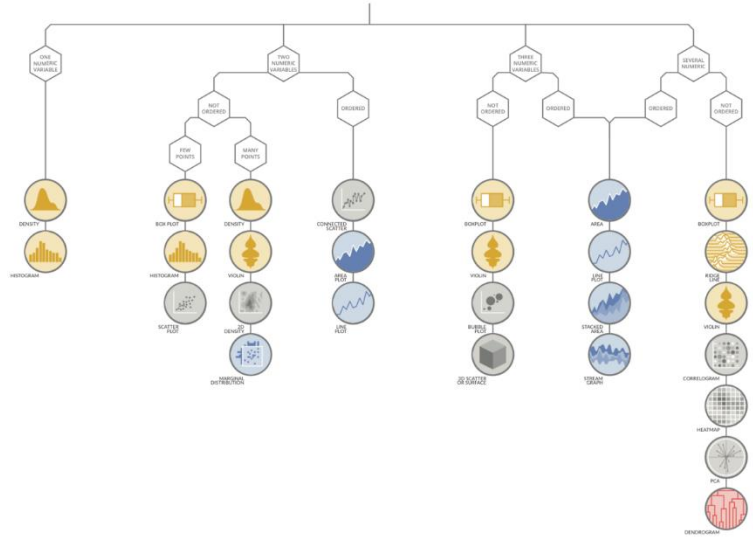
WHAT DO YOU WANT TO SHOW ?

- Distribution
- Correlation
- Ranking
- Part of a whole
- Evolution
- Maps
- Flow

