ARISTOTLE UNIVERSITY OF THESSALONIKI

FACULTY OF HEALTH SCIENCES - SCHOOL OF MEDICINE

MSc Health Statistics and Data Analytics

R data visualization

KONSTANTINOS I. BOUGIOUKAS

PHYSICIST, BIOSTATISTICIAN AND RESEARCH METHODOLOGIST

THESSALONIKI 2021-22

MSc Health Statistics & Data Analytics

What is the propose of visualization?

1. Analysis

-Using visualizations to gain insight (understanding)

2. Communication

- Passing a message to others (connection)

3. Monitoring

- Tracking information about performance (stewardship)

4. Planning

- Predicting and preparing for the future (foresight)

Visual Processing Stages

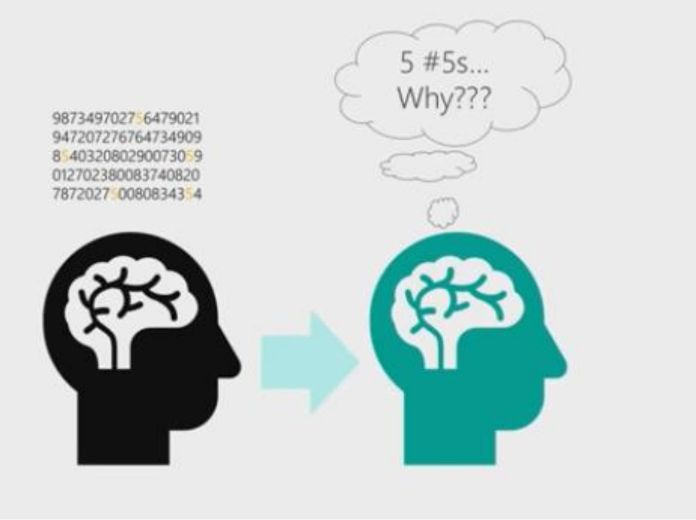
98723446325889899

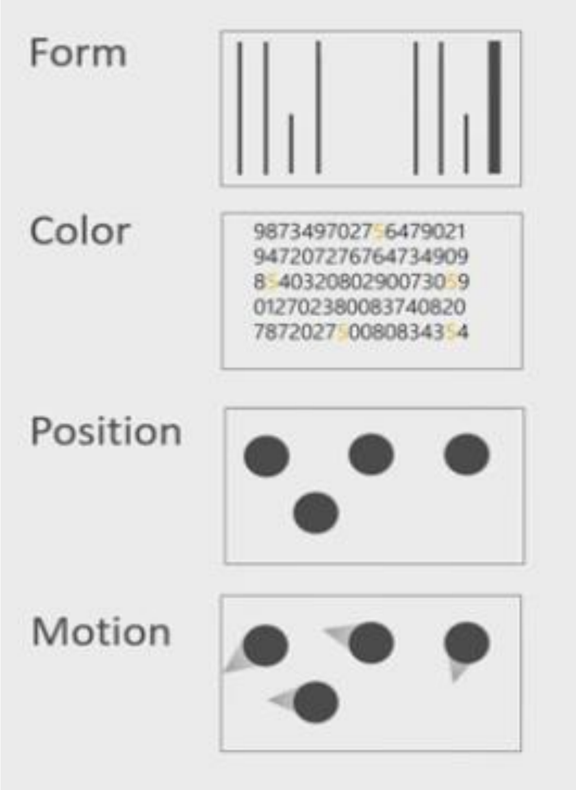
87867448679790 32

43435878675449 02

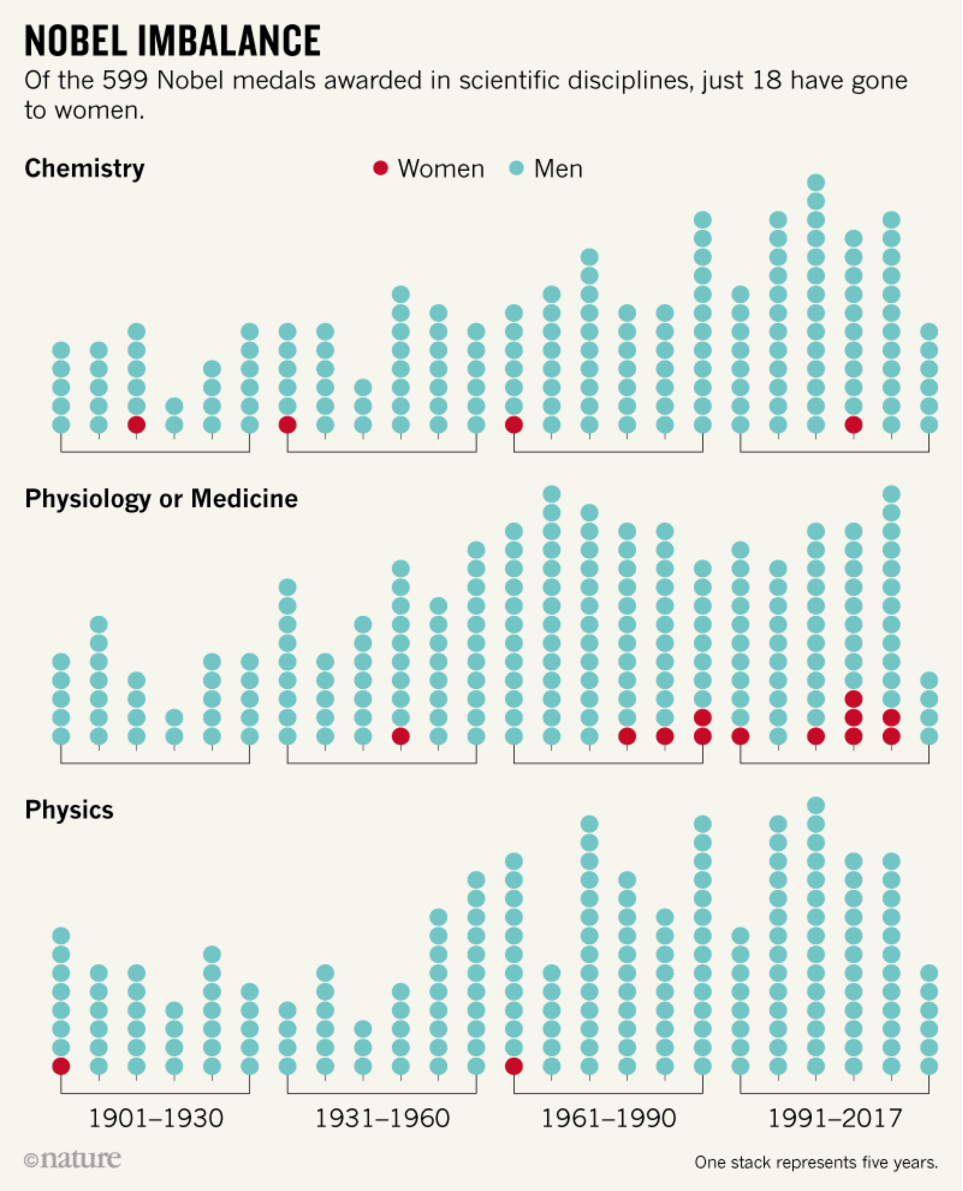
47894926797873 32

34358769322463 59

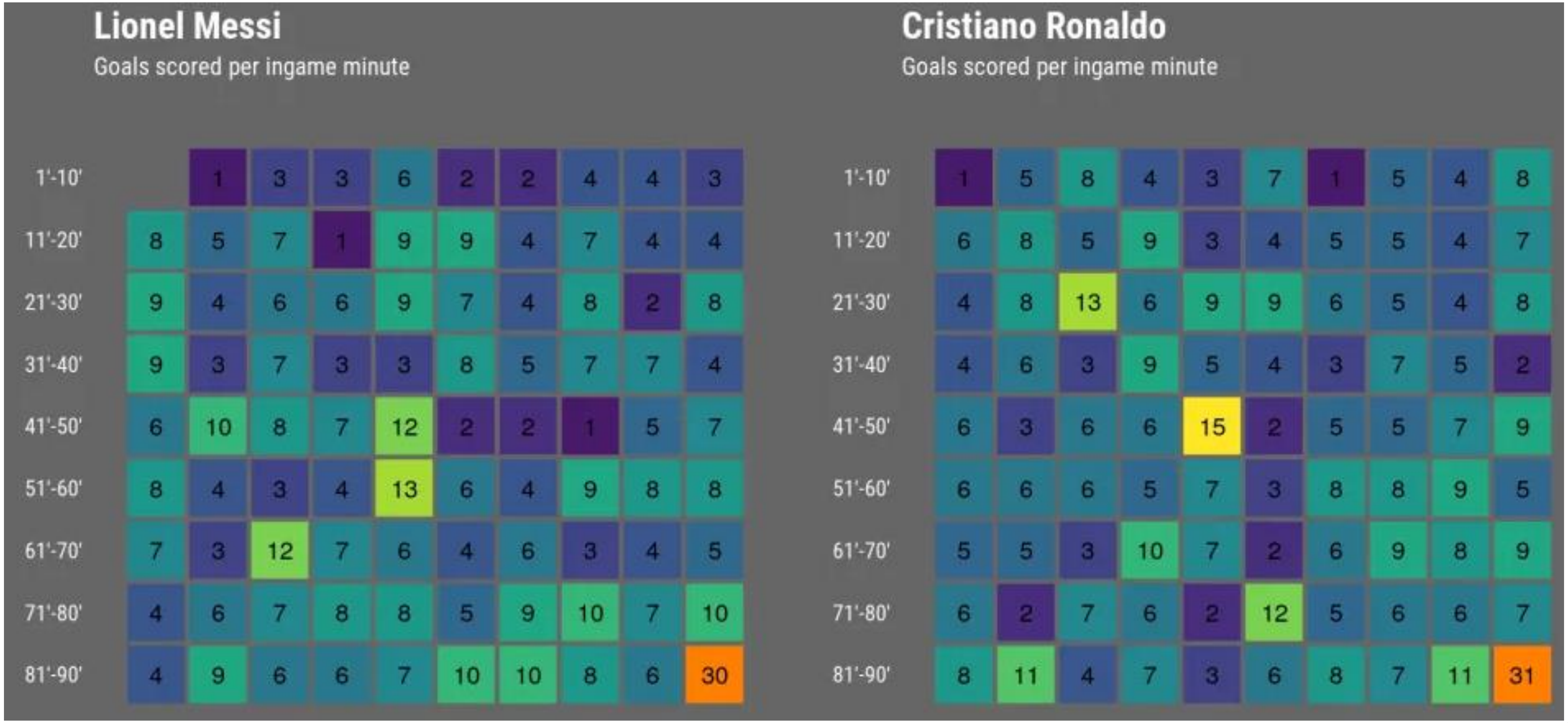




Nobel prizes



Goals in first 5’



Lionel Messi

Goals scored per ingame minute

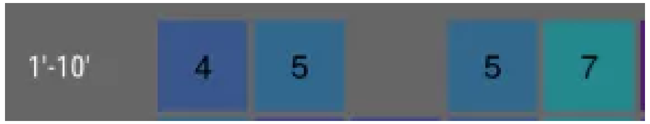
1'-10'		1	3	3	6	2	2	4	4	3
11'-20'	8	5	7	1	9	9	4	7	4	4
21'-30'	9	4	6	6	9	7	4	8	2	8
