Data visualization in R

(90% ggplot2)

Lara Vivian

3rd meet-up @RLadiesFra

2019/06/19 (updated: 2019-06-22)

Some background

> More than excited to share my favourite #ggplot2 functions, tips and tricks with @RLadiesFRA next week! https://t.co/BiMM8i5ATk
>
> — Lara Vivian (@LaravmVivian) 13 giugno 2019

About me and **R**:

- I started using **R** a few (7?) years ago
- still, I google basic stuff pretty much every day
- moved from *base* to *data.table* to *tidyverse*
- strong preference for whatever gets the job done

But first.. data

Preliminary steps

We are going to use the tidyverse syntax, knitr, kableExtra, and a great package for retrieving data from Eurostat link to data

- few packages need to be installed

```
install.packages("tidyverse") #* syntax and most of the functons
install.packages("knitr")     #* tables and much more
install.packages("eurostat")  #* retrieves data from eurostat
```

- and loaded

```
library(tidyverse)
library(knitr)
library(eurostat)
```

What are the datasets that contain unemployment?

```
"unemployment" %>%
  search_eurostat()
```

title	code	type	last update of data	last table structure change	data start
Harmonised unemployment (1 000) - monthly data	ei_lmhu_m	dataset	20.06.2019	20.06.2019	1983M
Harmonised unemployment rates (%) - monthly data	ei_lmhr_m	dataset	20.06.2019	20.06.2019	1983M
Long-term unemployment (12 months and more) by NUTS 2 regions	lfst_r_lfu2ltu	dataset	23.05.2019	29.04.2019	1999

Download the data

Youth unemployment ratio by sex and age 'yth_empl_140'

```
"yth_empl_140" %>% get_eurostat() -> data_raw
```

Have a look at the data

```
data_raw %>% glimpse()
data_raw %>% select_if(is.factor) %>% map(~unique(.)) #* unique value
```

Filter and select variables

```
data_raw %>%
  filter(
    sex %in% c("F", "M"),                          #* either females or ma
    geo %in% c("DE","ES","FR","IT"),               #* four largest economie
    age %in% c("Y15-19", "Y20-24", "Y25-29") #* three age brackets
          ) %>%
  select(-unit                                     #* unit is constant
         )-> data_values
```

TIP: labelling values and variables is a good practice and can be particularly helpful with graphs

```
data_values %>%
  label_eurostat() %>%
  mutate(geo = recode(geo, `Germany (until 1990 former territory of t
         geo = structure(geo, label = "country"),
         values = structure(values, label = "unemployement rate"),
         time = structure(time, label = "year")
         ) -> data_labels
```

Additional options (examples not run)

- Recode/label values

```
#* recode
data_values %>%
  mutate(geo = recode(geo, DE =

#* factor
data_values %>%
  mutate(geo = factor(geo, value
                      levels = c
```

- Rename/label variables

```
#* rename
data_values %>%
  rename("country" = geo)

#* factor (used above)
data_values %>%
  mutate(geo = structure(geo, la
```

Data values VS data labels

First six observations

data_values

sex	age	geo	time	values
F	Y15-19	DE	2018-01-01	1.8
F	Y15-19	ES	2018-01-01	5.1
F	Y15-19	FR	2018-01-01	3.2
F	Y15-19	IT	2018-01-01	3.1
F	Y20-24	DE	2018-01-01	3.1
F	Y20-24	ES	2018-01-01	15.4

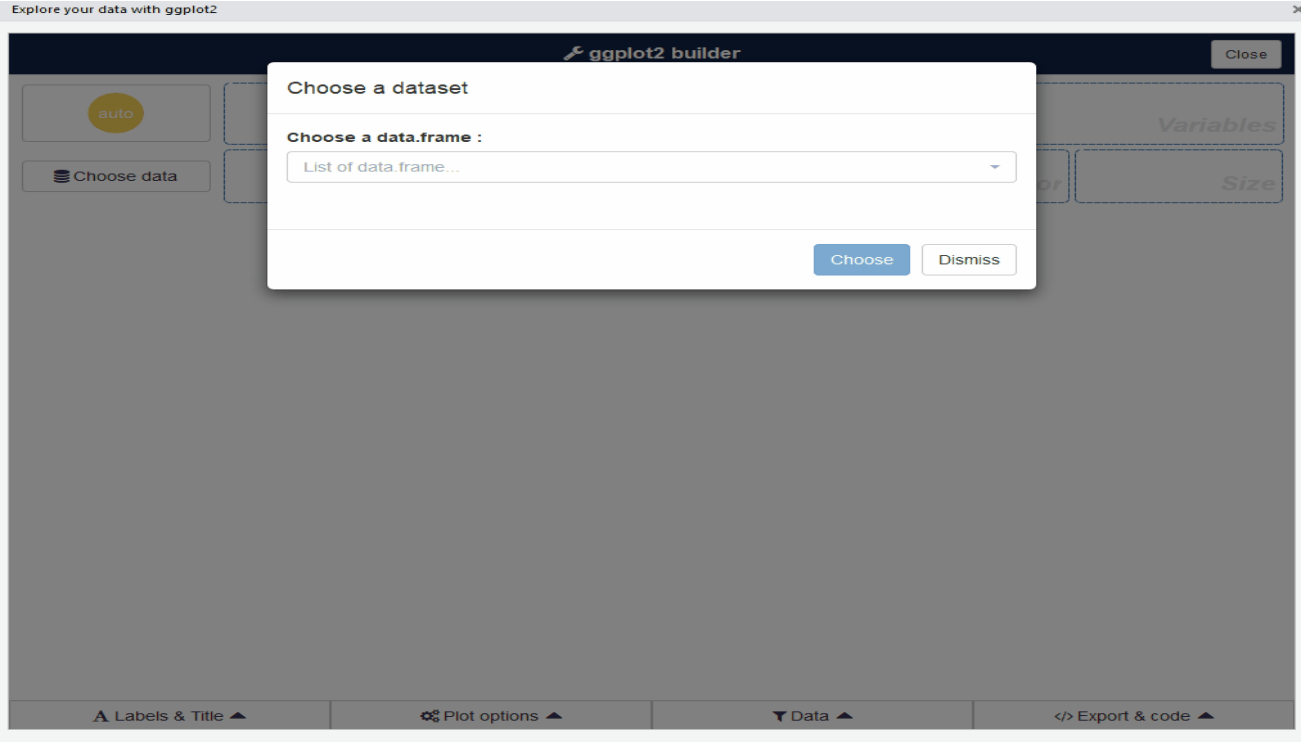
data_labels

sex	age	geo	time
Females	From 15 to 19 years	Germany	2018-01-01
Females	From 15 to 19 years	Spain	2018-01-01
Females	From 15 to 19 years	France	2018-01-01