CATEGORIC AND NUMERIC

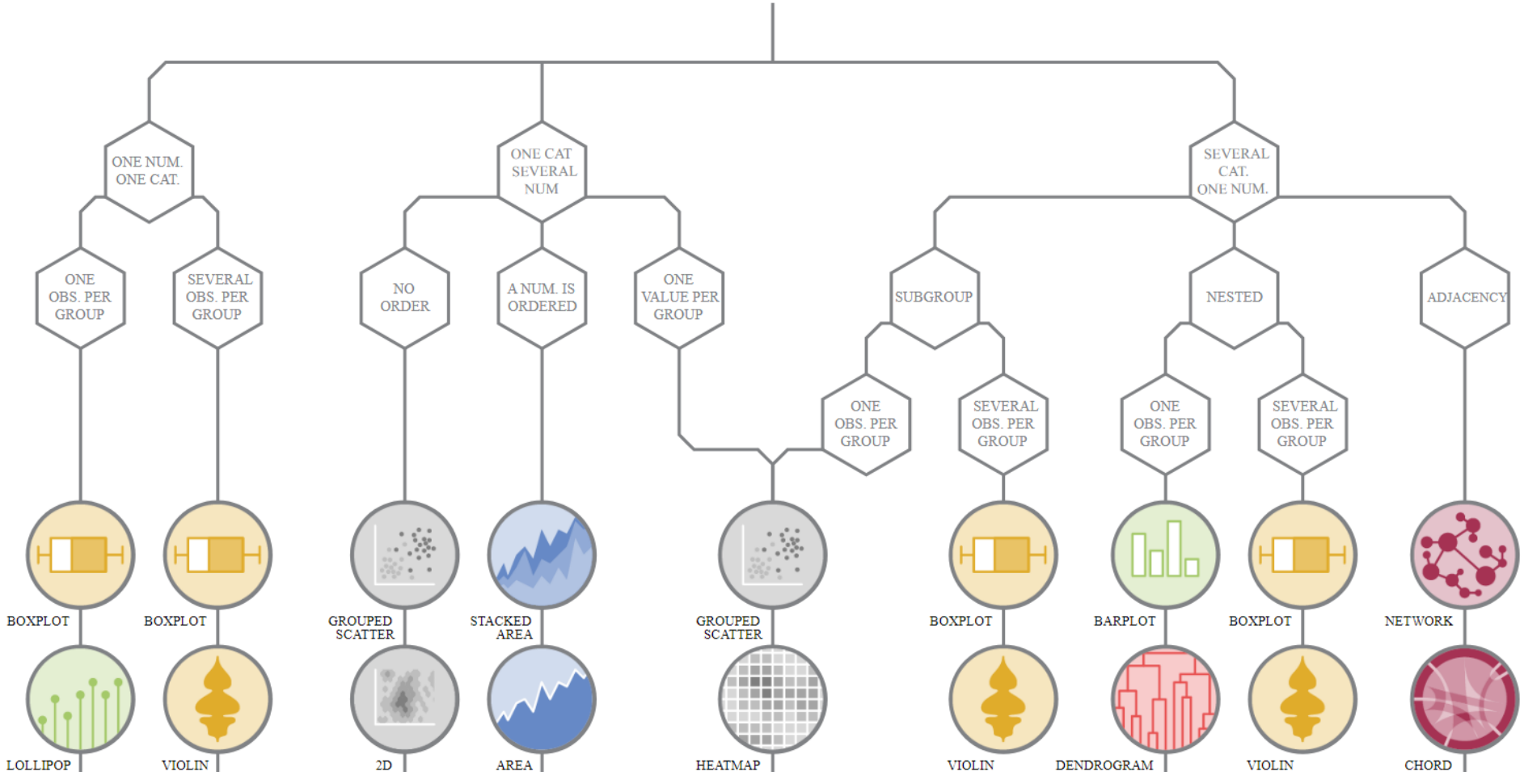


NUMERIC

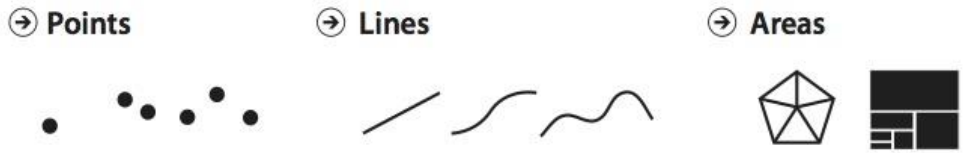


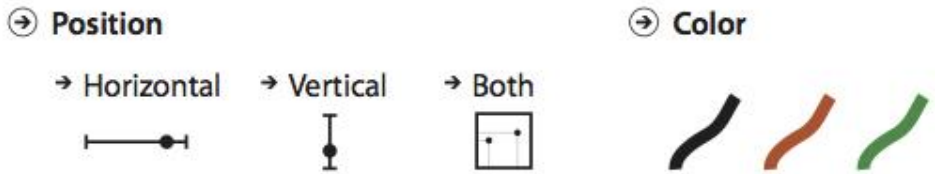
What kind of data do you have? Pick the main type using the buttons below. Then let the decision tree guide you toward your graphic possibilities.

