COVID-19 Data Analysis with Tidyverse and Ggplot2 - World Wide

Yanchang Zhao
yanchang@RDataMining.com
http://RDataMining.com

11 March 2020

Contents

1 Introduction

This is an analysis report of the Novel Coronavirus (COVID-19) around the world, to demonstrate data processing and visualisation with R, *tidyverse* and *ggplot2*. This report will be updated from time to time, with new data and more analysis. Please find its latest version at http://www.rdatamining.com/docs/Coronavirus-data-analysis-world.pdf.

I have also writen a similar report for COVID-19 in China only. If you are particually interested what has happened in China, please find that report at http://www.rdatamining.com/docs/Coronavirus-data-analysis-china.pdf.

1.1 Data Source

The data source used for this analysis is the 2019 Novel Coronavirus COVID-19 (2019-nCoV) Data Repository[1] built by the Center for Systems Science and Engineering, Johns Hopkins University.

1.2 R Packages

Blow is a list of R packages used for this analysis. Package *magrittr* is for pipe operations like `%>%` and `%<>%` and *lubridate* for date operations. Package *tidyverse* is a collection of R packages for data science, including *dplyr* and *tidyr* for data processing and *ggplot2* for graphics. Package *gridExtra* is for arranging multiple

[1] https://github.com/CSSEGISandData/COVID-19

grid-based plots on a page and *kableExtra* works together with `kable()` from *knitr* to build complex HTML or LaTeX tables.

```
library(magrittr)
library(lubridate)
library(tidyverse)
library(gridExtra)
library(kableExtra)
```

2 Loading Data

At first, the datasets, which are three CSV files, are downloaded and saved as local files, and then are loaded into R.

```
## source data files
filenames <- c('time_series_19-covid-Confirmed.csv',
               'time_series_19-covid-Deaths.csv',
               'time_series_19-covid-Recovered.csv')
url.path <- 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_c

## download files to local
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 <- read.csv('./data/time_series_19-covid-Confirmed.csv')
data.deaths <- read.csv('./data/time_series_19-covid-Deaths.csv')
data.recovered <- read.csv('./data/time_series_19-covid-Recovered.csv')

dim(data.confirmed)
```

```
## [1] 347  53
```

Each dataset has 347 rows, corresponding to country/region/province/state. It has 53 columns. Starting from column 5, each column corresponds to a single day. Here we draw a random sample of 10 rows and have a look at their first 10 columns.

```
data.confirmed[, 1:10] %>% sample_n(10) %>%
  kable("latex", booktabs=T, caption="Raw Data (Confirmed, First 10 Columns only)") %>%
  kable_styling(font_size=6, latex_options = c("striped", "hold_position", "repeat_header"))
```

Table 1: Raw Data (Confirmed, First 10 Columns only)

Province.State	Country.Region	Lat	Long	X1.22.20	X1.23.20	X1.24.20	X1.25.20	X1.26.20	X1.27.20
Jiangxi	Mainland China	27.6140	115.7221	2	7	18	18	36	72
San Benito, CA	US	36.5761	-120.9876	0	0	0	0	0	0
Cook County, IL	US	41.7377	-87.6976	0	0	1	1	1	1
	French Guiana	3.9339	-53.1258	0	0	0	0	0	0
Williamson County, TN	US	35.9179	-86.8622	0	0	0	0	0	0
	Viet Nam	16.0000	108.0000	0	2	2	2	2	2
Shanghai	Mainland China	31.2020	121.4491	9	16	20	33	40	53
	Panama	8.5380	-80.7821	0	0	0	0	0	0
Clark County, WA	US	45.7466	-122.5194	0	0	0	0	0	0
Norfolk County, MA	US	42.1767	-71.1449	0	0	0	0	0	0

Below we check the time frame of the data.

```
n.col <- ncol(data.confirmed)
## get dates from column names
dates <- names(data.confirmed)[5:n.col] %>% substr(2,8) %>% mdy()
range(dates)
```

```
## [1] "2020-01-22" "2020-03-10"
```

```
max.date <- max(dates)
```

It shows that the data was last updated on 10 March 2020 UTC.

3 Data Preparation

3.1 Data Cleaning

The three datesets are converted from wide to long format, and then are aggregated by country. After that, they are merged into one single dataset.

```
## data 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 %<>% cleanData() %>% rename(confirmed=count)
data.deaths %<>% cleanData() %>% rename(deaths=count)
data.recovered %<>% cleanData() %>% rename(recovered=count)

## merge above 3 datasets into one, by country and date
data <- data.confirmed %>% merge(data.deaths) %>% merge(data.recovered)

## first 10 records when it first broke out in China
data %>% filter(country=='Mainland China') %>% head(10) %>%
  kable("latex", booktabs=T, caption="Raw Data (with first 10 Columns Only)",
        format.args=list(big.mark=",")) %>%
  kable_styling(latex_options = c("striped", "hold_position", "repeat_header"))
```

3.2 Cases for the Whole World

The raw data provide the daily number of cases in every country. They are aggregated below to derive the daily stats of the whole world.

```
## counts for the whole world
data.world <- data %>% group_by(date) %>%
  summarise(country='World',
            confirmed = sum(confirmed),
```

Table 2: Raw Data (with first 10 Columns Only)

country	date	confirmed	deaths	recovered
Mainland China	2020-01-22	547	17	28
Mainland China	2020-01-23	639	18	30
Mainland China	2020-01-24	916	26	36
Mainland China	2020-01-25	1,399	42	39
Mainland China	2020-01-26	2,062	56	49
Mainland China	2020-01-27	2,863	82	58
Mainland China	2020-01-28	5,494	131	101
Mainland China	2020-01-29	6,070	133	120
Mainland China	2020-01-30	8,124	171	135
Mainland China	2020-01-31	9,783	213	214

```
            deaths = sum(deaths),
            recovered = sum(recovered))
data %<>% rbind(data.world)

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

3.3 Daily Increases and Death Rates

After that, the daily increases of death and cured cases and the death rates are calculated.

`rate.upper` is caculated with the total dead and cured cases. It is the upper bound of death rate and the reasons are

1) there were much more deaths than cured cases when the coronavirus broke out and when it was not contained, and
2) the daily number of death will decrease and that of cured will increase as it becomes contained and more effective measures and treatments are used.

`rate.lower` is caculated with total dead and confirmed cases. It is a lower bound of death rate, because there are and will be new deaths from the current confirmed cases. The final death rate is expected to be in between of the above two rates.

`rate.daily` is caculated with the daily dead and cured cases and therefore is more volatile than the above two. However, it can give us a clue of the current situlation: whether it is very serious or is getting better.

