COVID-19 Data Analysis with R - Worldwide

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

02 April 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.

A similar COVID-19 analysis report for China is available at http://www.rdatamining.com/docs/Coronavirus-data-analysis-china.pdf, if you are particually interested what has happened in China.

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

```
library(magrittr) # pipe operations
library(lubridate) # date operations
library(tidyverse) # ggplot2, tidyr, dplyr...
library(gridExtra) # multiple grid-based plots on a page
library(ggforce) # accelerating ggplot2
library(kableExtra) # complex tables
library(leaflet) # map
```

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_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
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
raw.data.confirmed <- read.csv('./data/time_series_covid19_confirmed_global.csv')
raw.data.deaths <- read.csv('./data/time_series_covid19_deaths_global.csv')
raw.data.recovered <- read.csv('./data/time_series_covid19_recovered_global.csv')

dim(raw.data.confirmed)
```

```
## [1] 256  75
```

Each dataset has 256 rows, corresponding to country/region/province/state. It has 75 columns. Starting from column 5, each column corresponds to a single day. Here we have a look at the first 10 rows and the first 10 columns.

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

```
raw.data.confirmed[1:10, 1:10] %>%
  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
	Afghanistan	33.0000	65.0000	0	0	0	0	0	0
	Albania	41.1533	20.1683	0	0	0	0	0	0
	Algeria	28.0339	1.6596	0	0	0	0	0	0
	Andorra	42.5063	1.5218	0	0	0	0	0	0
	Angola	-11.2027	17.8739	0	0	0	0	0	0
	Antigua and Barbuda	17.0608	-61.7964	0	0	0	0	0	0
	Argentina	-38.4161	-63.6167	0	0	0	0	0	0
	Armenia	40.0691	45.0382	0	0	0	0	0	0
Australian Capital Territory	Australia	-35.4735	149.0124	0	0	0	0	0	0
New South Wales	Australia	-33.8688	151.2093	0	0	0	0	3	4

Below we check the time frame of the data.

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

```
## [1] "2020-01-22" "2020-04-01"
```

```
min.date <- min(dates)
max.date <- max(dates)
min.date.txt <- min.date %>% format('%d %b %Y')
max.date.txt <- max.date %>% format('%d %b %Y')
```

It shows that the data was last updated on 01 Apr 2020 UTC and all the stats and charts in this report are based on that data.

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, na.rm=T)) %>% as.data.frame()
  return(data)
}

## clean the three datasets
data.confirmed <- raw.data.confirmed %>% cleanData() %>% rename(confirmed=count)
data.deaths <- raw.data.deaths %>% cleanData() %>% rename(deaths=count)
```

```r
data.recovered <- raw.data.recovered %>% 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)
# data %<>% mutate(recovered = ifelse(is.na(recovered), lag(recovered, 1), recovered))

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

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

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

country	date	confirmed	deaths	recovered
China	2020-01-22	548	17	28
China	2020-01-23	643	18	30
China	2020-01-24	920	26	36
China	2020-01-25	1,406	42	39
China	2020-01-26	2,075	56	49
China	2020-01-27	2,877	82	58
China	2020-01-28	5,509	131	101
China	2020-01-29	6,087	133	120
China	2020-01-30	8,141	171	135
China	2020-01-31	9,802	213	214

There are 180 countries with confirmed COVID-19 cases, as of 01 Apr 2020 UTC.

3.2 Worldwide Cases

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.

```r
## counts for the whole world
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(current.confirmed = confirmed - deaths - recovered)
```

3.3 Daily Increases and Death Rates

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

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

1) there were much more deaths than recovered cases when the coronavirus broke out and when it was not contained, and
2) the daily number of death will decrease and that of recovered 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 recovered 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 recovered cases
## set NA to the increases on day1
n <- nrow(data)
day1 <- min(data$date)
data %<>% mutate(new.confirmed = ifelse(date == day1, NA, confirmed - lag(confirmed, n=1)),
                 new.deaths = ifelse(date == day1, NA, deaths - lag(deaths, n=1)),
                 new.recovered = ifelse(date == day1, NA, recovered - lag(recovered, n=1)))

## change negative number of new cases to zero
data %<>% mutate(new.confirmed = ifelse(new.confirmed < 0, 0, new.confirmed),
                 new.deaths = ifelse(new.deaths < 0, 0, new.deaths),
                 new.recovered = ifelse(new.recovered < 0, 0, new.recovered))

## death rate based on total deaths and recovered 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/recovered on every single day
data %<>% mutate(rate.daily = (100 * new.deaths / (new.deaths + new.recovered)) %>% round(1))


## convert from wide to long format, for drawing area plots
data.long <- data %>%
  select(c(country, date, confirmed, current.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=recode_factor(type, confirmed='Total Confirmed',
                                         current.confirmed='Current Confirmed',
                                         recovered='Recovered',
                                         deaths='Deaths'))


## convert from wide to long format, for drawing area plots
rates.long <- data %>%
  # filter(country %in% top.countries) %>%
  select(c(country, date, rate.upper, rate.lower, rate.daily)) %>%
  # mutate(country=factor(country, levels=top.countries)) %>%
  gather(key=type, value=count, -c(country, date))
# set factor levels to show them in a desirable order
```

```
rates.long %<>% mutate(type=recode_factor(type, rate.daily='Daily',
                                          rate.lower='Lower bound',
                                          rate.upper='Upper bound'))
```

4 Worldwide Cases

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

4.1 World Map

Below is a world map of vconfirmed cases. An interactive map can be created if running the code in R or RStudio, or knitting it into a HTML file.

