

# Visualisation of evolution of novel coronavirus

Marcelo Avila

20 März, 2020

## Contents

load libraries	1
load data	1
Cumulative Cases for Selected Countries	2
Normalized comparison	4

## load libraries

## load data

```
cat_labels <- c("daily cases", "daily deaths",  
               "cumulative cases", "cumulative deaths")  
df <- df %>% mutate(category = factor(category,  
                                     labels = cat_labels),  
                   continent=countrycode(location,  
                                     origin = "country.name",  
                                     destination = "continent"))
```

```
## Warning in countrycode(location, origin = "country.name", destination = "continent"): Some values were
```

```
if (interactive()) write_csv(df, "../data/our_world_in_data_corona.csv")
```

There is data for 159 countries. The last update was on 17 of March

```
head(df)
```

```
## # A tibble: 6 x 8  
##   date      location category value cca2 pop2020 value_per_100k_~ continent  
##   <date>    <chr>    <fct>    <dbl> <chr>  <dbl>      <dbl> <chr>  
## 1 2020-02-25 Afghanis~ daily cas~    0 AF    3.89e7      0 Asia  
## 2 2020-02-25 Afghanis~ daily dea~    0 AF    3.89e7      0 Asia
```

```
## 3 2020-02-25 Afghanis~ cumulativ~      1 AF      3.89e7      0.00257 Asia
## 4 2020-02-25 Afghanis~ cumulativ~      0 AF      3.89e7      0      Asia
## 5 2020-02-26 Afghanis~ daily cas~      0 AF      3.89e7      0      Asia
## 6 2020-02-26 Afghanis~ daily dea~      0 AF      3.89e7      0      Asia
```

```
# for thousand separator for plotting
fun_dot <- function(x) format(x, big.mark = " ",
                             scientific = FALSE,
                             decimal.mark = ".")

# negative in
`%!in%` <- Negate(`%in%`)

# y scale
log10_minor_break = function (...){
  function(x) {
    minx      = floor(min(log10(x), na.rm=T))-1;
    maxx      = ceiling(max(log10(x), na.rm=T))+1;
    n_major    = maxx-minx+1;
    major_breaks = seq(minx, maxx, by=1)
    minor_breaks =
      rep(log10(seq(1, 9, by=1)), times = n_major)+
      rep(major_breaks, each = 9)
    return(10^(minor_breaks))
  }
}

calc_cagr <- function(ending, beginning, nr_periods) {
  return((ending / beginning)^(1/nr_periods))-1
}
```

