Silesian University of Technology

RESEARCH UNIVERSITY

EXCELLENCE INITIATIVE

Ministry of Science
and Higher Education

Programming for Cognitive Science

Lecture 2 – R basics. Data visualization in R.

Joanna Tobiasz, PhD
Anna Papiez, PhD

Department of Data Science and Engineering

Plan for today

01

R basics

Grouping functions, data sorting

02

Data visualization in R

Silesian
University
of Technology

RESEARCH
UNIVERSITY
EXCELLENCE INITIATIVE
Ministry of Science
and Higher Education

Part 1

Grouping functions

Grouping functions

For th better performance and preattier code, ”for” and ”while” loops can be substituted with grouping functions:

- apply
- sapply
- lapply
- tapply
- by
- aggregate

Grouping functions

For th better performance and preattier code, ”for” and ”while” loops can be substituted with grouping functions:

- apply
- sapply
- lapply
- tapply
- by
- aggregate

Grouping functions

```
for (i in 1:N) {
do_something_with_each_i
}
```

```
sapply(1:N, function_doing_sth_with_each_value)
lapply(1:N, function_doing_sth_with_each_value)
```

For each row or column of the data frame/matrix:

```
apply(object_name, margin, function_doing_sth_with_each_row/column)
```

1 for rows,
2 for columns

apply

Applies a function to matrix rows or columns

```
apply(X, MARGIN, FUN)
```

`X`	matrix
`MARGIN`	dimension over which the function will be applied
`FUN`	function

apply

```
M
     [,1] [,2] [,3] [,4]
[1,] 1    5     9   13
[2,] 2    6    10   14
[3,] 3    7    11   15
[4,] 4    8    12   16

apply(M, 1, min)
[1] 1 2 3 4
```

lapply

Applies a function to each list element, returns list

```
lapply(L, FUN)
```

`L`	list
`FUN`	function

lapply

```
L <- list(a = 1, b = 1:3, c = 10:100)
lapply(L, length)
   $a
   [1] 1
   $b
   [1] 3
   $c
   [1] 91
```

sapply

Applies a function to each list element, returns vector

```
lapply(L, FUN)
```

`L`	list
`FUN`	function

sapply

```
L <- list(a = 1, b = 1:3, c = 10:100)
sapply(L, length)
   a  b  c
   1  3 91
```

mapply

Applies a function to the 1st elements of each structure, and then the 2nd elements of each, etc. Returns a vector.

```
mapply(FUN, ...)
```

`FUN`	function
...	arguments (vectors, lists, etc.)

mapply

```
mapply(sum, 1:5, 1:5, 1:5)
[1]  3  6  9 12 15
```

tapply

Applies a function to subsets of a vector defined by a factor

```
tapply(X, INDEX, FUN)
```

`X`	vector
`INDEX`	factor
`FUN`	function

tapply

```
x <- 1:20
y <- factor(rep(letters[1:5], each = 4))

tapply(x, y, sum)
    a  b  c  d  e
   10 26 42 58 74
```

by

Applies a function to subsets of a data frame defined by a factor. Returns list.

```
by(data, INDICES, FUN)
```

`data`	data frame
`INDICES`	factor
`FUN`	function

by

```
data("iris")
attach(iris)
head(iris)
```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width
     Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

by

```
by(iris, list(Species=iris$Species), function(x){
    y <- subset(x, select= -Species)
    apply(y, 2, mean)
      } )
```

```
Species: setosa
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       5.006        3.428        1.462        0.246
---------------------------------------------------------------------------
Species: versicolor
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       5.936        2.770        4.260        1.326
---------------------------------------------------------------------------
Species: virginica
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       6.588        2.974        5.552        2.026
```

aggregate

Applies a function to subsets of a data frame defined by a list. Returns data frame.

```
aggregate(data, by, FUN)
```

`data`	data frame
`by`	list of grouping elements
`FUN`	function

aggregate

```
iris.x <- subset(iris, select= -Species)
iris.s <- subset(iris, select= Species)
aggregate(iris.x, iris.s, mean)
```

```
        Species Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa        5.006       3.428        1.462       0.246
2 versicolor        5.936       2.770        4.260       1.326
3  virginica        6.588       2.974        5.552       2.026
```

Data sorting

- sort
- arrange
- order

Data sorting

```
ssl <- sort(iris$Sepal.Length)
head(ssl)
      [1] 4.3 4.4 4.4 4.4 4.5 4.6
```

```
sIris <- sort(iris$Sepal.Length, index.return=TRUE)
head(iris[sIris$ix,])
         Sepal.Length Sepal.Width Petal.Length Petal.Width Species
14            4.3         3.0          1.1         0.1   setosa
9             4.4         2.9          1.4         0.2   setosa
39            4.4         3.0          1.3         0.2   setosa
43            4.4         3.2          1.3         0.2   setosa
42            4.5         2.3          1.3         0.3   setosa
4             4.6         3.1          1.5         0.2   setosa
```

arrange

```
library(dplyr)
```

```
sIris <- arrange(iris, Sepal.Length)
head(sIris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          4.3         3.0          1.1         0.1  setosa
2          4.4         2.9          1.4         0.2  setosa
3          4.4         3.0          1.3         0.2  setosa
4          4.4         3.2          1.3         0.2  setosa
5          4.5         2.3          1.3         0.3  setosa
6          4.6         3.1          1.5         0.2  setosa
```

Let’s move on to coding...