```
## select last column, which is the number of latest confirmed cases
x <- raw.data.confirmed
x$confirmed <- x[, ncol(x)]
x %<>% select(c(Country.Region, Province.State, Lat, Long, confirmed)) %>%
  mutate(txt=paste0(Country.Region, ' - ', Province.State, ': ', confirmed))

m <- leaflet(width=1200, height=800) %>% addTiles()
# circle marker (units in pixels)
m %<>% addCircleMarkers(x$Long, x$Lat,
                        radius=2+log2(x$confirmed), stroke=F,
                        color='red', fillOpacity=0.3,
                        popup=x$txt)

# world
m
```

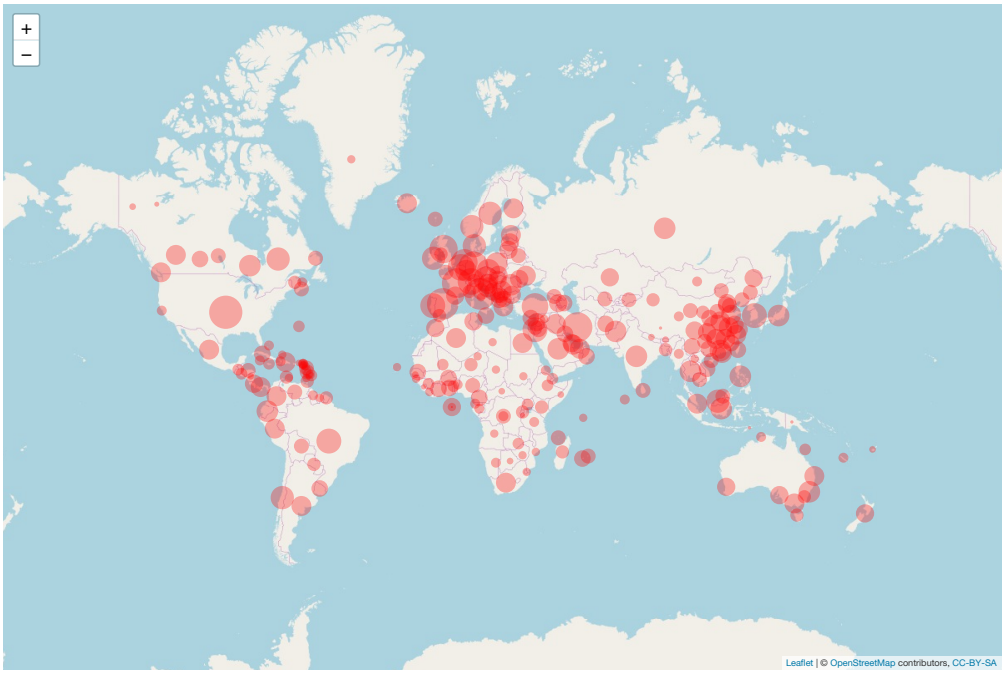


Figure 1: World Map

Views of some specific countries or regions can be produced with the script below.

```
## China
m %>% setView(95, 35, zoom=4)
## Australia and New Zealand
m %>% setView(135, -27, zoom=4)
## US and Canada
m %>% setView(-105, 40, zoom=4)
## Europe
m %>% setView(10, 50, zoom=4)
```

4.2 Number of Cases

In the rest of this section, we will focuse on the cases worldwide. Similar analysis for a single country can be done by filter the data with the corresponding country name.

```
# data %<>% filter(country=='China')
# data %<>% filter(country=='Australia')
world.long <- data.long %>% filter(country == 'World')

## cases - 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('Numbers of 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('Numbers of 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')