Data visualization

Graphs without coding?

```
install.packages("esquisse")
library(esquisse)
esquisser(viewer = "browser")
```



credits: https://github.com/dreamRs/esquisse

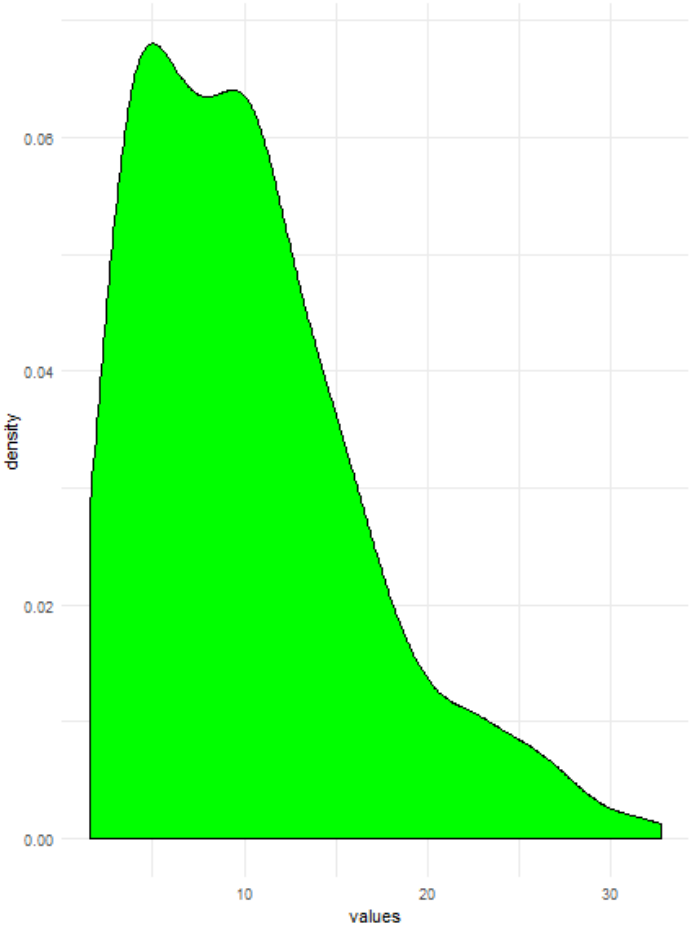
Base

```
ggplot( data = data_labels ) +
  aes( x = values ) +
  theme_minimal()
```

(some) geoms

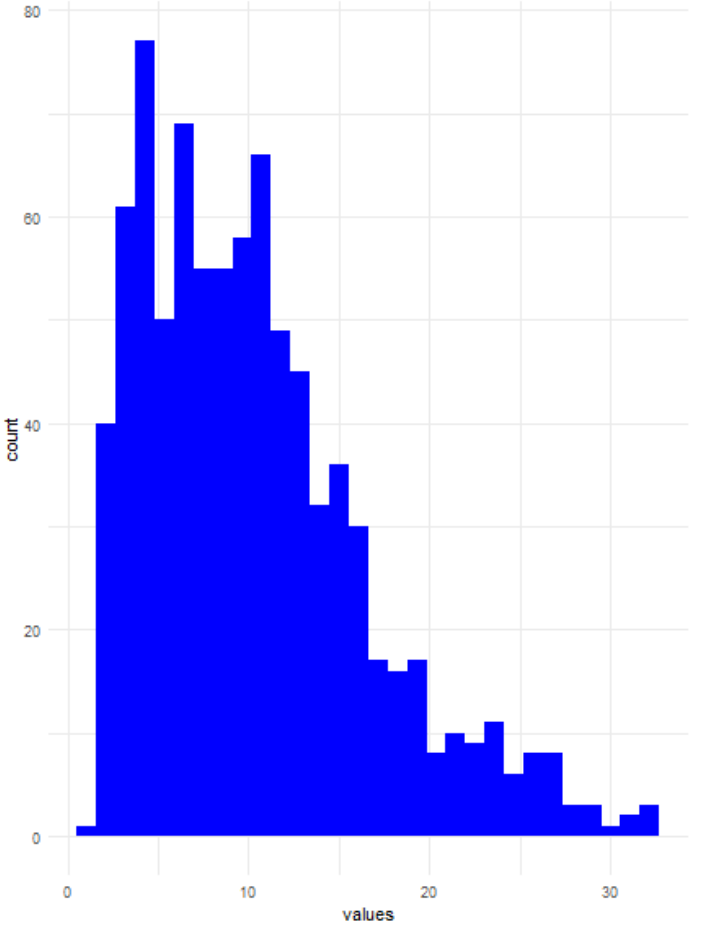
geom_density

```
ggplot( data = data_labels ) +
  aes(x = values) +
  theme_minimal() +
  geom_density(adjust = 1, fill
```



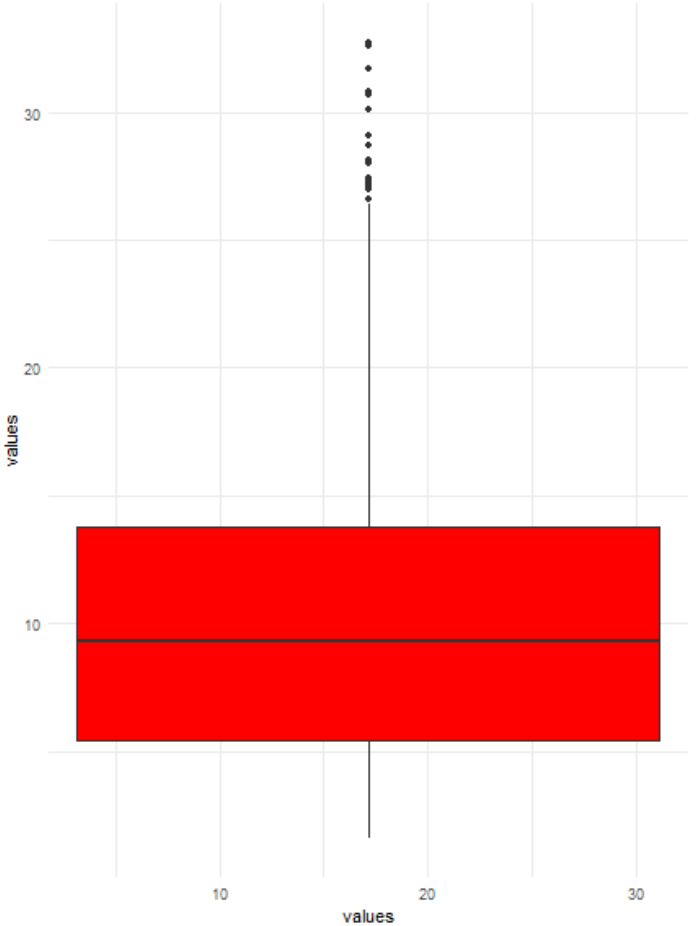
geom_histogram

```
ggplot( data = data_labels ) +
  aes(x = values) +
  theme_minimal() +
  geom_histogram(bins = 30, fill
```



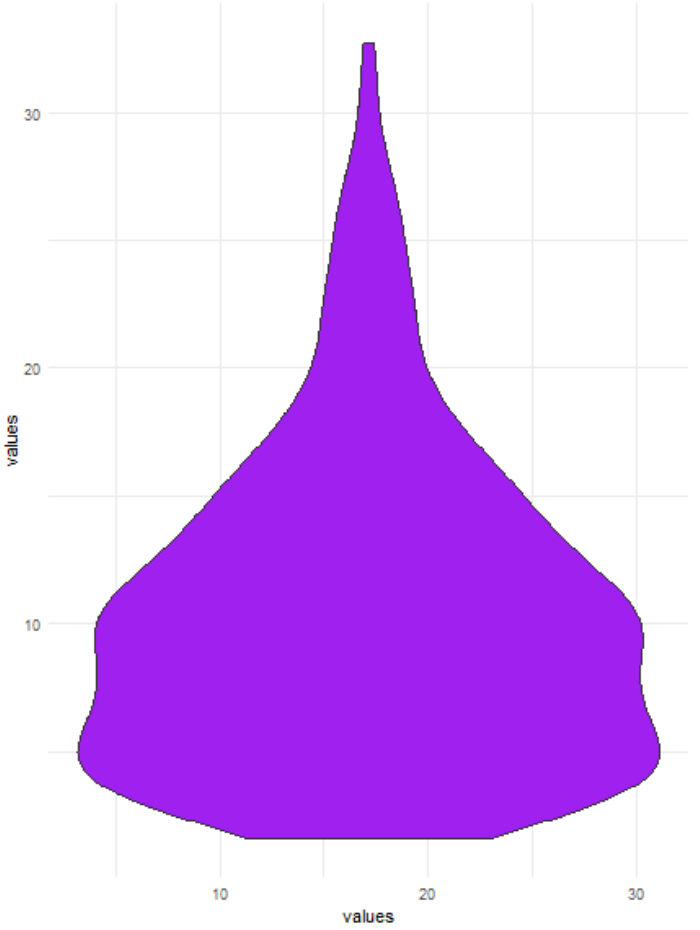
geom_boxplot

```
ggplot( data = data_labels ) +
  aes(x = values, y = values) +
  theme_minimal() +
  geom_boxplot(fill = "red")
```



geom_violin

```
ggplot( data = data_labels ) +
  aes(x = values, y = values) +
  theme_minimal() +
  geom_violin(fill = "purple")
```

