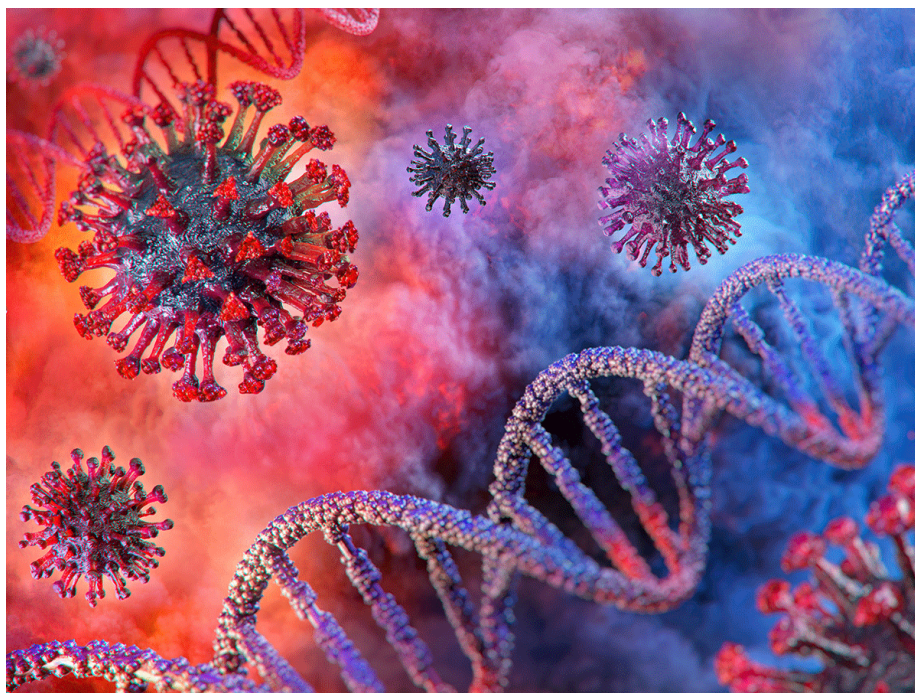


Data Science - COVID-19

Carlos Utrilla Guerrero

3/30/2020



Analysis coronavirus disease (COVID-19).

This is an R Markdown document. It is intended to publicly illustrate how R statistics can help you out to output data science pipeline.

About this data

It changes rapidly

It doesn't include all cases

Confirmed cases aren't all cases. They only include people who tested positive.
Testing rules and availability vary by country.

Data Repository: Johns Hopkins University.

```
# This is an analysis report of the Novel Coronavirus (COVID-19)
# Aim for data processing, visualisation and statistics
# Source code: http://yanchang.rdatamining.com/
# set directory
# Data Source: 2019 Data Repository https://github.com/CSSEGISandData/COVID-19
# R Packages:
library(magrittr) # pipeline operations
library(lubridate) # date operation
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##   date
```

```
library(tidyverse) # data science pips
```

```
## -- Attaching packages -----
```

```
## v ggplot2 3.3.0      v purrr   0.3.3
## v tibble  2.1.3      v dplyr   0.8.5
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0
```

```
## -- Conflicts -----
```

```
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()       masks base::date()
## x tidyr::extract()        masks magrittr::extract()
## x dplyr::filter()         masks stats::filter()
## x lubridate::intersect()  masks base::intersect()
## x dplyr::lag()            masks stats::lag()
## x purrr::set_names()      masks magrittr::set_names()
## x lubridate::setdiff()    masks base::setdiff()
## x lubridate::union()      masks base::union()
```

```
library(gridExtra) # grid based plots
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##      combine
```

```
library(dplyr)
library(leaflet)
library(ggforce)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##      group_rows
```

```
# Loading data
# At first, three CSV files, are downloaded and saved as local files
# and then loaded into R
# source data files changes everytime
filenames <- c('time_series_covid19_confirmed_global.csv',
               'time_series_covid19_deaths_global.csv',
               'time_series_covid19_recovered_global.csv')
url.path <- paste0('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/',
                  'master/csse_covid_19_data/csse_covid_19_time_series/')

#download files to local folder
download <- function(filename) {
  url <- file.path(url.path, filename)
  dest <- file.path('./data', filename)
  download.file(url, dest)
}
bin <- lapply(filenames, download)

# load data into R
data.confirmed.original <- read.csv('./data/time_series_covid19_confirmed_global.csv')
data.deaths.original <- read.csv('./data/time_series_covid19_deaths_global.csv')
data.recovered.original <- read.csv('./data/time_series_covid19_recovered_global.csv')
```

```
# check dimension of data confirmed
dim(data.confirmed.original)
```

```
## [1] 253 72
```

Below we check the time frame of data set

```
# check time frame of the data
n.col <- ncol(data.confirmed.original) # 58 variables
# get dates from column names
dates <- names(data.confirmed.original)[5:n.col] %>% substr(2,8) %>% mdy()
range(dates)
```

```
## [1] "2020-01-22" "2020-03-29"
```

```
min.date <- min(dates)
max.date <- max(dates)
max.date.txt <- max.date %>% format('%d %b %Y')
min.date.txt <- min.date %>% format('%d %b %Y')
# last update on 26 March 2020 max.date
```

```
# Data Preparation steps:
# 1.From wide to long format
# 2.Aggregate by country
# 3. merge into a single dataset
# cleaning and transformation
cleanData <- function(data) {
  ## remove some columns
  data %<>% select(-c(Province.State, Lat, Long)) %>% rename(country=Country.Region)
  ## convert from wide to long format
  data %<>% gather(key=date, value=count, -country)
  ## convert from character to date
  data %<>% mutate(date = date %>% substr(2,8) %>% mdy())
  ## aggregate by country
  data %<>% group_by(country, date) %>% summarise(count=sum(count)) %>% as.data.frame()
  return(data)
}
# clean the three datasets
data.confirmed <- data.confirmed.original %>% cleanData() %>% rename(confirmed=count)
data.deaths <- data.deaths.original %>% cleanData() %>% rename(deaths=count)
data.recovered <- data.recovered.original %>% cleanData() %>% rename(recovered=count)

# merge above 3 datasets into one, by country and date
```

```
data <- data.confirmed %>% merge(data.deaths, all = T) %>% merge(data.recovered, all = T)

# countries/regions with confirmed cases (excl cruise ships)
countries <- data %>% pull(country) %>% setdiff('Cruise Ship')

# last 10 records when it first broke out in Spain
data %>% filter(country == 'Spain') %>% tail(10)
```

```
##      country      date confirmed deaths recovered
## 59   Spain 2020-03-20     20410     1043      1588
## 60   Spain 2020-03-21     25374     1375      2125
## 61   Spain 2020-03-22     28768     1772      2575
## 62   Spain 2020-03-23     35136     2311      2575
## 63   Spain 2020-03-24     39885     2808      3794
## 64   Spain 2020-03-25     49515     3647      5367
## 65   Spain 2020-03-26     57786     4365      7015
## 66   Spain 2020-03-27     65719     5138      9357
## 67   Spain 2020-03-28     73235     5982     12285
## 68   Spain 2020-03-29     80110     6803     14709
```

```
# counts for worldwide
data.world <- data %>% group_by(date) %>%
  summarise(country='World',
             confirmed=sum(confirmed, na.rm = T),
             deaths=sum(deaths, na.rm = T),
             recovered=sum(recovered, na.rm = T))

data %<>% rbind(data.world)

# current confirmed cases
data %<>% mutate(remaining.confirmed = confirmed - deaths - recovered)
```

```
# Visualisation
# After preparing the data, we portrait it in various graphs

# TOP Ten Countries
# ranking by confirmed cases
data.latest.all <- data %>% filter(date == max(date)) %>%
  select(country, date,
         confirmed, confirmed.new, remaining.confirmed, recovered, deaths.new, deaths, deaths.new)

# top 20 countries incl 11 World
k<- 20
top.countries <- data.latest.all %>% filter(ranking <= k+1) %>%
  arrange(ranking) %>% pull(country) %>% as.character()
top.countries %>% setdiff('World') %>% print()
```

```
## [1] "US" "Italy" "China" "Spain"
## [5] "Germany" "France" "Iran" "United Kingdom"
## [9] "Switzerland" "Netherlands" "Belgium" "Korea, South"
## [13] "Turkey" "Austria" "Canada" "Portugal"
## [17] "Norway" "Brazil" "Israel" "Australia"
```

```
names(data.latest.all)
```

```
## [1] "country" "date" "confirmed"
## [4] "confirmed.new" "remaining.confirmed" "recovered"
## [7] "deaths.new" "deaths" "death.rate"
## [10] "ranking"
```

```
## add 'Others'
top.countries %<>% c('Others')
## put all others in a single group of 'Others'
data.latest <- data.latest.all %>% filter(!is.na(country)) %>%
mutate(country=ifelse(ranking <= k + 1, as.character(country), 'Others')) %>%
mutate(country=country %>% factor(levels=c(top.countries)))

data.latest %<>% group_by(country) %>%
  summarise(confirmed=sum(confirmed), confirmed.new=sum(confirmed.new), remaining.confirmed=
    mutate(death.rate=(100*deaths/confirmed) %>% round(1))
data.latest %<>% select(c(country, confirmed, deaths,death.rate, confirmed.new, deaths.new,1
data.latest %>% mutate(death.rate=death.rate %>% format(nsmall=1)) %>% paste0('%')) %>% kable
```