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

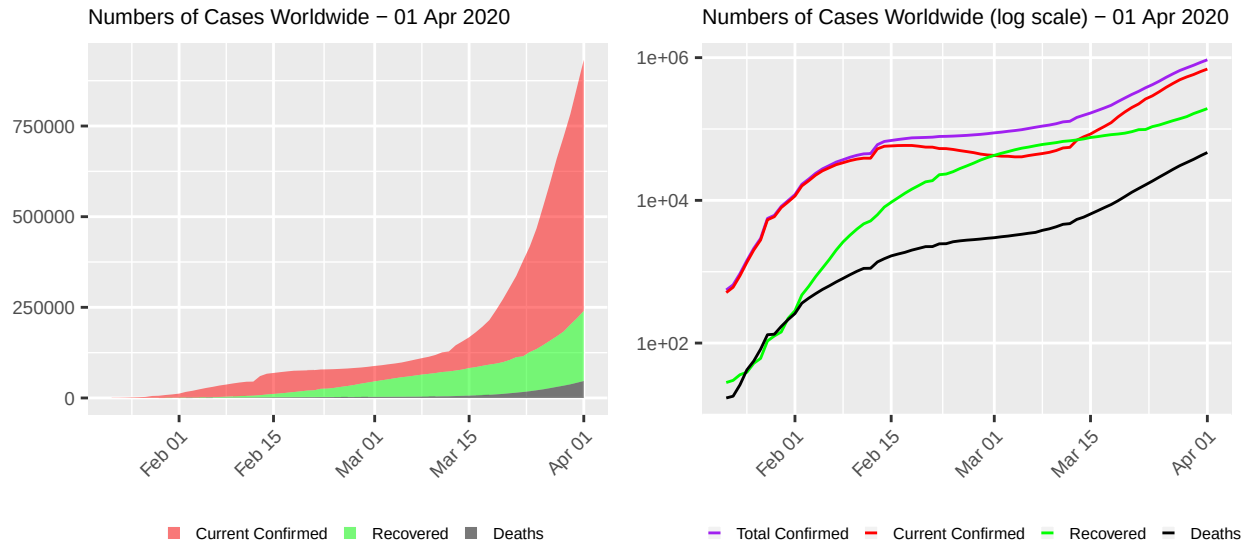


Figure 2: COVID-19 Cases Worldwide

4.3 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=current.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=new.confirmed)) +
  geom_point() + geom_smooth() +
  xlab('') + ylab('Count') + labs(title='Daily New Confirmed Cases') +
  theme(axis.text.x=element_text(angle=45, hjust=1))
## show two plots side by side
grid.arrange(plot1, plot2, ncol=2)
```

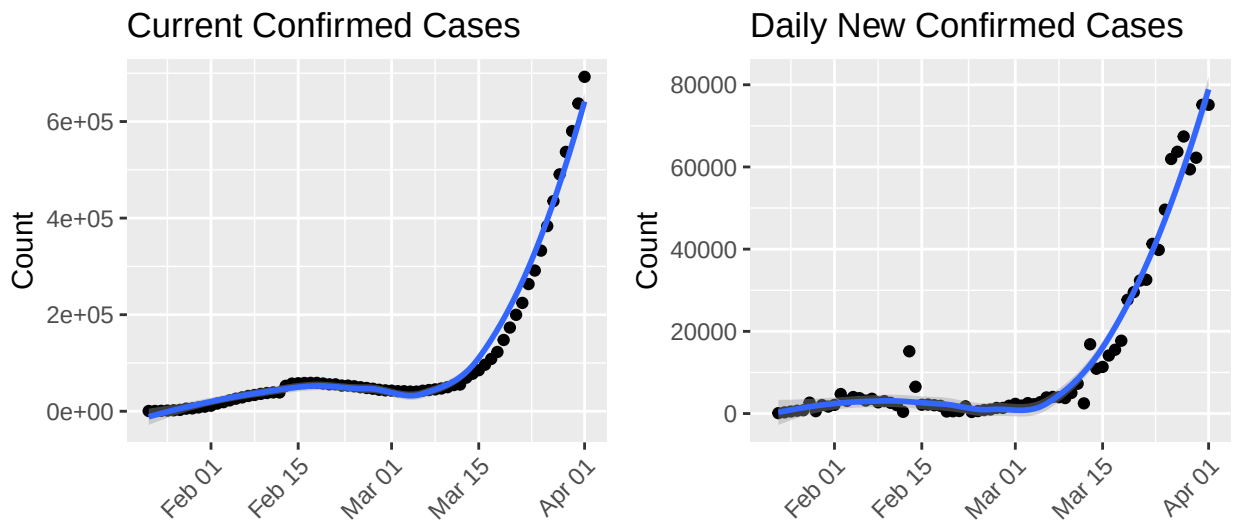


Figure 3: Current Confirmed Cases

Figure 3 shows the numbers of current (see left chart) and new (see right chart) confirmed cases. The blue lines are smoothed conditional means and the grey band around them show the 95% confidence interval.

4.4 Deaths and Recovered Cases

```
## a scatter plot with a smoothed line and vertical x-axis labels
plot1 <- ggplot(data.world, aes(x=date, y=deaths)) +
  geom_point() + geom_smooth() +
  xlab('') + ylab('Count') + labs(title='Accumulative Deaths') +
  theme(axis.text.x=element_text(angle=45, hjust=1))
plot2 <- ggplot(data.world, aes(x=date, y=recovered)) +
  geom_point() + geom_smooth() +
  xlab('') + ylab('Count') + labs(title='Accumulative Recovered Cases') +
  theme(axis.text.x=element_text(angle=45, hjust=1))
plot3 <- ggplot(data.world, aes(x=date, y=new.deaths))  +
  geom_point() + geom_smooth()  +
  xlab('') + ylab('Count') + labs(title='New Deaths') +
  theme(axis.text.x=element_text(angle=45, hjust=1))
plot4 <- ggplot(data.world, aes(x=date, y=new.recovered)) +
  geom_point() + geom_smooth()  +
  xlab('') + ylab('Count') + labs(title='New Recovered Cases') +
  theme(axis.text.x=element_text(angle=45, hjust=1))
## show four plots together, with 2 plots in each row
grid.arrange(plot1, plot2, plot3, plot4, nrow=2)
```

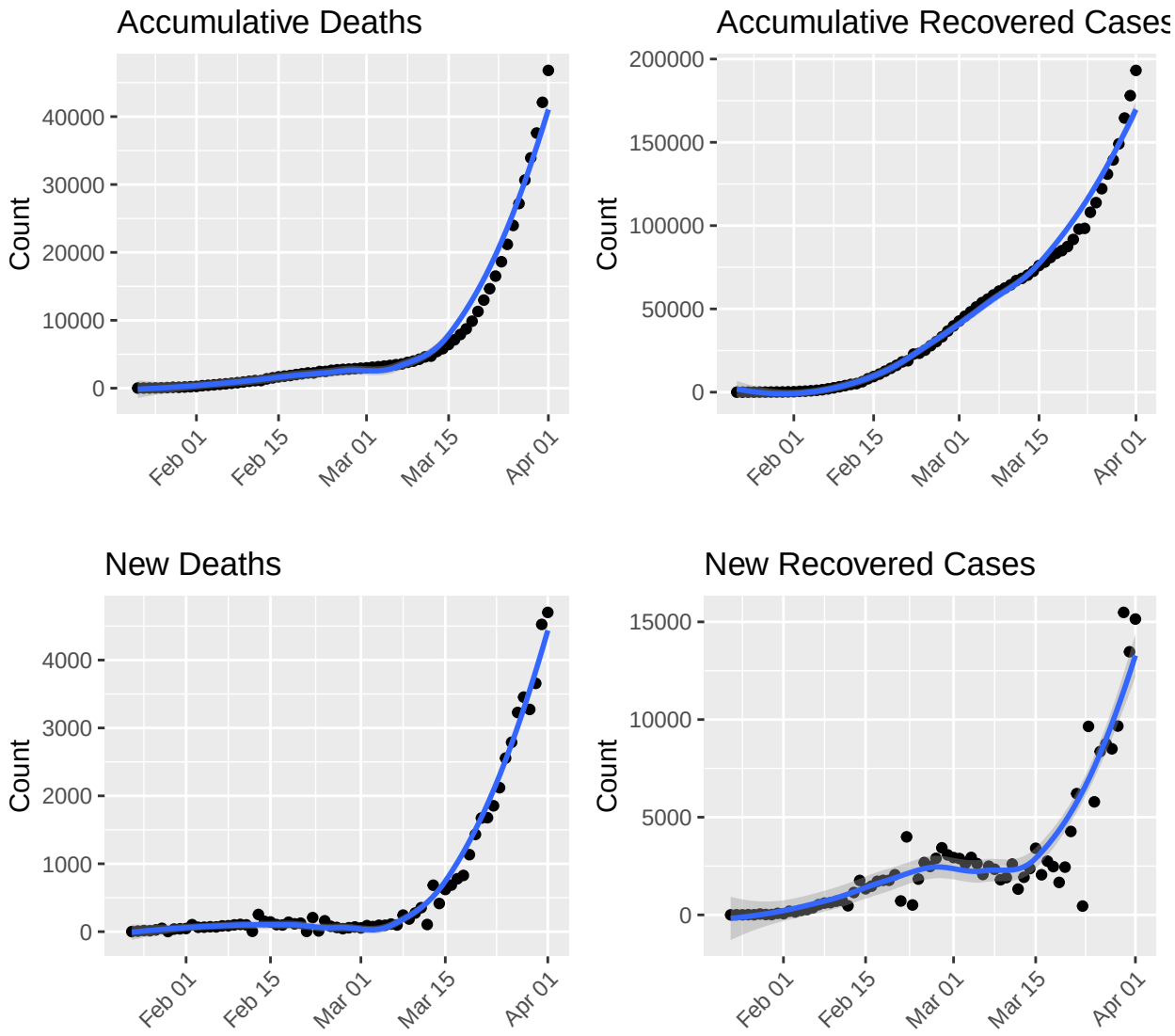


Figure 4: Deaths and Recovered Cases

4.5 Death Rates

Figure 5 shows death rates caculated in three different ways (see Section 3.3 for details). The left chart shows the death rates from 22 Jan 2020 to 01 Apr 2020 and the right one is a zoom-in view of the rates in last two weeks.

In the right chart, the upper bound (in blue) is decreasing, as there will be more recovered 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 01 Apr 2020 UTC, it will be between 5% and 19.5%.

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 recent outbreak of coronavirus in Italy, Iran and some other European countries.