```
## sort by country and date
data %<>% arrange(country, date)

## daily increases of deaths and cured cases
## set NA to the increases on day1
n <- nrow(data)
day1 <- min(data$date)
data %<>% mutate(confirmed.inc = ifelse(date == day1, NA, confirmed - lag(confirmed, n=1)),
                 deaths.inc = ifelse(date == day1, NA, deaths - lag(deaths, n=1)),
                 recovered.inc = ifelse(date == day1, NA, recovered - lag(recovered, n=1)))

## death rate based on total deaths and cured cases
data %<>% mutate(rate.upper = (100 * deaths / (deaths + recovered)) %>% round(1))
## lower bound: death rate based on total confirmed cases
```

```
data %<>% mutate(rate.lower = (100 * deaths / confirmed) %>% round(1))
## death rate based on the number of death/cured on every single day
data %<>% mutate(rate.daily = (100 * deaths.inc / (deaths.inc + recovered.inc)) %>% round(1))
```

4 Visualisation

After tidying up the data, we visualise it with various charts.

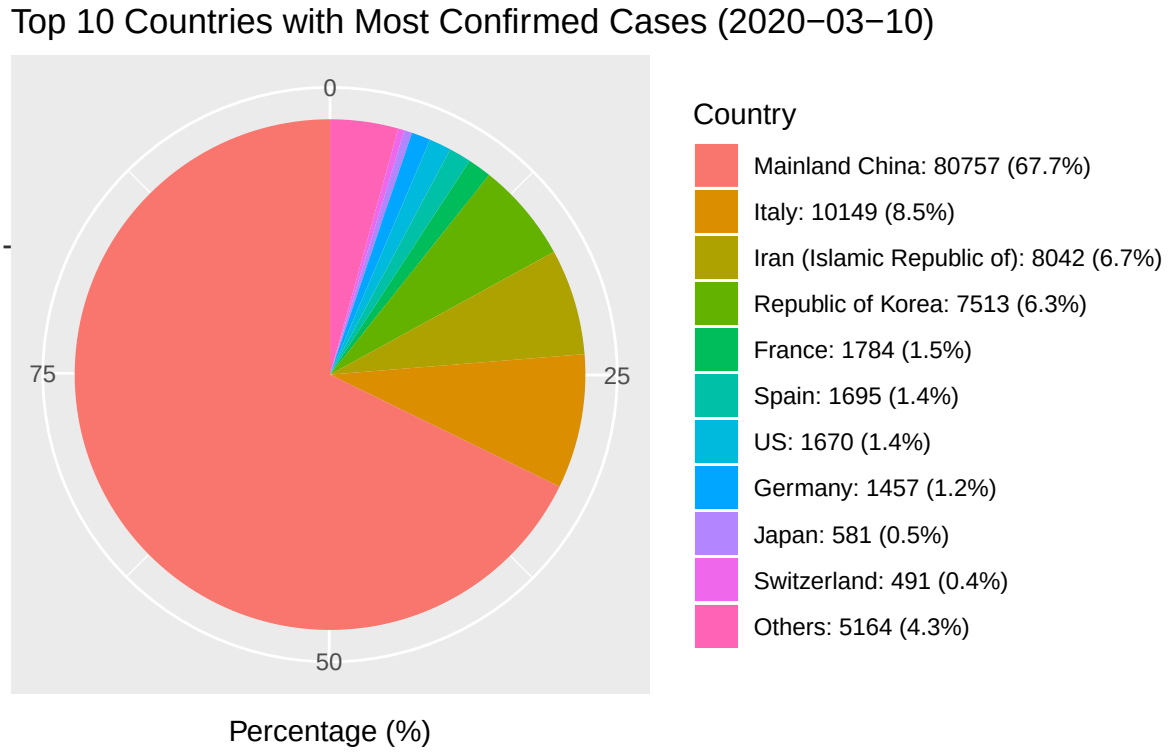
4.1 Top 10 countries with most confirmed cases

The line arnd area plots show the numbers of dead, cured, current confirmed and suppected cases. Note that, in the area plot, the total number of confirmed cases is represented by the total areas of confirmed, cured and dead.