Silesian
University
of Technology

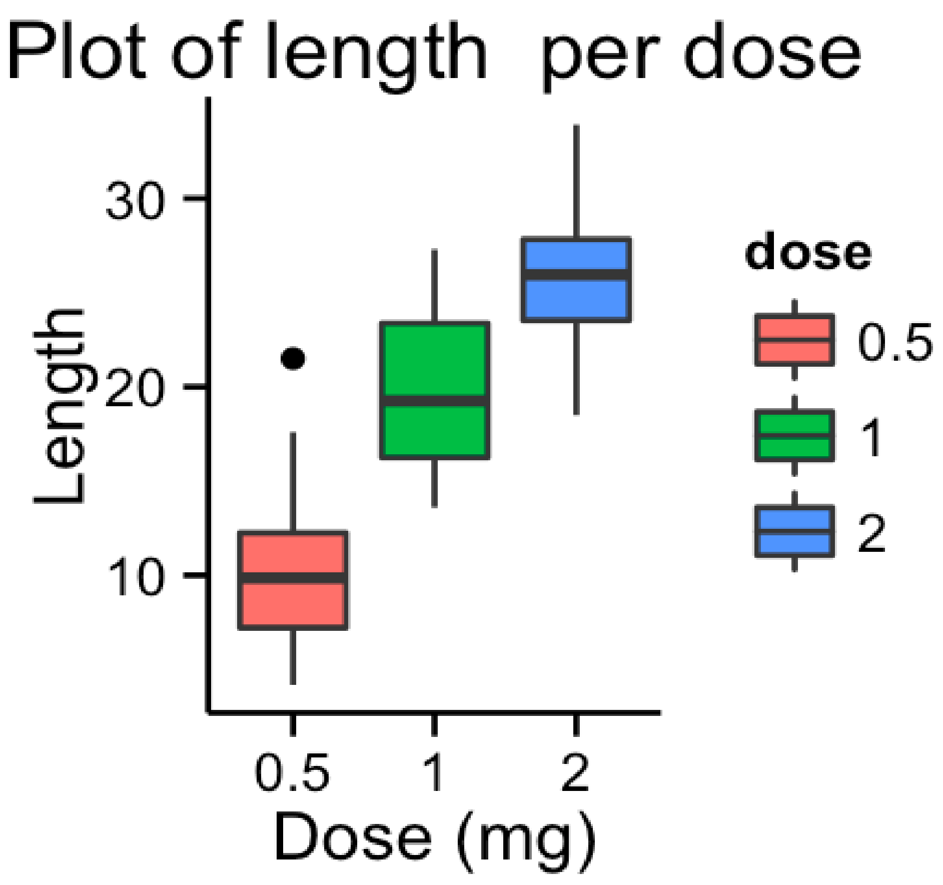
RESEARCH
UNIVERSITY
EXCELLENCE INITIATIVE
Ministry of Science
and Higher Education