## Cumulative Cases for Selected Countries

```
sel_category <- "cumulative cases" # new cases, new deaths,
                                     # cumulative cases, cumulative deaths
selected_countries <- c("Germany",
                        "Italy",
                        "United Kingdom",
                        "Brazil",
                        "China",
                        "Spain",
                        "South Korea",
                        "United States")

df %>%
  # wrangle
  filter(location %in% selected_countries, category==sel_category) %>%
  mutate(
    location=fct_reorder2(location, date, value),
    label = ifelse(date == "2020-02-29", #max(date),
                   as.character(location),
```

```

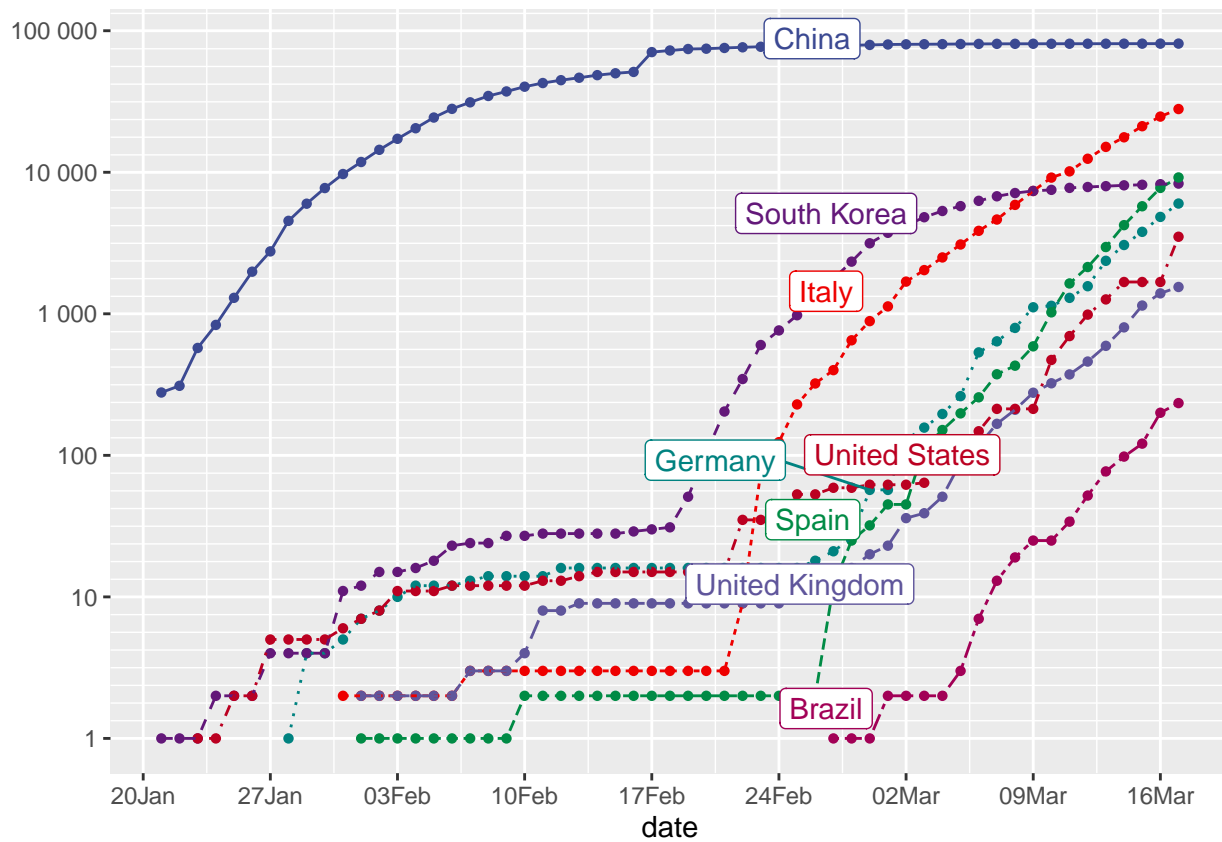
NA_character_)) %>%

# plot
ggplot(aes(x=date, y=value, color=location)) +
  geom_point(aes(shape="o")) +
  geom_line(aes(linetype=location)) +

# labels
ggrepel::geom_label_repel(aes(label = label),
                          nudge_x = -1, nudge_y = 1/10,
                          na.rm = TRUE) +

# legend and scales
scale_y_log10(name="(log scale)", labels=fun_dot,
              breaks=10^(0:9),
              #minor_breaks=log10_minor_break()
              minor_breaks=(1 * 1.33333^(0:10000))
              ) +
scale_x_date(date_breaks = "1 week",
             date_labels="%d%b") +
ggsci::scale_color_aas() +
theme(legend.position = "none",
      axis.title.y = element_blank())