```
## three death rates
plot1 <- ggplot(data.world, 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('') + ylab('Death Rate (%)') + labs(title='Overall') +
  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))
## focusing on last 2 weeks
y.max <- data.world[n-(14:0), ] %>% select(rate.upper, rate.lower, rate.daily) %>% max()
plot2 <- ggplot(data.world[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('') + ylab('Death Rate (%)') + labs(title='Last two weeks') +
  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, y.max))
grid.arrange(plot1, plot2, ncol=2)
```

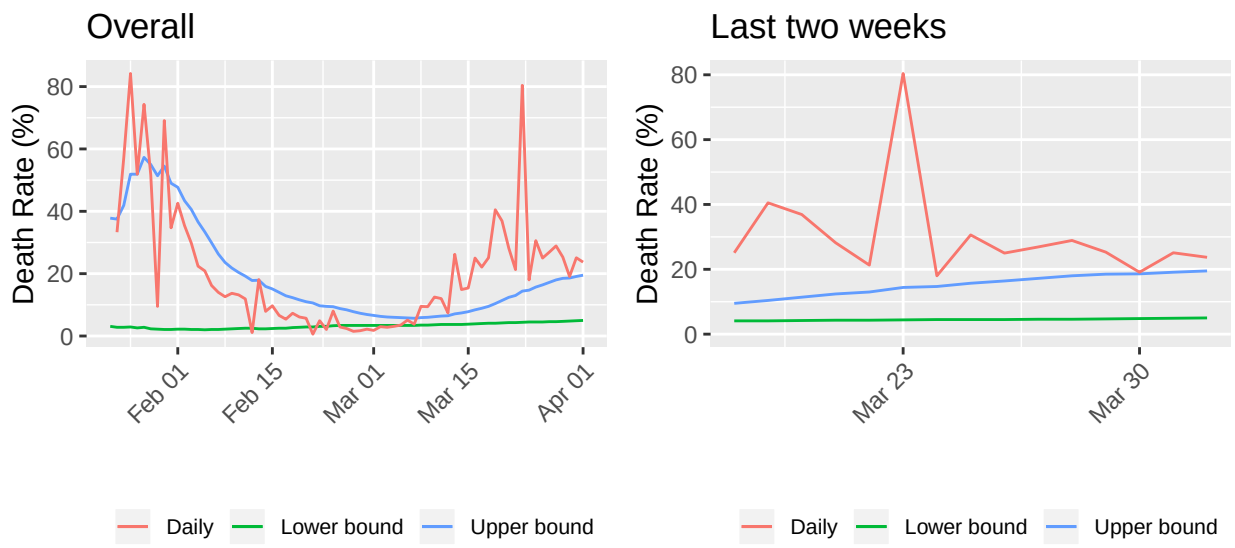


Figure 5: Death Rate

5 Top Twenty Countries

Next, we will have a look at the top 20 countries in total confirmed cases.

```
## ranking by confirmed cases
data.latest.all <- data %>% filter(date == max(date)) %>%
  select(country, date,
         confirmed, new.confirmed, current.confirmed,
         recovered, deaths, new.deaths, death.rate=rate.lower) %>%
  mutate(ranking = dense_rank(desc(confirmed)))
```

```r
k <- 20
## top 20 countries: 21 incl. 'World'
top.countries <- data.latest.all %>% filter(ranking <= k + 1) %>%
  arrange(ranking) %>% pull(country) %>% as.character()
top.countries %>% setdiff('World') %>% print()
```

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

```r
## 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), new.confirmed=sum(new.confirmed),
            current.confirmed=sum(current.confirmed),
            recovered=sum(recovered), deaths=sum(deaths), new.deaths=sum(new.deaths)) %>%
  mutate(death.rate=(100 * deaths/confirmed) %>% round(1))
data.latest %<>% select(c(country, confirmed, deaths, death.rate,
                 new.confirmed, new.deaths, current.confirmed))
```

```r
data.latest %>% mutate(death.rate=death.rate %>% format(nsmall=1) %>% paste0('%')) %>%
  kable('latex', booktabs=T, row.names=T, align=c('l', rep('r', 6)),
        caption=paste0('Cases in Top 20 Countries - ', max.date.txt,
                       '. See a complete list of all infected countries at the end of this report.'),
        format.args=list(big.mark=',')) %>%
  kable_styling(font_size=7, latex_options=c('striped', 'hold_position', 'repeat_header'))
```

```r
## convert from wide to long format, for drawing area plots
data.latest.long <- data.latest %>% filter(country!='World') %>%
  gather(key=type, value=count, -country)
## set factor levels to show them with proper text and in a desirable order
data.latest.long %<>% mutate(type=recode_factor(type,
                                     confirmed='Total Confirmed',
                                     deaths='Total Deaths',
                                     death.rate='Death Rate (%)',
                                     new.confirmed='New Confirmed (compared with one day before)',
                                     new.deaths='New Deaths (compared with one day before)',
                                     current.confirmed='Current Confirmed'))
```