Worldmap

```
x <- data.confirmed.original
x$confirmed <- x[, ncol(x)]
x %>% select(c(Country.Region, Province.State, Lat, Long, confirmed)) %>%
  mutate(txt=paste0(Country.Region, '-', Province.State, ':', confirmed))
```

```
## Country.Region Province.State Lat
## 1 Afghanistan 33.000000
## 2 Albania 41.153300
## 3 Algeria 28.033900
## 4 Andorra 42.506300
## 5 Angola -11.202700
## 6 Antigua and Barbuda 17.060800
## 7 Argentina -38.416100
## 8 Armenia 40.069100
## 9 Australia Australian Capital Territory -35.473500
```

Table 1: Cases in Top 20 Countries - 29 Mar 2020.

	country	confirmed	deaths	death.rate	confirmed.new	deaths.new	remaining.confirmed
1	World	720,117	33,925	4.7%	59,411	3,273	537,110
2	US	140,886	2,467	1.8%	19,408	441	135,754
3	Italy	97,689	10,779	11.0%	5,217	756	73,880
4	China	82,122	3,304	4.0%	123	5	3,236
5	Spain	80,110	6,803	8.5%	6,875	821	58,598
6	Germany	62,095	533	0.9%	4,400	100	52,351
7	France	40,708	2,611	6.4%	2,603	294	30,871
8	Iran	38,309	2,640	6.9%	2,901	123	23,278
9	United Kingdom	19,780	1,231	6.2%	2,468	210	18,398
10	Switzerland	14,829	300	2.0%	753	36	12,934
11	Netherlands	10,930	772	7.1%	1,111	132	9,905
12	Belgium	10,836	431	4.0%	1,702	78	9,046
13	Korea, South	9,583	152	1.6%	105	8	4,398
14	Turkey	9,217	131	1.4%	1,815	23	8,981
15	Austria	8,788	86	1.0%	517	18	8,223
16	Canada	6,280	64	1.0%	704	3	5,750
17	Portugal	5,962	119	2.0%	792	19	5,800
18	Norway	4,284	25	0.6%	269	2	4,252
19	Brazil	4,256	136	3.2%	352	25	4,114
20	Israel	4,247	15	0.4%	628	3	4,100
21	Australia	3,984	16	0.4%	344	2	3,724
22	Others	65,222	1,310	2.0%	6,324	174	59,517

## 10	Australia	New South Wales	-33.868800
## 11	Australia	Northern Territory	-12.463400
## 12	Australia	Queensland	-28.016700
## 13	Australia	South Australia	-34.928500
## 14	Australia	Tasmania	-41.454500
## 15	Australia	Victoria	-37.813600
## 16	Australia	Western Australia	-31.950500
## 17	Austria		47.516200
## 18	Azerbaijan		40.143100
## 19	Bahamas		25.034300
## 20	Bahrain		26.027500
## 21	Bangladesh		23.685000
## 22	Barbados		13.193900
## 23	Belarus		53.709800
## 24	Belgium		50.833300
## 25	Benin		9.307700
## 26	Bhutan		27.514200
## 27	Bolivia		-16.290200
## 28	Bosnia and Herzegovina		43.915900
## 29	Brazil		-14.235000
## 30	Brunei		4.535300
## 31	Bulgaria		42.733900

## 32	Burkina Faso		12.238300
## 33	Cabo Verde		16.538800
## 34	Cambodia		11.550000
## 35	Cameroon		3.848000
## 36	Canada	Alberta	53.933300
## 37	Canada	British Columbia	49.282700
## 38	Canada	Grand Princess	37.648900
## 39	Canada	Manitoba	53.760900
## 40	Canada	New Brunswick	46.565300
## 41	Canada	Newfoundland and Labrador	53.135500
## 42	Canada	Nova Scotia	44.682000
## 43	Canada	Ontario	51.253800
## 44	Canada	Prince Edward Island	46.510700
## 45	Canada	Quebec	52.939900
## 46	Canada	Saskatchewan	52.939900
## 47	Central African Republic		6.611100
## 48	Chad		15.454200
## 49	Chile		-35.675100
## 50	China	Anhui	31.825700
## 51	China	Beijing	40.182400
## 52	China	Chongqing	30.057200
## 53	China	Fujian	26.078900
## 54	China	Gansu	37.809900
## 55	China	Guangdong	23.341700
## 56	China	Guangxi	23.829800
## 57	China	Guizhou	26.815400
## 58	China	Hainan	19.195900
## 59	China	Hebei	39.549000
## 60	China	Heilongjiang	47.862000
## 61	China	Henan	33.882000
## 62	China	Hong Kong	22.300000
## 63	China	Hubei	30.975600
## 64	China	Hunan	27.610400
## 65	China	Inner Mongolia	44.093500
## 66	China	Jiangsu	32.971100
## 67	China	Jiangxi	27.614000
## 68	China	Jilin	43.666100
## 69	China	Liaoning	41.295600
## 70	China	Macau	22.166700
## 71	China	Ningxia	37.269200
## 72	China	Qinghai	35.745200
## 73	China	Shaanxi	35.191700
## 74	China	Shandong	36.342700
## 75	China	Shanghai	31.202000
## 76	China	Shanxi	37.577700
## 77	China	Sichuan	30.617100

## 78	China	Tianjin	39.305400
## 79	China	Tibet	31.692700
## 80	China	Xinjiang	41.112900
## 81	China	Yunnan	24.974000
## 82	China	Zhejiang	29.183200
## 83	Colombia		4.570900
## 84	Congo (Brazzaville)		-4.038300
## 85	Congo (Kinshasa)		-4.038300
## 86	Costa Rica		9.748900
## 87	Cote d'Ivoire		7.540000
## 88	Croatia		45.100000
## 89	Diamond Princess		0.000000
## 90	Cuba		22.000000
## 91	Cyprus		35.126400
## 92	Czechia		49.817500
## 93	Denmark	Faroe Islands	61.892600
## 94	Denmark	Greenland	71.706900
## 95	Denmark		56.263900
## 96	Djibouti		11.825100
## 97	Dominican Republic		18.735700
## 98	Ecuador		-1.831200
## 99	Egypt		26.000000
## 100	El Salvador		13.794200
## 101	Equatorial Guinea		1.500000
## 102	Eritrea		15.179400
## 103	Estonia		58.595300
## 104	Eswatini		-26.522500
## 105	Ethiopia		9.145000
## 106	Fiji		-17.713400
## 107	Finland		64.000000
## 108	France	French Guiana	3.933900
## 109	France	French Polynesia	-17.679700
## 110	France	Guadeloupe	16.250000
## 111	France	Mayotte	-12.827500
## 112	France	New Caledonia	-20.904300
## 113	France	Reunion	-21.135100
## 114	France	Saint Barthelemy	17.900000
## 115	France	St Martin	18.070800
## 116	France	Martinique	14.641500
## 117	France		46.227600
## 118	Gabon		-0.803700
## 119	Gambia		13.443200
## 120	Georgia		42.315400
## 121	Germany		51.000000
## 122	Ghana		7.946500
## 123	Greece		39.074200

## 124	Guatemala	15.783500
## 125	Guinea	9.945600
## 126	Guyana	5.000000
## 127	Haiti	18.971200
## 128	Holy See	41.902900
## 129	Honduras	15.200000
## 130	Hungary	47.162500
## 131	Iceland	64.963100
## 132	India	21.000000
## 133	Indonesia	-0.789300
## 134	Iran	32.000000
## 135	Iraq	33.000000
## 136	Ireland	53.142400
## 137	Israel	31.000000
## 138	Italy	43.000000
## 139	Jamaica	18.109600
## 140	Japan	36.000000
## 141	Jordan	31.240000
## 142	Kazakhstan	48.019600
## 143	Kenya	-0.023600
## 144	Korea, South	36.000000
## 145	Kuwait	29.500000
## 146	Kyrgyzstan	41.204400
## 147	Latvia	56.879600
## 148	Lebanon	33.854700
## 149	Liberia	6.428100
## 150	Liechtenstein	47.140000
## 151	Lithuania	55.169400
## 152	Luxembourg	49.815300
## 153	Madagascar	-18.766900
## 154	Malaysia	2.500000
## 155	Maldives	3.202800
## 156	Malta	35.937500
## 157	Mauritania	21.007900
## 158	Mauritius	-20.200000
## 159	Mexico	23.634500
## 160	Moldova	47.411600
## 161	Monaco	43.733300
## 162	Mongolia	46.862500
## 163	Montenegro	42.500000
## 164	Morocco	31.791700
## 165	Namibia	-22.957600
## 166	Nepal	28.166700
## 167	Netherlands	Aruba 12.518600
## 168	Netherlands	Curacao 12.169600
## 169	Netherlands	Sint Maarten 18.042500