31'-40'	9	3	7	3	3	8	5	7	7	4
41'-50'	6	10	8	7	12	2	2	1	5	7
51'-60'	8	4	3	4	13	6	4	9	8	8
61'-70'	7	3	12	7	6	4	6	3	4	5
71'-80'	4	6	7	8	8	5	9	10	7	10
81'-90'	4	9	6	6	7	10	10	8	6	30

Cristiano Ronaldo

Goals scored per ingame minute

1'-10'	1	5	8	4	3	7	1	5	4	8
11'-20'	6	8	5	9	3	4	5	5	4	7
21'-30'	4	8	13	6	9	9	6	5	4	8
31'-40'	4	6	3	9	5	4	3	7	5	2
41'-50'	6	3	6	6	15	2	5	5	7	9
51'-60'	6	6	6	5	7	3	8	8	9	5
61'-70'	5	5	3	10	7	2	6	9	8	9
71'-80'	6	2	7	6	2	12	5	6	6	7
81'-90'	8	11	4	7	3	6	8	7	11	31

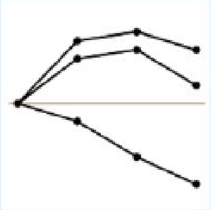
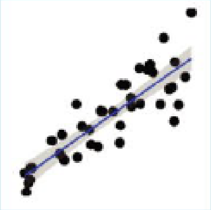
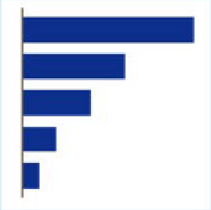
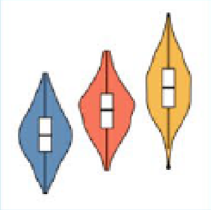
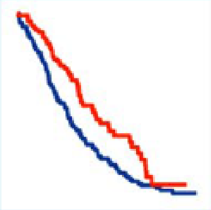
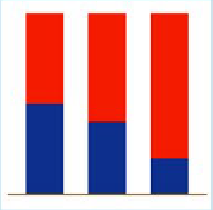
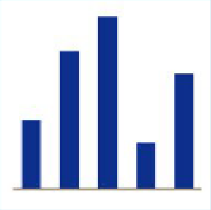
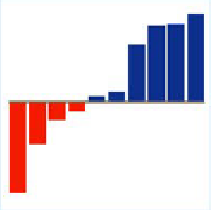
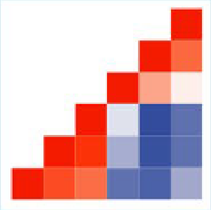
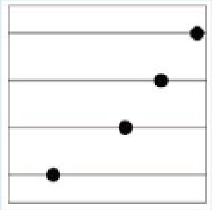
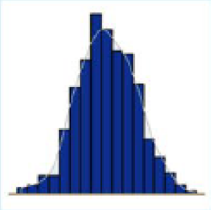
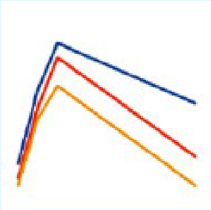
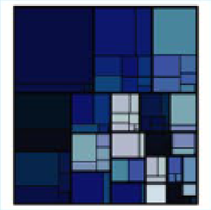
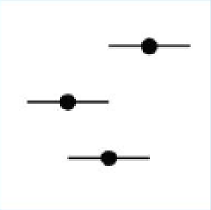
Vs Zlatan Ibrahimovic