```



## Normalized comparison

```
df$continent %>% unique()
```

```
## [1] "Asia"      "Europe"    "Africa"    "Americas" "Oceania"  NA
```

```
df$category %>% unique() %>% as.character()
```

```
## [1] "daily cases"      "daily deaths"      "cumulative cases"
## [4] "cumulative deaths"
```

```
make_plot <- function(dta, sel_category="cumulative cases",
                      sel_continent="Europe",
                      sel_country=NULL,
                      threshold=100) {

  # extract singular form of category
  singular_category <- scan(text = sel_category, what = "", quiet=TRUE)[2] %>%
    substr(1, nchar(.)-1)

  sel_plot_title <- ifelse(length(sel_continent)>1,
                          yes = "selected continents",
                          no = sel_continent)

  if (!is.null(sel_country)) {
    dta <- dta %>% filter(location %in% sel_country)
    sel_continent <- unique(dta$continent)

    sel_plot_title <- ifelse(length(sel_country)>1,
                          yes = "selected countries",
                          no = sel_country)
  }

  dta %>%
    filter(category == sel_category,
           continent %in% sel_continent) %>%
    group_by(location) %>%
    filter(value>=threshold) %>%
    mutate(date_shift=0:(n()-1)) %>% # generate date shifted after threshold
    ungroup(location) %>%
    arrange(date_shift) %>%
    # mutate for legend ordering and ggrepel postioning
    mutate(location=fct_reorder2(location, date_shift, value),
           # for ggrepel
           label = ifelse(date == max(date),
                          as.character(location),
                          NA_character_)) %>%

  # plot
  ggplot(aes(x=date_shift, y=value, colour=location)) +
  # geoms
```

```

geom_point() +
geom_line(aes(linetype=location, colour=location)) +
#scale_color_grey(start = 0, end = .3) +

# ggrepel
ggrepel::geom_label_repel(
  aes(label = label), nudge_x = 1, na.rm = TRUE) +

# legend and scales
scale_y_log10(
  name = paste(sel_category, "(log scale)"),
  labels=fun_dot, breaks = 10^(0:9), minor_breaks=log10_minor_break()) +
scale_x_continuous(
  name = paste0("Days past since ", threshold, "th ", singular_category),
  breaks = seq(0, 1e5, 7), minor_breaks = 1:1e3) +

theme(legend.position = "none", axis.title.y = element_blank()) +
#title
ggtitle(paste("Evolution of novel coronavirus in", sel_plot_title),
  subtitle = paste(sel_category, "(log scale)"))
}

```

```

countries <- c("Brazil", "United States", "Italy", "Iran", "South Korea",
  "Singapore", "Japan", "Australia", "India", "Germany", "Spain",
  "Portugal", "Switzerland", "Serbia", "France")
assertthat::are_equal(1, mean(countries %in% df$location))

```

```
## [1] TRUE
```

```

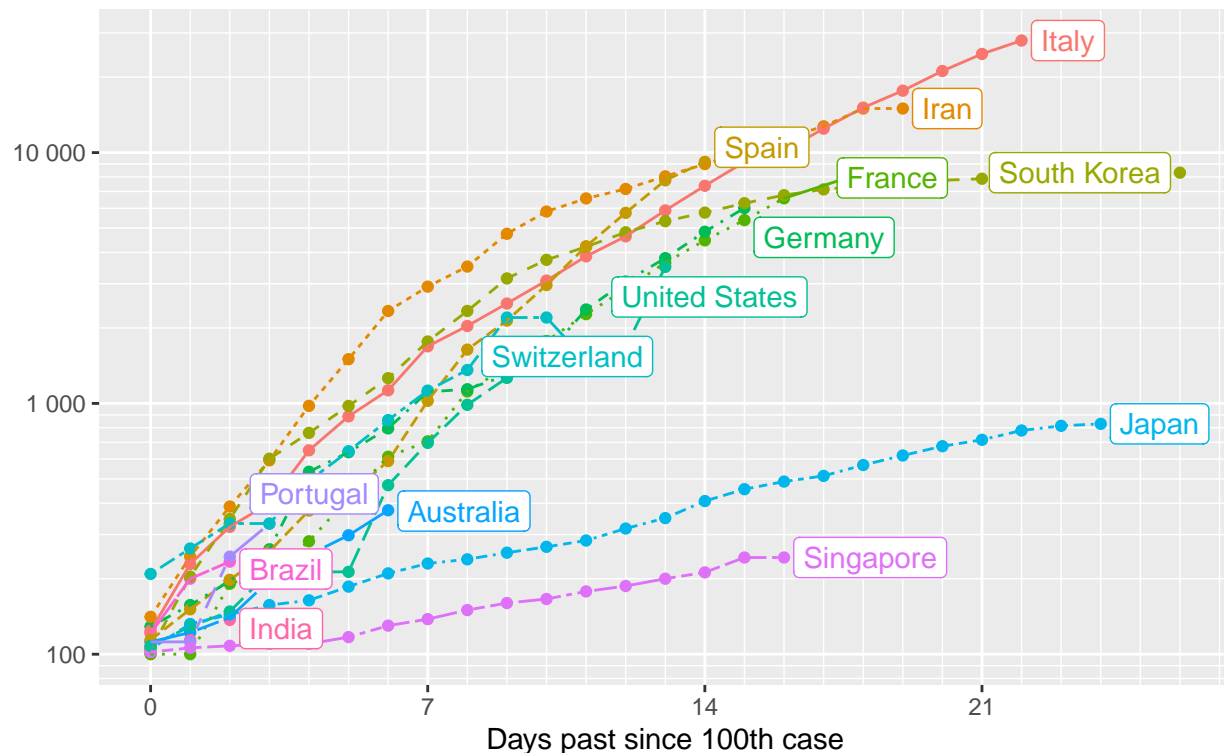
#countries <- df$location %>% unique
conts <- "Europe"

make_plot(df,
  sel_country = countries,
  sel_continent = NA,
  threshold = 100,
  sel_category = "cumulative cases")