```r
## 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_fill_discrete(name='Country', labels=aes(count)) +
  theme(legend.title=element_blank(),
        legend.position='none',
        plot.title=element_text(size=11),
```

Table 3: Cases in Top 20 Countries - 01 Apr 2020. See a complete list of all infected countries at the end of this report.

	country	confirmed	deaths	death.rate	new.confirmed	new.deaths	current.confirmed
1	World	932,605	46,809	5.0%	75,118	4,702	692,619
2	US	213,372	4,757	2.2%	25,200	884	200,141
3	Italy	110,574	13,155	11.9%	4,782	727	80,572
4	Spain	104,118	9,387	9.0%	8,195	923	72,084
5	China	82,361	3,316	4.0%	82	7	2,640
6	Germany	77,872	920	1.2%	6,064	145	58,252
7	France	57,749	4,043	7.0%	4,922	511	42,653
8	Iran	47,593	3,036	6.4%	2,988	138	29,084
9	United Kingdom	29,865	2,357	7.9%	4,384	564	27,329
10	Switzerland	17,768	488	2.7%	1,163	55	14,313
11	Turkey	15,679	277	1.8%	2,148	63	15,069
12	Belgium	13,964	828	5.9%	1,189	123	11,004
13	Netherlands	13,696	1,175	8.6%	1,029	135	12,261
14	Austria	10,711	146	1.4%	531	18	9,129
15	Korea, South	9,887	165	1.7%	101	3	4,155
16	Canada	9,560	109	1.1%	1,033	8	8,127
17	Portugal	8,251	187	2.3%	808	27	8,021
18	Brazil	6,836	240	3.5%	1,119	39	6,469
19	Israel	6,092	26	0.4%	734	6	5,825
20	Sweden	4,947	239	4.8%	512	59	4,605
21	Norway	4,863	44	0.9%	222	5	4,806
22	Others	86,847	1,914	2.2%	7,912	262	76,080