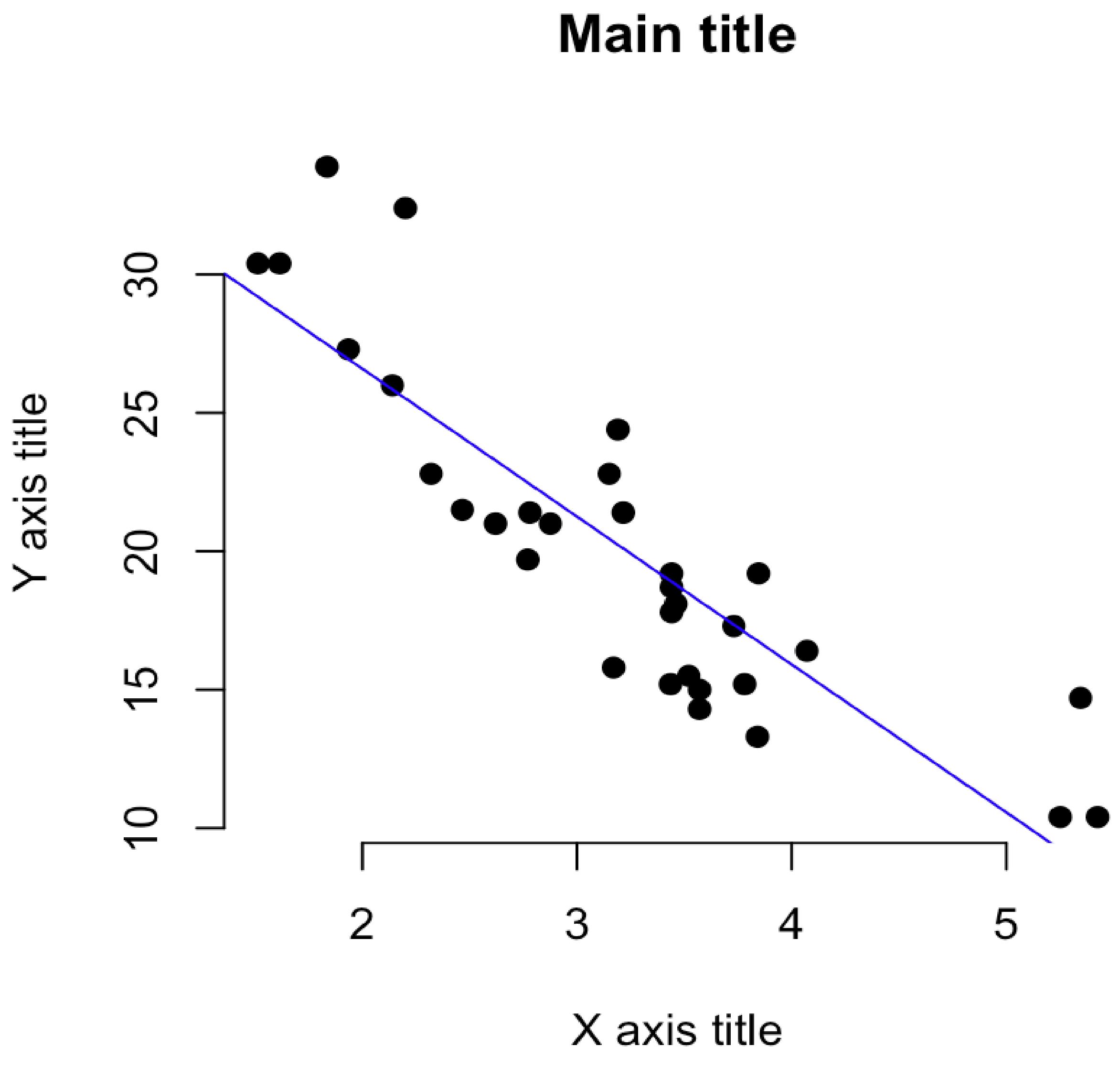
Part 2

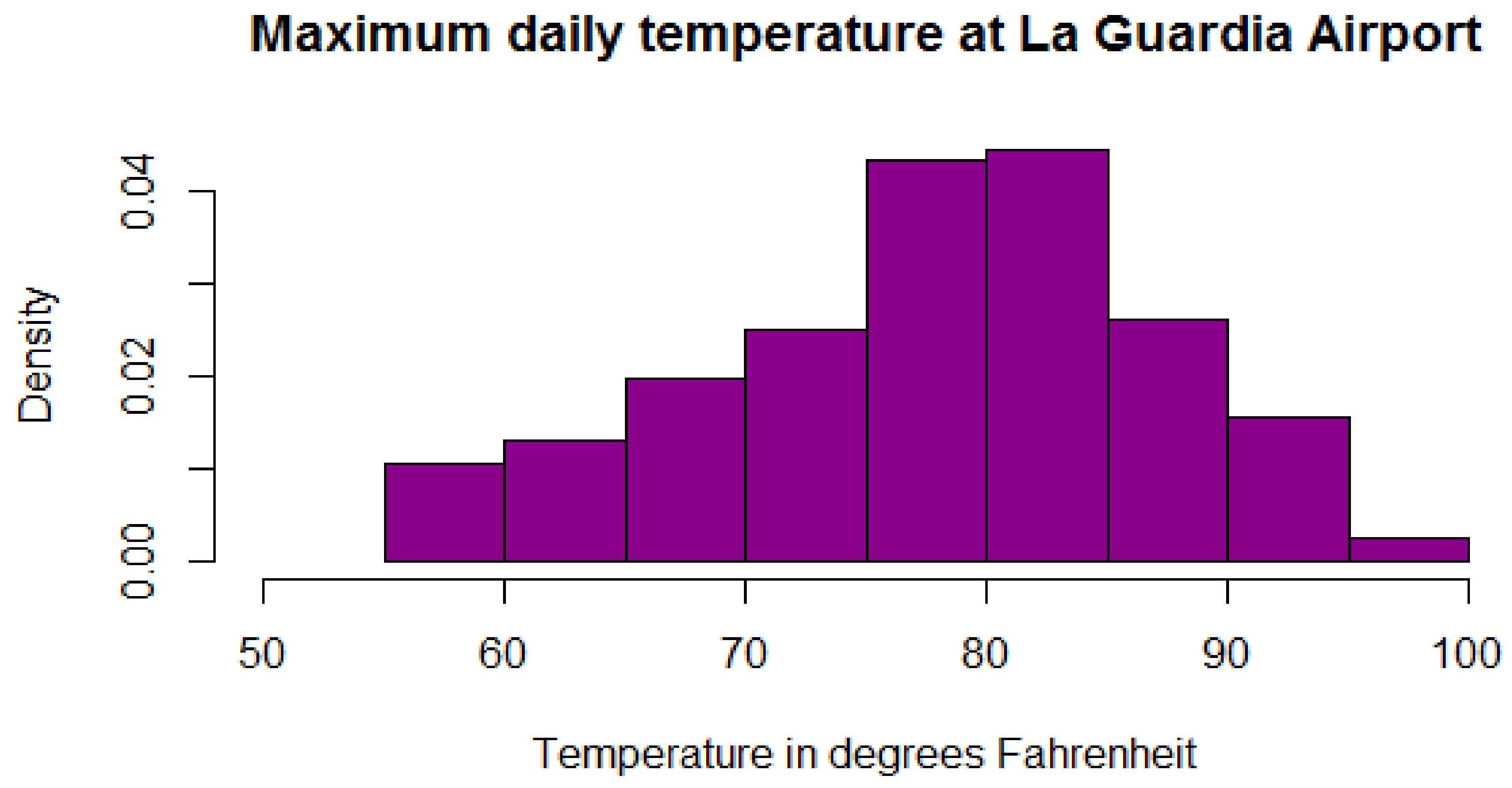
Data visualization in R

Main types of graphs in data analysis

- Scatterplots
- Barplots
- Histograms
- Density plots
- Boxplots

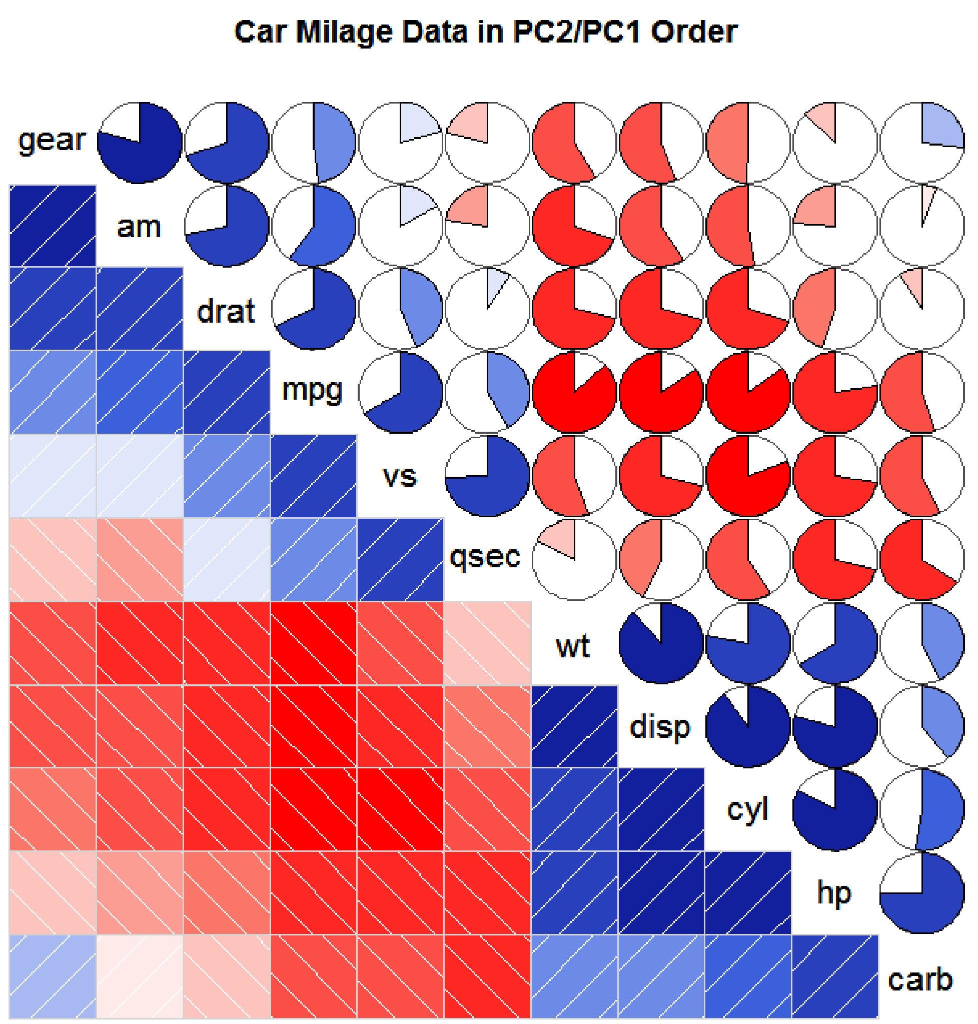


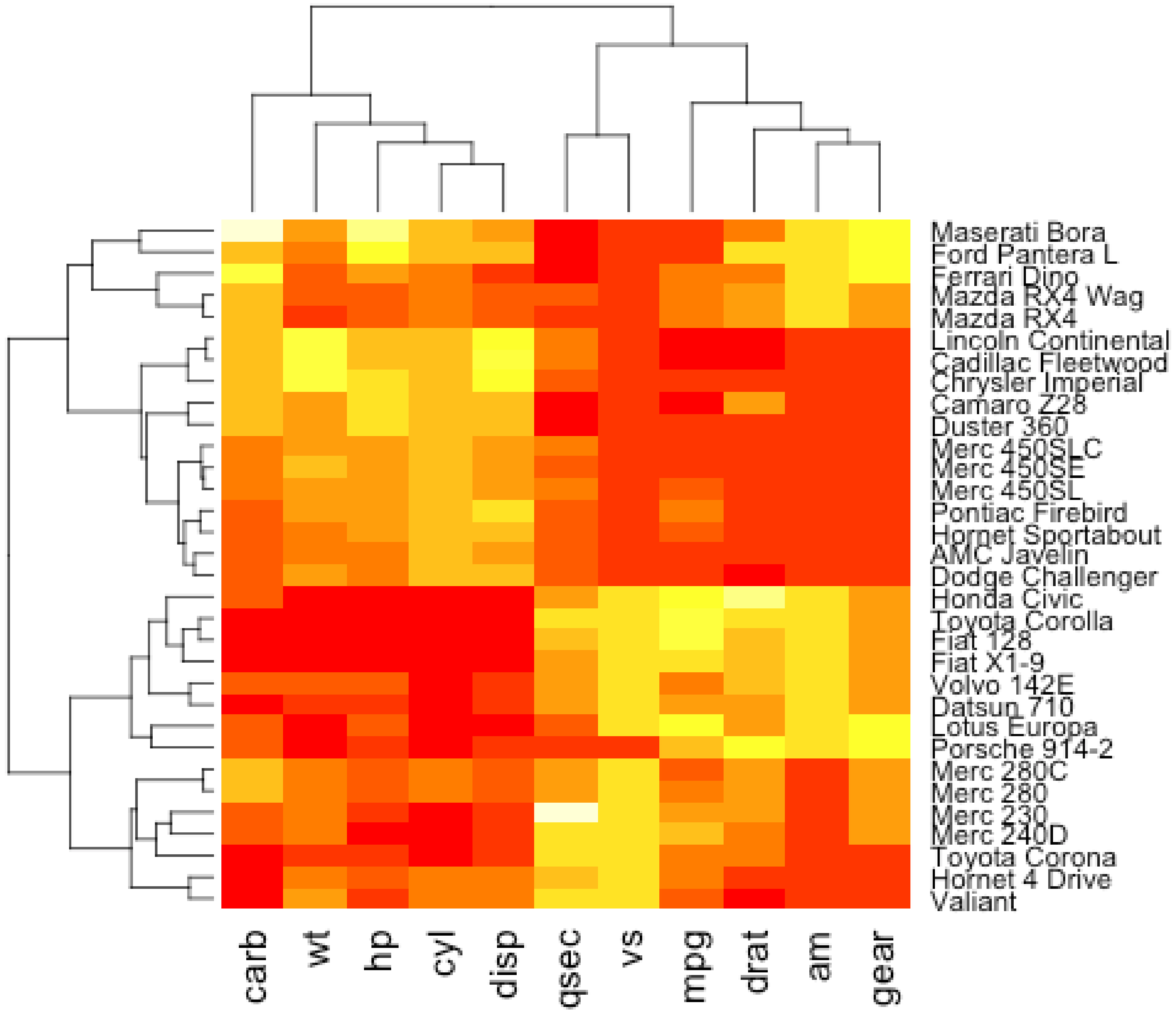