## 170	Netherlands	52.132600
## 171	New Zealand	-40.900600
## 172	Nicaragua	12.865400
## 173	Niger	17.607800
## 174	Nigeria	9.082000
## 175	North Macedonia	41.608600
## 176	Norway	60.472000
## 177	Oman	21.000000
## 178	Pakistan	30.375300
## 179	Panama	8.538000
## 180	Papua New Guinea	-6.315000
## 181	Paraguay	-23.442500
## 182	Peru	-9.190000
## 183	Philippines	13.000000
## 184	Poland	51.919400
## 185	Portugal	39.399900
## 186	Qatar	25.354800
## 187	Romania	45.943200
## 188	Russia	60.000000
## 189	Rwanda	-1.940300
## 190	Saint Lucia	13.909400
## 191	Saint Vincent and the Grenadines	12.984300
## 192	San Marino	43.942400
## 193	Saudi Arabia	24.000000
## 194	Senegal	14.497400
## 195	Serbia	44.016500
## 196	Seychelles	-4.679600
## 197	Singapore	1.283300
## 198	Slovakia	48.669000
## 199	Slovenia	46.151200
## 200	Somalia	5.152100
## 201	South Africa	-30.559500
## 202	Spain	40.000000
## 203	Sri Lanka	7.000000
## 204	Sudan	12.862800
## 205	Suriname	3.919300
## 206	Sweden	63.000000
## 207	Switzerland	46.818200
## 208	Taiwan*	23.700000
## 209	Tanzania	-6.369000
## 210	Thailand	15.000000
## 211	Togo	8.619500
## 212	Trinidad and Tobago	10.691800
## 213	Tunisia	34.000000
## 214	Turkey	38.963700
## 215	Uganda	1.000000

## 216	Ukraine	48.379400
## 217	United Arab Emirates	24.000000
## 218	United Kingdom	Bermuda 32.307800
## 219	United Kingdom	Cayman Islands 19.313300
## 220	United Kingdom	Channel Islands 49.372300
## 221	United Kingdom	Gibraltar 36.140800
## 222	United Kingdom	Isle of Man 54.236100
## 223	United Kingdom	Montserrat 16.742500
## 224	United Kingdom	55.378100
## 225	Uruguay	-32.522800
## 226	US	37.090200
## 227	Uzbekistan	41.377500
## 228	Venezuela	6.423800
## 229	Vietnam	16.000000
## 230	Zambia	-15.416700
## 231	Zimbabwe	-20.000000
## 232	Canada	Diamond Princess 0.000000
## 233	Dominica	15.415000
## 234	Grenada	12.116500
## 235	Mozambique	-18.665695
## 236	Syria	34.802075
## 237	Timor-Leste	-8.874217
## 238	Belize	13.193900
## 239	Canada	Recovered 0.000000
## 240	Laos	19.856270
## 241	Libya	26.335100
## 242	West Bank and Gaza	31.952200
## 243	Guinea-Bissau	11.803700
## 244	Mali	17.570692
## 245	Saint Kitts and Nevis	17.357822
## 246	Canada	Northwest Territories 64.825500
## 247	Canada	Yukon 64.282300
## 248	Kosovo	42.602636
## 249	Burma	21.916200
## 250	United Kingdom	Anguilla 18.220600
## 251	United Kingdom	British Virgin Islands 18.420700
## 252	United Kingdom	Turks and Caicos Islands 21.694000
## 253	MS Zaandam	0.000000
##	Long confirmed	txt
## 1	65.000000	120 Afghanistan-:120
## 2	20.168300	212 Albania-:212
## 3	1.659600	511 Algeria-:511
## 4	1.521800	334 Andorra-:334
## 5	17.873900	7 Angola-:7
## 6	-61.796400	7 Antigua and Barbuda-:7
## 7	-63.616700	745 Argentina-:745

## 8	45.038200	424	Armenia-:424
## 9	149.012400	77	Australia-Australian Capital Territory:77
## 10	151.209300	1791	Australia-New South Wales:1791
## 11	130.845600	15	Australia-Northern Territory:15
## 12	153.400000	656	Australia-Queensland:656
## 13	138.600700	299	Australia-South Australia:299
## 14	145.970700	66	Australia-Tasmania:66
## 15	144.963100	769	Australia-Victoria:769
## 16	115.860500	311	Australia-Western Australia:311
## 17	14.550100	8788	Austria-:8788
## 18	47.576900	209	Azerbaijan-:209
## 19	-77.396300	11	Bahamas-:11
## 20	50.550000	499	Bahrain-:499
## 21	90.356300	48	Bangladesh-:48
## 22	-59.543200	33	Barbados-:33
## 23	27.953400	94	Belarus-:94
## 24	4.000000	10836	Belgium-:10836
## 25	2.315800	6	Benin-:6
## 26	90.433600	4	Bhutan-:4
## 27	-63.588700	81	Bolivia-:81
## 28	17.679100	323	Bosnia and Herzegovina-:323
## 29	-51.925300	4256	Brazil-:4256
## 30	114.727700	126	Brunei-:126
## 31	25.485800	346	Bulgaria-:346
## 32	-1.561600	222	Burkina Faso-:222
## 33	-23.041800	6	Cabo Verde-:6
## 34	104.916700	103	Cambodia-:103
## 35	11.502100	139	Cameroon-:139
## 36	-116.576500	621	Canada-Alberta:621
## 37	-123.120700	884	Canada-British Columbia:884
## 38	-122.665500	13	Canada-Grand Princess:13
## 39	-98.813900	72	Canada-Manitoba:72
## 40	-66.461900	66	Canada-New Brunswick:66
## 41	-57.660400	135	Canada-Newfoundland and Labrador:135
## 42	-63.744300	122	Canada-Nova Scotia:122
## 43	-85.323200	1355	Canada-Ontario:1355
## 44	-63.416800	11	Canada-Prince Edward Island:11
## 45	-73.549100	2840	Canada-Quebec:2840
## 46	-106.450900	156	Canada-Saskatchewan:156
## 47	20.939400	3	Central African Republic-:3
## 48	18.732200	3	Chad-:3
## 49	-71.543000	2139	Chile-:2139
## 50	117.226400	990	China-Anhui:990
## 51	116.414200	577	China-Beijing:577
## 52	107.874000	579	China-Chongqing:579
## 53	117.987400	338	China-Fujian:338

## 54	101.058300	138	China-Gansu:138
## 55	113.424400	1475	China-Guangdong:1475
## 56	108.788100	254	China-Guangxi:254
## 57	106.874800	146	China-Guizhou:146
## 58	109.745300	168	China-Hainan:168
## 59	116.130600	319	China-Hebei:319
## 60	127.761500	484	China-Heilongjiang:484
## 61	113.614000	1276	China-Henan:1276
## 62	114.200000	641	China-Hong Kong:641
## 63	112.270700	67801	China-Hubei:67801
## 64	111.708800	1018	China-Hunan:1018
## 65	113.944800	95	China-Inner Mongolia:95
## 66	119.455000	644	China-Jiangsu:644
## 67	115.722100	937	China-Jiangxi:937
## 68	126.192300	98	China-Jilin:98
## 69	122.608500	134	China-Liaoning:134
## 70	113.550000	37	China-Macau:37
## 71	106.165500	75	China-Ningxia:75
## 72	95.995600	18	China-Qinghai:18
## 73	108.870100	253	China-Shaanxi:253
## 74	118.149800	772	China-Shandong:772
## 75	121.449100	492	China-Shanghai:492
## 76	112.292200	136	China-Shanxi:136
## 77	102.710300	550	China-Sichuan:550
## 78	117.323000	166	China-Tianjin:166
## 79	88.092400	1	China-Tibet:1
## 80	85.240100	76	China-Xinjiang:76
## 81	101.487000	180	China-Yunnan:180
## 82	120.093400	1254	China-Zhejiang:1254
## 83	-74.297300	702	Colombia-:702
## 84	21.758700	19	Congo (Brazzaville)-:19
## 85	21.758700	65	Congo (Kinshasa)-:65
## 86	-83.753400	314	Costa Rica-:314
## 87	-5.547100	165	Cote d'Ivoire-:165
## 88	15.200000	713	Croatia-:713
## 89	0.000000	712	Diamond Princess-:712
## 90	-80.000000	139	Cuba-:139
## 91	33.429900	214	Cyprus-:214
## 92	15.473000	2817	Czechia-:2817
## 93	-6.911800	159	Denmark-Faroe Islands:159
## 94	-42.604300	10	Denmark-Greenland:10
## 95	9.501800	2395	Denmark-:2395
## 96	42.590300	18	Djibouti-:18
## 97	-70.162700	859	Dominican Republic-:859
## 98	-78.183400	1924	Ecuador-:1924
## 99	30.000000	609	Egypt-:609