```
      axis.text=element_text(size=7),
      axis.text.x=element_text(angle=45, hjust=1)) +
facet_wrap(~type, ncol=1, scales='free_y')
```

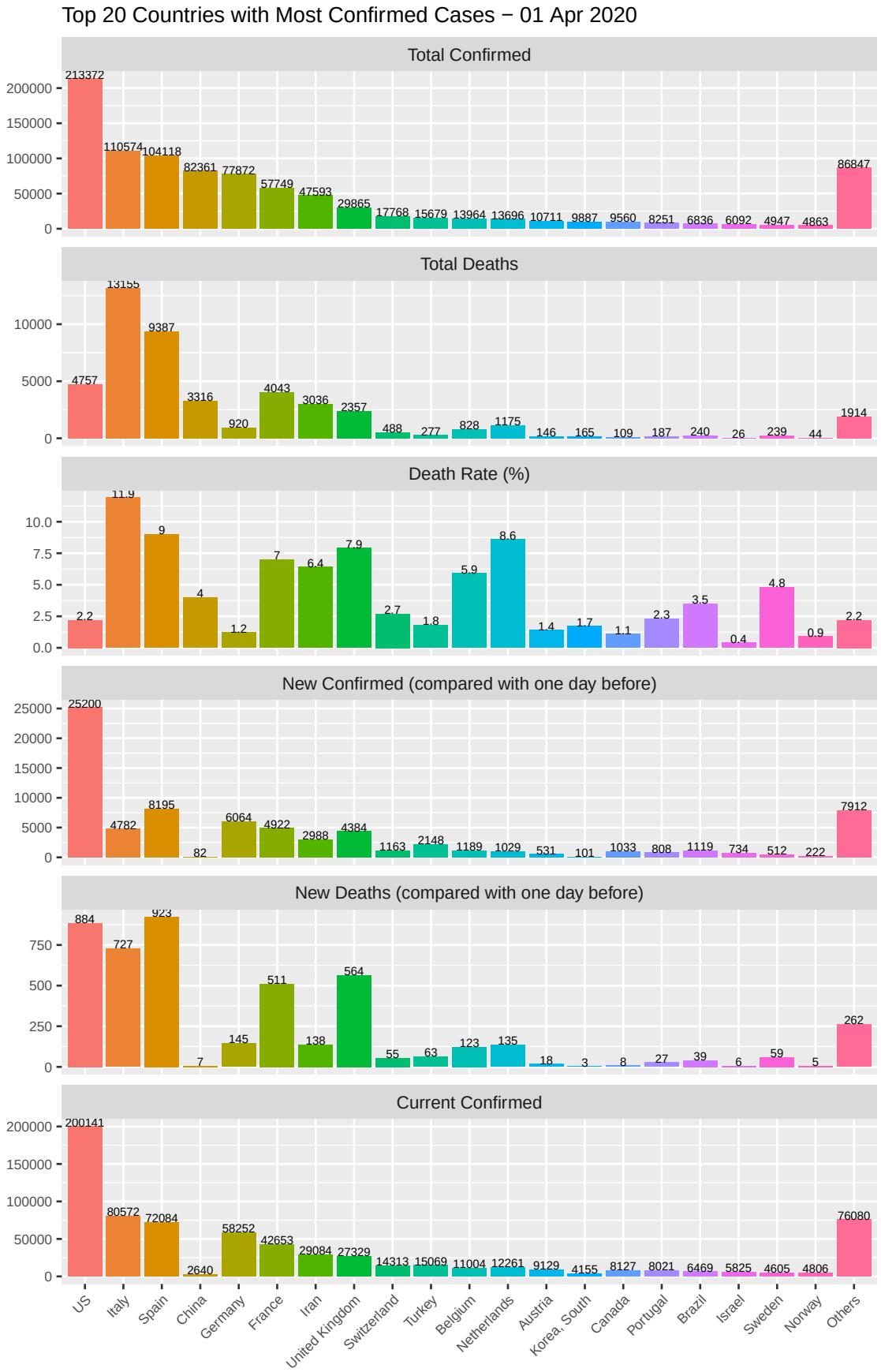
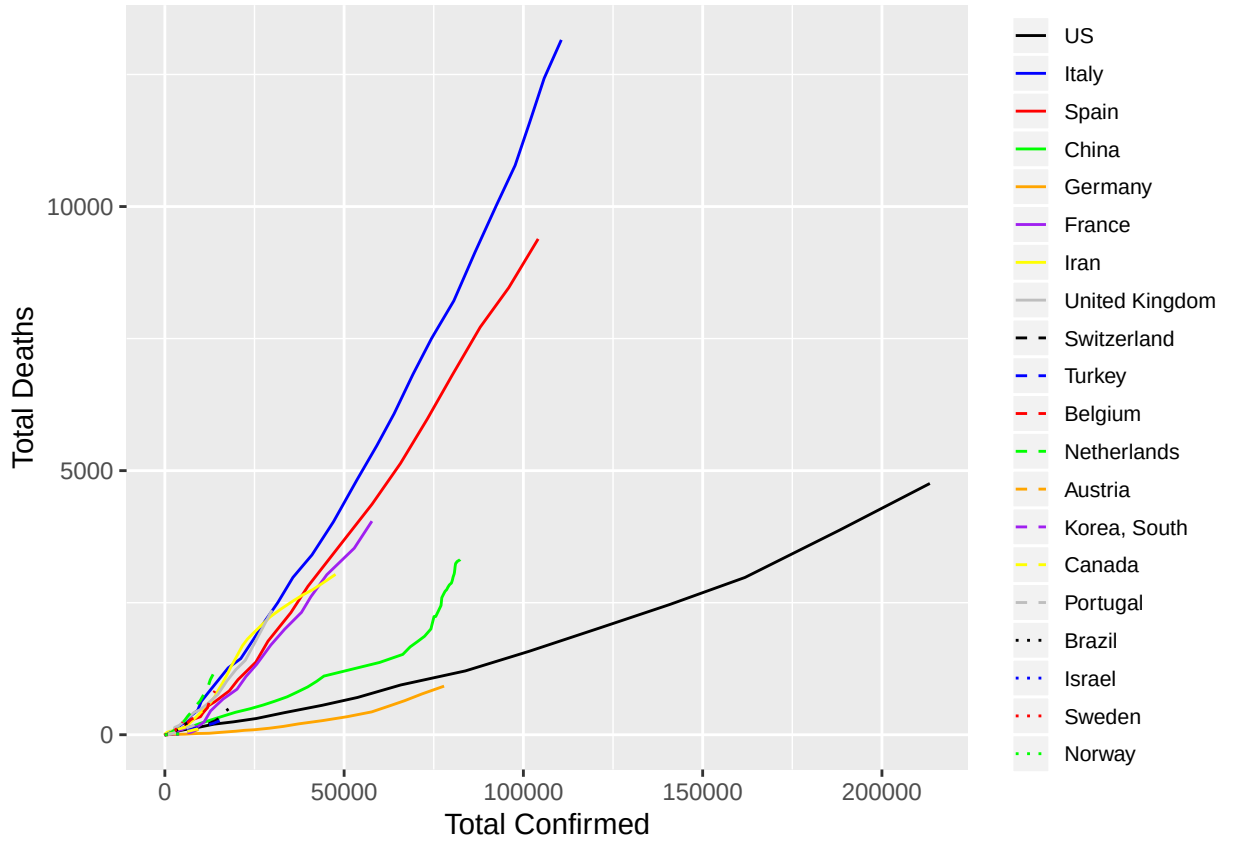


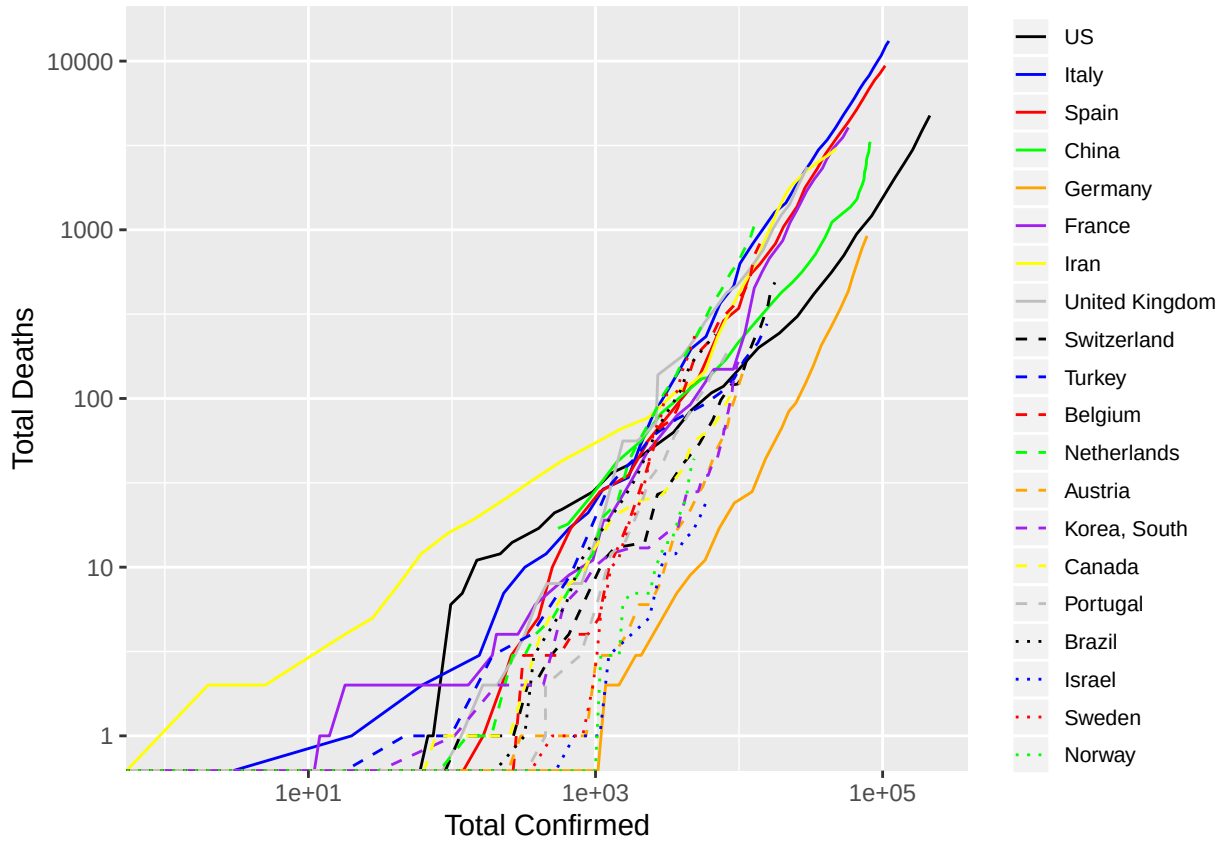
Figure 6: Top 20 Countries with Most Confirmed Cases

5.1 Confirmed vs Deaths

```
# linetypes <- rep(c("dotted", "dashed", "solid"), each=8)
# colors <- rep(c('grey', 'yellow', 'purple', 'orange', 'green', 'red', 'blue', 'black'), 3)
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)))
p <- 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'))
p
```