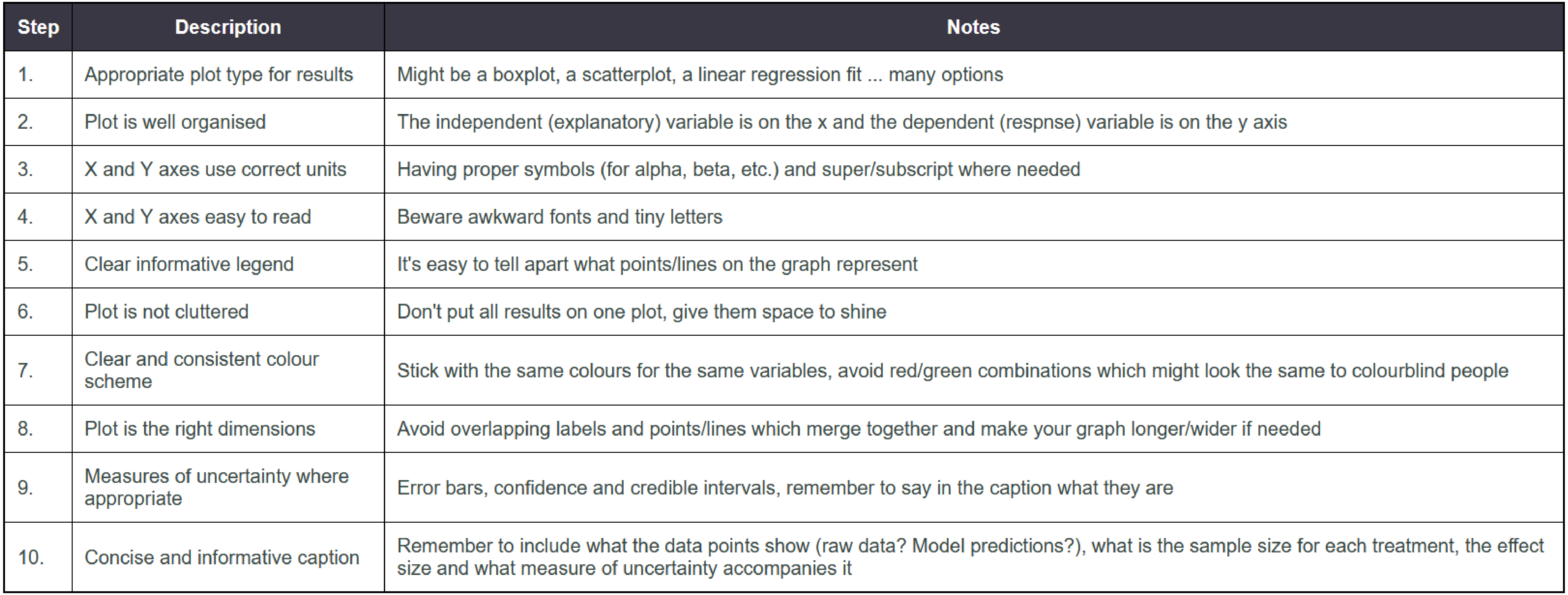
1'-10'	4	5		5	7

https://www.r-bloggers.com/analyzing-the-greatest-strikers-in-football-ii-visualizing-data/

Chart types

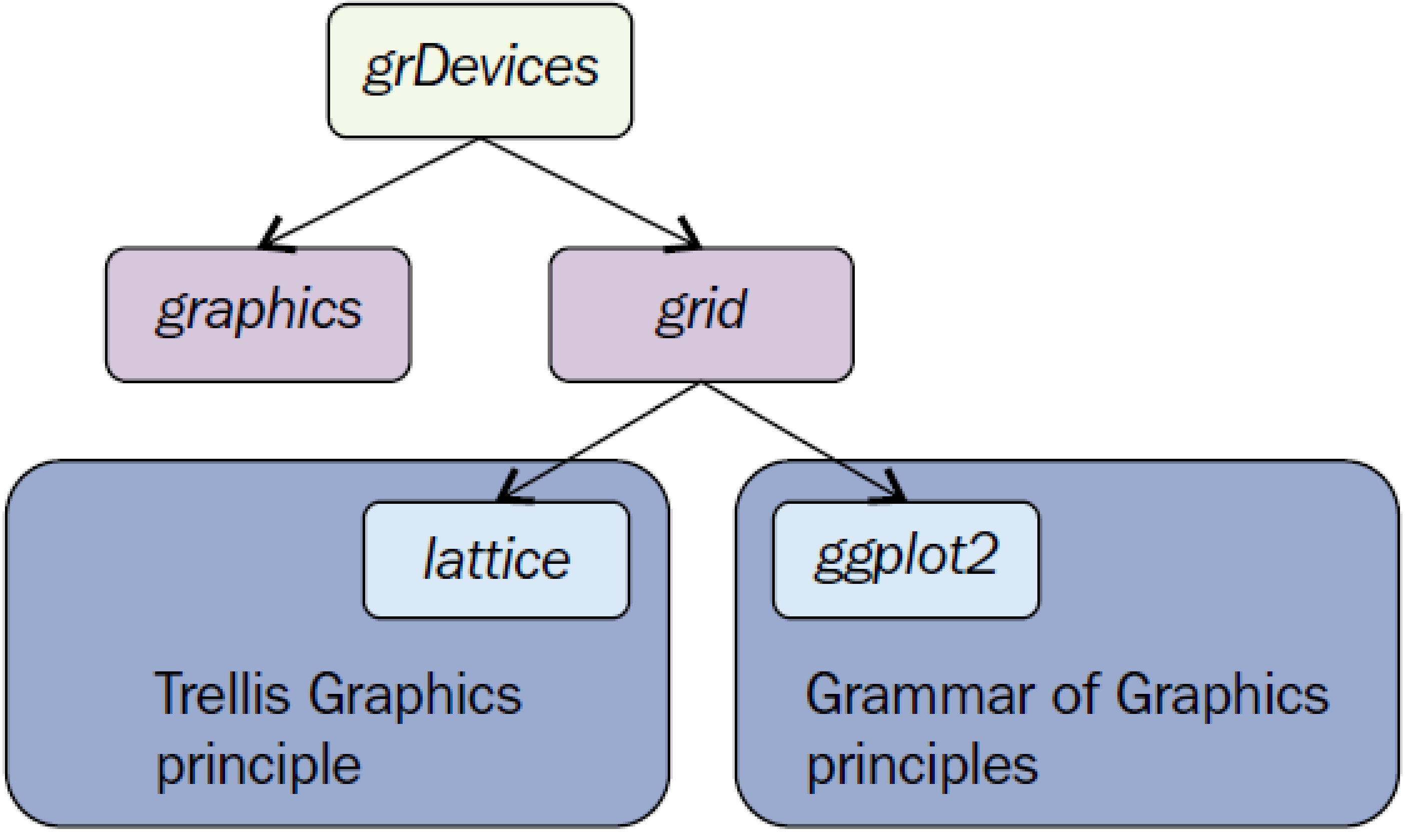
Deviation	Correlation	Ranking	Distribution	Evolution	Part-to-whole	Magnitude
Chg. from baseline	Scatter plot	Horizontal bar chart	Boxplot	Kaplan Meier	Stacked bar chart	Vertical bar chart
						
Waterfall	Heat map	Dotplot	Histogram	Line plot	Tree map	Forest plot
						

Good data visualisation



Step	Description	Notes
1.	Appropriate plot type for results	Might be a boxplot, a scatterplot, a linear regression fit ... many options
2.	Plot is well organised	The independent (explanatory) variable is on the x and the dependent (respnse) variable is on the y axis
3.	X and Y axes use correct units	Having proper symbols (for alpha, beta, etc.) and super/subscript where needed
4.	X and Y axes easy to read	Beware awkward fonts and tiny letters
5.	Clear informative legend	It's easy to tell apart what points/lines on the graph represent
6.	Plot is not cluttered	Don't put all results on one plot, give them space to shine
7.	Clear and consistent colour scheme	Stick with the same colours for the same variables, avoid red/green combinations which might look the same to colourblind people
8.	Plot is the right dimensions	Avoid overlapping labels and points/lines which merge together and make your graph longer/wider if needed
9.	Measures of uncertainty where appropriate	Error bars, confidence and credible intervals, remember to say in the caption what they are
10.	Concise and informative caption	Remember to include what the data points show (raw data? Model predictions?), what is the sample size for each treatment, the effect size and what measure of uncertainty accompanies it

The plotting environments in R



What is ggplot2 ?

- Based on the Grammar of Graphics, a coherent system for describing and building graphs.

- ggplot2 enables you to compose simple to complex graphs by combining independent components.