```

## Evolution of novel coronavirus in selected countries

cumulative cases (log scale)



```
df_dates <- read_csv("../data/threshold_dates.csv")
```

```
## Parsed with column specification:
## cols(
##   location = col_character(),
##   category = col_character(),
##   th_1 = col_date(format = ""),
##   th_5 = col_date(format = ""),
##   th_10 = col_date(format = ""),
##   th_25 = col_date(format = ""),
##   th_50 = col_date(format = ""),
##   th_100 = col_date(format = "")
## )
```

```
df_restaurant <- read_csv("../data/opentable_state_of_industry_data.csv") %>%
  pivot_longer( -c(Type, Name), names_to = "Date", values_to = "Value") %>%
  mutate(Date = as.Date(Date, format="%m/%d"))
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Type = col_character(),
##   Name = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

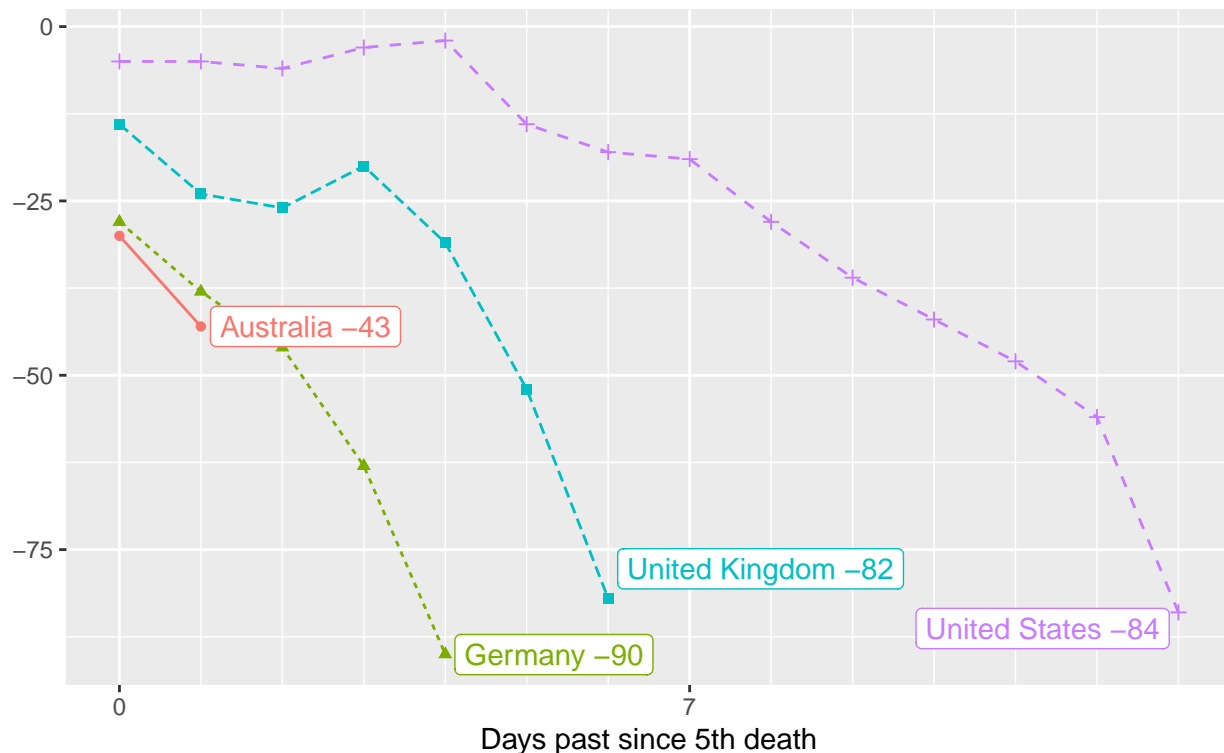
```
df_join <- full_join(df_restaurant, df_dates,
                     by=c(Name="location")) %>%
  filter(Name != "Global", category == "cumulative deaths") %>%
  mutate(xx = as.integer(Date - th_5)) %>%
  group_by(Name) %>%
  mutate(label = ifelse(xx == max(xx),
                        paste(as.character(Name), Value),
                        NA_character_)) %>%
  filter(xx >= 0)