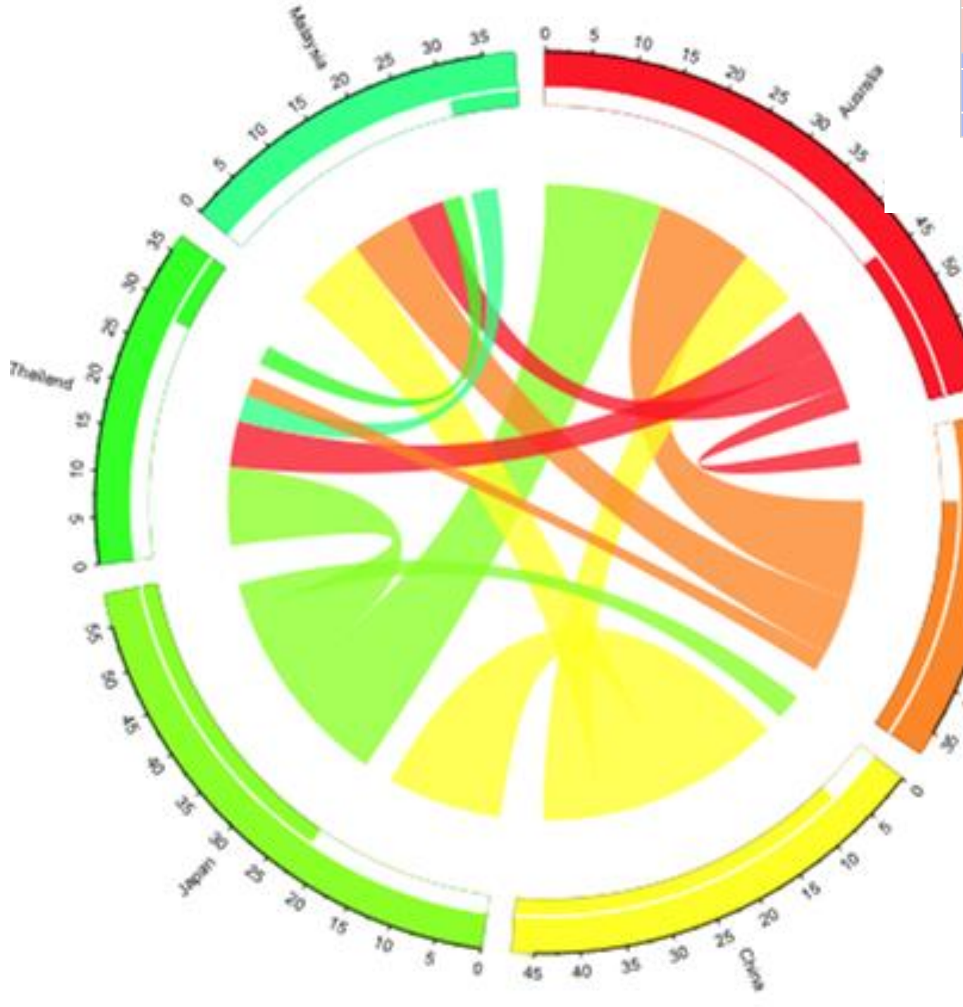


Fancy types of graphs

- Heatmaps
- Correlograms
- Dendrograms
- Circos plots
- Spatial plots
- …

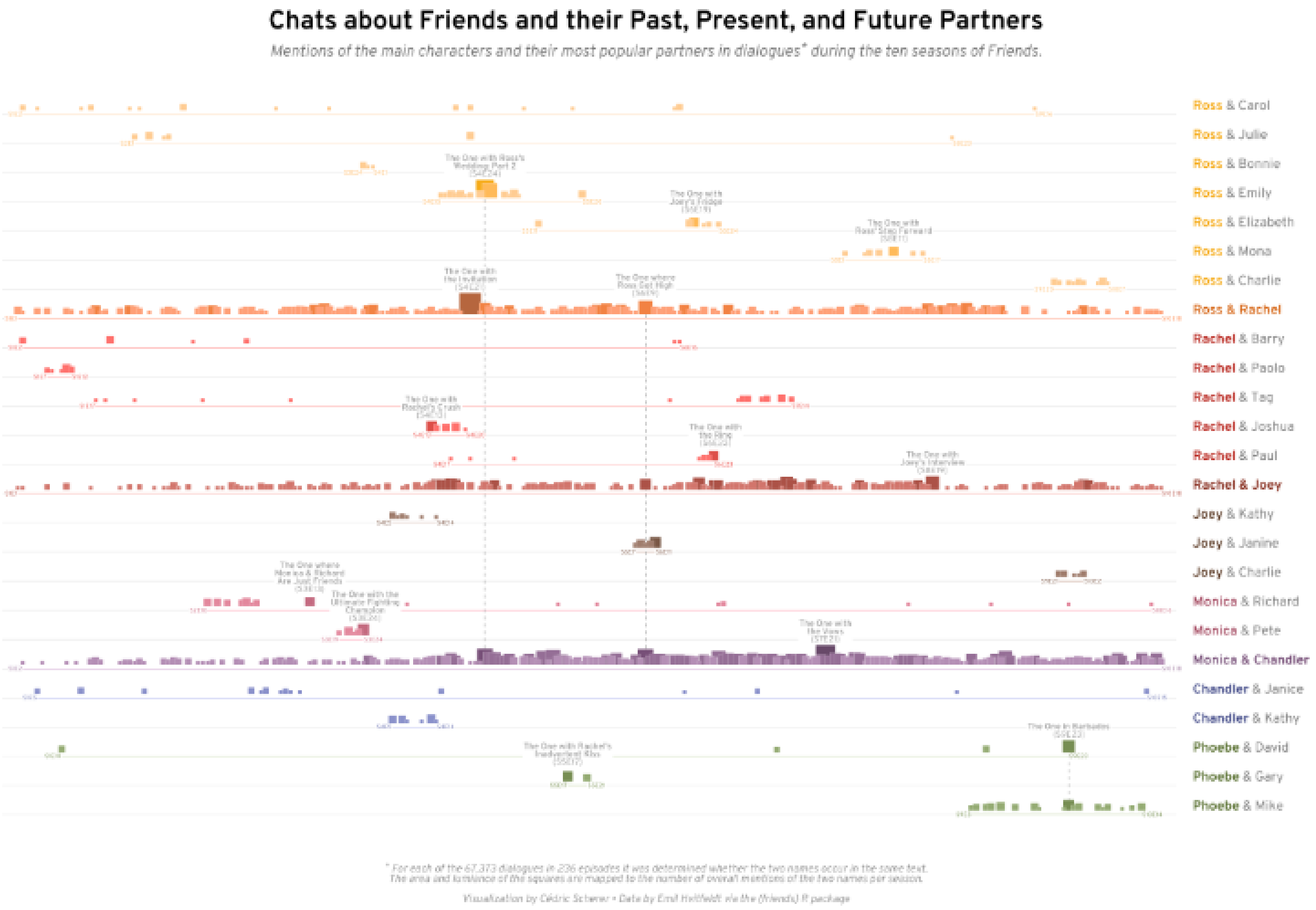






Extra fancy types of graphs

www.cedricscherer.com



Chats about Friends and their Past, Present, and Future Partners

Mentions of the main characters and their most popular partners in dialogues during the ten seasons of Friends.*

* For each of the 67,373 dialogues in 236 episodes it was determined whether the two names occur in the same text. The area and luminance of the squares are mapped to the number of overall mentions of the two names per season.