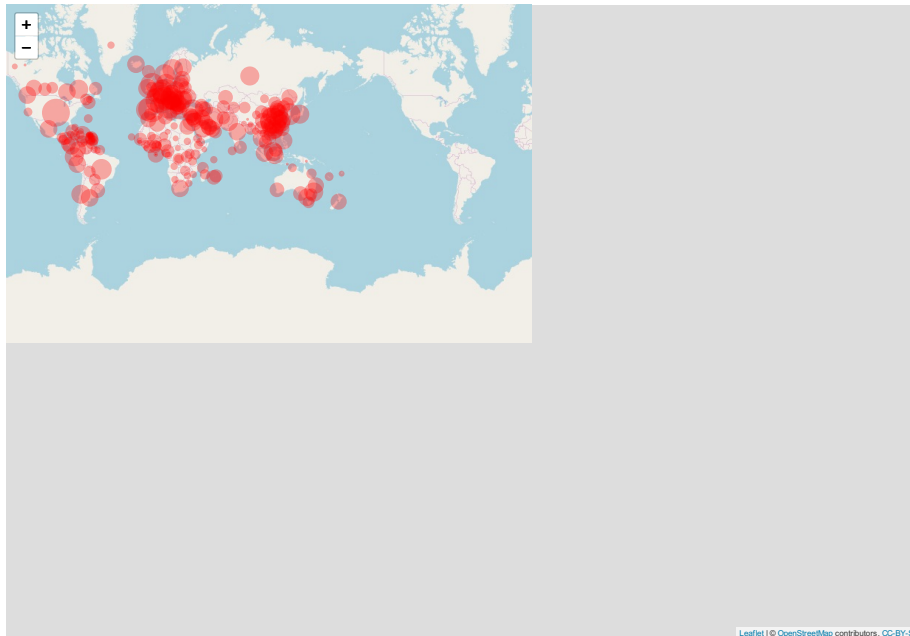
## 100	-88.896500	24	El Salvador-:24
## 101	10.000000	12	Equatorial Guinea-:12
## 102	39.782300	12	Eritrea-:12
## 103	25.013600	679	Estonia-:679
## 104	31.465900	9	Eswatini-:9
## 105	40.489700	21	Ethiopia-:21
## 106	178.065000	5	Fiji-:5
## 107	26.000000	1240	Finland-:1240
## 108	-53.125800	28	France-French Guiana-:28
## 109	149.406800	30	France-French Polynesia-:30
## 110	-61.583300	106	France-Guadeloupe-:106
## 111	45.166200	63	France-Mayotte-:63
## 112	165.618000	15	France-New Caledonia-:15
## 113	55.247100	183	France-Reunion-:183
## 114	-62.833300	5	France-Saint Barthelemy-:5
## 115	-63.050100	11	France-St Martin-:11
## 116	-61.024200	93	France-Martinique-:93
## 117	2.213700	40174	France-:40174
## 118	11.609400	7	Gabon-:7
## 119	-15.310100	4	Gambia-:4
## 120	43.356900	91	Georgia-:91
## 121	9.000000	62095	Germany-:62095
## 122	-1.023200	152	Ghana-:152
## 123	21.824300	1156	Greece-:1156
## 124	-90.230800	34	Guatemala-:34
## 125	-9.696600	16	Guinea-:16
## 126	-58.750000	8	Guyana-:8
## 127	-72.285200	15	Haiti-:15
## 128	12.453400	6	Holy See-:6
## 129	-86.241900	110	Honduras-:110
## 130	19.503300	408	Hungary-:408
## 131	-19.020800	1020	Iceland-:1020
## 132	78.000000	1024	India-:1024
## 133	113.921300	1285	Indonesia-:1285
## 134	53.000000	38309	Iran-:38309
## 135	44.000000	547	Iraq-:547
## 136	-7.692100	2615	Ireland-:2615
## 137	35.000000	4247	Israel-:4247
## 138	12.000000	97689	Italy-:97689
## 139	-77.297500	32	Jamaica-:32
## 140	138.000000	1866	Japan-:1866
## 141	36.510000	259	Jordan-:259
## 142	66.923700	284	Kazakhstan-:284
## 143	37.906200	42	Kenya-:42
## 144	128.000000	9583	Korea, South-:9583
## 145	47.750000	255	Kuwait-:255

## 146	74.766100	84	Kyrgyzstan-:84
## 147	24.603200	347	Latvia-:347
## 148	35.862300	438	Lebanon-:438
## 149	-9.429500	3	Liberia-:3
## 150	9.550000	56	Liechtenstein-:56
## 151	23.881300	460	Lithuania-:460
## 152	6.129600	1950	Luxembourg-:1950
## 153	46.869100	39	Madagascar-:39
## 154	112.500000	2470	Malaysia-:2470
## 155	73.220700	17	Maldives-:17
## 156	14.375400	151	Malta-:151
## 157	10.940800	5	Mauritania-:5
## 158	57.500000	107	Mauritius-:107
## 159	-102.552800	848	Mexico-:848
## 160	28.369900	263	Moldova-:263
## 161	7.416700	46	Monaco-:46
## 162	103.846700	12	Mongolia-:12
## 163	19.300000	85	Montenegro-:85
## 164	-7.092600	479	Morocco-:479
## 165	18.490400	11	Namibia-:11
## 166	84.250000	5	Nepal-:5
## 167	-70.035800	50	Netherlands-Aruba:50
## 168	-68.990000	8	Netherlands-Curacao:8
## 169	-63.054800	6	Netherlands-Sint Maarten:6
## 170	5.291300	10866	Netherlands-:10866
## 171	174.886000	514	New Zealand-:514
## 172	-85.207200	4	Nicaragua-:4
## 173	8.081700	18	Niger-:18
## 174	8.675300	111	Nigeria-:111
## 175	21.745300	259	North Macedonia-:259
## 176	8.468900	4284	Norway-:4284
## 177	57.000000	167	Oman-:167
## 178	69.345100	1597	Pakistan-:1597
## 179	-80.782100	901	Panama-:901
## 180	143.955500	1	Papua New Guinea-:1
## 181	-58.443800	59	Paraguay-:59
## 182	-75.015200	852	Peru-:852
## 183	122.000000	1418	Philippines-:1418
## 184	19.145100	1862	Poland-:1862
## 185	-8.224500	5962	Portugal-:5962
## 186	51.183900	634	Qatar-:634
## 187	24.966800	1815	Romania-:1815
## 188	90.000000	1534	Russia-:1534
## 189	29.873900	70	Rwanda-:70
## 190	-60.978900	9	Saint Lucia-:9
## 191	-61.287200	1	Saint Vincent and the Grenadines-:1