df_join %>%
  ggplot(aes(x=xx, y=Value, colour=Name)) +
  geom_line(aes(linetype=Name)) +
  geom_point(aes(shape=Name)) +

  ggtitle("Reastaurant Reservations (% Year-over-year variation)",
          "OpenTable data (https://www.opentable.com/state-of-industry)") +
  theme(legend.position = "none",
        axis.title.y = element_blank()) +
  scale_x_continuous(
    name = paste0("Days past since ", "5th ", "death"),
    breaks = seq(0, 1e5, 7), minor_breaks = 1:1e3) +
  ggrepel::geom_label_repel(
    aes(label = label), nudge_x = 1, na.rm = TRUE)
```

## Reastaurant Reservations (% Year-over-year variation)

OpenTable data (<https://www.opentable.com/state-of-industry>)



```

df_join <- full_join(df_restaurant, df_dates,
                     by=c(Name="location")) %>%
  filter(Name != "Global", category == "cumulative cases") %>%
  mutate(xx = as.integer(Date - th_50)) %>%
  group_by(Name) %>%
  mutate(label = ifelse(xx == max(xx),
                        paste(as.character(Name), Value),
                        NA_character_)) %>%

  filter(xx >= 0)

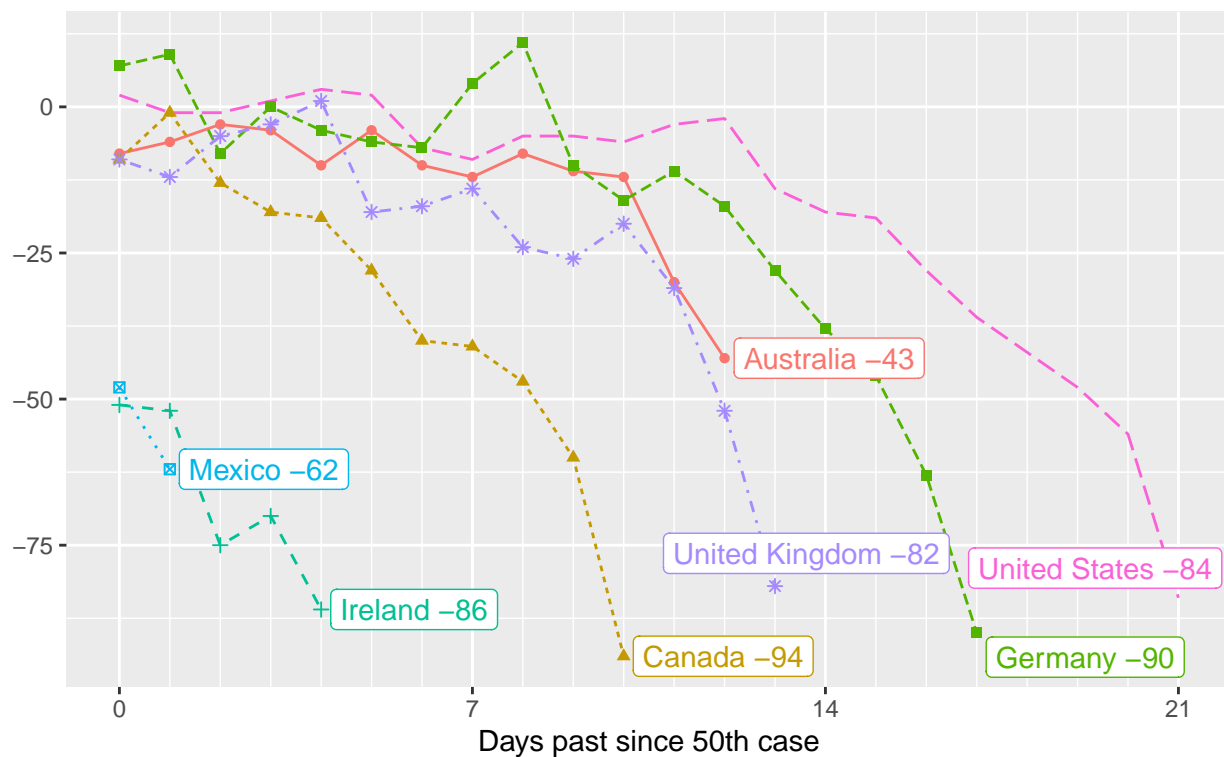
df_join %>%
  ggplot(aes(x=xx, y=Value, colour=Name)) +
  geom_line(aes(linetype=Name)) +
  geom_point(aes(shape=Name)) +

  ggtitle("Reastaurant Reservations (% Year-over-year variation)",
          "OpenTable data (https://www.opentable.com/state-of-industry)") +
  theme(legend.position = "none",
        axis.title.y = element_blank()) +
  scale_x_continuous(
    name = paste0("Days past since ", "50th ", "case"),
    breaks = seq(0, 1e5, 7), minor_breaks = 1:1e3) +
  ggrepel::geom_label_repel(
    aes(label = label), nudge_x = 1, na.rm = TRUE)

```