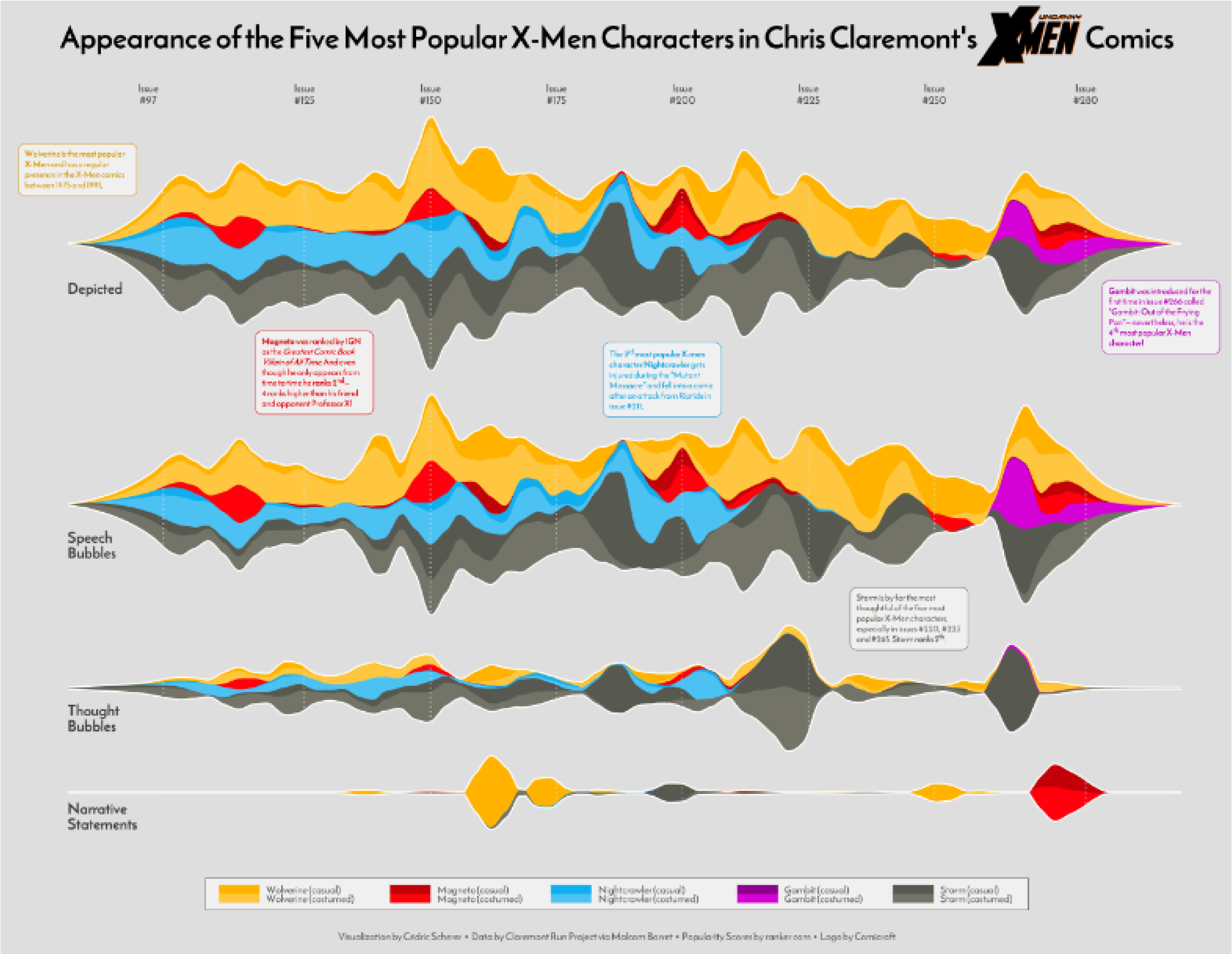
Visualization by Cédric Scherer • Data by Emil Hvitfeldt via the {friends} R package

Silesian University of Technology

RESEARCH UNIVERSITY
EXCELLENCE INITIATIVE

Extra fancy types of graphs

www.cedricscherer.com



Appearance of the Five Most Popular X-Men Characters in Chris Claremont's X-MEN Comics

Wolverine is the most popular X-Men and has a regular presence in the X-Men comics between 1975 and 1991.

Magneto was ranked by IGN as the *Greatest Comic Book Villain of All Time*. And even though he only appears from time to time he ranks 2nd – 4 ranks higher than his friend and opponent Professor X!

The 3rd most popular X-men character Nightcrawler gets injured during the "Mutant Massacre" and fell into a coma after an attack from Riptide in issue #211.