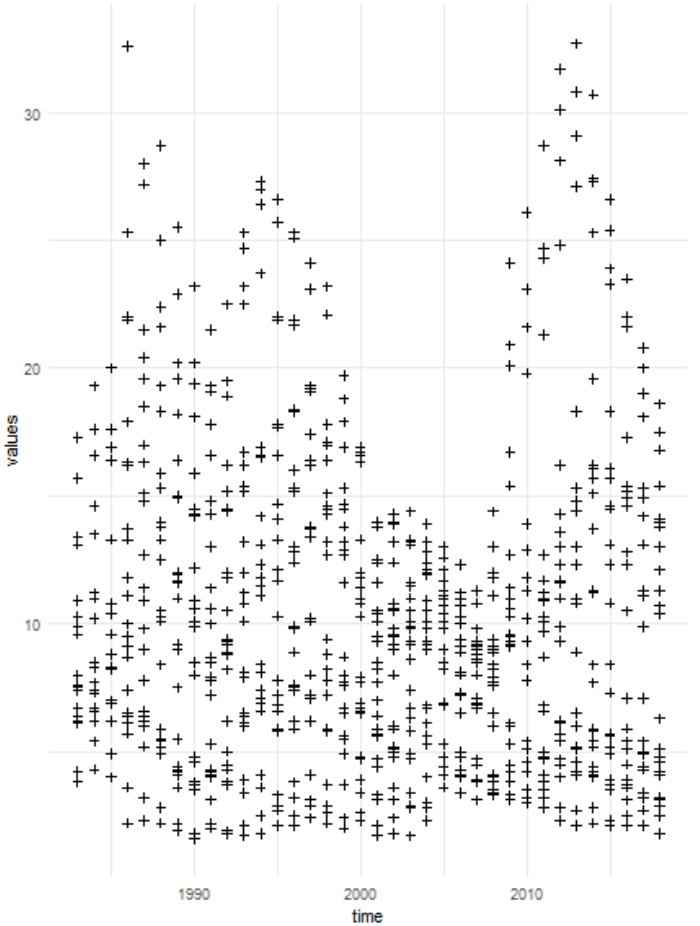


geom_bar

```
ggplot( data = data_labels ) +
  aes(x = geo) +
  theme_minimal() +
  geom_bar( fill = "yellow" )
```

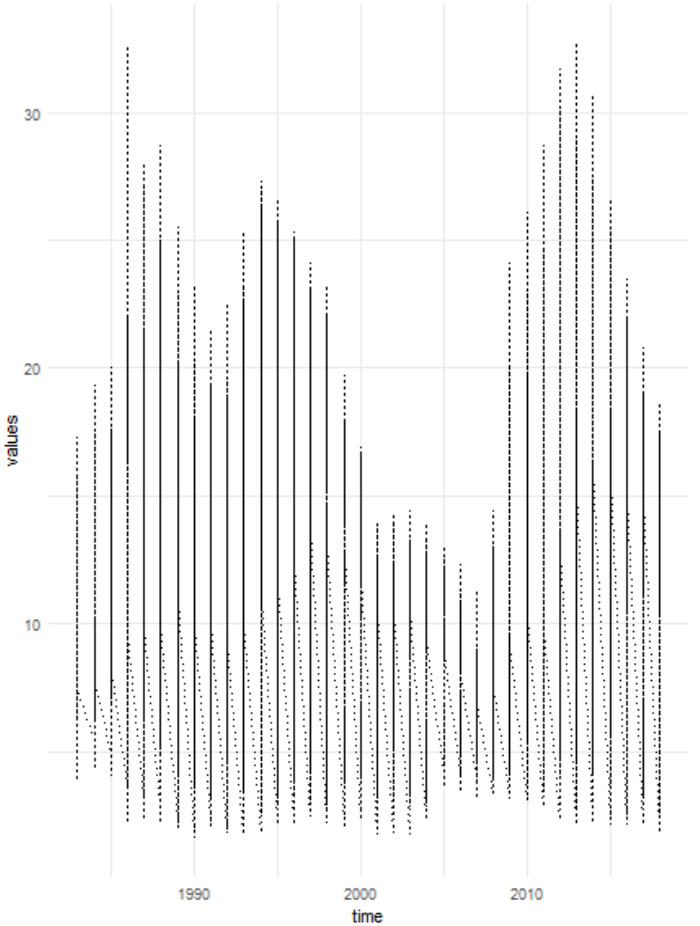
geom_point

```
ggplot( data = data_labels ) +
  aes(x = time, y = values) +
  theme_minimal() +
  geom_point( shape = 3 )
```



geom_line

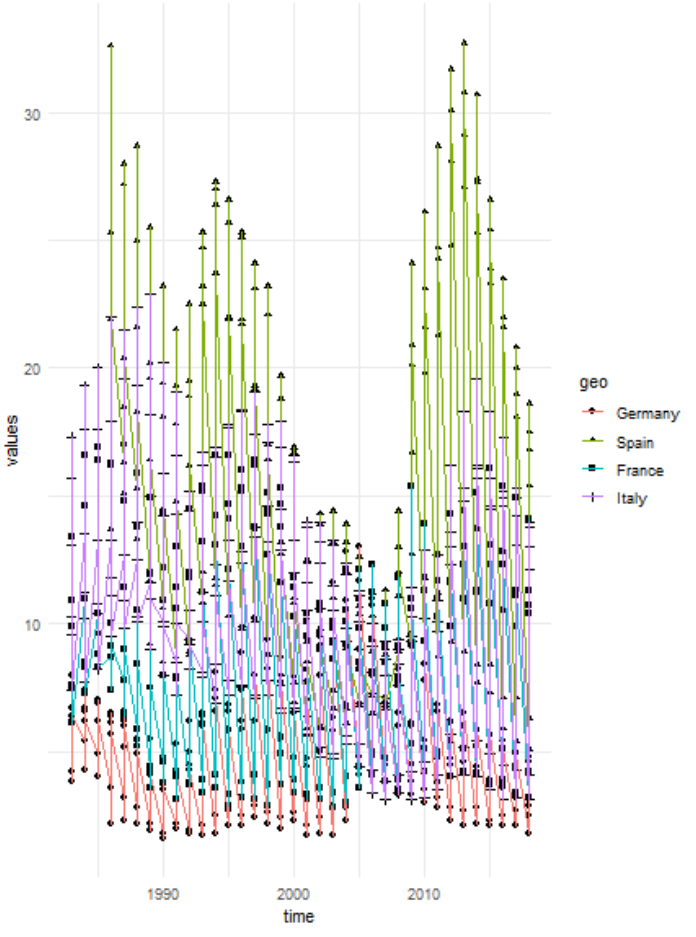
```
ggplot( data = data_labels ) +
  aes(x = time, y = values) +
  theme_minimal() +
  geom_line( linetype = "dotted"
```