## Reastaurant Reservations (% Year-over-year variation)

OpenTable data (<https://www.opentable.com/state-of-industry>)





```

df_de <- read_csv("../data/COVID_19_de_wiki.csv")

## Parsed with column specification:
## cols(
##   state = col_character(),
##   Date = col_date(format = ""),
##   value = col_double()
## )

update <- max(df_de$Date)
threshold <- 25
df_de %>%
  group_by(state) %>%
  filter(value>=threshold) %>%
  mutate(date_shift=0:(n()-1)) %>% # generate date shifted after threshold
  ungroup(state) %>%
  arrange(date_shift) %>%
  # mutate for legend ordering and ggrepel postioning
  mutate(state=fct_reorder2(state, date_shift, value),
         # for ggrepel
         label = ifelse(Date == max(Date),
                        as.character(state),
                        NA_character_)) %>%

  # plot
  ggplot(aes(x=date_shift, y=value, colour=state)) +
  # geoms
  geom_point() +
  geom_line(aes()) +
  #scale_color_grey(start = 0, end = .3) +

  # ggrepel
  ggrepel::geom_label_repel(
    aes(label = label), nudge_x = 5, na.rm = TRUE) +

  # legend and scales
  scale_y_log10(
    name = paste(sel_category, "(log scale)"),
    labels=fun_dot, breaks = 10^(0:9), minor_breaks=log10_minor_break()) +
  scale_x_continuous(
    name = paste0("Days past since ", threshold, "th ", "case"),
    breaks = seq(0, 1e5, 7), minor_breaks = 1:1e3) +

  theme(legend.position = "none", axis.title.y = element_blank()) +
  #title
  ggtitle(paste("Evolution of COVID-19 cases in Germany"),
          subtitle = paste0("Cumulative cases (log scale, updated: ", update, ")"))

```

Evolution of COVID-19 cases in Germany  
 Cumulative cases (log scale, updated: 2020-03-20)