Basic elements to build a ggplot graph

Data

dataset

Aesthetics

color

position

size

Geometry

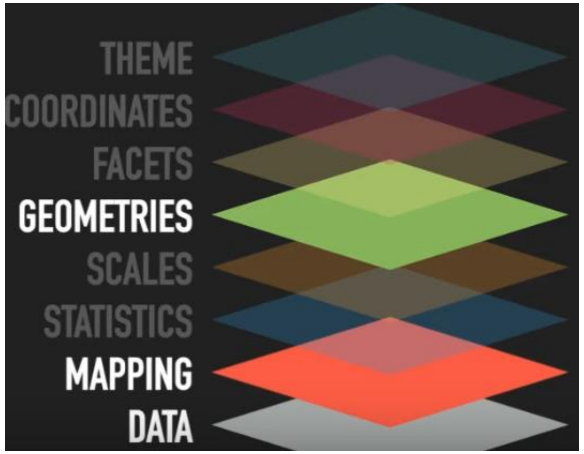
points

lines

bars

```
ggplot(dataset, aes(x = var1, y = var2)) +
   geom_point()
```

Plot

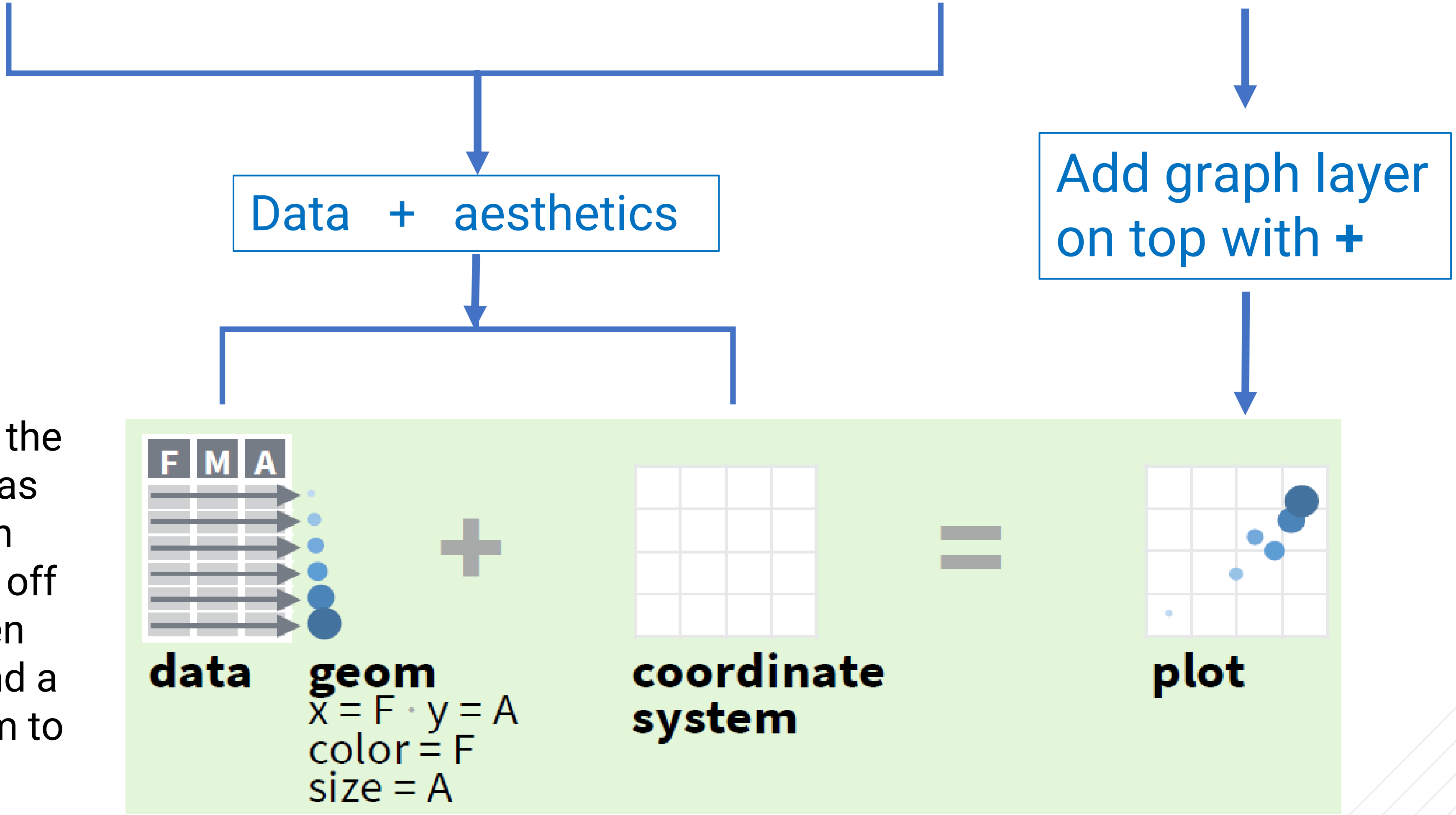


`ggplot()`

- Initializes a ggplot object
- Adds layers to create graph
- No defaults, more control

ggplot() function

```
ggplot(dataset, aes(x = var1, y = var2)) + geom_point()
```



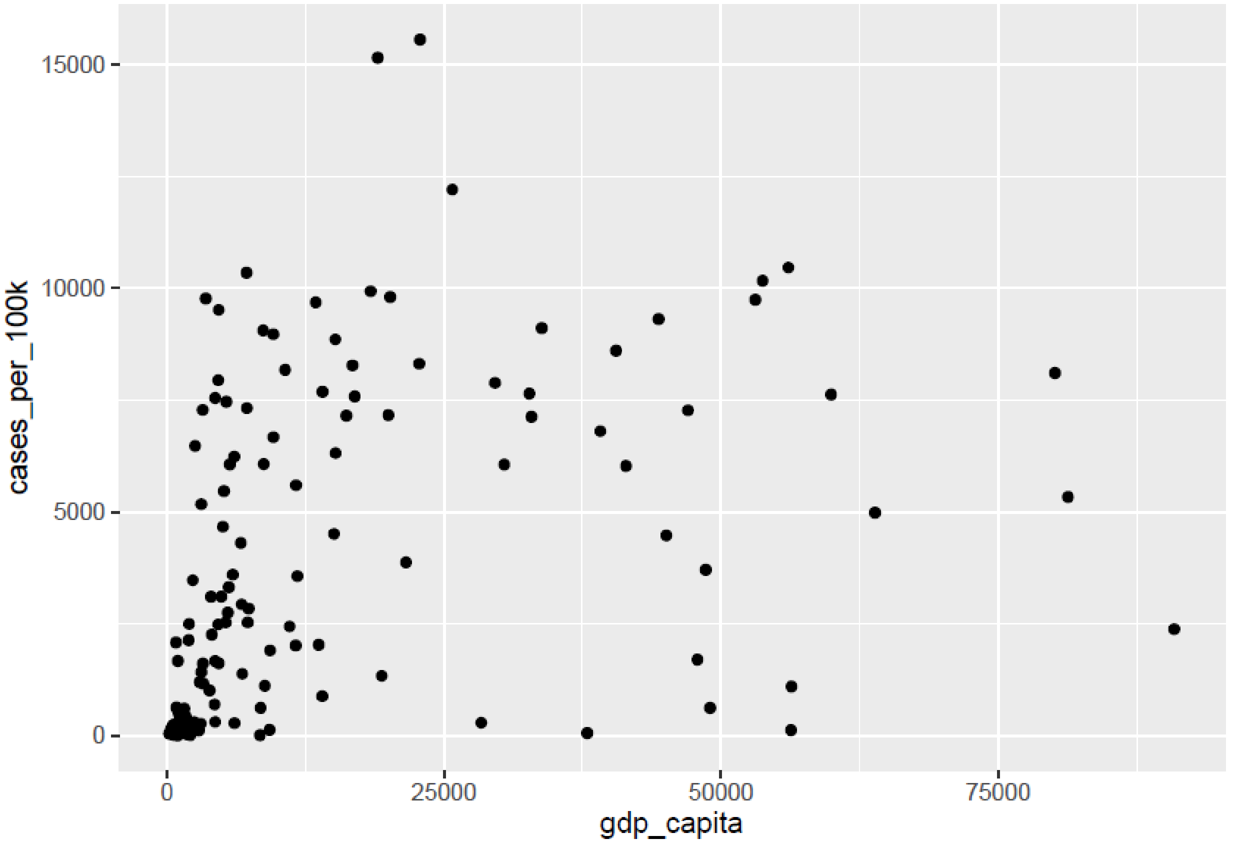
Data + aesthetics

Add graph layer on top with **+**

You can visualize the ggplot() function as drawing on a plain canvas. You start off with data, and then you add points and a coordinate system to get a final plot.