Numeric | Categoric | Num & Cat | Maps | Network | Time series

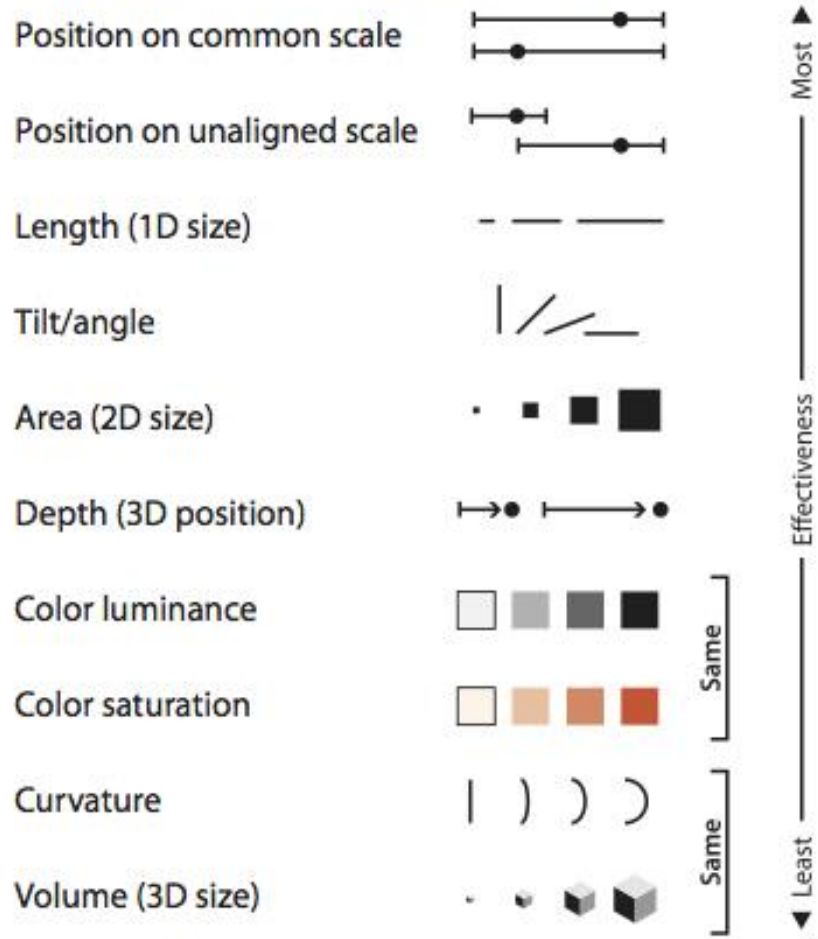


Značky a kanály

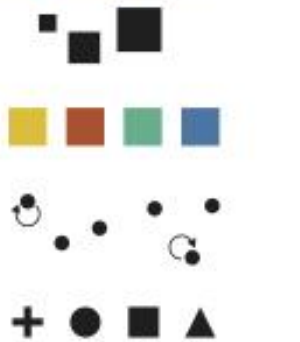




Magnitude Channels: Ordered Attributes



Identity Channels: Categorical Attributes



Napoleonovo tažení do Ruska 1812

- Charles Joseph Minard, 1869
  - množství vojáků, místa, řeky, teplota
  - „re-vize“: https://www.datavis.ca/gallery/re-minard.php
  - také mapa k Hannibalovu tažení: https://www.edwardtufte.com/tufte/minard-hannibal

Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.

Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.

Paris, le 20 Novembre 1869.

Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. ___ Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.

Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow en ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.

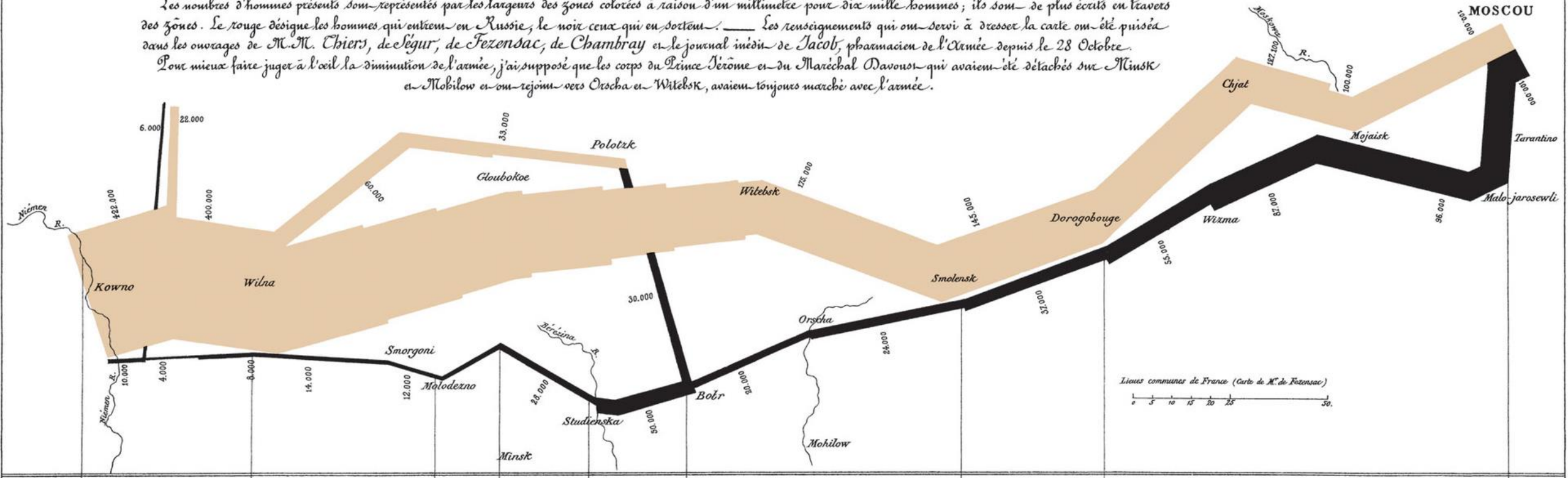
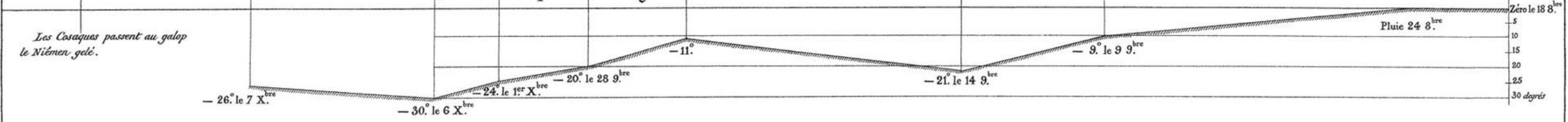


TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.