```
## ranking by confirmed cases
data.latest <- data %>% filter(date == max(date)) %>%
  select(country, date, confirmed, deaths, recovered, remaining.confirmed) %>%
  mutate(ranking = dense_rank(desc(confirmed)))
## top 10 countries: 12 incl. 'World' and 'Others'
top.countries <- data.latest %>% filter(ranking <= 12) %>%
  arrange(ranking) %>% pull(country) %>% as.character()
## move 'Others' to the end
top.countries %<>% setdiff('Others') %>% c('Others')
top.countries
```

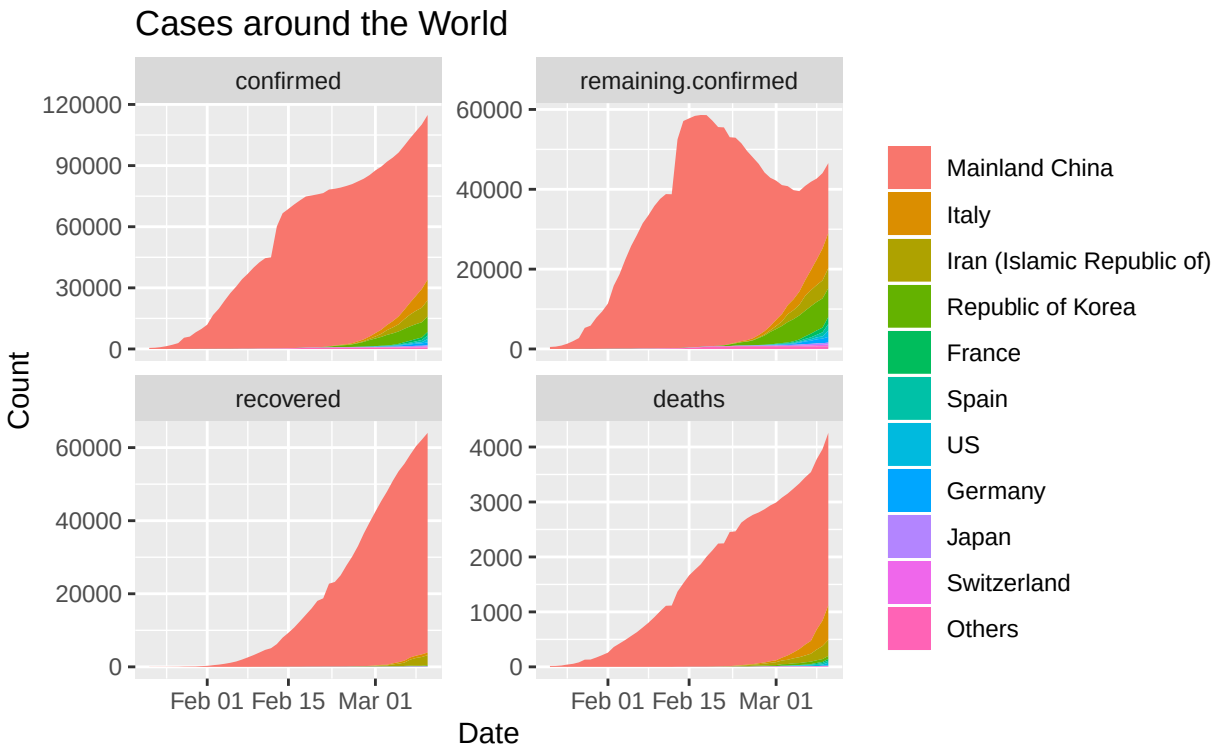
```
##  [1] "World"                     "Mainland China"
##  [3] "Italy"                     "Iran (Islamic Republic of)"
##  [5] "Republic of Korea"         "France"
##  [7] "Spain"                     "US"
##  [9] "Germany"                   "Japan"
## [11] "Switzerland"               "Others"
```

```
# a <- data %>% group_by(country) %>% tally()
## put all others in a single group of 'Others'
df <- data.latest %>% filter(!is.na(country) & country!='World') %>%
  mutate(country=ifelse(ranking <= 12, as.character(country), 'Others')) %>%
  mutate(country=country %>% factor(levels=c(top.countries)))
df %<>% group_by(country) %>% summarise(confirmed=sum(confirmed))
## precentage and label
df %<>% mutate(per = (100*confirmed/sum(confirmed)) %>% round(1)) %>%
  mutate(txt = paste0(country, ': ', confirmed, ' (', per, '%)'))
# pie(df$confirmed, labels=df$txt, cex=0.7)
df %>% ggplot(aes(fill=country)) +
  geom_bar(aes(x='', y=per), stat='identity') +
  coord_polar("y", start=0) +
  xlab('') + ylab('Percentage (%)') +
  labs(title=paste0('Top 10 Countries with Most Confirmed Cases (', max.date, ')')) +
  scale_fill_discrete(name='Country', labels=df$txt)
```

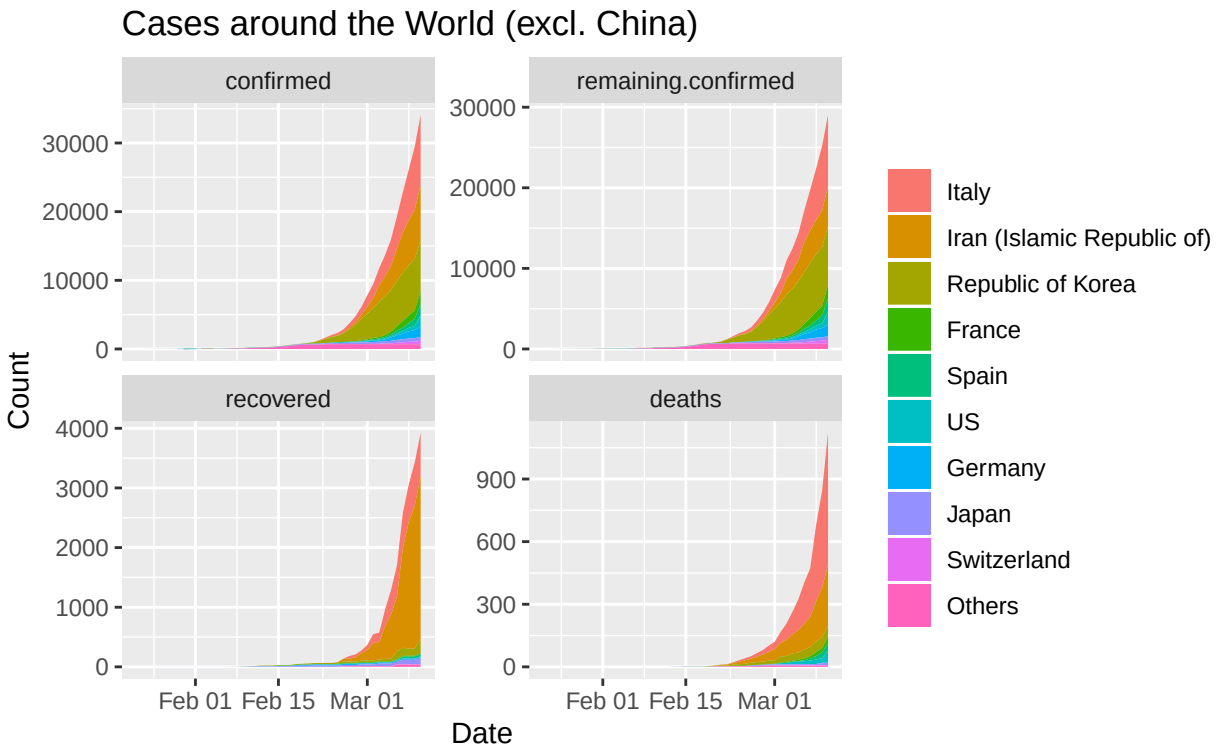