Example: scatter plot

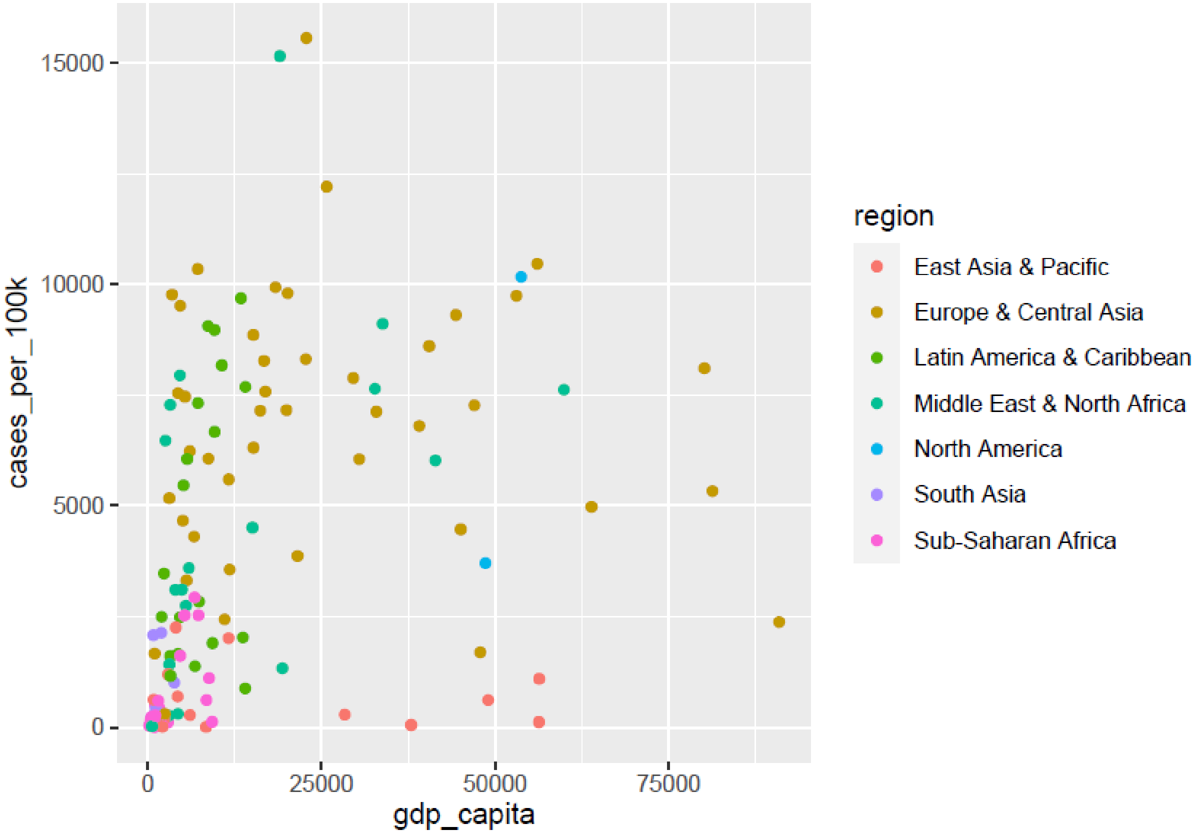
```
ggplot(data = dat, mapping = aes(x = gdp_capita, y = cases_per_100k)) +
geom_point()
```



In this case, geom_point() **inherits** the *x* and y aesthetics from the ggplot() function

Example: scatter plot- Color aesthetic

```
ggplot(dat, aes(x = gdp_capita, y = cases_per_100k)) +
geom_point(aes(color = region))
```



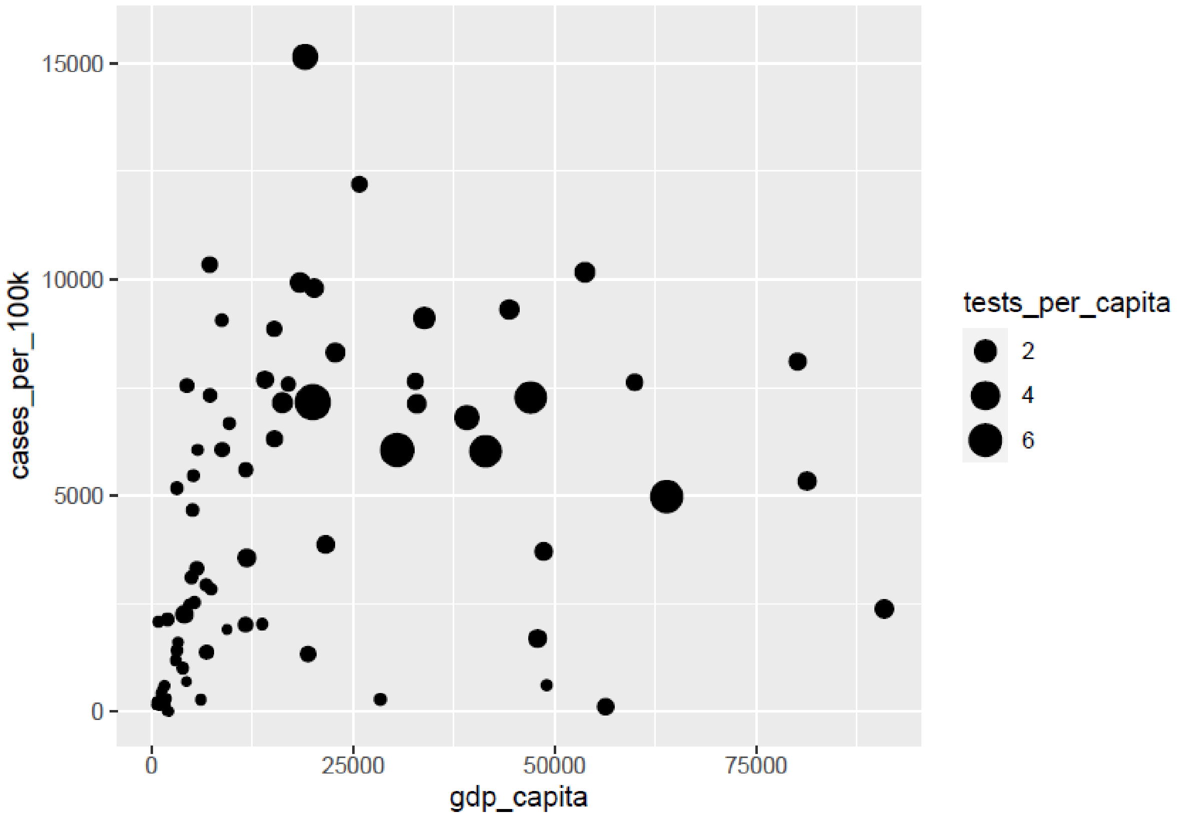
Here, we added **inside** the aes() of geom_point the color aesthetic.

The data of variable ***region*** mapped to color aesthetic of geom_point.

Obviously, the qualitative scale was applied automatically by ggplot2

Example: scatter plot- Size aesthetic

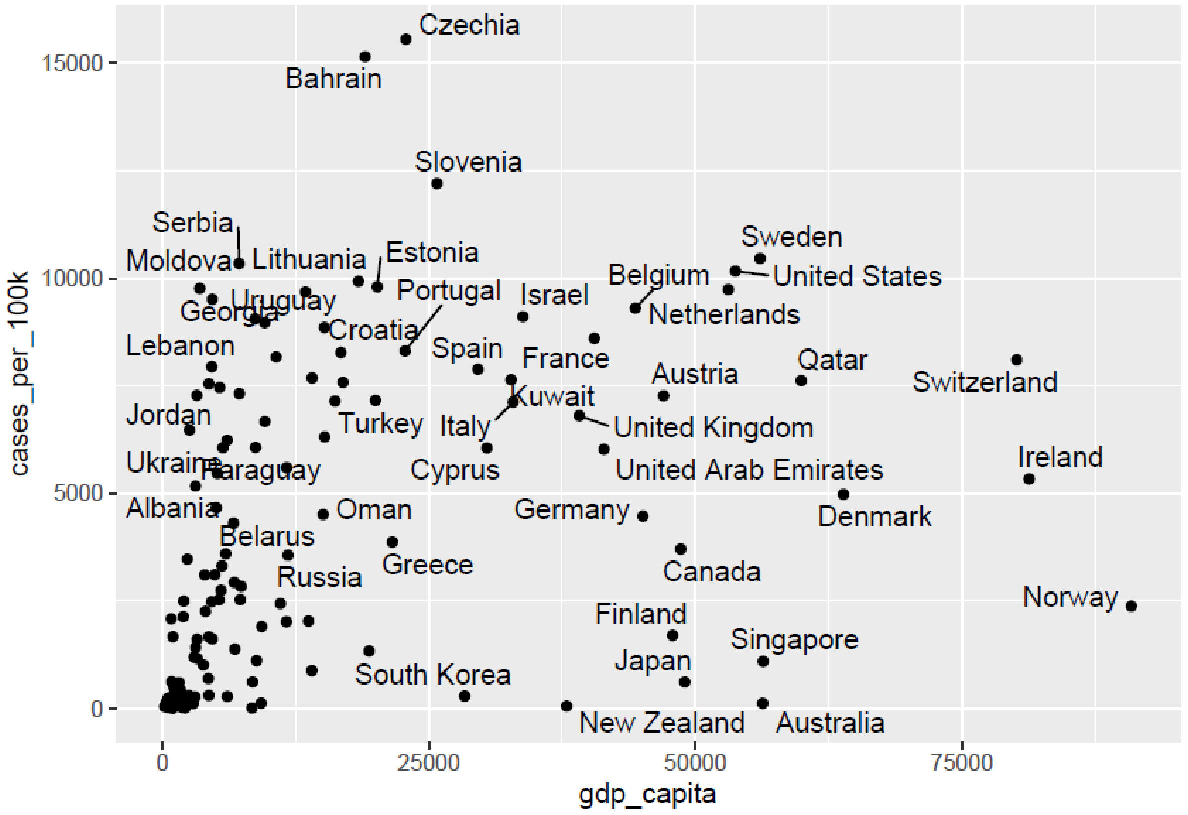
```
ggplot(dat, aes(x = gdp_capita, y = cases_per_100k)) +
geom_point(aes(size = tests_per_capita))
```