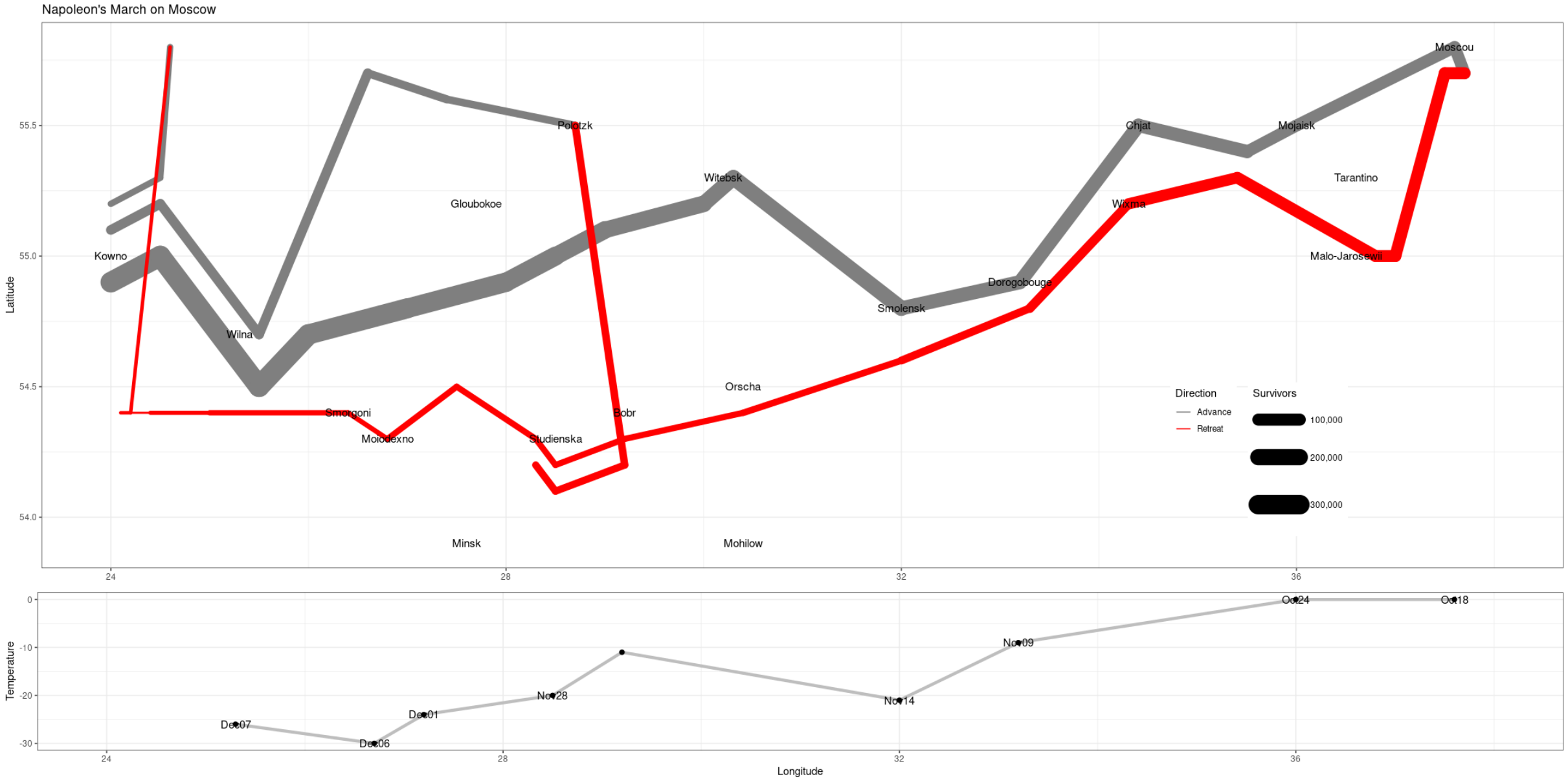
Autog. par Regnier, 8. Pas. Ste Marie St Gain à Paris.

Imp. Lith. Regnier et Dourdet.

Minardův graf v ggplot2

```
> install.packages(HistData)
```

- viz soubor v Moodlu
  - minard.R


Napoleon's March on Moscow
Moscou
Chjat
Mojaisk
Polotzk
Witebsk
Tarantino
Gloubokoe
Wixma
Kowno
Malo-Jarosewli
Dorogobouge
Smolensk
Wilna
Orscha
Smorgoni
Bobr
Moiodexno
Studienska
Minsk
Mohilow
Direction
Advance
Retreat
Survivors
100,000
200,000
300,000
Latitude
55.5
55.0
54.5
54.0
24
28
32
36
Oct24
Oct18
Nov09
Nov28
Nov14
Dec01
Dec07
Dec06
Temperature
0
-10
-20
-30
Longitude