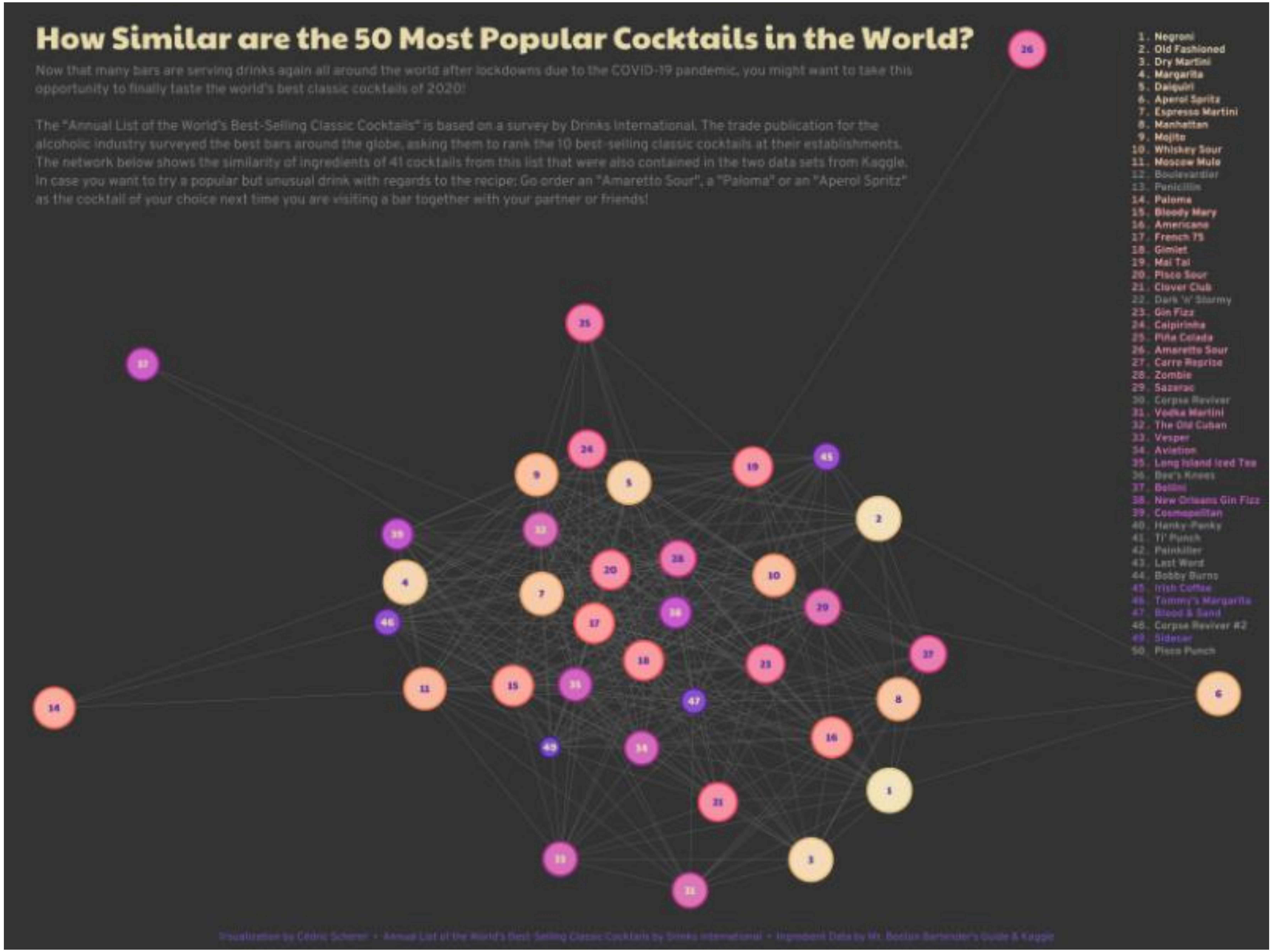
Gambit was introduced for the first time in issue #266 called "Gambit: Out of the Frying Pan" – nevertheless, he is the 4th most popular X-Men character!

Storm is by far the most thoughtful of the five most popular X-Men characters, especially in issues #220, #223 and #265. Storm ranks 5th.

Visualization by Cédric Scherer • Data by Claremont Run Project via Malcom Barret • Popularity Scores by ranker.com • Logo by Comicraft

Extra fancy types of graphs

www.cedricscherer.com



How Similar are the 50 Most Popular Cocktails in the World?

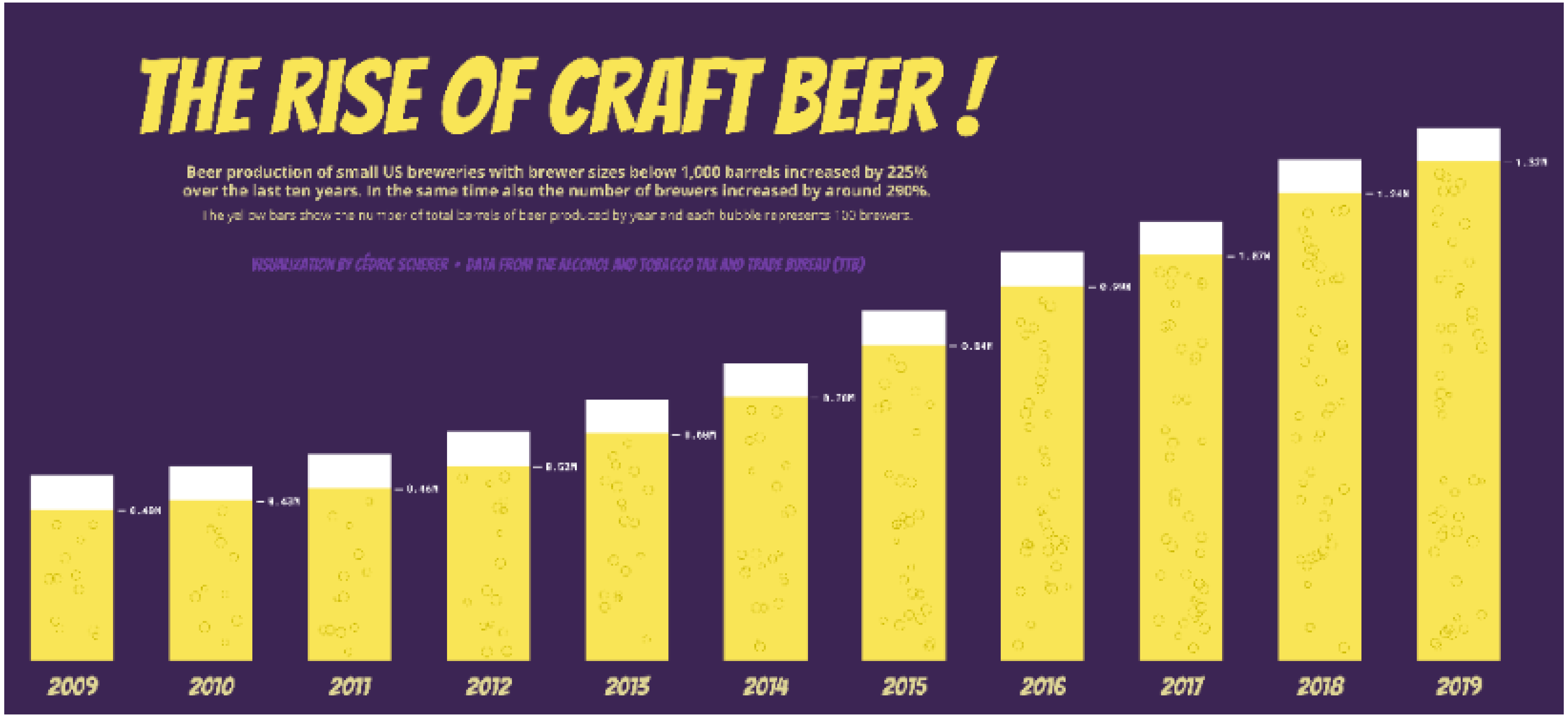
Now that many bars are serving drinks again all around the world after lockdowns due to the COVID-19 pandemic, you might want to take this opportunity to finally taste the world's best classic cocktails of 2020!