We can add a third variable ***tests_per_capita*** using the **size** aesthetic

Example: scatter plot- Add a new geom

```
ggplot(dat, aes(x = gdp_capita, y = cases_per_100k)) +
geom_point() +
geom_text_repel(aes(label = country))
```

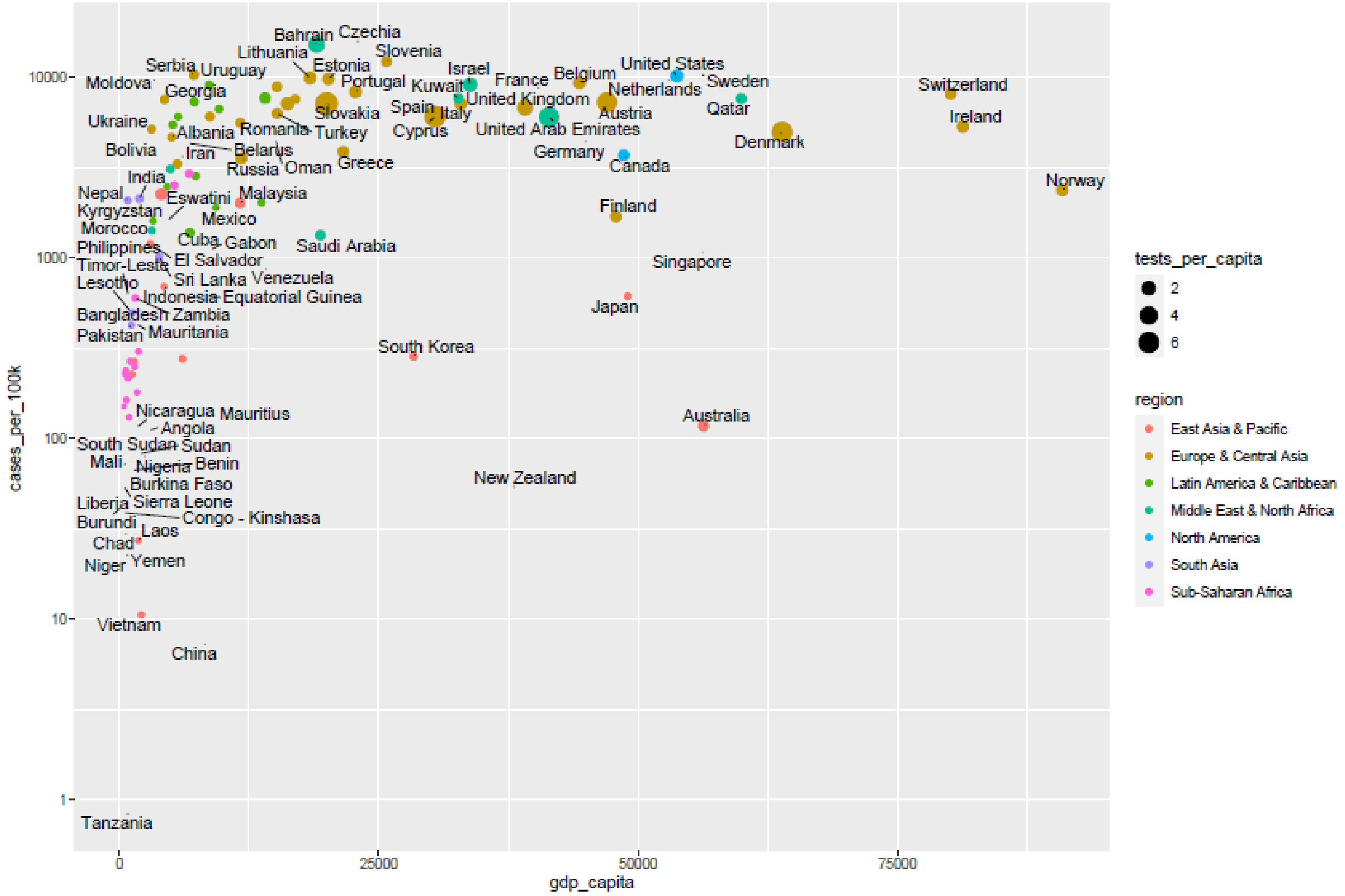


We can add the name of the ***country*** for each data point with a new geometry:

geom_text_repel()

Example: scatter plot- Change the scale of the y-axis

```
ggplot(dat, aes(x = gdp_capita, y = cases_per_100k)) +
  geom_point(aes(size = tests_per_capita, color = region)) +
  geom_text_repel(aes(label = country),
                            min.segment.length = 0, seed = 42,
                             box.padding = 0.1) +
scale_y_continuous(trans = "log10")
```



We transform the scale to log10 with scale_y_continuous()

Note: Continuous variable ***tests_per_capital*** mapped to **size** and categorical variable ***region*** mapped to **color**.

Theme elements

ggplot2 theme elements reference

Set `minimal` as the baseline theme:

```
theme_minimal() +
theme(theme.element = element_type())
```

Use `element_blank()` to **remove an element**

Axis titles, **text**, **ticks**, and **lines** can be specified per axis using theme inheritance by putting `.x`/`.y` at the end of the theme element.

```
axis.line.y = element_line()

axis.title.y = element_text()

panel.grid.major = element_line()

panel.grid.minor = element_line()

axis.text.y

axis.text = element_text()
```

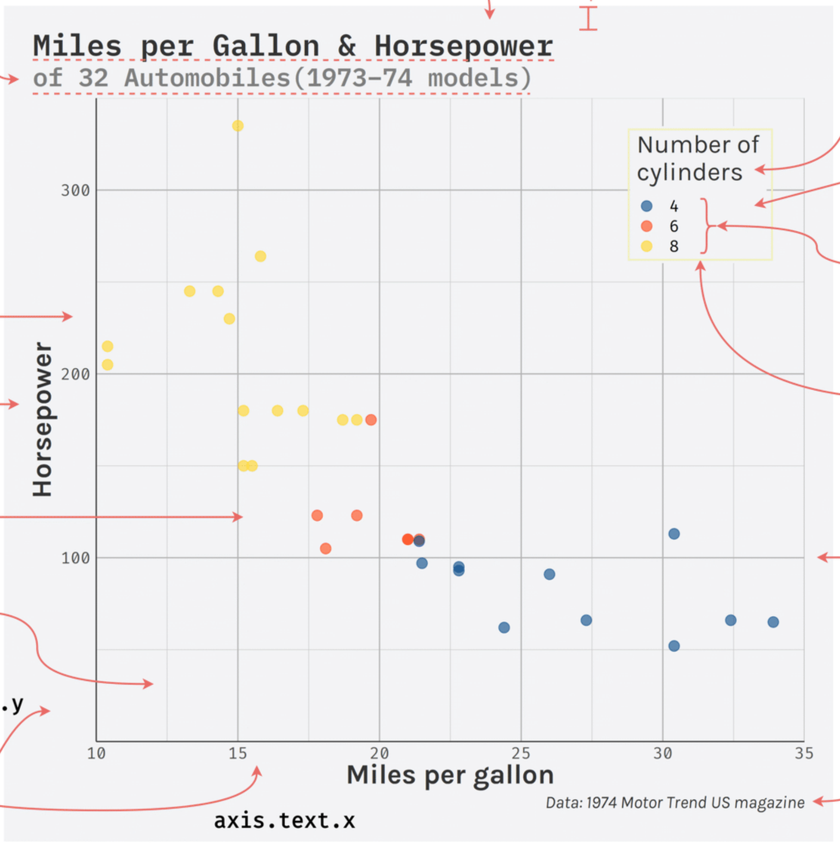
```
plot.title.position = "plot"
plot.caption.position = "plot"
```