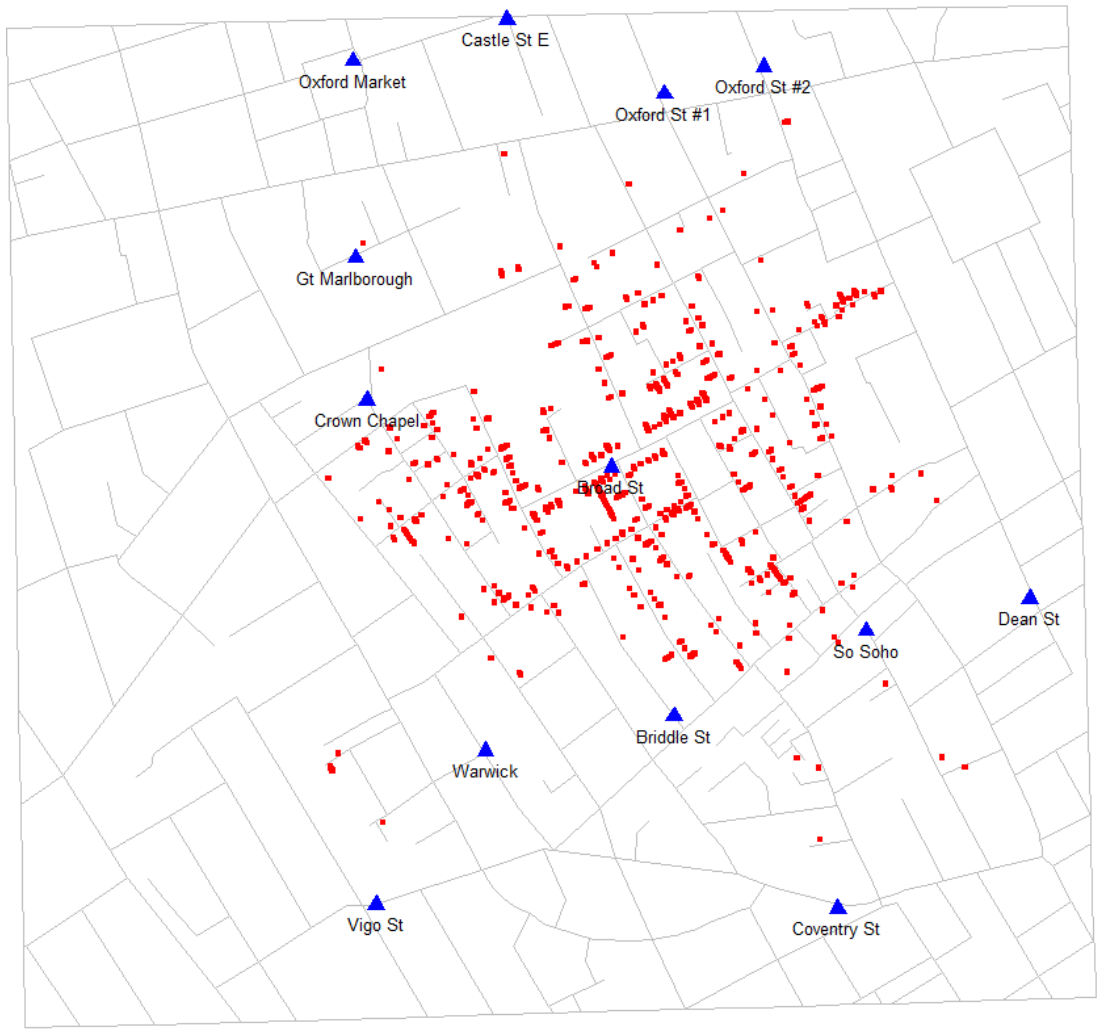
Cholera v Londýně, 1854

- John Snow
  - mapa: https://www.ph.ucla.edu/epi/snow/snowmap1_1854_lge.htm
  - Wikipedie: https://en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak
  - nová reprezentace dat:
    - https://www.leahmeisterlin.com/remappingsnow
    - https://www.r-bloggers.com/2013/03/john-snows-cholera-data-in-more-formats/