```
## convert from wide to long format, for purpose of drawing a area plot
data.long <- data %>% select(c(country, date, confirmed, remaining.confirmed, recovered, deaths)) %>%
  gather(key=type, value=count, -c(country, date))
## set factor levels to show them in a desirable order
data.long %<>% mutate(type = factor(type, c('confirmed', 'remaining.confirmed', 'recovered', 'deaths')))

## cases by type
df <- data.long %>% filter(country %in% top.countries) %<>%
  mutate(country=country %>% factor(levels=c(top.countries)))
df %>% filter(country != 'World') %>%
  ggplot(aes(x=date, y=count, fill=country)) +
  geom_area() + xlab('Date') + ylab('Count') +
  labs(title='Cases around the World') +
  theme(legend.title=element_blank()) +
  facet_wrap(~type, ncol=2, scales='free_y')
```

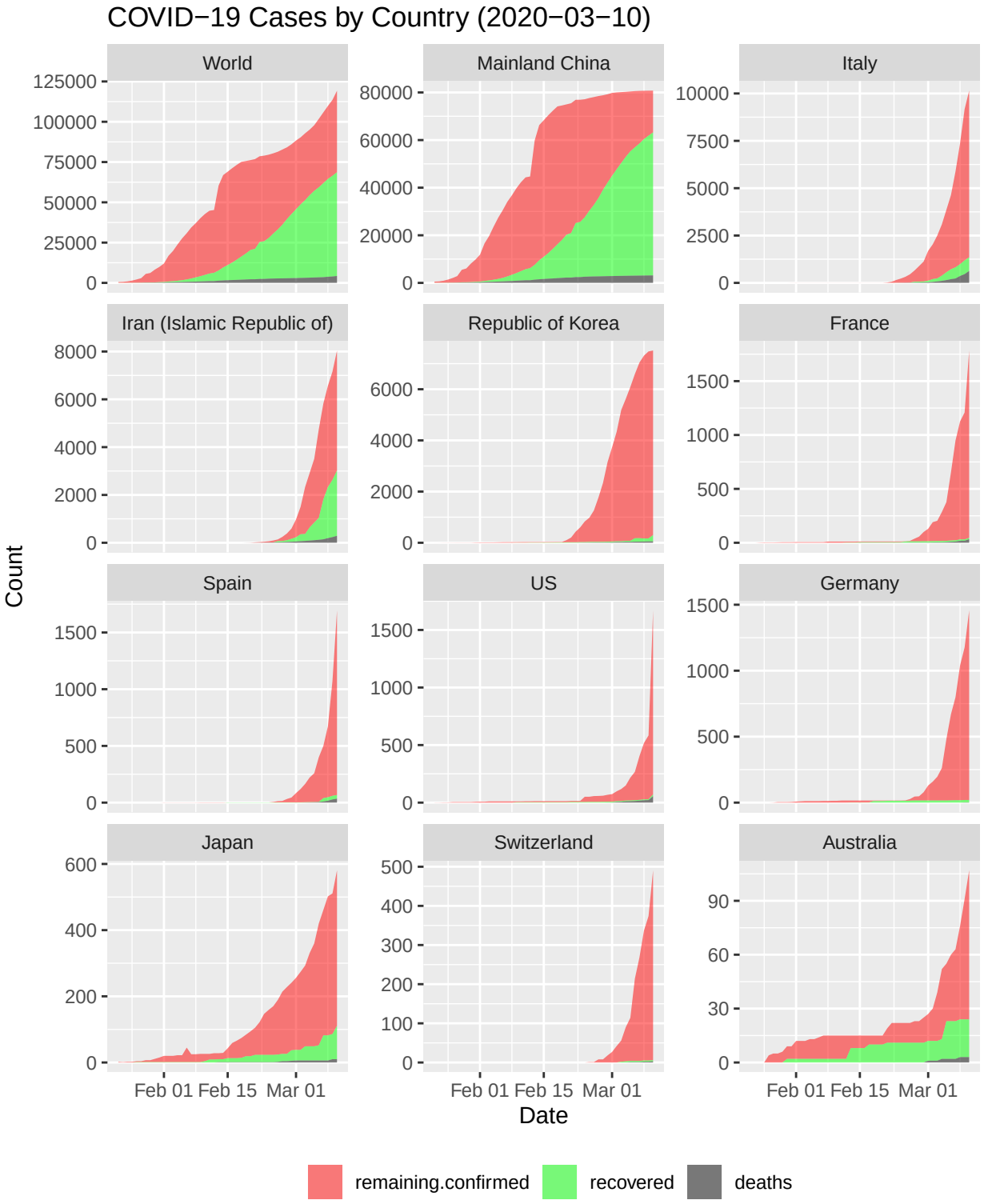



```
## excluding Mainland China
df %>% filter(!(country %in% c('World', 'Mainland China'))) %>%
  ggplot(aes(x=date, y=count, fill=country)) +
  geom_area() + xlab('Date') + ylab('Count') +
  labs(title='Cases around the World (excl. China)') +
  theme(legend.title=element_blank()) +
  facet_wrap(~type, ncol=2, scales='free_y')
```




```
## if Australia in not in top 10, add it in and remove 'Others'
if(!('Australia' %in% top.countries)) {
  top.countries %<>% setdiff('Others') %>% c('Australia')
  df <- data.long %>% filter(country %in% top.countries) %<>%
    mutate(country=country %>% factor(levels=c(top.countries)))
}