Groups

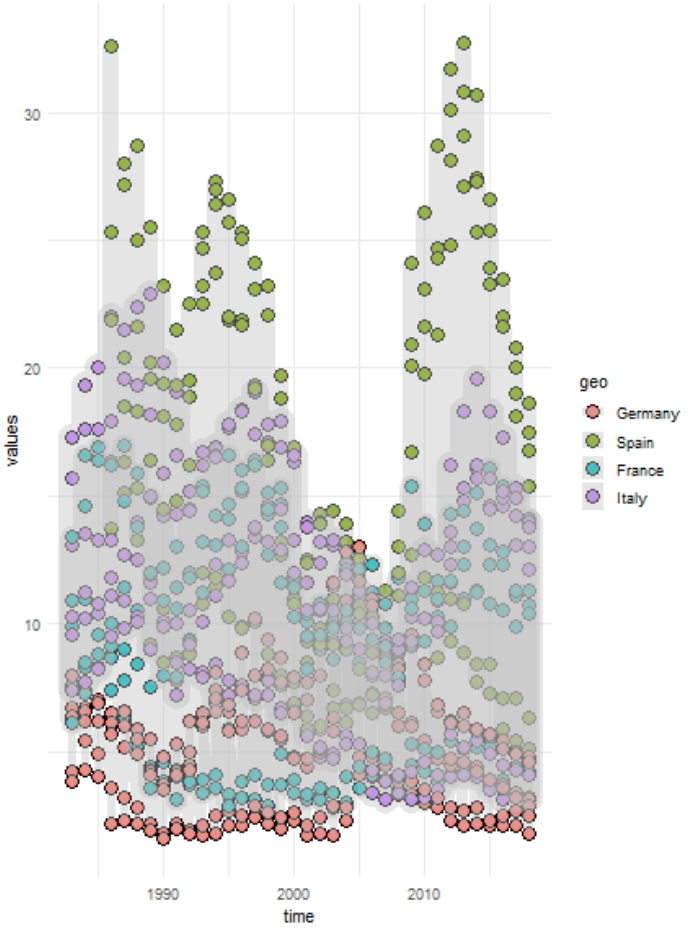
Shapes and colors

```
ggplot( data = data_labels ) +
  aes(x = time, y = values, grou
  theme_minimal() +
  geom_point(aes(shape = geo)) +
  geom_line(aes(color = geo))
```



Fill and size

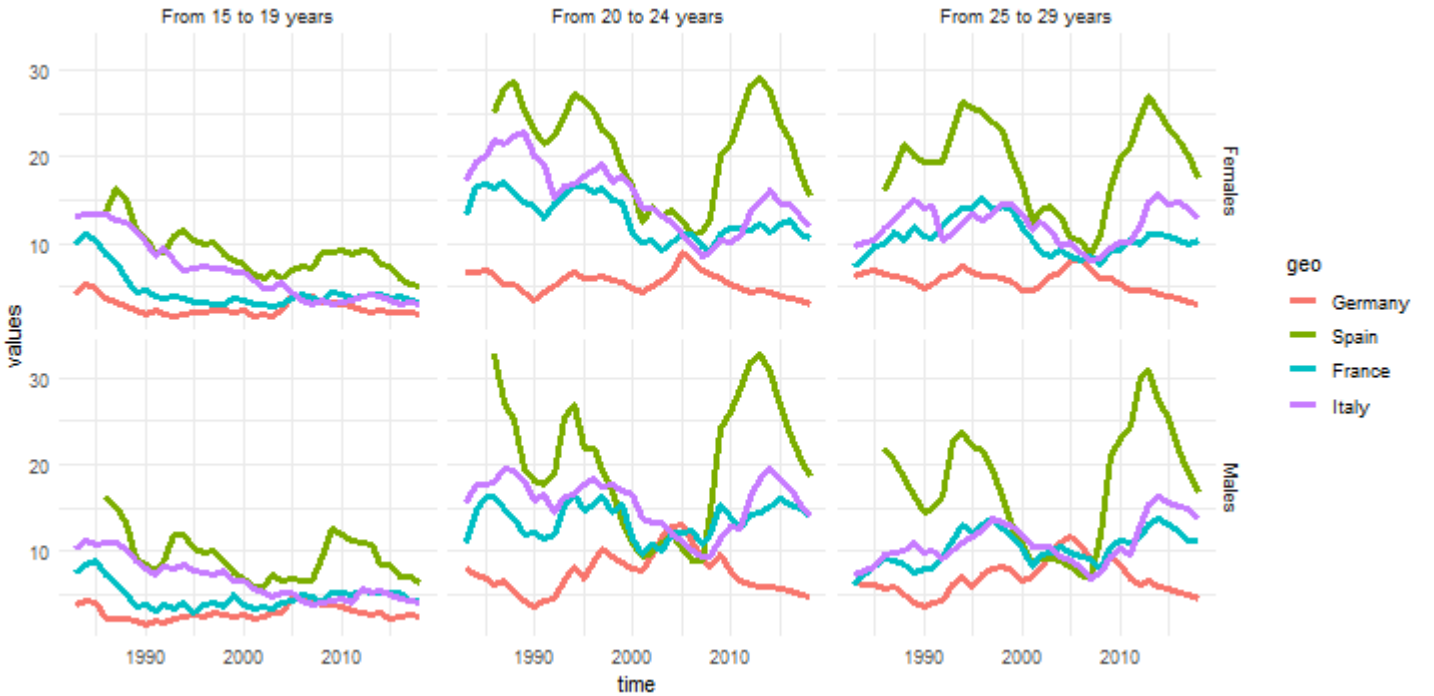
```
ggplot( data = data_labels ) +
  aes(x = time, y = values, grou
  theme_minimal() +
  geom_point(aes(fill = geo), sh
  geom_line(aes(size = geo), col
```