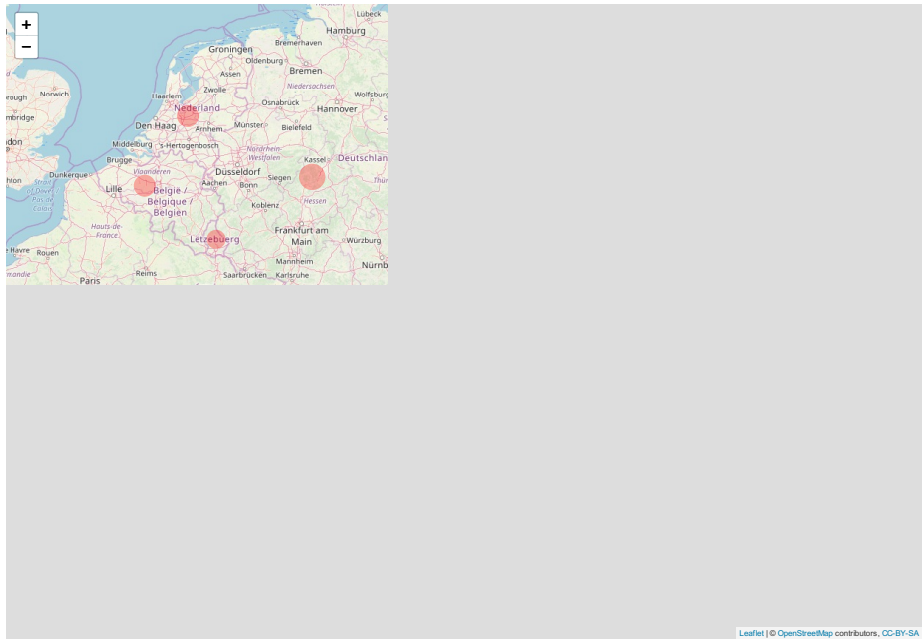
## 192	12.457800	224	San Marino-:224
## 193	45.000000	1299	Saudi Arabia-:1299
## 194	-14.452400	142	Senegal-:142
## 195	21.005900	741	Serbia-:741
## 196	55.492000	8	Seychelles-:8
## 197	103.833300	844	Singapore-:844
## 198	19.699000	314	Slovakia-:314
## 199	14.995500	730	Slovenia-:730
## 200	46.199600	3	Somalia-:3
## 201	22.937500	1280	South Africa-:1280
## 202	-4.000000	80110	Spain-:80110
## 203	81.000000	117	Sri Lanka-:117
## 204	30.217600	6	Sudan-:6
## 205	-56.027800	8	Suriname-:8
## 206	16.000000	3700	Sweden-:3700
## 207	8.227500	14829	Switzerland-:14829
## 208	121.000000	298	Taiwan*-:298
## 209	34.888800	14	Tanzania-:14
## 210	101.000000	1388	Thailand-:1388
## 211	0.824800	25	Togo-:25
## 212	-61.222500	78	Trinidad and Tobago-:78
## 213	9.000000	312	Tunisia-:312
## 214	35.243300	9217	Turkey-:9217
## 215	32.000000	33	Uganda-:33
## 216	31.165600	475	Ukraine-:475
## 217	54.000000	570	United Arab Emirates-:570
## 218	-64.750500	22	United Kingdom-Bermuda:22
## 219	-81.254600	8	United Kingdom-Cayman Islands:8
## 220	-2.364400	108	United Kingdom-Channel Islands:108
## 221	-5.353600	65	United Kingdom-Gibraltar:65
## 222	-4.548100	42	United Kingdom-Isle of Man:42
## 223	-62.187400	5	United Kingdom-Montserrat:5
## 224	-3.436000	19522	United Kingdom-:19522
## 225	-55.765800	304	Uruguay-:304
## 226	-95.712900	140886	US-:140886
## 227	64.585300	144	Uzbekistan-:144
## 228	-66.589700	119	Venezuela-:119
## 229	108.000000	188	Vietnam-:188
## 230	28.283300	29	Zambia-:29
## 231	30.000000	7	Zimbabwe-:7
## 232	0.000000	0	Canada-Diamond Princess:0
## 233	-61.371000	11	Dominica-:11
## 234	-61.679000	9	Grenada-:9
## 235	35.529562	8	Mozambique-:8
## 236	38.996815	9	Syria-:9
## 237	125.727539	1	Timor-Leste-:1

## 238	-59.543200	2	Belize-:2
## 239	0.000000	0	Canada-Recovered:0
## 240	102.495496	8	Laos-:8
## 241	17.228331	8	Libya-:8
## 242	35.233200	109	West Bank and Gaza-:109
## 243	-15.180400	2	Guinea-Bissau-:2
## 244	-3.996166	18	Mali-:18
## 245	-62.782998	2	Saint Kitts and Nevis-:2
## 246	-124.845700	1	Canada-Northwest Territories:1
## 247	-135.000000	4	Canada-Yukon:4
## 248	20.902977	94	Kosovo-:94
## 249	95.956000	10	Burma-:10
## 250	-63.068600	2	United Kingdom-Anguilla:2
## 251	-64.640000	2	United Kingdom-British Virgin Islands:2
## 252	-71.797900	4	United Kingdom-Turks and Caicos Islands:4
## 253	0.000000	2	MS Zaandam-:2

```
map <- leaflet() %>% addTiles()
#marker
map %<>% addCircleMarkers(x$Long, x$Lat, radius = 2+log2(x$confirmed), stroke = F,
                           color = 'red', fillOpacity = 0.3, popup = x$txt)
map
```



```
map %>% setView(5, 52, zoom = 6)
```



Number of cases:

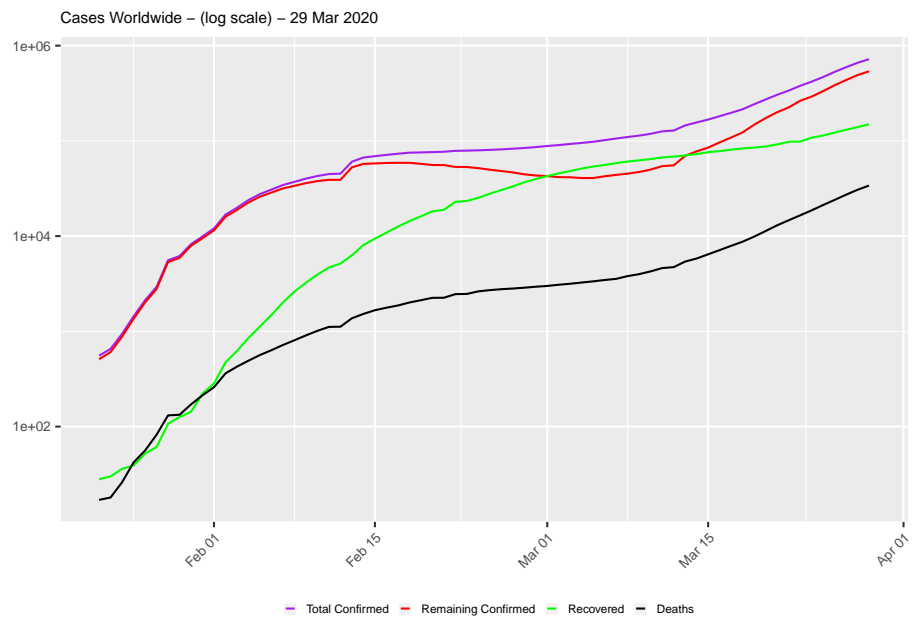
```
world.long <- data.long %>% filter(country == 'World') # can be also filtered for different
```

```
# area plot
plot1 <- world.long %>% filter(type != 'Total Confirmed') %>%
  ggplot(aes(x=date, y=count)) +
  geom_area(aes(fill=type), alpha=0.5) +
  labs(title=paste0('Cases Worldwide - ', max.date.txt)) +
  scale_fill_manual(values=c('red', 'green', 'black')) +
  theme(legend.title=element_blank(), legend.position='bottom',
        plot.title = element_text(size=8),
        axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        legend.key.size=unit(0.2, 'cm'),
        legend.text=element_text(size=6),
        axis.text=element_text(size=7),
        axis.text.x=element_text(angle=45, hjust=1))

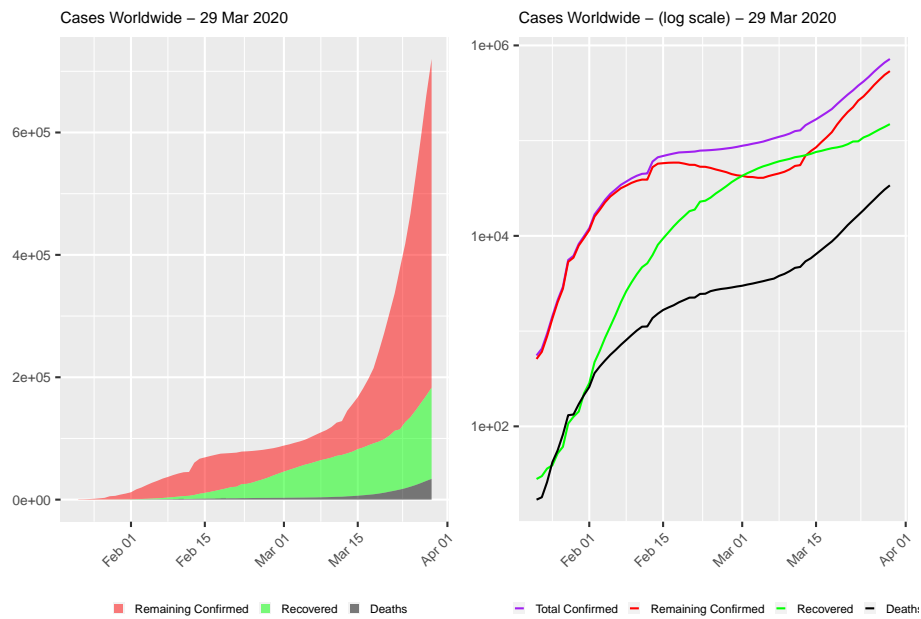
plot2 <- world.long %>%
  ggplot(aes(x=date, y=count)) +
```

```
geom_line(aes(color=type)) +
labs(title = paste0('Cases Worldwide - (log scale) - ', max.date.txt)) +
scale_color_manual(values=c('purple', 'red', 'green', 'black')) +
theme(legend.title=element_blank(), legend.position='bottom',
      plot.title = element_text(size = 8),
      axis.title.x=element_blank(),
      axis.title.y = element_blank(),
      legend.key.size = unit(0.2, 'cm'),
      legend.text = element_text(size = 6),
      axis.text = element_text(size = 7),
      axis.text.x =element_text(angle = 45, hjust = 1)) +
scale_y_continuous(trans = 'log10')
```

plot2



```
grid.arrange(plot1, plot2, ncol=2)
```



Current confirmed Cases:

```
data.world <- data %>% filter(country == 'World')
n <- nrow(data.world)

##current confirmed and daily new confirmed
plot1 <- ggplot(data.world, aes(x=date, y=remaining.confirmed)) +
  geom_point()+geom_smooth()+
  xlab('') + ylab('Count') + labs(title = 'Current Confirmed Cases') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot2 <- ggplot(data.world, aes(x=date, y=confirmed.new))+ geom_point() + geom_smooth() + xlab('') + ylab('Count') +
  theme(axis.text.x = element_text(angle = 45, hjust=1))
```

```
## List of 1
## $ axis.text.x:List of 11
## ..$ family      : NULL
## ..$ face         : NULL
## ..$ colour       : NULL
## ..$ size         : NULL
## ..$ hjust        : num 1
## ..$ vjust        : NULL
## ..$ angle        : num 45
## ..$ lineheight   : NULL
## ..$ margin       : NULL
```