## cases by country
df %>% filter(type != 'confirmed') %>%
  ggplot(aes(x=date, y=count, fill=type)) +
  geom_area(alpha=0.5) + xlab('Date') + ylab('Count') +
  labs(title=paste0('COVID-19 Cases by Country (', max.date, ')')) +
  scale_fill_manual(values=c('red', 'green', 'black')) +
  theme(legend.title=element_blank(), legend.position='bottom') +
  facet_wrap(~country, ncol=3, scales='free_y')
```



From the above figure (based on official stats), the coronavirus seems to be under control in China, with an increase of recovered cases (in green) every day and a shrinking of the remaining confrimed cases (in red). However, in the rest of the world (i.e., outside of China), the confirmed cases are surging up in many other countries, which suggests that the virus has broken out there.

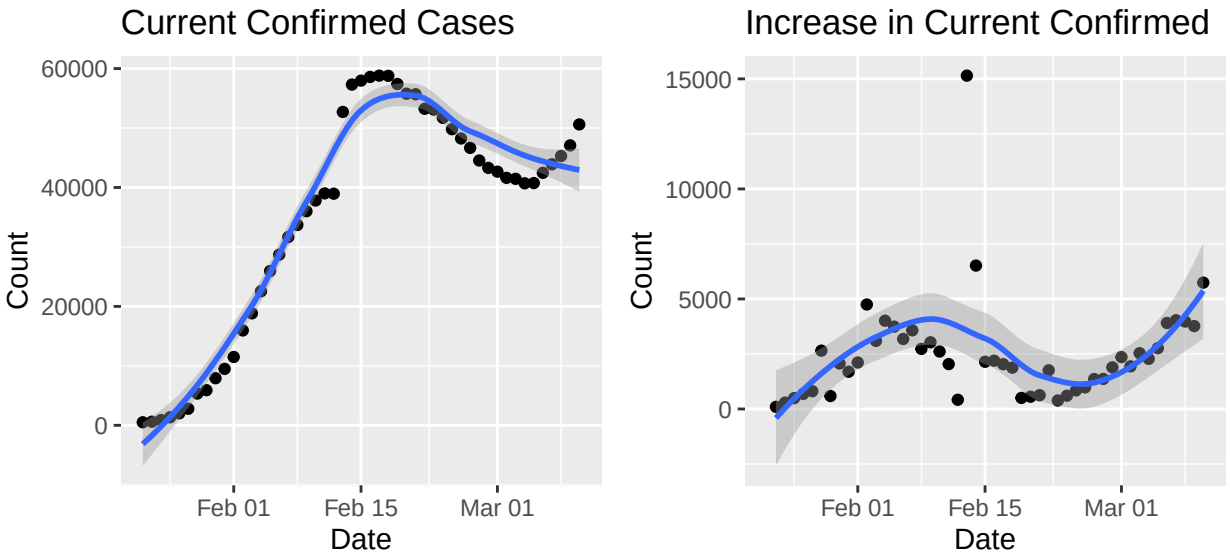
4.2 Current (or Remaining) Confirmed Cases

Now we focuse on all cases world wide.

```
# data %<>% filter(country=='Mainland China')
# data %<>% filter(country=='Australia')
data %<>% filter(country=='World')
n <- nrow(data)
```

```
## current confirmed and its increase
plot1 <- ggplot(data, aes(x=date, y=remaining.confirmed)) +
  geom_point() + geom_smooth() +
  xlab('Date') + ylab('Count') + labs(title='Current Confirmed Cases')
plot2 <- ggplot(data, aes(x=date, y=confirmed.inc)) +
  geom_point() + geom_smooth() +
  xlab('Date') + ylab('Count') + labs(title='Increase in Current Confirmed')
# +  ylim(0, 4500)
grid.arrange(plot1, plot2, ncol=2)
```

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

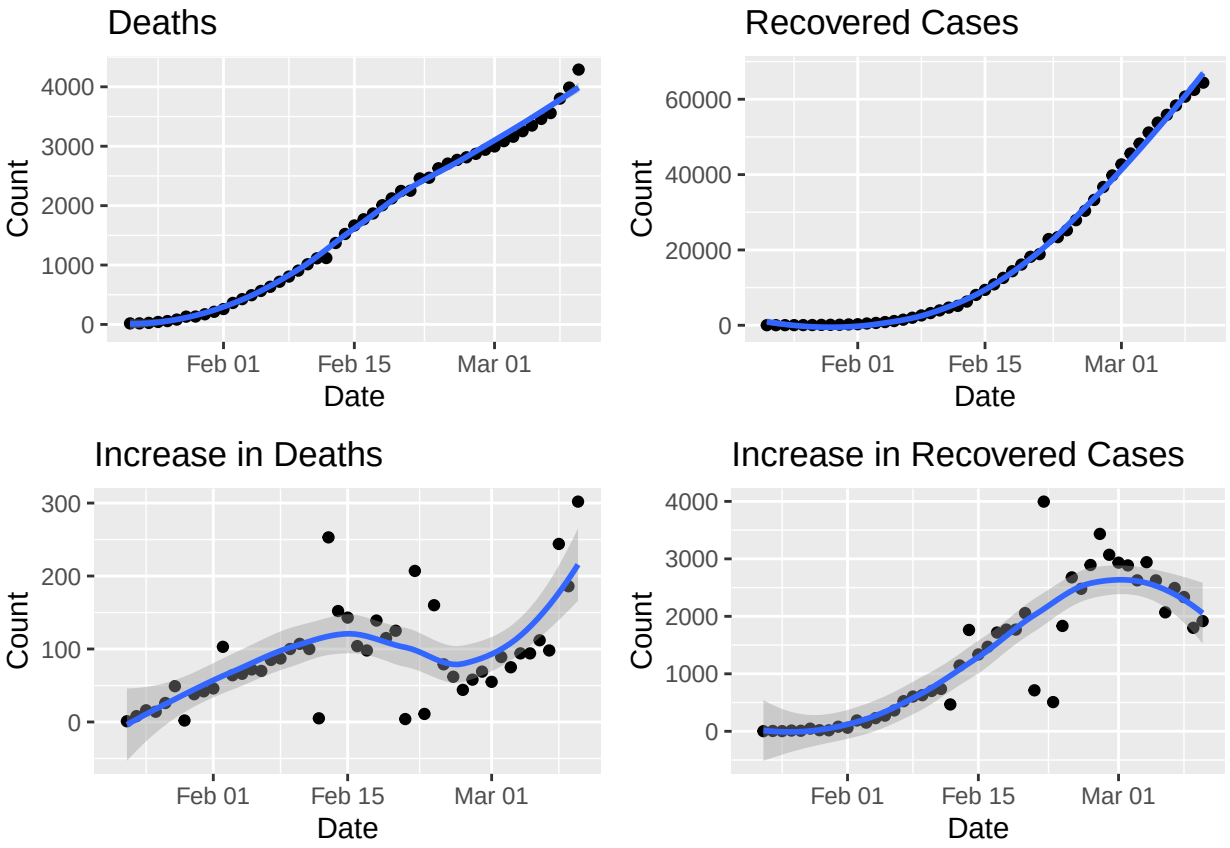


4.3 Deaths and Cured Cases