Facetting

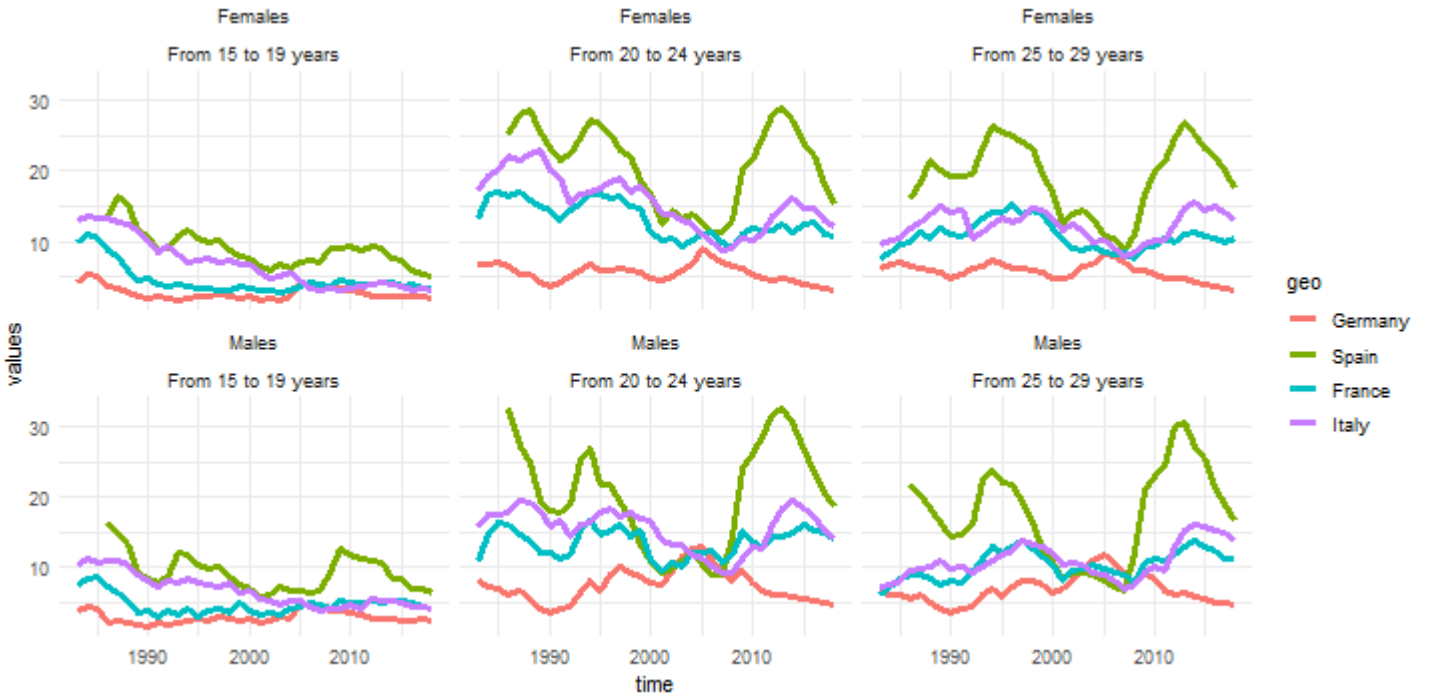
facet_grid

```
ggplot( data = data_labels ) +
  aes(x = time, y = values) +
  theme_minimal() +
  geom_line(aes(color = geo), size = 1.2) +
  facet_grid(sex~age)
```



facet_wrap

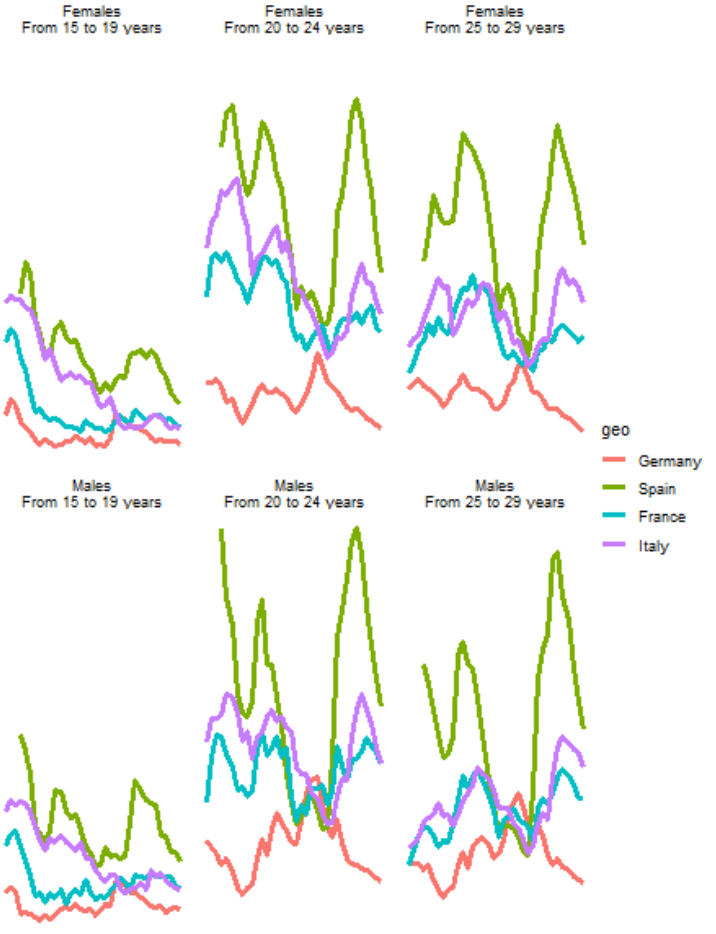
```
ggplot( data = data_labels ) +
  aes(x = time, y = values) +
  theme_minimal() +
  geom_line(aes(color = geo), size = 1.2) +
  facet_wrap(sex~age)
```



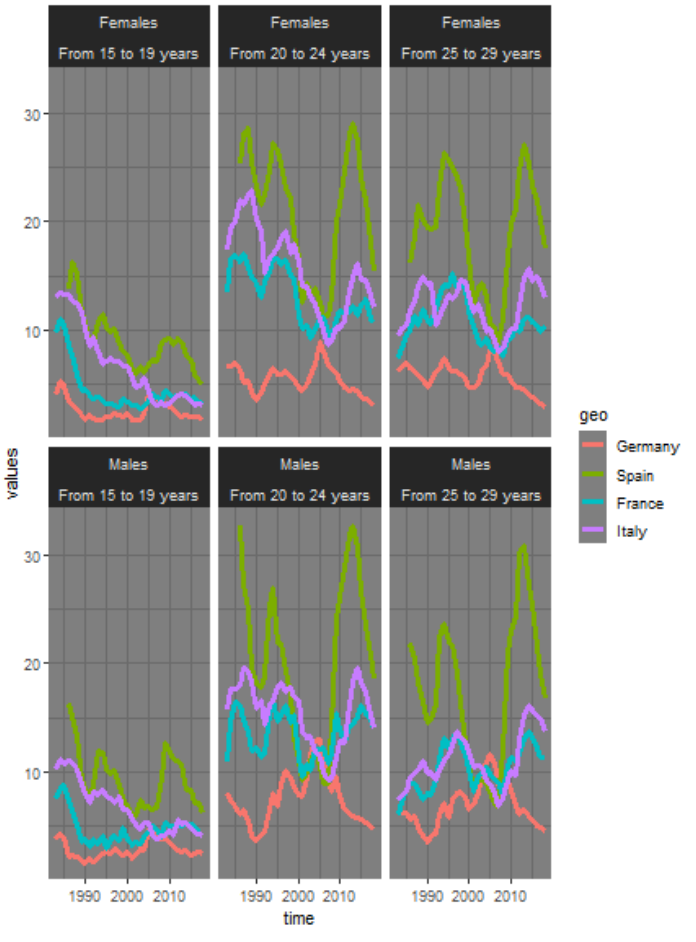
Themes and colors

e.g. from ggplot2

- **theme_void**

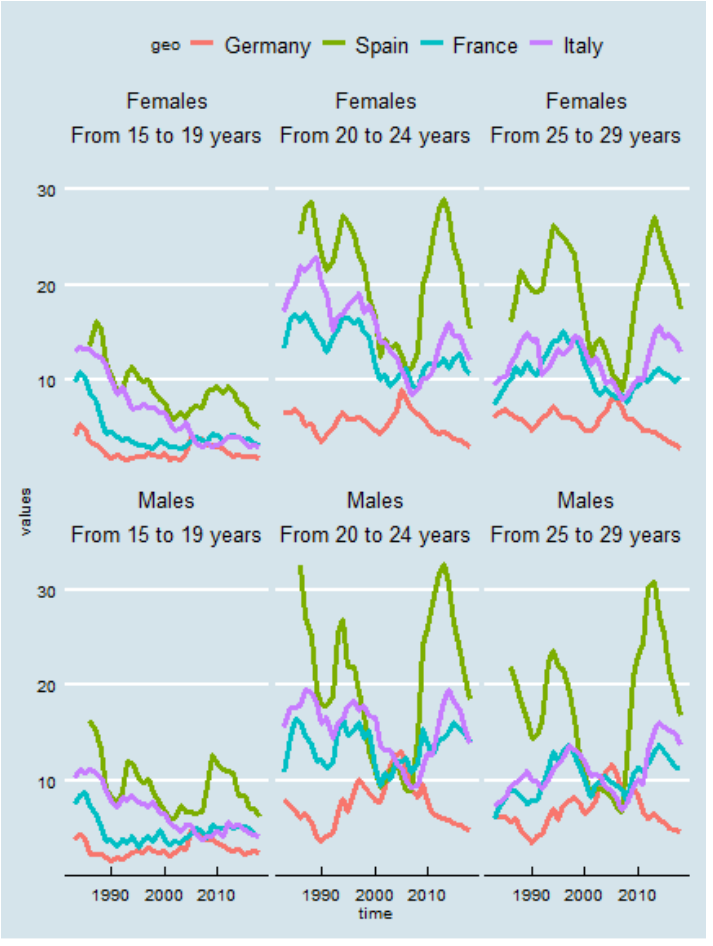


- **theme_dark**



e.g. from ggthems library(ggthemes)