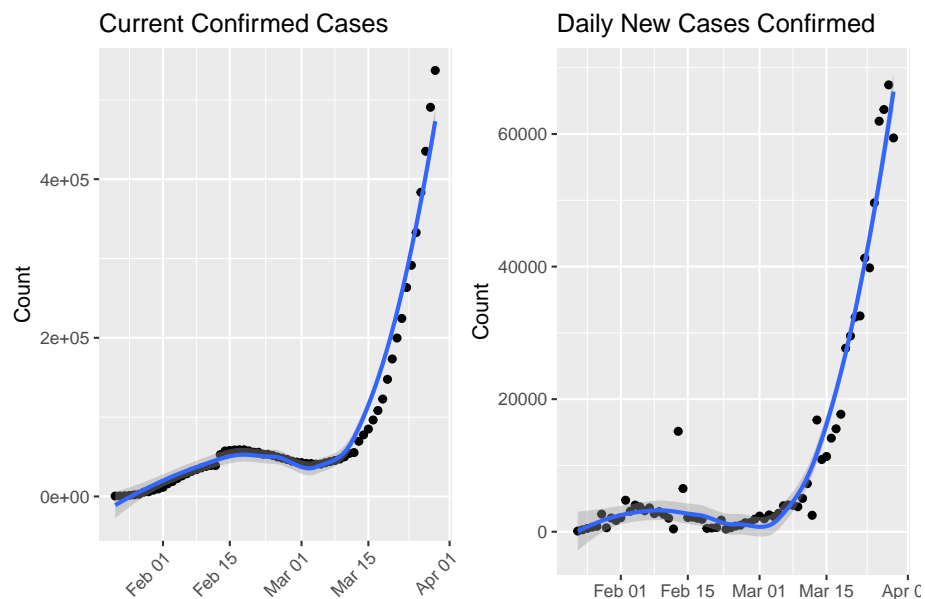
```
## ..$ debug          : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi FALSE
## - attr(*, "validate")= logi TRUE
```

```
grid.arrange(plot1, plot2, ncol=2)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



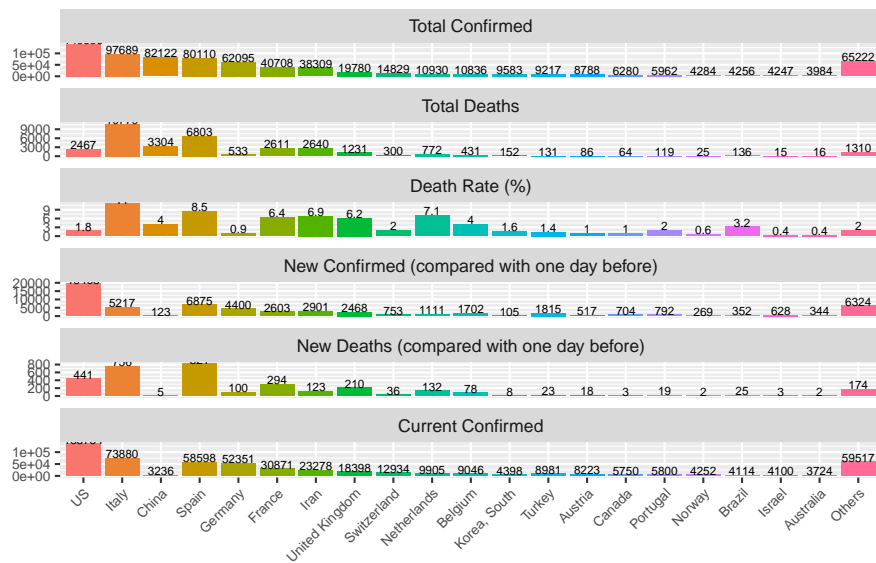
Bar Chart

```
data.latest.long <- data.latest %>% filter(country!='World') %>% gather(key=type, value=coun

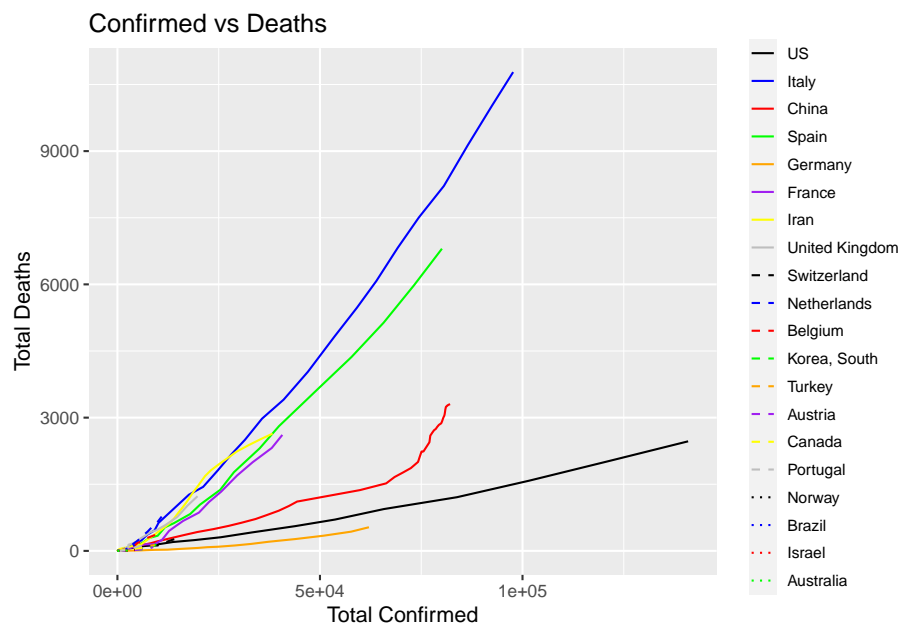
data.latest.long %<>% mutate(type=recode_factor(type, confirmed='Total Confirmed', deaths='')
```

```
## bar chart
data.latest.long %>% ggplot(aes(x=country, y=count, fill=country, group=country)) +
  geom_bar(stat='identity') +
  geom_text(aes(label=count, y=count), size=2, vjust=0) +
  xlab('') + ylab('') +
  labs(title=paste0('Top 20 Countries with Most Confirmed Cases - ', max.date.txt)) + scale_y_log10()
theme(legend.title=element_blank(),
      legend.position='none',
      plot.title=element_text(size=11), axis.text=element_text(size=7), axis.text.x=element_text(size=7))
```

Top 20 Countries with Most Confirmed Cases – 29 Mar 2020



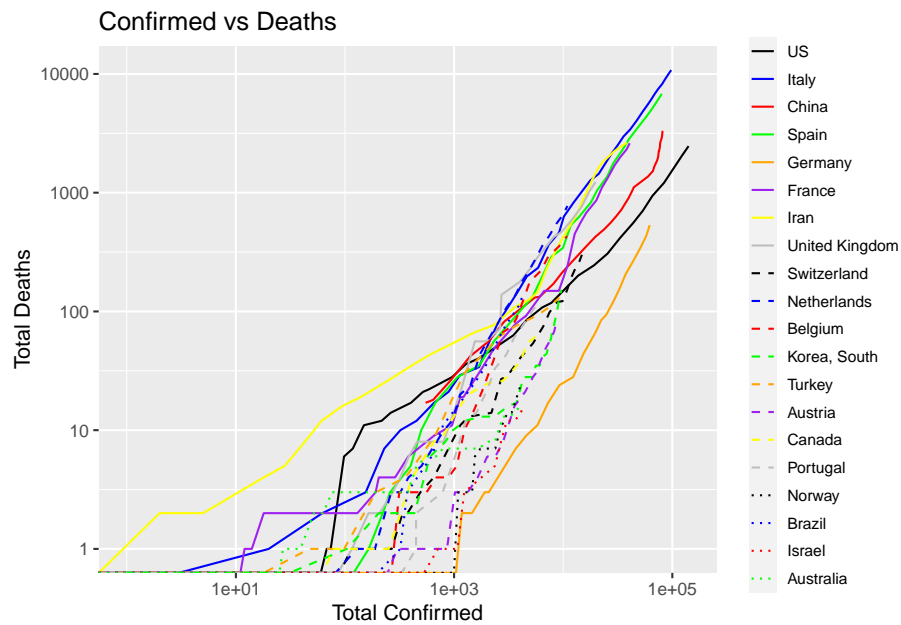
```
# Confirmed versus Deaths
linetypes <- rep(c("solid", "dashed", "dotted"), each=8)
colors <- rep(c('black', 'blue', 'red', 'green', 'orange', 'purple', 'yellow', 'grey'), 3)
df <- data %>% filter(country %in% setdiff(top.countries, c('World', 'Others')) %>%
  mutate(country=country %>% factor(levels=c(top.countries)))
vs <- df %>% ggplot(aes(x=confirmed, y=deaths, group=country)) +
  geom_line(aes(color=country, linetype=country)) +
  xlab('Total Confirmed') + ylab('Total Deaths') +
  scale_linetype_manual(values=linetypes) +
  scale_color_manual(values=colors) +
  theme(legend.title=element_blank(),
        legend.text=element_text(size=8),
        legend.key.size=unit(0.5, 'cm')) + ggtitle('Confirmed vs Deaths')
vs
```