```
## a scatter plot with a smoothed line and vertical x-axis labels
plot1 <- ggplot(data, aes(x=date, y=deaths)) +
  geom_point() + geom_smooth() +
  xlab('Date') + ylab('Count') + labs(title='Deaths')
plot2 <- ggplot(data, aes(x=date, y=recovered)) +
  geom_point() + geom_smooth() +
  xlab('Date') + ylab('Count') + labs(title='Recovered Cases')
plot3 <- ggplot(data, aes(x=date, y=deaths.inc))  +
  geom_point() + geom_smooth()  +
  xlab('Date') + ylab('Count') + labs(title='Increase in Deaths')
plot4 <- ggplot(data, aes(x=date, y=recovered.inc)) +
  geom_point() + geom_smooth()  +
  xlab('Date') + ylab('Count') + labs(title='Increase in Recovered Cases')
```

```
## show four plots together, with 2 plots in each row
grid.arrange(plot1, plot2, plot3, plot4, nrow=2)
```

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



4.4 Death Rates

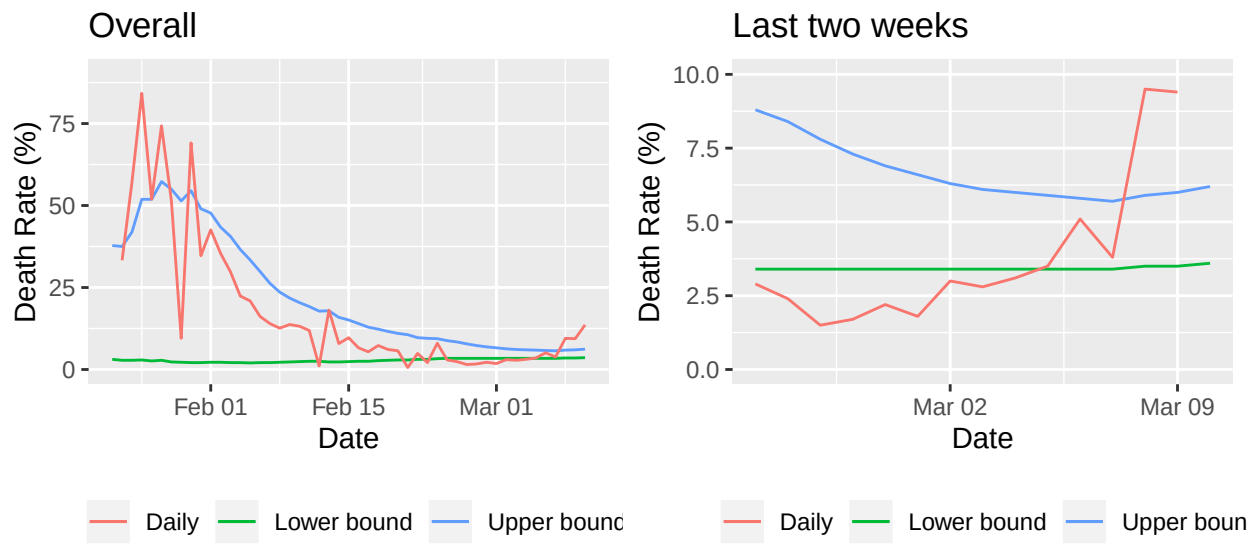
Below are death rates caculated in three different ways. See Section 3.3 for the details of caculating the three rates.

In the right chart, the upper bound (in blue) is decreasing, as there will be more cured cases and fewer dead ones daily as time goes on. However, the lower bound (in green) keeps going up, as there are and will be new deaths from the current confirmed cases. Therefore, the final death rate is expected to be in-between of those two rates, and based on the latest data retrieved as of 10 March 2020, it will be between 3.6% and 6.2% (see the last row in the table at the end of this report).

A surge in the daily death rate (in red) suggests that the situlation is changing dramatically (actually, getting worse) and that above lower/upper bounds are likely to increase shortly. A likely reason of that surge is the outbreak of coronavirus in Italy and Iran recently.

```
## three death rates
plot1 <- ggplot(data, aes(x=date)) +
  geom_line(aes(y=rate.upper, colour='Upper bound')) +
  geom_line(aes(y=rate.lower, colour='Lower bound')) +
  geom_line(aes(y=rate.daily, colour='Daily')) +
```

```
  xlab('Date') + ylab('Death Rate (%)') + labs(title='Overall') +
  theme(legend.position='bottom', legend.title=element_blank()) +
  ylim(0, 90)
## focusing on last 2 weeks
plot2 <- ggplot(data[n-(14:0),], aes(x=date)) +
  geom_line(aes(y=rate.upper, colour='Upper bound')) +
  geom_line(aes(y=rate.lower, colour='Lower bound')) +
  geom_line(aes(y=rate.daily, colour='Daily')) +
  xlab('Date') + ylab('Death Rate (%)') + labs(title='Last two weeks') +
  theme(legend.position='bottom', legend.title=element_blank()) +
  ylim(0, 10)
grid.arrange(plot1, plot2, ncol=2)
```



5 Processed Data

Blow is the processed data for this analysis and visualisation.