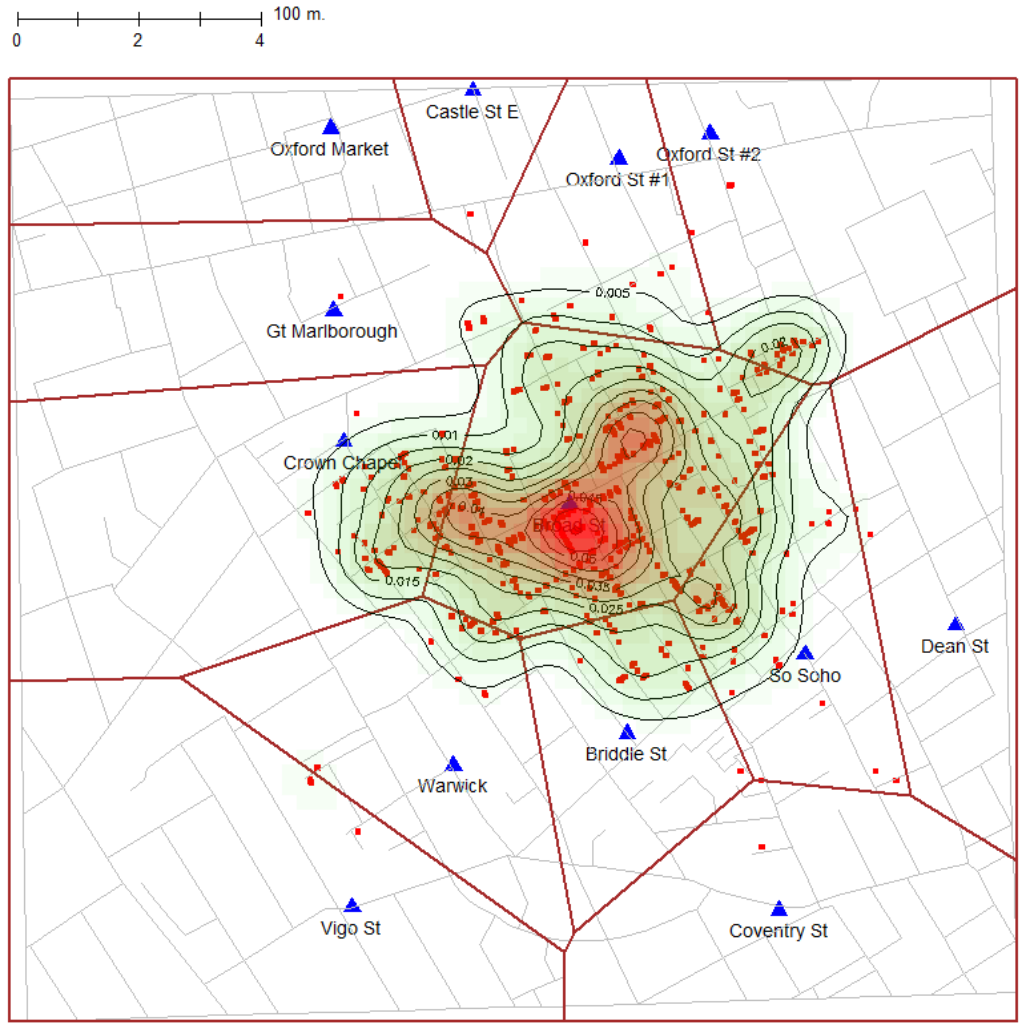
Snowova mapa v ggplot2

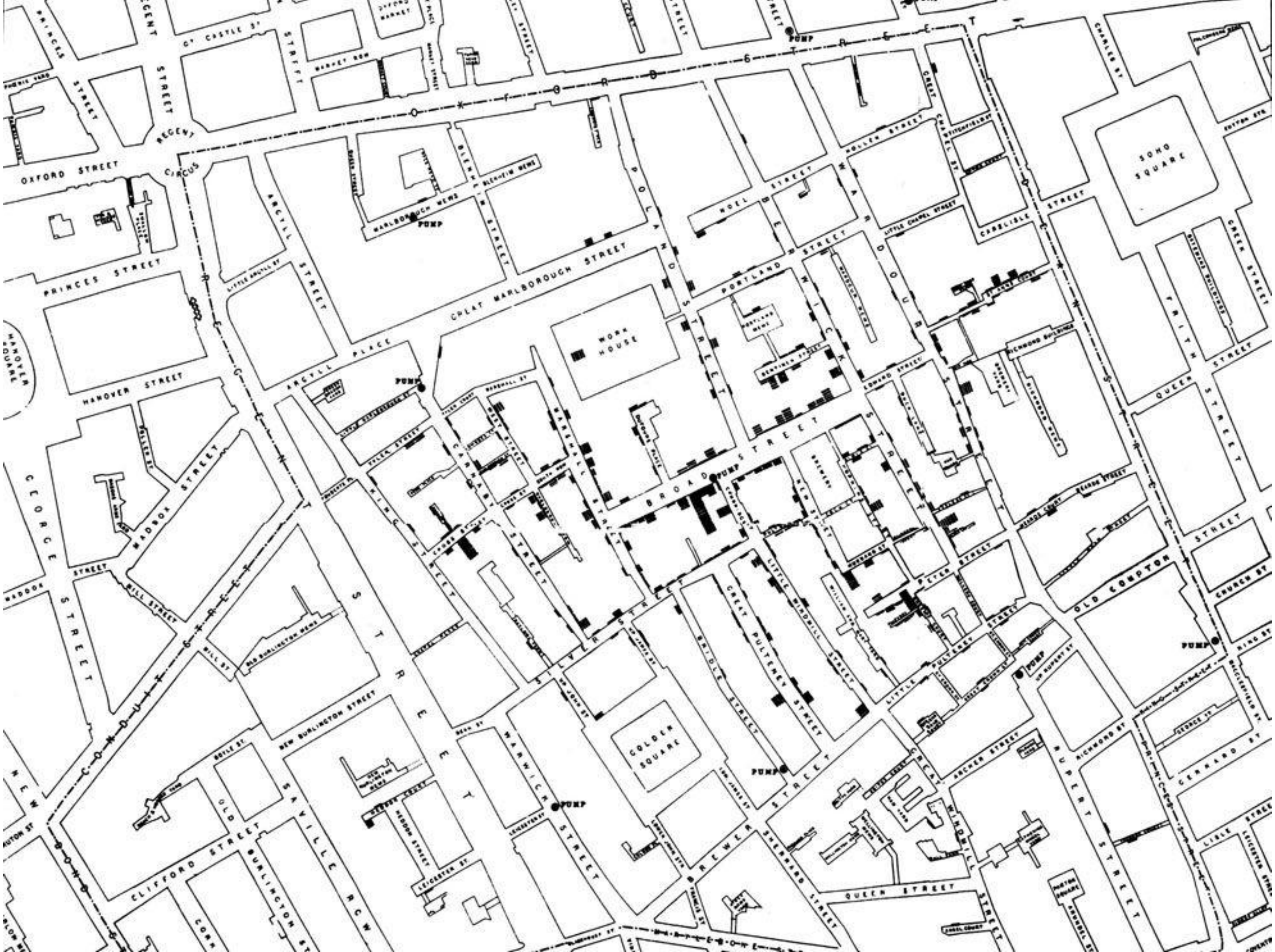
- viz soubory v Moodlu
  - snow_map1.R
  - snow_map2.R

Snow's Cholera Map of London (sp)



Snow's Cholera Map of London




OXFORD STREET
REGENT CIRCUS
PRINCES STREET
HANOVER SQUARE
HANOVER STREET
GEORGE STREET
MADDOX STREET
ARGYLL STREET
ARGYLL PLACE
LITTLE ARGYLL ST
GREAT MARLBOROUGH STREET
MARLBOROUGH MEWS
BLENHEIM MEWS
PUMP
POLAND STREET
NOEL STREET
PORTLAND STREET
WORK HOUSE
BROAD STREET
BERWICK STREET
WARDOUR STREET
HOLLEN STREET
LITTLE CHAPEL STREET
CARLISLE STREET
SOHO SQUARE
CHARLES ST
FRITH STREET
GREEK STREET
QUEEN STREET
OLD COMPTON STREET
CHURCH STREET
KING STREET
MARSHALL STREET
SILVER STREET
PETER STREET
GREAT PULTENEY STREET
LITTLE PULTENEY STREET
LITTLE WINDMILL STREET
BRIDLE STREET
GOLDEN SQUARE
BREWER STREET
WARWICK STREET
ARCHER STREET
RUPERT STREET
RICHMOND ST
GERRARD ST
LISLE STREET
REGENT STREET
CONDUIT STREET
NEW BOND ST
MILL STREET
OLD BURLINGTON STREET
NEW BURLINGTON STREET
BOYLE ST
CLIFFORD STREET
SAVILLE ROW
HEDDON STREET

Tidyverse a vizualizace dat

The **tidyverse** is a powerful collection of R packages that are actually data tools for transforming and visualizing data. All packages of the tidyverse share an underlying philosophy and common APIs.

The core packages are:

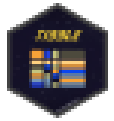
• **ggplot2**, which implements the grammar of graphics. You can use it to visualize your data.

• **dplyr** is a grammar of data manipulation. You can use it to solve the most common data manipulation challenges.

• **tidyr** helps you to create tidy data or data where each variable is in a column, each observation is a row end each value is a cell.

• **readr** is a fast and friendly way to read rectangular data.

• **purrr** enhances R's functional programming (FP) toolkit by providing a complete and consistent set of tools for working with functions and vectors.



• **tibble** is a modern re-imaginging of the data frame.