- **theme_economist**

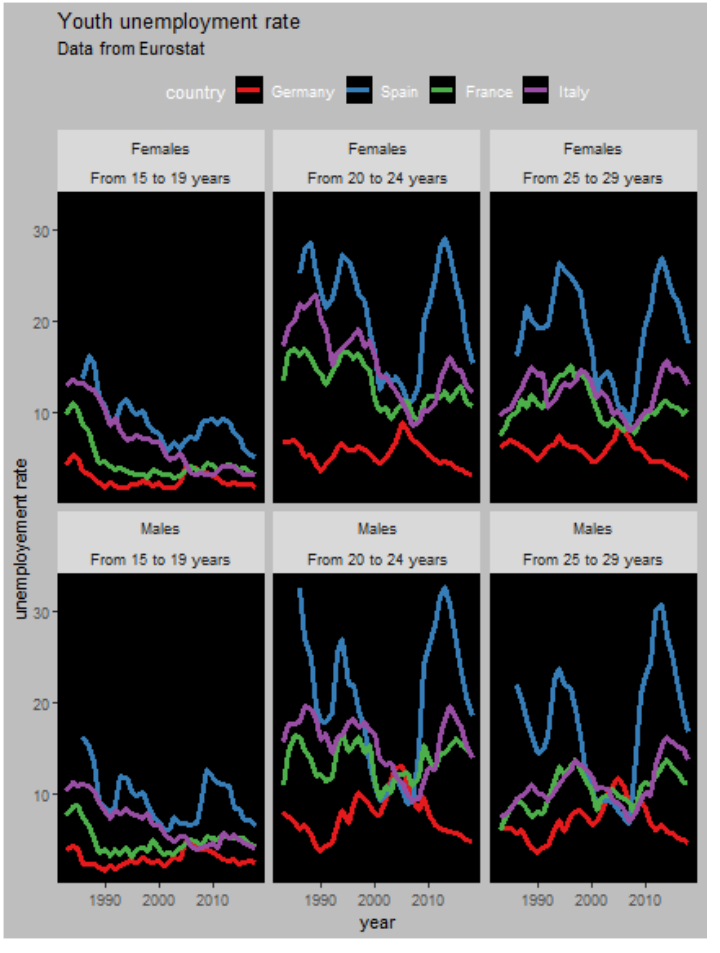


- **theme_solarized**

Our own?

```
ggplot( data = data_labels ) + a

  #* titles
  ggtitle("Youth unemployment ra

  theme(
    #* legend position
    legend.position = "top", leg
    #* get rid of panel grids
    panel.grid.major = element_b
    #* plot and panel backgrounc
    plot.background=element_rect
  ) +

  #* colors lines
  scale_color_brewer(palette = "
```

TIP (follow-up): variable labels come in handy for titles

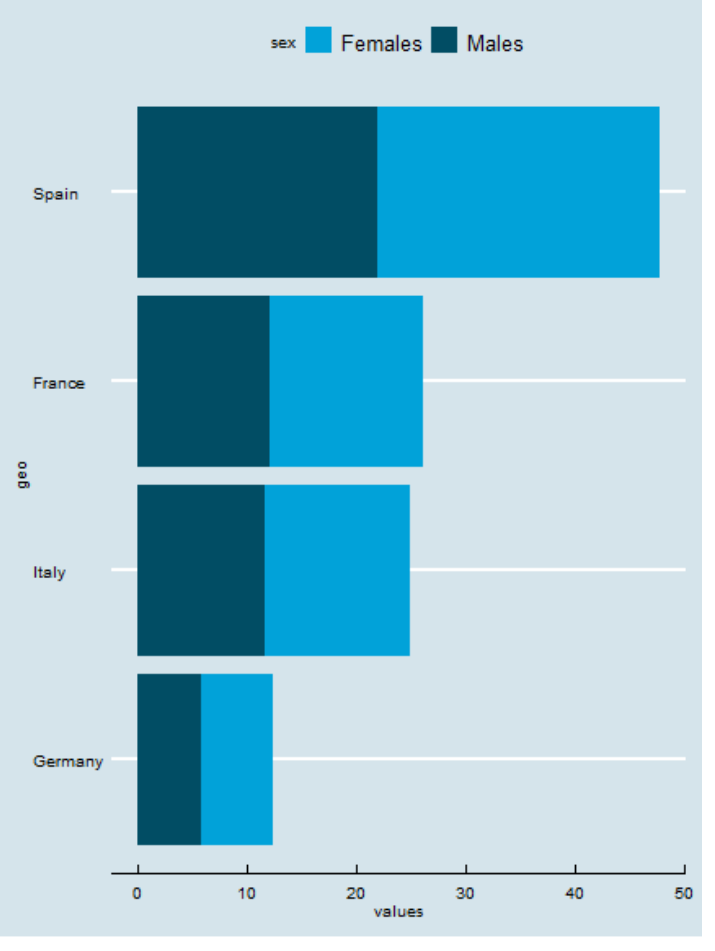


More?

Barplot

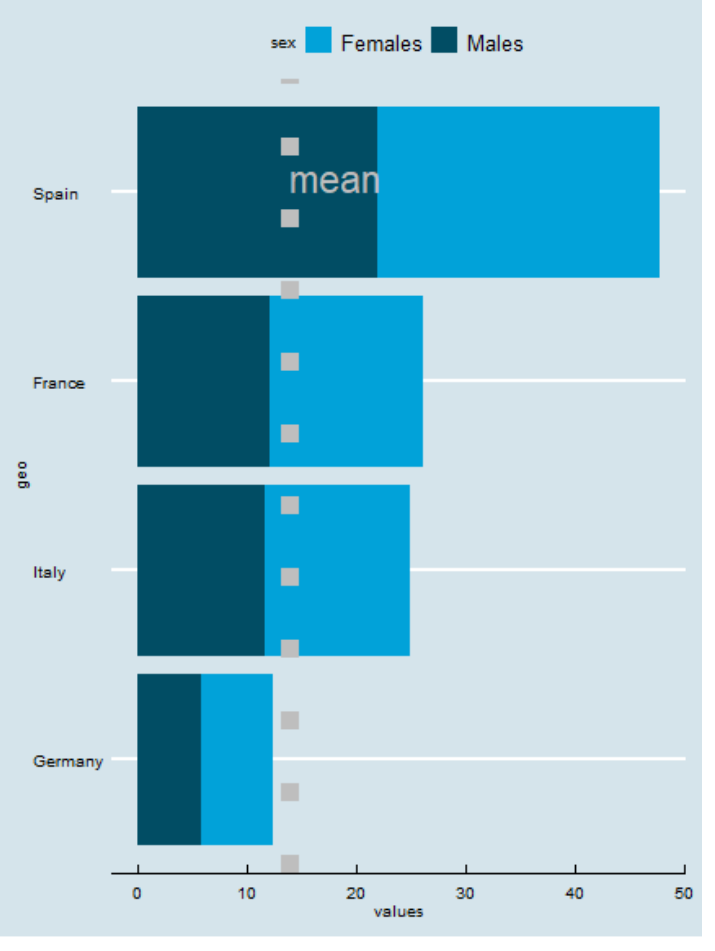
```
data_labels %>%
  filter(time == "1995-01-01" &
  mutate(geo = reorder(geo, valu

ggplot( data = data_sel ) +
  aes(x = geo, y = values) +
  geom_bar( aes(fill = sex), sta
  theme_economist() +
  scale_fill_economist() +
  coord_flip() -> bar

bar
```