```
## re-order columns
# deadIncr, curedIncr,
data %<>% select(c(date, confirmed, deaths, recovered, remaining.confirmed,
                   confirmed.inc, deaths.inc, recovered.inc, rate.upper, rate.daily, rate.lower))
## to make column names shorter for output purpose only
names(data) %<>% gsub(pattern='Count', replacement='')
## output as a table
data %>% kable("latex", booktabs=T, longtable=T, caption="Cases in the Whole World",
               format.args=list(big.mark=",")) %>%
    kable_styling(font_size=5, latex_options = c("striped", "hold_position", "repeat_header"))
```

Table 3: Cases in the Whole World

date	confirmed	deaths	recovered	remaining.confirmed	confirmed.inc	deaths.inc	recovered.inc	rate.upper	rate.daily	rate.lower
2020-01-22	555	17	28	510				37.8		3.1
2020-01-23	653	18	30	605	98	1	2	37.5	33.3	2.8
2020-01-24	941	26	36	879	288	8	6	41.9	57.1	2.8
2020-01-25	1,434	42	39	1,353	493	16	3	51.9	84.2	2.9
2020-01-26	2,118	56	52	2,010	684	14	13	51.9	51.9	2.6
2020-01-27	2,927	82	61	2,784	809	26	9	57.3	74.3	2.8
2020-01-28	5,578	131	107	5,340	2,651	49	46	55.0	51.6	2.3
2020-01-29	6,166	133	126	5,907	588	2	19	51.4	9.5	2.2

Table 3: Cases in the Whole World *(continued)*

date	confirmed	deaths	recovered	remaining.confirmed	confirmed.inc	deaths.inc	recovered.inc	rate.upper	rate.daily	rate.lower
2020-01-30	8,234	171	143	7,920	2,068	38	17	54.5	69.1	2.1
2020-01-31	9,927	213	222	9,492	1,693	42	79	49.0	34.7	2.1
2020-02-01	12,038	259	284	11,495	2,111	46	62	47.7	42.6	2.2
2020-02-02	16,787	362	472	15,953	4,749	103	188	43.4	35.4	2.2
2020-02-03	19,881	426	623	18,832	3,094	64	151	40.6	29.8	2.1
2020-02-04	23,892	492	852	22,548	4,011	66	229	36.6	22.4	2.1
2020-02-05	27,635	564	1,124	25,947	3,743	72	272	33.4	20.9	2.0
2020-02-06	30,817	634	1,487	28,696	3,182	70	363	29.9	16.2	2.1
2020-02-07	34,391	719	2,011	31,661	3,574	85	524	26.3	14.0	2.1
2020-02-08	37,120	806	2,616	33,698	2,729	87	605	23.6	12.6	2.2
2020-02-09	40,150	906	3,244	36,000	3,030	100	628	21.8	13.7	2.3
2020-02-10	42,762	1,013	3,946	37,803	2,612	107	702	20.4	13.2	2.4
2020-02-11	44,802	1,113	4,683	39,006	2,040	100	737	19.2	11.9	2.5
2020-02-12	45,221	1,118	5,150	38,953	419	5	467	17.8	1.1	2.5
2020-02-13	60,368	1,371	6,295	52,702	15,147	253	1,145	17.9	18.1	2.3
2020-02-14	66,885	1,523	8,058	57,304	6,517	152	1,763	15.9	7.9	2.3
2020-02-15	69,030	1,666	9,395	57,969	2,145	143	1,337	15.1	9.7	2.4
2020-02-16	71,224	1,770	10,865	58,589	2,194	104	1,470	14.0	6.6	2.5
2020-02-17	73,258	1,868	12,583	58,807	2,034	98	1,718	12.9	5.4	2.5
2020-02-18	75,136	2,007	14,352	58,777	1,878	139	1,769	12.3	7.3	2.7
2020-02-19	75,639	2,122	16,121	57,396	503	115	1,769	11.6	6.1	2.8
2020-02-20	76,197	2,247	18,177	55,773	558	125	2,056	11.0	5.7	2.9
2020-02-21	76,823	2,251	18,890	55,682	626	4	713	10.6	0.6	2.9
2020-02-22	78,579	2,458	22,886	53,235	1,756	207	3,996	9.7	4.9	3.1
2020-02-23	78,965	2,469	23,394	53,102	386	11	508	9.5	2.1	3.1
2020-02-24	79,568	2,629	25,227	51,712	603	160	1,833	9.4	8.0	3.3
2020-02-25	80,413	2,708	27,905	49,800	845	79	2,678	8.8	2.9	3.4
2020-02-26	81,395	2,770	30,384	48,241	982	62	2,479	8.4	2.4	3.4
2020-02-27	82,754	2,814	33,277	46,663	1,359	44	2,893	7.8	1.5	3.4
2020-02-28	84,120	2,872	36,711	44,537	1,366	58	3,434	7.3	1.7	3.4
2020-02-29	86,011	2,941	39,782	43,288	1,891	69	3,071	6.9	2.2	3.4
2020-03-01	88,369	2,996	42,716	42,657	2,358	55	2,934	6.6	1.8	3.4
2020-03-02	90,306	3,085	45,602	41,619	1,937	89	2,886	6.3	3.0	3.4
2020-03-03	92,840	3,160	48,228	41,452	2,534	75	2,626	6.1	2.8	3.4
2020-03-04	95,120	3,254	51,170	40,696	2,280	94	2,942	6.0	3.1	3.4
2020-03-05	97,882	3,348	53,796	40,738	2,762	94	2,626	5.9	3.5	3.4
2020-03-06	101,789	3,460	55,865	42,464	3,907	112	2,069	5.8	5.1	3.4
2020-03-07	105,822	3,558	58,358	43,906	4,033	98	2,493	5.7	3.8	3.4
2020-03-08	109,795	3,802	60,694	45,299	3,973	244	2,336	5.9	9.5	3.5
2020-03-09	113,562	3,988	62,494	47,080	3,767	186	1,800	6.0	9.4	3.5
2020-03-10	119,303	4,290	64,411	50,602	5,741	302	1,917	6.2	13.6	3.6