```
vs + scale_x_log10() + scale_y_log10()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

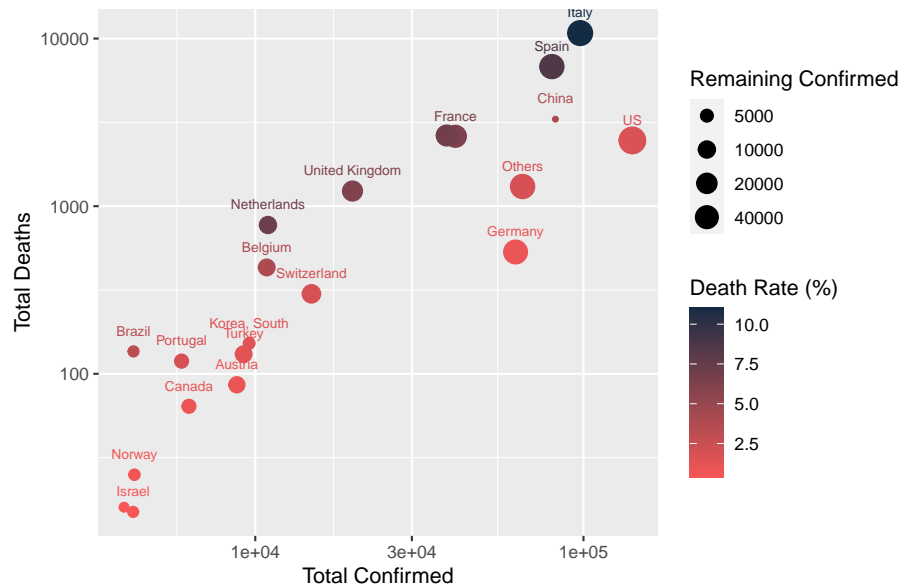



Number of confirmed cases and deaths in top 20 countries.

```
df <- data.latest %>% filter(country %in% setdiff(top.countries, 'World'))

plot1 <- df %>% ggplot(aes(x=confirmed, y=deaths, col=death.rate, size=remaining.confirmed))
  scale_size(name='Remaining Confirmed', trans='log2', breaks=c(1e3, 2e3, 5e3, 1e4, 2e4, 4e4))
  geom_text(aes(label=country), size=2.5, check_overlap=T, vjust=-1.6) +
  geom_point() +
  xlab('Total Confirmed') + ylab('Total Deaths') +
  labs(col="Death Rate (%)") +
  scale_color_gradient(low='#f75656', high='#132B43') +
  scale_x_log10() + scale_y_log10()
plot1
```

Number of confirmed cases and deaths in top 20 countries.

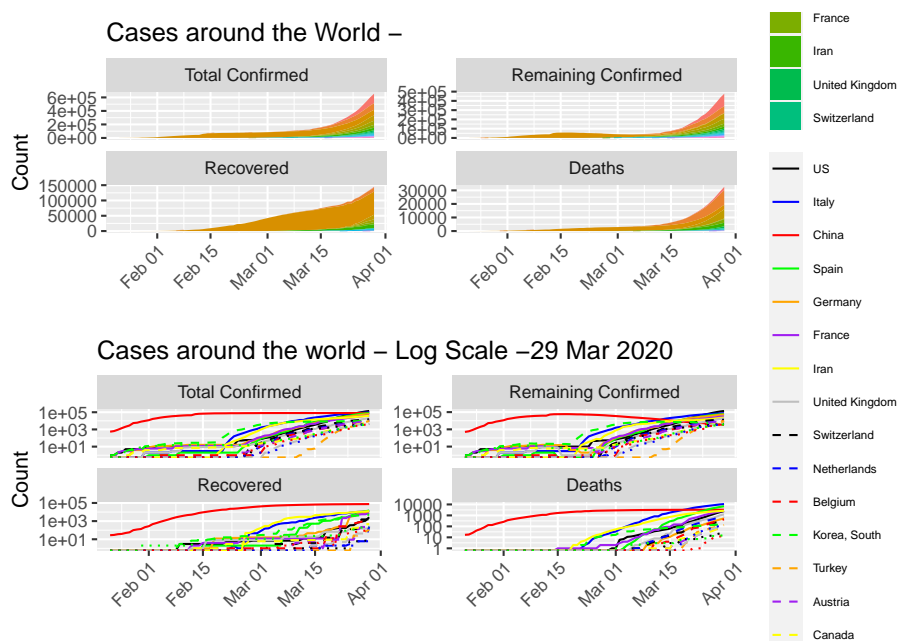


```
df <- data.long %>% filter(country %in% top.countries) %>% mutate(country=country %>% factor())

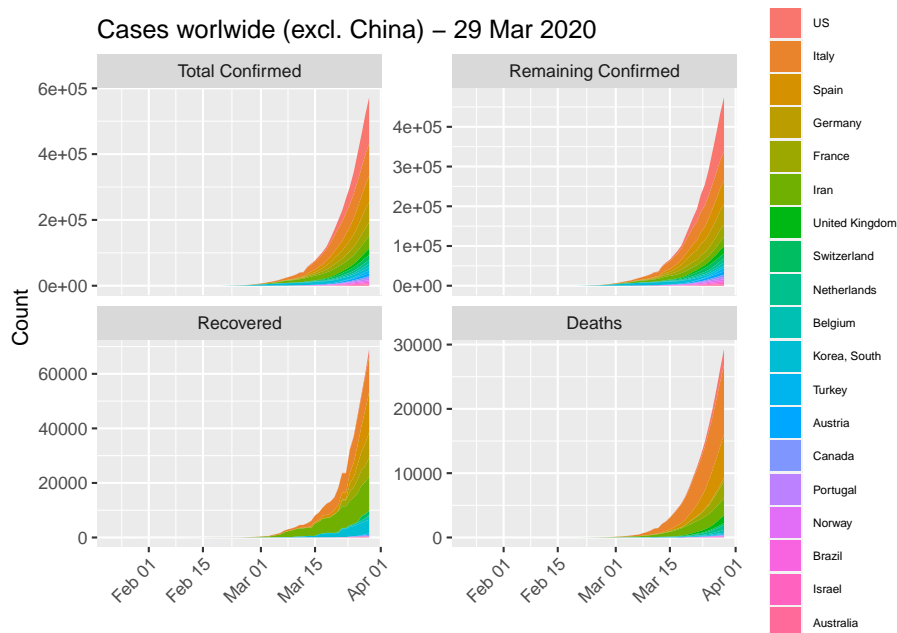
### CASES AROUND WORLD
p <- df %>% filter(country != 'World') %>%
  ggplot(aes(x=date, y=count)) + xlab('') + ylab('Count') +
  theme(legend.title=element_blank(),
        legend.text = element_text(size=6),
        legend.key.size=unit(0.6, 'cm'),
        axis.text.x=element_text(angle = 45, hjust=1)) +
  facet_wrap(~type, ncol = 2, scale='free_y')
# area plot
plot1 <- p + geom_area(aes(fill=country)) +
  labs(title='Cases around the World - ', max.date.txt)

# line plot and in log scale
#linetypes <- rep(c('solid','dashed','dotted'), each=8)
#colors <- rep(c('black','blue','red','green','orange','purple','yellow','grey'), 3)
plot2 <- p + geom_line(aes(color=country, linetype=country)) +
  scale_linetype_manual(values = linetypes) +
  scale_color_manual(values = colors) +
  labs(title =paste0('Cases around the world - Log Scale -', max.date.txt)) +
  scale_y_continuous(trans = 'log10')
grid.arrange(plot1, plot2, ncol=1)
```