"plot" means that they will be aligned to the entire plot (instead of the panel)

```
plot.title = element_text()

plot.subtitle = element_text()

plot.margin = margin(25, 25, 25, 25)
```



```
axis.text.x

legend.title
= element_text()

legend.background
= element_rect()

legend.text
= element_text()

legend.position
= c(.85,.85) / "none"/
 "left"/ "right"/
 "bottom"/ "top"

plot.background
= element_rect()

plot.caption
= element_text()
```

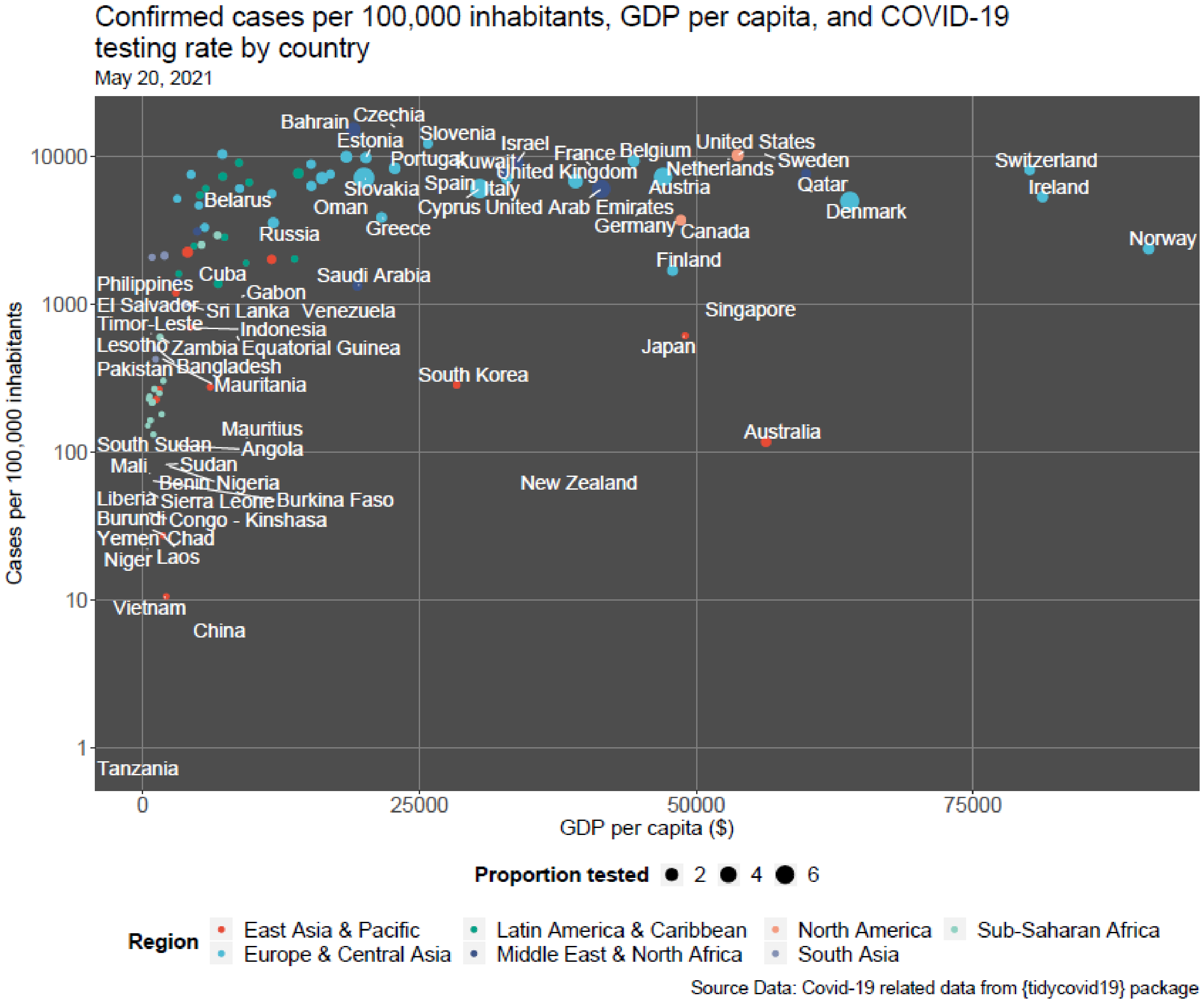
isabella-b

`text = element_text()` ← modifications will be applied to all text elements

Full list of elements at ggplot2.tidyverse.org/reference/theme

Themes can be used to give plots a consistent customized look to the **non-data elements** of the plot. Modify a single plot's theme using **theme()**

Modify theme elements

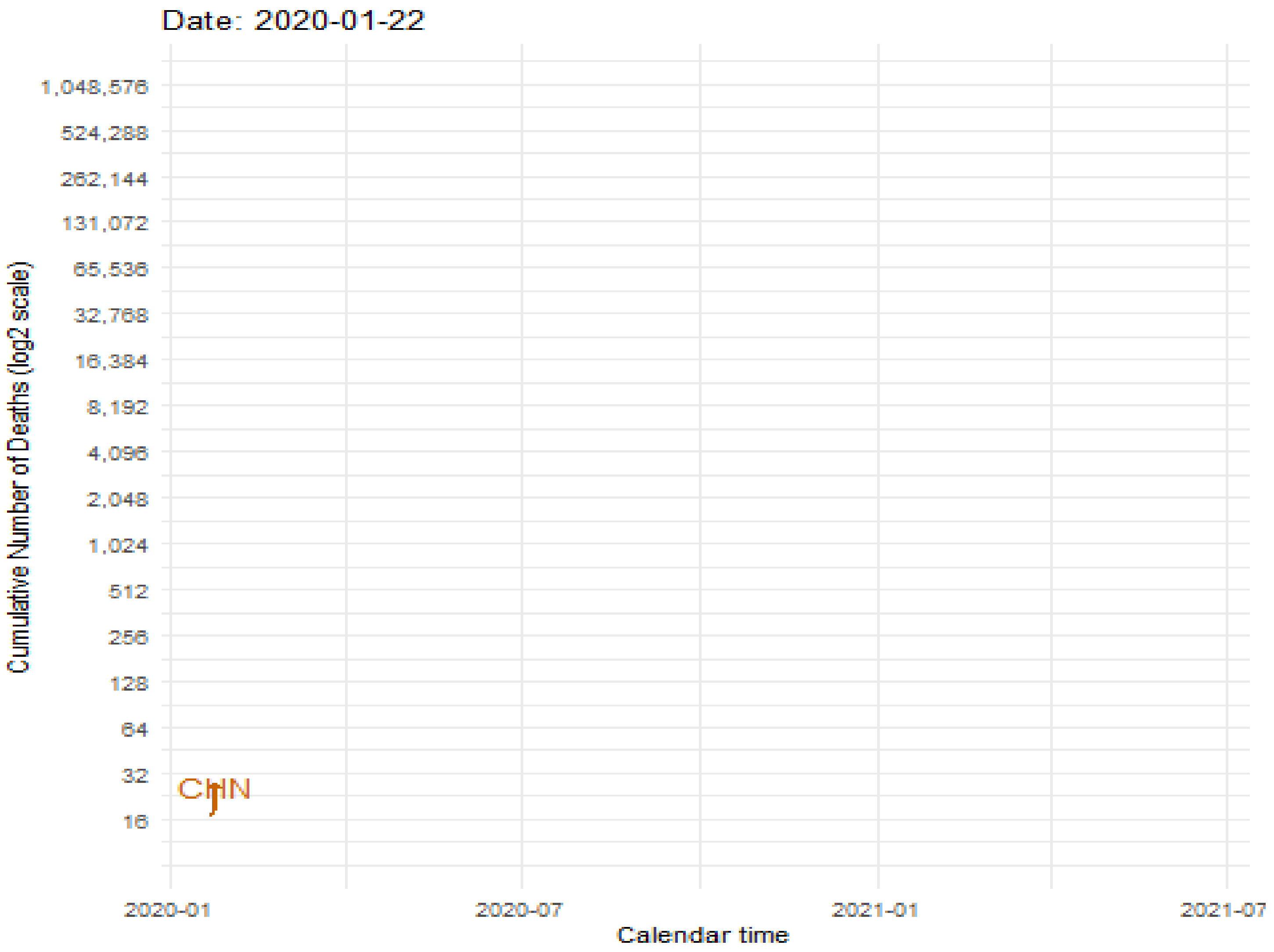


Confirmed cases per 100,000 inhabitants, GDP per capita, and COVID-19 testing rate by country