• **stringr** provides a cohesive set of functions designed to make working with strings as easy as possible

• **forcats** provide a suite of useful tools that solve common problems with factors.

dplyr

Filter

`filter()` allows you to select a subset of rows in a data frame.

```
> iris %>% #Select iris data of species "virginica"
     filter(Species=="virginica")
> iris %>% #Select iris data of species "virginica" and sepal length greater than 6.
     filter(Species=="virginica",
     Sepal.Length > 6)
```

Arrange

`arrange()` sorts the observations in a dataset in ascending or descending order based on one of its variables.

```
> iris %>% #Sort in ascending order of sepal length
     arrange(Sepal.Length)
> iris %>% #Sort in descending order of sepal length
     arrange(desc(Sepal.Length))
```

Combine multiple `dplyr` verbs in a row with the pipe operator `%>%`:

```
> iris %>% #Filter for species "virginica" then arrange in descending order of sepal length
     filter(Species=="virginica") %>%
     arrange(desc(Sepal.Length))
```

Mutate

`mutate()` allows you to update or create new columns of a data frame.

```
> iris %>% #Change Sepal.Length to be in millimeters
     mutate(Sepal.Length=Sepal.Length*10)
> iris %>% #Create a new column called SLMm
     mutate(SLMm=Sepal.Length*10)
```

Combine the verbs `filter()`, `arrange()`, and `mutate()`:

```
> iris %>%
     filter(Species=="Virginica") %>%
     mutate(SLMm=Sepal.Length*10) %>%
     arrange(desc(SLMm))
```

Summarize

`summarize()` allows you to turn many observations into a single data point.

```
> iris %>% #Summarize to find the median sepal length
     summarize(medianSL=median(Sepal.Length))
> iris %>% #Filter for virginica then summarize the median sepal length
     filter(Species=="virginica") %>%
     summarize(medianSL=median(Sepal.Length))
```

You can also summarize multiple variables at once:

```
> iris %>%
     filter(Species=="virginica") %>%
     summarize(medianSL=median(Sepal.Length),
               maxSL=max(Sepal.Length))
```

`group_by()` allows you to summarize within groups instead of summarizing the entire dataset:

```
> iris %>% #Find median and max sepal length of each species
     group_by(Species) %>%
     summarize(medianSL=median(Sepal.Length),
     maxSL=max(Sepal.Length))
> iris %>% #Find median and max petal length of each species with sepal length > 6
     filter(Sepal.Length>6) %>%
     group_by(Species) %>%
     summarize(medianPL=median(Petal.Length),
     maxPL=max(Petal.Length))
```

ggplot2

Scatter plot

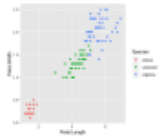
Scatter plots allow you to compare two variables within your data. To do this with ggplot2, you use `geom_point()`

```
> iris_small ← iris %>%
 filter(Sepal.Length > 5)
> ggplot(iris_small, aes(x=Petal.Length, #Compare petal width and length
                         y=Petal.Width)) +  geom_point()
```

Additional Aesthetics

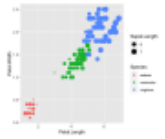
Color

```
> ggplot(iris_small, aes(x=Petal.Length,
        y=Petal.Width,
        color=Species)) +
   geom_point()
```



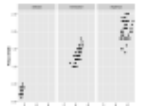
Size

```
> ggplot(iris_small, aes(x=Petal.Length,
                         y=Petal.Width,
                         color=Species,
                         size=Sepal.Length)) +
   geom_point()
```



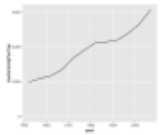
Faceting

```
> ggplot(iris_small, aes(x=Petal.Length,
                         y=Petal.Width)) +
    geom_point()+
    facet_wrap(~Species)
```



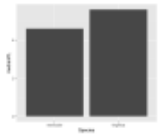
Line Plots

```
> by_year ← gapminder %>%
    group_by(year) %>%
    summarize(medianGdpPerCap=median(gdpPercap))
> ggplot(by_year, aes(x=year,
                      y=medianGdpPerCap))+
    geom_line()+
    expand_limits(y=0)
```



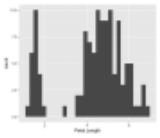
Bar Plots

```
> by_species ← iris %>%
    filter(Sepal.Length>6) %>%
    group_by(Species) %>%
    summarize(medianPL=median(Petal.Length))
> ggplot(by_species, aes(x=Species,
                         y=medianPL)) +
    geom_col()
```



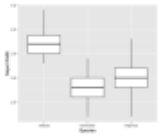
Histograms

```
> ggplot(iris_small, aes(x=Petal.Length))+
    geom_histogram()
```



Box Plots

```
> ggplot(iris_small, aes(x=Species,
                         y=Sepal.Width))+
    geom_boxplot()
```



Exkurs: dataset ggplot2movies