```
data.latest %>% arrange(desc(confirmed)) %>%
  select(-ranking) %>% filter(country!='Others') %>%
  kable("latex", booktabs=T, longtable=T, row.names=T, caption="Cases by Country",
        format.args=list(big.mark=",")) %>%
  kable_styling(font_size=7, latex_options = c("striped", "hold_position", "repeat_header"))
```

Table 4: Cases by Country

	country	date	confirmed	deaths	recovered	remaining.confirmed
1	World	2020-03-10	119,303	4,290	64,411	50,602
2	Mainland China	2020-03-10	80,757	3,136	60,106	17,515
3	Italy	2020-03-10	10,149	631	724	8,794
4	Iran (Islamic Republic of)	2020-03-10	8,042	291	2,731	5,020
5	Republic of Korea	2020-03-10	7,513	54	247	7,212
6	France	2020-03-10	1,784	33	12	1,739
7	Spain	2020-03-10	1,695	35	32	1,628
8	US	2020-03-10	1,670	56	15	1,599
9	Germany	2020-03-10	1,457	2	18	1,437
10	Japan	2020-03-10	581	10	101	470
11	Switzerland	2020-03-10	491	3	3	485
12	Norway	2020-03-10	400	0	1	399
13	Netherlands	2020-03-10	382	4	0	378
14	UK	2020-03-10	382	6	18	358
15	Sweden	2020-03-10	355	0	1	354
16	Belgium	2020-03-10	267	0	1	266

Table 4: Cases by Country *(continued)*

	country	date	confirmed	deaths	recovered	remaining.confirmed
17	Denmark	2020-03-10	262	0	1	261
18	Austria	2020-03-10	182	0	4	178
19	Singapore	2020-03-10	160	0	78	82
20	Malaysia	2020-03-10	129	0	24	105
21	Hong Kong SAR	2020-03-10	120	3	65	52
22	Bahrain	2020-03-10	110	0	22	88
23	Australia	2020-03-10	107	3	21	83
24	Greece	2020-03-10	89	0	0	89
25	Canada	2020-03-10	79	1	8	70
26	United Arab Emirates	2020-03-10	74	0	12	62
27	Iraq	2020-03-10	71	7	3	61
28	Iceland	2020-03-10	69	0	1	68
29	Kuwait	2020-03-10	69	0	1	68
30	Egypt	2020-03-10	59	1	1	57
31	Israel	2020-03-10	58	0	4	54
32	India	2020-03-10	56	0	4	52
33	Thailand	2020-03-10	53	1	33	19
34	San Marino	2020-03-10	51	2	0	49
35	Taipei and environs	2020-03-10	47	1	17	29
36	Czech Republic	2020-03-10	41	0	0	41
37	Lebanon	2020-03-10	41	1	1	39
38	Portugal	2020-03-10	41	0	0	41
39	Finland	2020-03-10	40	0	1	39
40	Ireland	2020-03-10	34	0	0	34
41	Philippines	2020-03-10	33	1	2	30
42	Brazil	2020-03-10	31	0	0	31
43	Slovenia	2020-03-10	31	0	0	31
44	Viet Nam	2020-03-10	31	0	16	15
45	Indonesia	2020-03-10	27	0	2	25
46	occupied Palestinian territory	2020-03-10	25	0	0	25
47	Romania	2020-03-10	25	0	3	22
48	Qatar	2020-03-10	24	0	0	24
49	Poland	2020-03-10	22	0	0	22
50	Algeria	2020-03-10	20	0	0	20
51	Russian Federation	2020-03-10	20	0	3	17
52	Saudi Arabia	2020-03-10	20	0	1	19
53	Oman	2020-03-10	18	0	9	9
54	Argentina	2020-03-10	17	1	0	16
55	Pakistan	2020-03-10	16	0	1	15
56	Ecuador	2020-03-10	15	0	0	15
57	Georgia	2020-03-10	15	0	0	15
58	Croatia	2020-03-10	14	0	0	14
59	Chile	2020-03-10	13	0	0	13
60	Estonia	2020-03-10	12	0	0	12
61	Azerbaijan	2020-03-10	11	0	0	11
62	Peru	2020-03-10	11	0	0	11
63	Albania	2020-03-10	10	0	0	10
64	Macao SAR	2020-03-10	10	0	10	0
65	Belarus	2020-03-10	9	0	3	6
66	Costa Rica	2020-03-10	9	0	0	9
67	Hungary	2020-03-10	9	0	0	9
68	Latvia	2020-03-10	8	0	1	7
69	Mexico	2020-03-10	7	0	4	3
70	North Macedonia	2020-03-10	7	0	0	7
71	Slovakia	2020-03-10	7	0	0	7
72	South Africa	2020-03-10	7	0	0	7
73	Maldives	2020-03-10	6	0	0	6
74	Afghanistan	2020-03-10	5	0	0	5
75	Bosnia and Herzegovina	2020-03-10	5	0	0	5
76	Dominican Republic	2020-03-10	5	0	0	5
77	French Guiana	2020-03-10	5	0	0	5

Table 4: Cases by Country *(continued)*

	country	date	confirmed	deaths	recovered	remaining.confirmed
78	Luxembourg	2020-03-10	5	0	0	5
79	Malta	2020-03-10	5	0	0	5
80	New Zealand	2020-03-10	5	0	0	5
81	Serbia	2020-03-10	5	0	0	5
82	Tunisia	2020-03-10	5	0	0	5
83	Bulgaria	2020-03-10	4	0	0	4
84	Senegal	2020-03-10	4	0	1	3
85	Bangladesh	2020-03-10	3	0	0	3
86	Colombia	2020-03-10	3	0	0	3
87	Cyprus	2020-03-10	3	0	0	3
88	Morocco	2020-03-10	3	1	0	2
89	Republic of Moldova	2020-03-10	3	0	0	3
90	Cambodia	2020-03-10	2	0	1	1
91	Cameroon	2020-03-10	2	0	0	2
92	Faroe Islands	2020-03-10	2	0	0	2
93	Martinique	2020-03-10	2	0	0	2
94	Nigeria	2020-03-10	2	0	0	2
95	Saint Martin	2020-03-10	2	0	0	2
96	Andorra	2020-03-10	1	0	0	1
97	Armenia	2020-03-10	1	0	0	1
98	Bhutan	2020-03-10	1	0	0	1
99	Brunei	2020-03-10	1	0	0	1
100	Burkina Faso	2020-03-10	1	0	0	1
101	Channel Islands	2020-03-10	1	0	0	1
102	Gibraltar	2020-03-10	1	0	1	0
103	Holy See	2020-03-10	1	0	0	1
104	Jordan	2020-03-10	1	0	0	1
105	Liechtenstein	2020-03-10	1	0	0	1
106	Lithuania	2020-03-10	1	0	0	1
107	Monaco	2020-03-10	1	0	0	1
108	Mongolia	2020-03-10	1	0	0	1
109	Nepal	2020-03-10	1	0	1	0
110	Panama	2020-03-10	1	0	0	1
111	Paraguay	2020-03-10	1	0	0	1
112	Saint Barthelemy	2020-03-10	1	0	0	1
113	Sri Lanka	2020-03-10	1	0	1	0
114	Togo	2020-03-10	1	0	0	1
115	Ukraine	2020-03-10	1	0	0	1