Warning: Transformation introduced infinite values in continuous y-axis



```
# Plot: excluding China
p <- df %>% filter(!(country %in% c('World', 'China'))) %>%
  ggplot(aes(x=date, y=count)) + xlab('') + ylab('Count') +
  theme(legend.title=element_blank(),
        legend.text = element_text(size=6),
        legend.key.size=unit(0.6, 'cm'),
        axis.text.x=element_text(angle = 45, hjust=1)) +
  facet_wrap(~type, ncol = 2, scale='free_y')
p + geom_area(aes(fill=country)) +
  labs(title=paste0('Cases worldwide (excl. China) - ', max.date.txt))
```



```

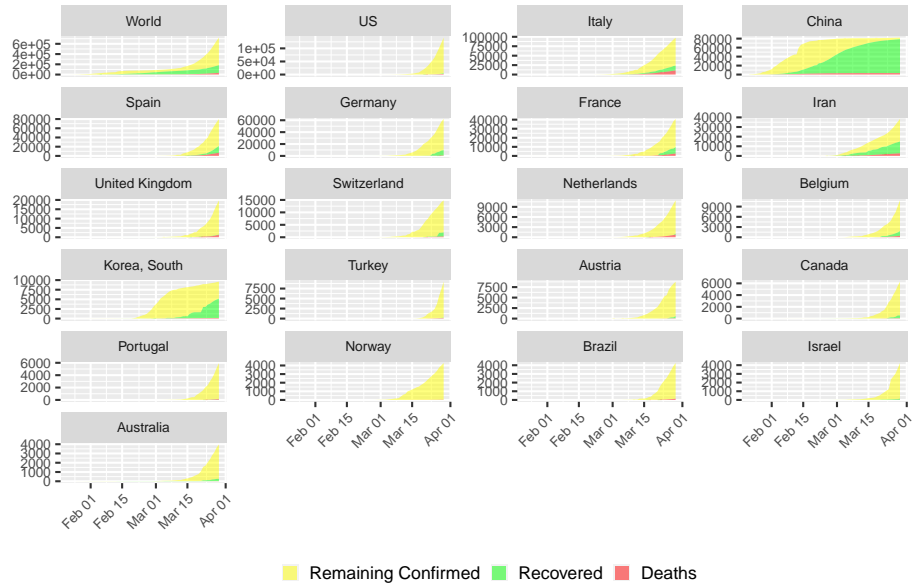
### list(countries) == 'Netherlands'

## If The Netherland is not top 20, add it in and remove 'Others'
if(!('Netherlands' %in% top.countries)) {
  top.countries %<>% setdiff('Others') %>% c('Netherlands')
  df <- data.long %>% filter(country %in% top.countries) %>%
    mutate(country=country %>% factor(levels = c(top.countries)))
}

# cases by country - area plot
df %>% filter(type != 'World' & type != 'Total Confirmed') %>%
  ggplot(aes(x=date, y=count, fill=type)) +
  geom_area(alpha=0.5) +
  labs(title = paste0('COVID - 19 Cases in Countries TOP 20 (incl. Netherlands) - ', max.date),
        scale_fill_manual(values=c('yellow','green','red')) +
  theme(legend.title=element_blank(), legend.position='bottom',
        plot.title= element_text(size = 9),
        axis.title.x=element_blank(),
        axis.title.y = element_blank(),
        legend.key.size = unit(0.3, 'cm'),
        strip.text.x = element_text(size=7),
        axis.text=element_text(size = 7),
        axis.text.x = element_text(angle=45, hjust=1)) +
  facet_wrap(~country, ncol=4, scale='free_y') + facet_wrap(~country, ncol=4, scales = 'free_y')

```

COVID – 19 Cases in Countries TOP 20 (incl. Netherlands) – 29 Mar 2020



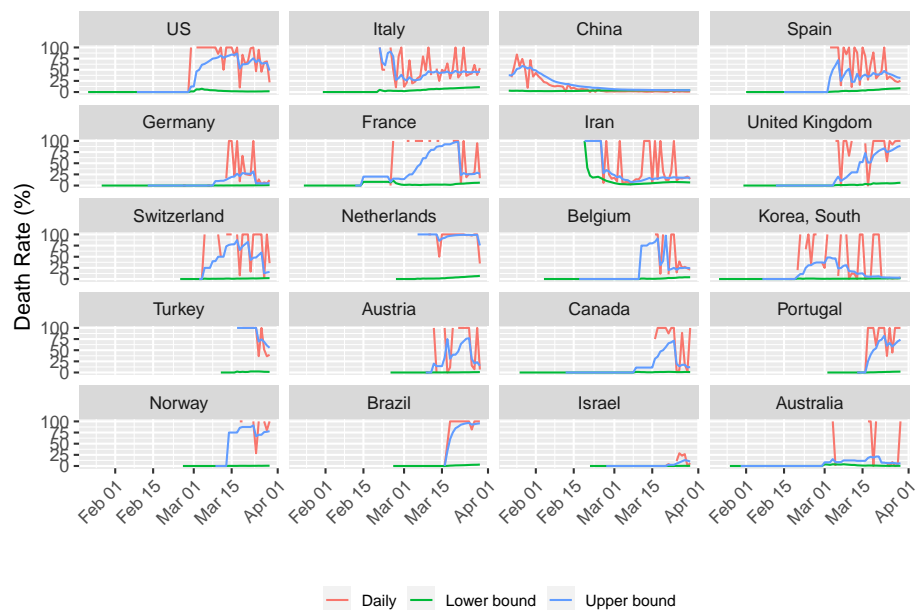
Deaths rate:

```
rate.max <- rates.long$count %>% max(na.rm=T)

df <- rates.long %>% filter(country %in% setdiff(top.countries, 'World')) %>%
  mutate(country=factor(country, levels=top.countries))

df %>% ggplot(aes(x=date, y=count, color=type)) +
  geom_line() +
  xlab('') + ylab('Death Rate (%)') +
  theme(legend.position='bottom', legend.title=element_blank(),
        legend.text=element_text(size=8),
        legend.key.size=unit(0.5, 'cm'),
        axis.text.x=element_text(angle=45, hjust=1)) +
  ylim(c(0, 100)) +
  facet_wrap(~country, ncol=4)
```

Warning: Removed 36 row(s) containing missing values (geom_path).



Countries with Highest Death Rates

```
## sort the latest data by death rate, and if tie, by confirmed
df <- data %>% filter(date == max(date) & country != 'World' & confirmed >= 100) %>%
  select(country, confirmed, confirmed.new, remaining.confirmed,
         recovered, deaths, deaths.new, death.rate=rate.lower) %>%
  arrange(desc(death.rate, confirmed))

df %>% head(20) %>%
  mutate(death.rate=death.rate %>% format(nsmall=1) %>% paste0('%')) %>%
  kable('latex', booktabs=T, row.names=T, align=c('l', rep('r', 7)),
        caption=paste0('Top 20 Countries with Highest Death Rates - ', max.date.txt), format
  kable_styling(font_size=7, latex_options=c('striped', 'hold_position', 'repeat_header'))
```

Note that this is an developing story. Check back for updates.

Table 2: Top 20 Countries with Highest Death Rates - 29 Mar 2020

	country	confirmed	confirmed.new	remaining.confirmed	recovered	deaths	deaths.new	death.rate
1	Italy	97,689	5,217	73,880	13,030	10,779	756	11.0%
2	San Marino	224	0	196	6	22	0	9.8%
3	Indonesia	1,285	130	1,107	64	114	12	8.9%
4	Spain	80,110	6,875	58,598	14,709	6,803	821	8.5%
5	Iraq	547	41	362	143	42	0	7.7%
6	Netherlands	10,930	1,111	9,905	253	772	132	7.1%
7	Iran	38,309	2,901	23,278	12,391	2,640	123	6.9%
8	Egypt	609	33	437	132	40	4	6.6%
9	France	40,708	2,603	30,871	7,226	2,611	294	6.4%
10	United Kingdom	19,780	2,468	18,398	151	1,231	210	6.2%
11	Algeria	511	57	449	31	31	2	6.1%
12	Burkina Faso	222	15	187	23	12	1	5.4%
13	Morocco	479	77	440	13	26	1	5.4%
14	Philippines	1,418	343	1,305	42	71	3	5.0%
15	Albania	212	15	169	33	10	0	4.7%
16	Dominican Republic	859	140	817	3	39	11	4.5%
17	Cameroon	139	48	128	5	6	4	4.3%
18	Belgium	10,836	1,702	9,046	1,359	431	78	4.0%
19	China	82,122	123	3,236	75,582	3,304	5	4.0%
20	Afghanistan	120	10	114	2	4	0	3.3%