Disclaimer: graph not really meaningful as we sum unemployment rates



Text

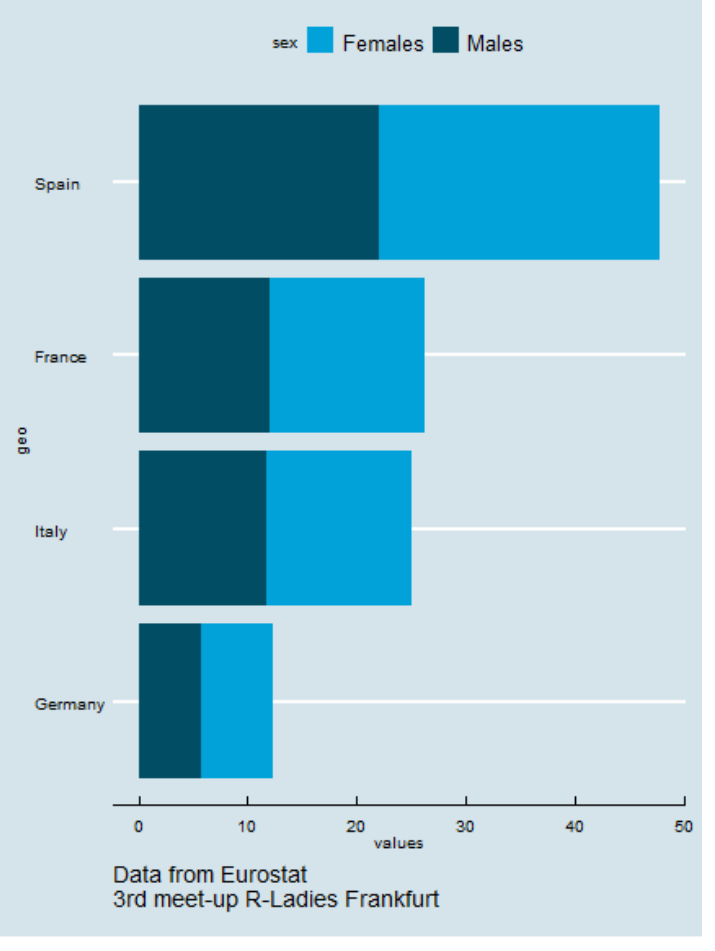
```
bar +
  geom_hline(yintercept=mean(dat
  annotate("text", x = "Spain",
```



Text below

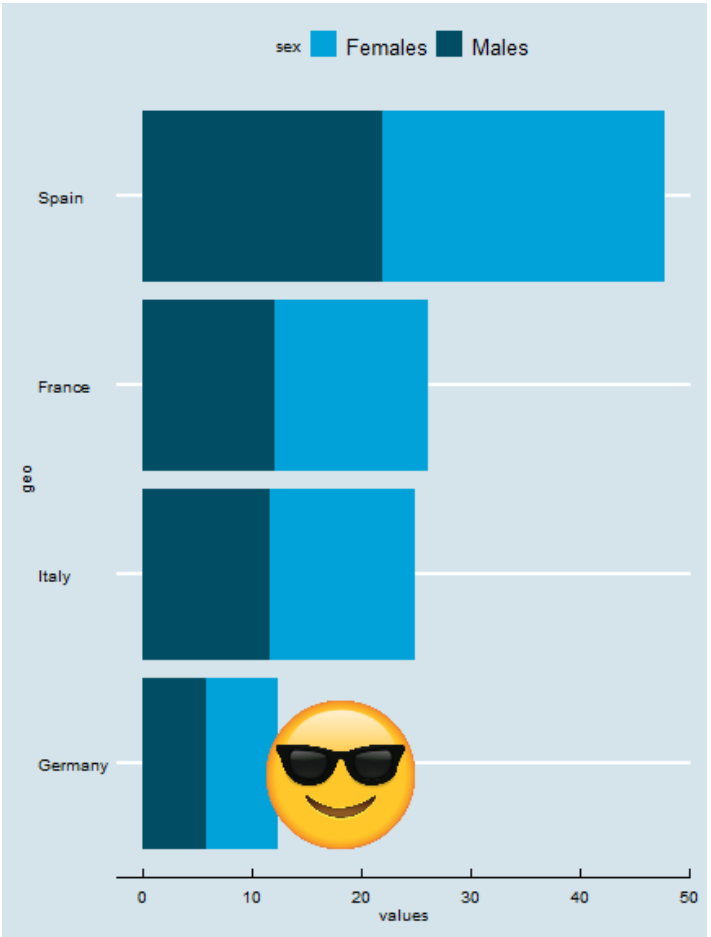
```
library(cowplot) #* great for cc

bar %>%
  add_sub("Data from Eurostat \n
  ggdraw()
```

Have a look at library(ggrepel) as well



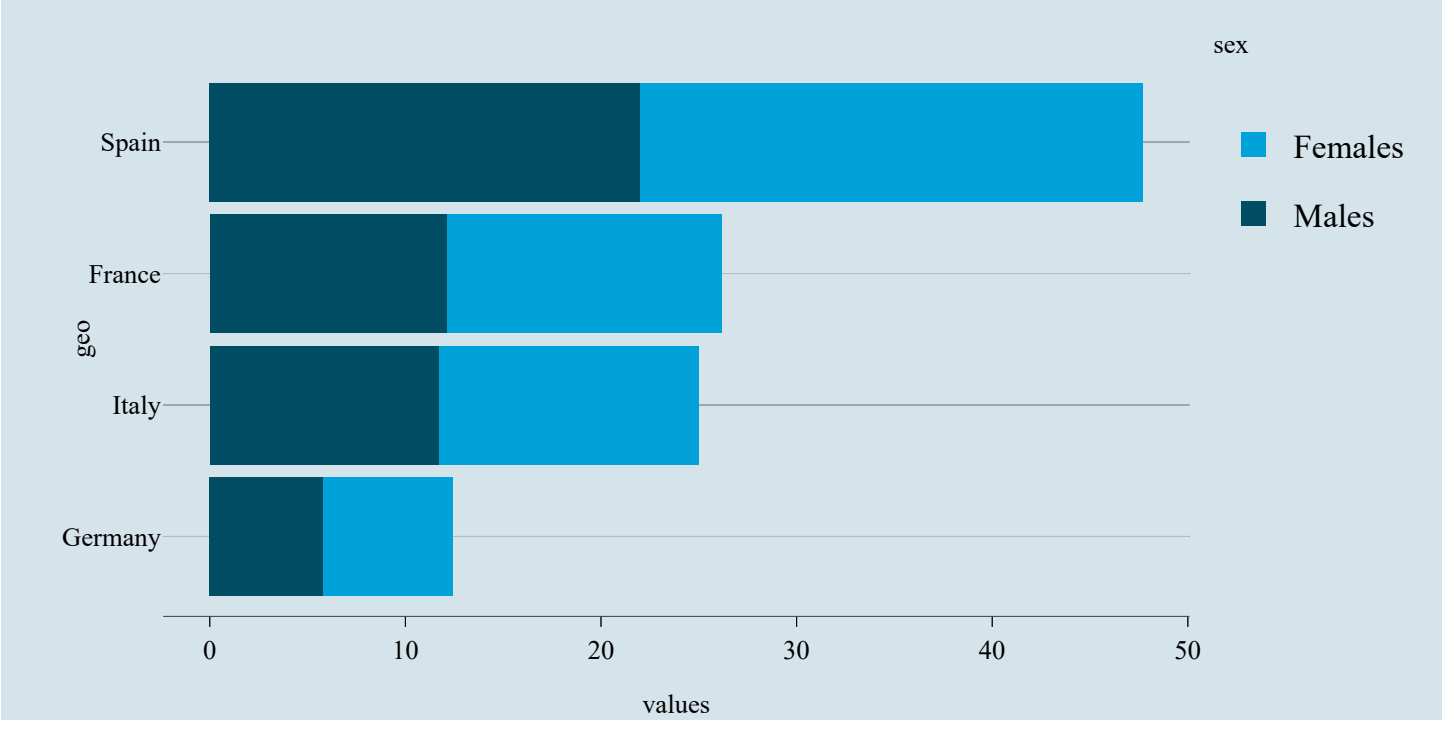
Images

```
library(grid)     #* pre-installe
library(magick)   #* read png

"https://bitbucket.org/laravmviv
  image_read() %>%
  rasterGrob(interpolate=TRUE) -

bar +
  annotation_custom(img, xmin =
```