```
p + scale_x_log10() + scale_y_log10()
```



The two figures below show the numbers of confirmed cases and deaths of top 20 countries, as well as the death rates up to 01 Apr 2020.

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

plot1 <- df %>% ggplot(aes(x=confirmed, y=deaths, col=death.rate, size=current.confirmed)) +
  scale_size(name='Current 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='#56B1F7', high='#132B43') +
  scale_x_log10() + scale_y_log10()

plot2 <- df %>% ggplot(aes(x=new.confirmed, y=new.deaths, col=death.rate, size=current.confirmed)) +
  scale_size(name='Current 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('New Confirmed') + ylab('New Deaths') +
  labs(col="Death Rate (%)") +
  scale_color_gradient(low='#56B1F7', high='#132B43') +
  scale_x_log10() + scale_y_log10()

grid.arrange(plot1, plot2, ncol=1)
```

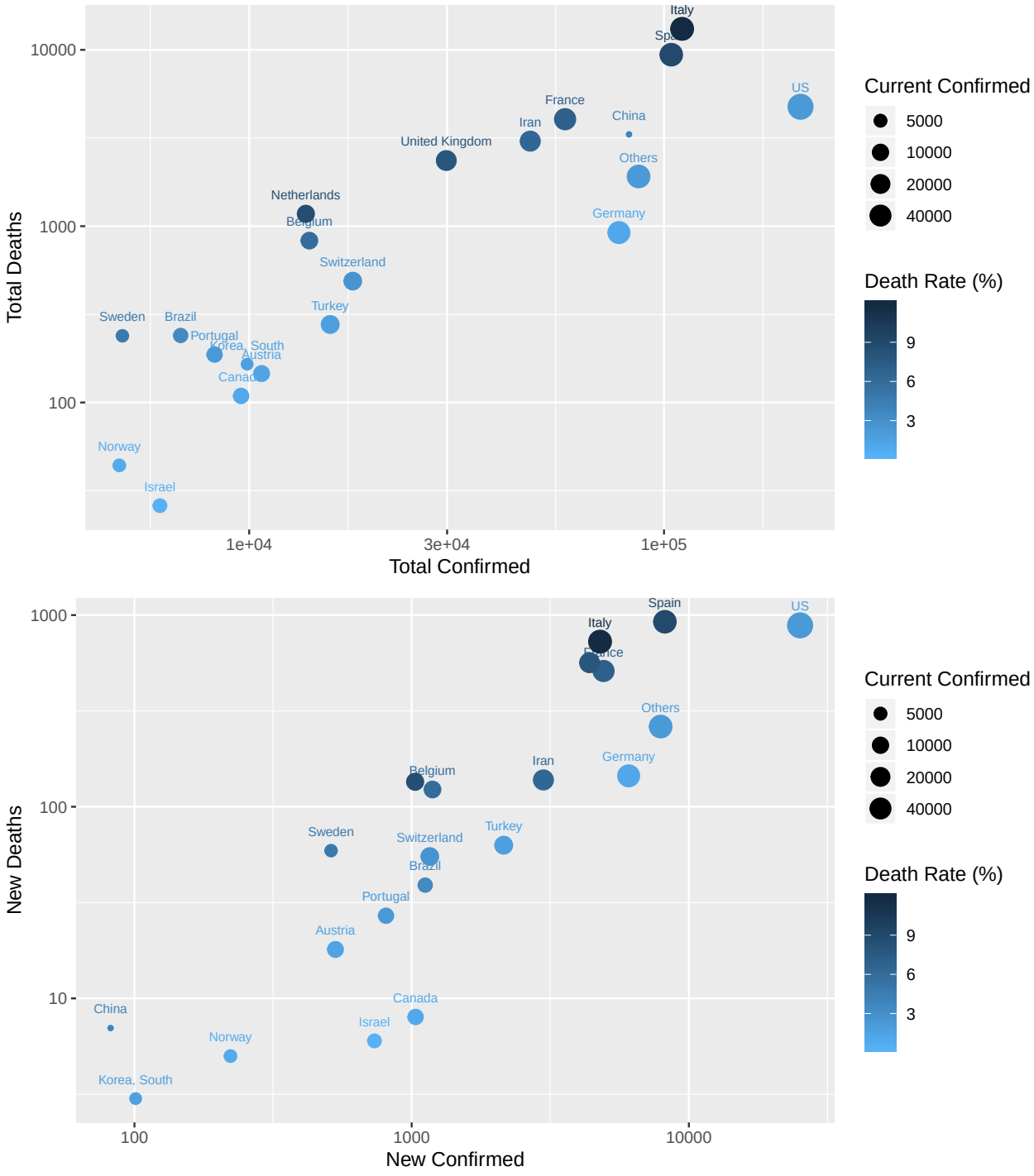


Figure 7: Top 20 Countries

5.2 Comparison across Countries

The area plots blow show the numbers of dead, recovered, total and current confimed cases. Note that, in the area plot, the total number of total confirmed cases is represented by the total areas of current confirmed, recovered and dead.

```r
## plot: cases by type
df <- data.long %>% filter(country %in% top.countries) %<>%
  mutate(country=country %>% factor(levels=c(top.countries)))

p <- df %>% filter(country != 'World') %>%
  ggplot(aes(x=date, y=count)) + xlab('') + ylab('Count')  +
  theme(legend.title=element_blank(),
        legend.text=element_text(size=8),
        legend.key.size=unit(0.5, 'cm'),
        plot.title=element_text(size=11),
        axis.text.x=element_text(angle=45, hjust=1)) +
  facet_wrap(~type, ncol=2, scales='free_y')

## area plot
plot1 <- p + geom_area(aes(fill=country)) +
  labs(title=paste0('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)
```