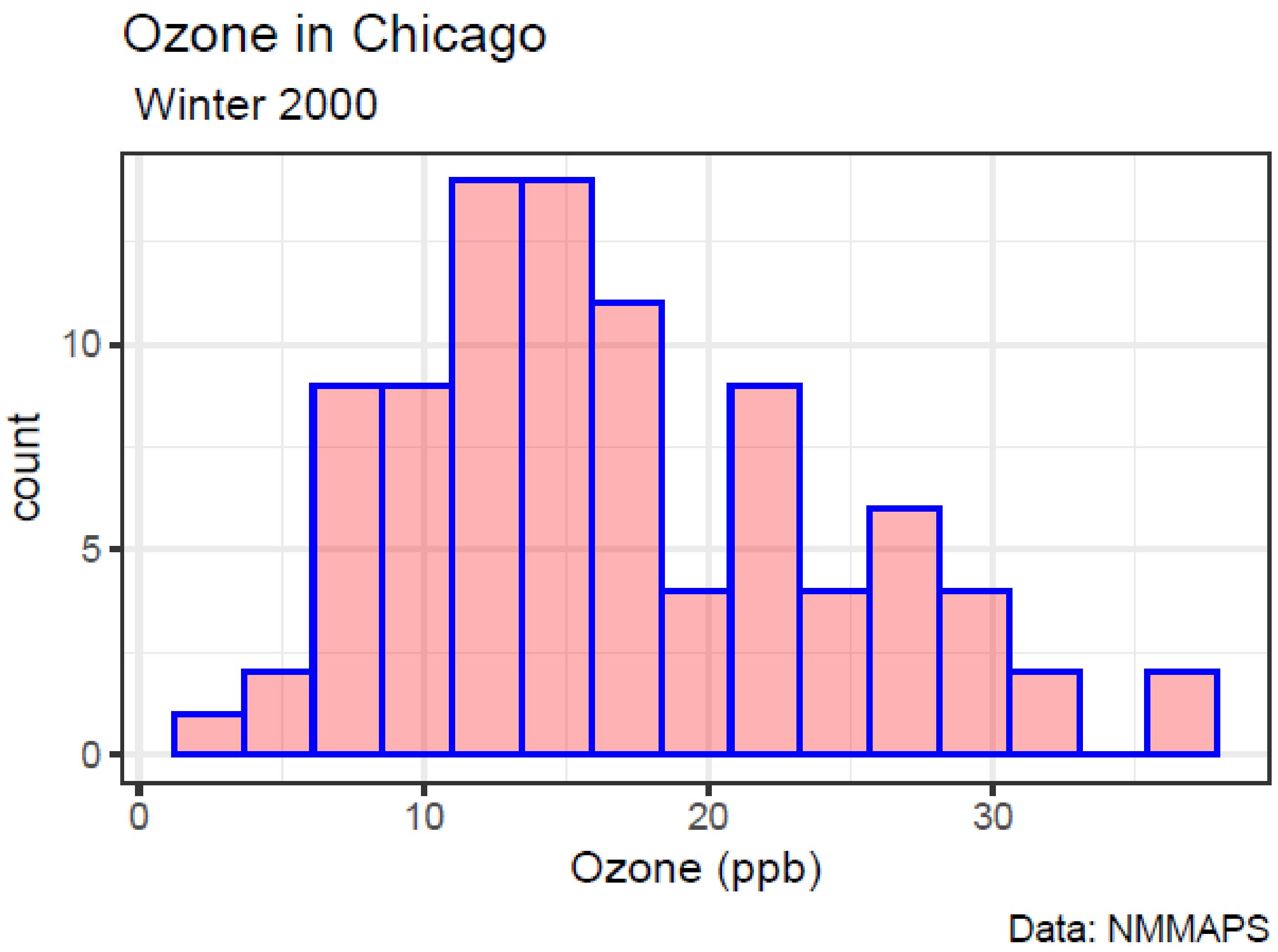
May 20, 2021

Source Data: Covid-19 related data from {tidycovid19} package

Example: Animated plots with gganimate

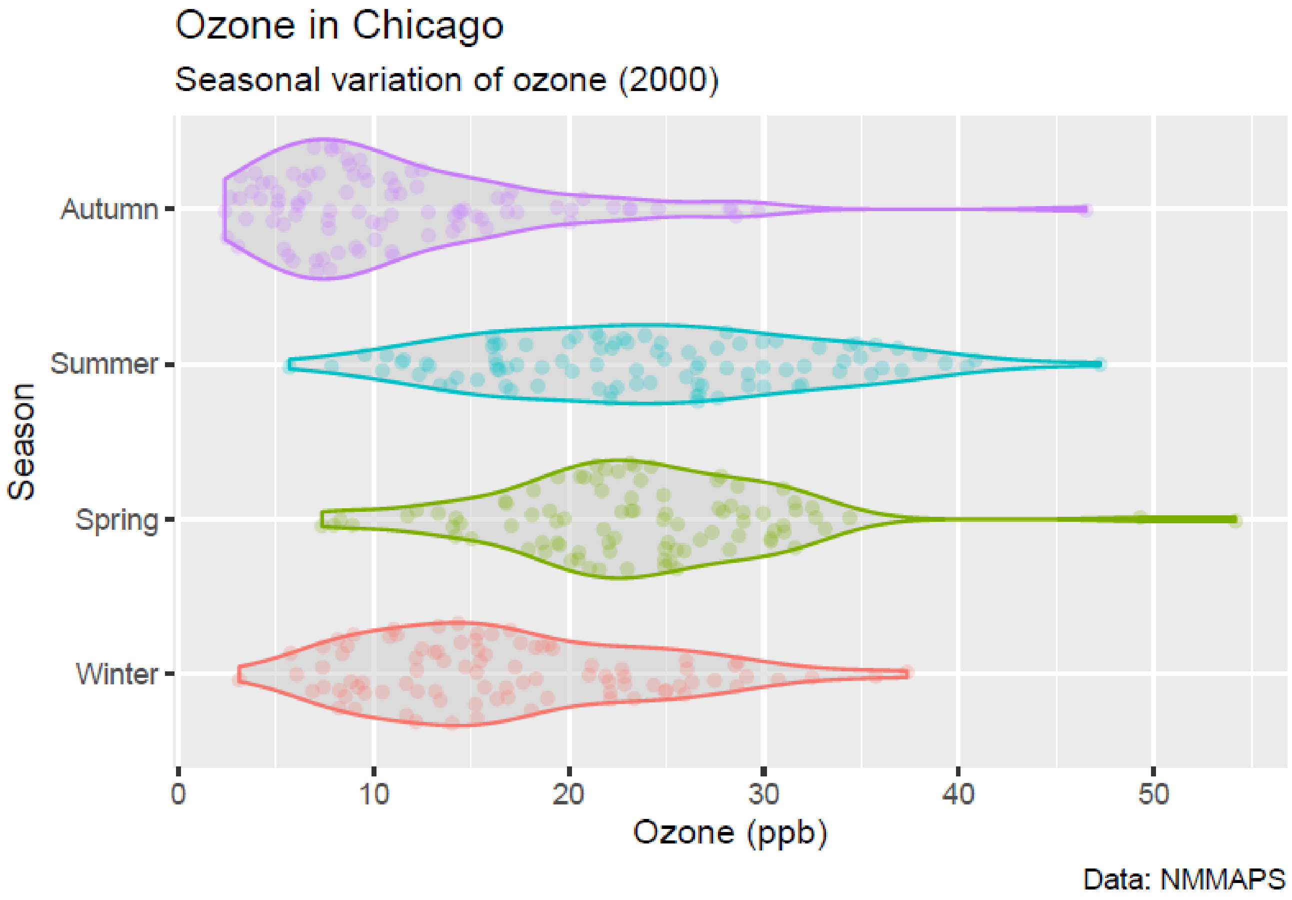


Example: Histogram



Geometry: geom_histogram (color + fill)

Example: Violin dotplot



Geometry: geom_violin (color + fill) + geom_point