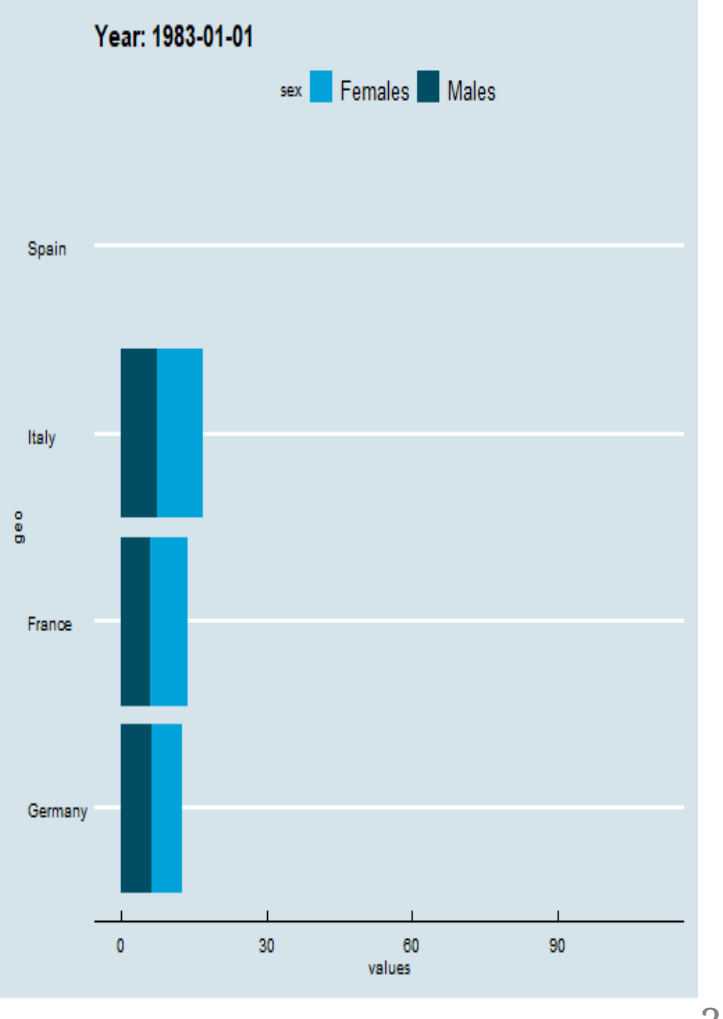
Interactive

```
library(plotly)

bar %>%
  ggplotly()
```



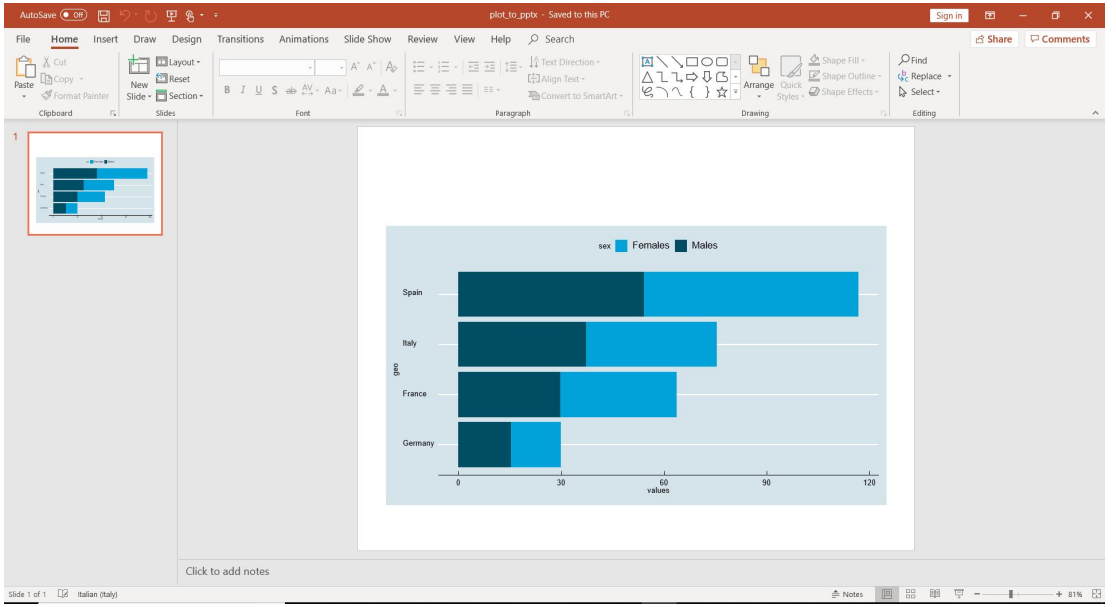
Animation

```
library(gganimate)

data_labels %>% filter(age == "F

ggplot( data = data_labels_anim

  #* animation starts here
  labs(title = 'Year: {frame_tim

animate(p_anim) #* View the grap
animate(p_anim, nframes = 24, re
```



Export to PowerPoint

```
library(officer); library(rvg)

read_pptx() %>% #* this code is likely to be outdated
  add_slide(layout = "Title and Content", master = "Office Theme") %>
  ph_with_vg(code = print(bar), type = "body") %>%
  print(target = "plot_to_pptx.pptx")
```



Great resources out there

- practise with TidyTuesday
- ggplot2
- ggplot2 ext
- https://www.r-graph-gallery.com/
- https://www.williamrchase.com/slides/ggplot_intro.html#1
- https://evamaerey.github.io/ggplot_flipbook/ggplot_flipbook_xaringan.html#1
- tidyverse team and RLadies groups
- and many more!

Thank you!

Slides created via the R package **xaringan**, **knitr**, and R Markdown.

Additional packages used tidyverse, eurostat, kableExtra, esquisse, ggthemes, magick, cowplot, plotly, officer, rvg

Ps: do not forget to save your plots with ggsave()!