The "Annual List of the World's Best-Selling Classic Cocktails" is based on a survey by Drinks International. The trade publication for the alcoholic industry surveyed the best bars around the globe, asking them to rank the 10 best-selling classic cocktails at their establishments. The network below shows the similarity of ingredients of 41 cocktails from this list that were also contained in the two data sets from Kaggle. In case you want to try a popular but unusual drink with regards to the recipe: Go order an "Amaretto Sour", a "Paloma" or an "Aperol Spritz" as the cocktail of your choice next time you are visiting a bar together with your partner or friends!

1. Negroni
2. Old Fashioned
3. Dry Martini
4. Margarita
5. Daiquiri
6. Aperol Spritz
7. Espresso Martini
8. Manhattan
9. Mojito
10. Whiskey Sour
11. Moscow Mule
12. Boulevardier
13. Penicillin
14. Paloma
15. Bloody Mary
16. Americano
17. French 75
18. Gimlet
19. Mai Tai
20. Pisco Sour
21. Clover Club
22. Dark 'n' Stormy
23. Gin Fizz
24. Caipirinha
25. Piña Colada
26. Amaretto Sour
27. Carre Reprise
28. Zombie
29. Sazerac
30. Corpse Reviver
31. Vodka Martini
32. The Old Cuban
33. Vesper
34. Aviation
35. Long Island Iced Tea
36. Bee's Knees
37. Bellini
38. New Orleans Gin Fizz
39. Cosmopolitan
40. Hanky-Panky
41. Ti' Punch
42. Painkiller
43. Last Word
44. Bobby Burns
45. Irish Coffee
46. Tommy's Margarita
47. Blood & Sand
48. Corpse Reviver #2
49. Sidecar
50. Pisco Punch

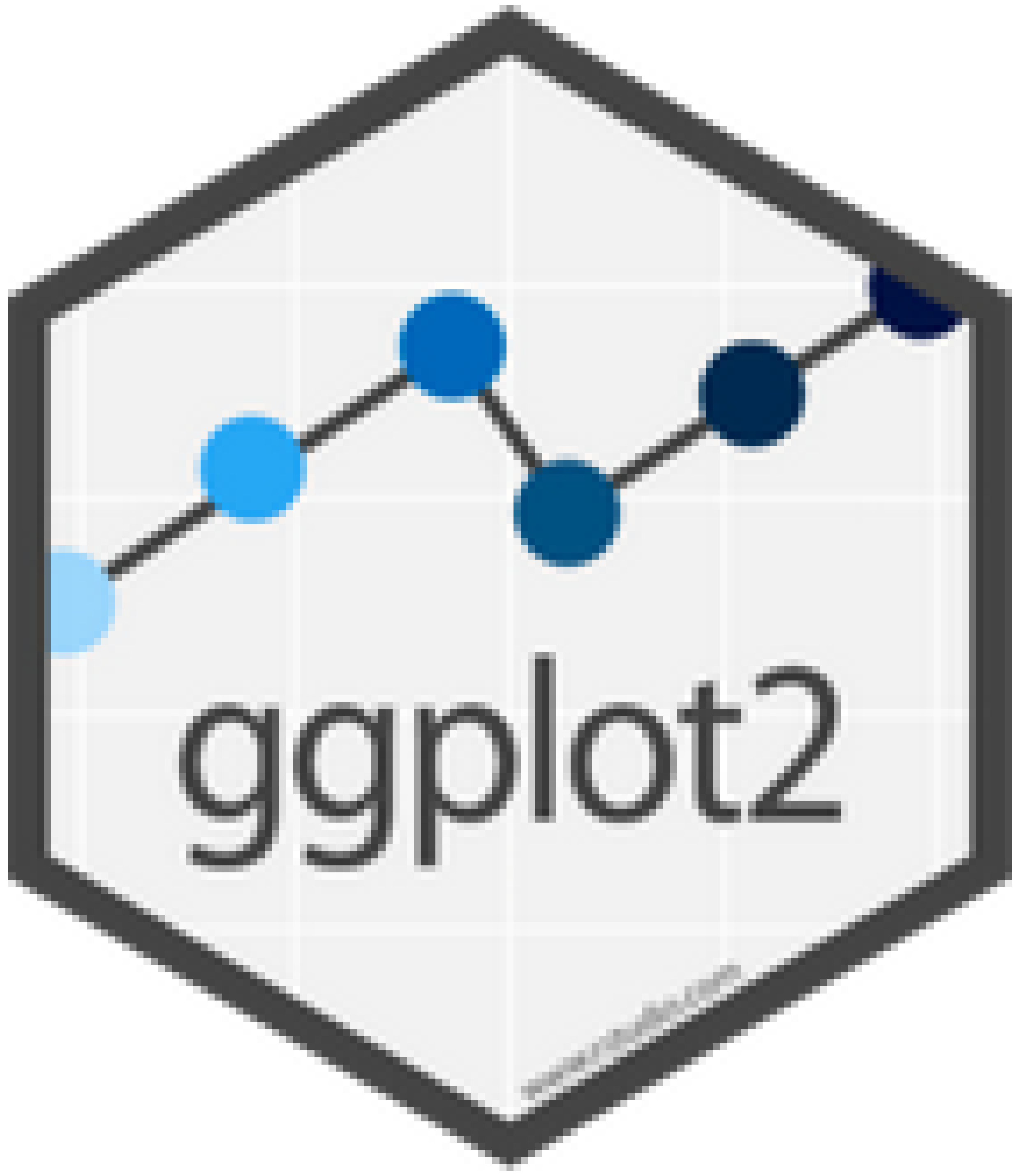
Visualization by Cédric Scherer • Annual List of the World's Best-Selling Classic Cocktails by Drinks International • Ingredient Data by Mr. Boston Bartender's Guide & Kaggle

Extra fancy types of graphs



www.cedricscherer.com

How to make’em?



Let’s move on to coding...

Silesian
University
of Technology

RESEARCH
UNIVERSITY
EXCELLENCE INITIATIVE
Ministry of Science
and Higher Education

I APPRECIATE YOUR ATTENTION