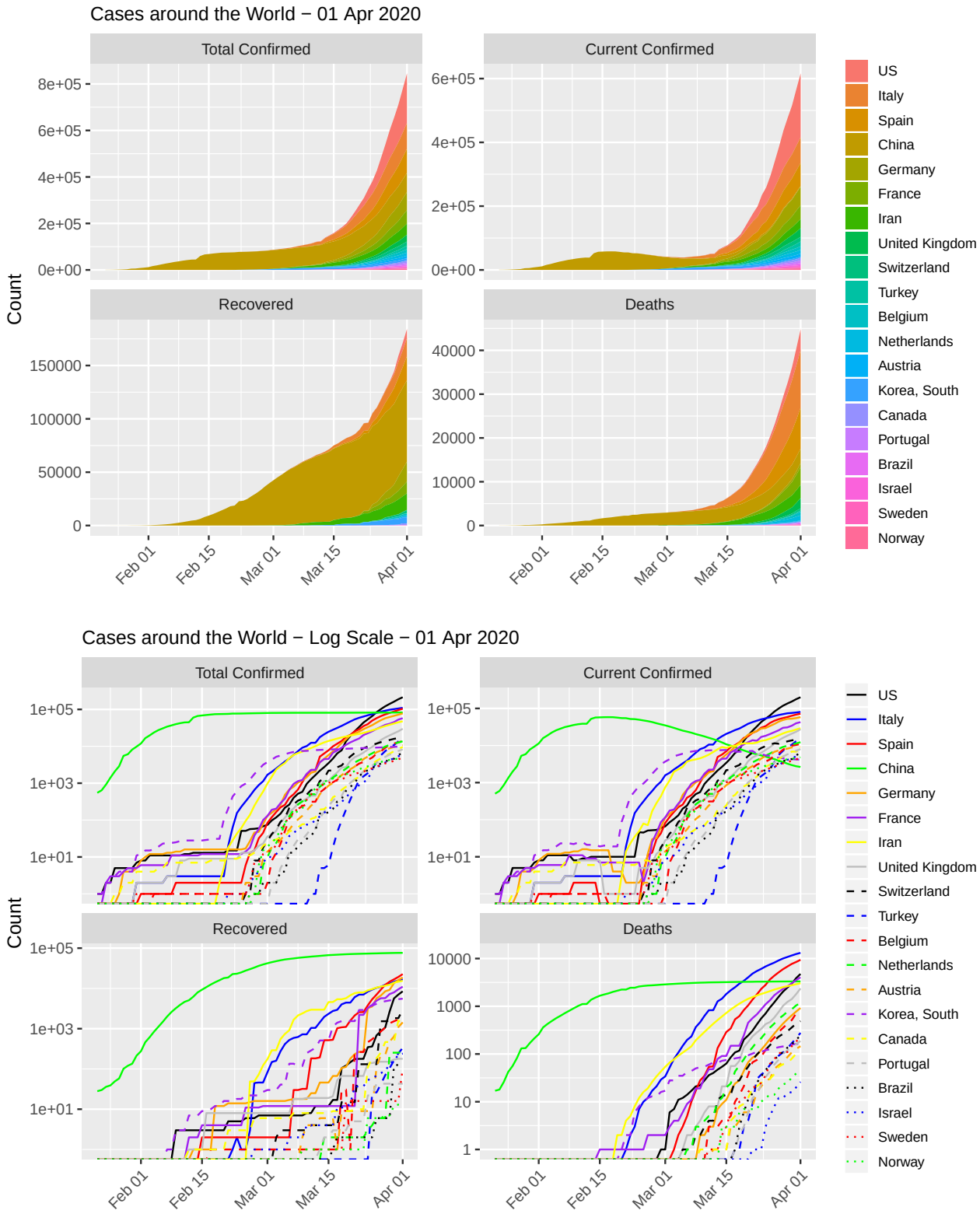


Figure 8: Cases around the World

```
## 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=8),
        legend.key.size=unit(0.5, 'cm'),
        plot.title=element_text(size=11),
        axis.text.x=element_text(angle=45, hjust=1)) +
  facet_wrap(~type, ncol=2, scales='free_y')
p + geom_area(aes(fill=country)) +
  labs(title=paste0('Cases around the World (excl. China) - ', max.date.txt))
```

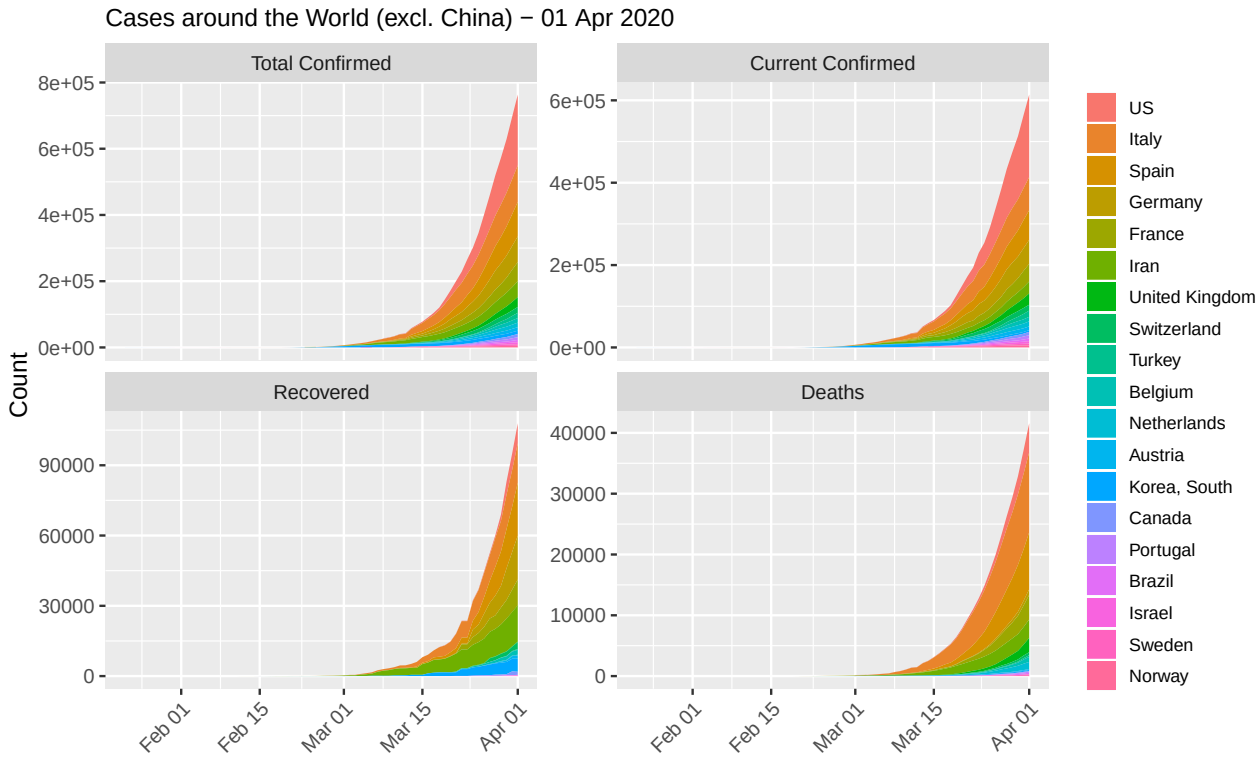


Figure 9: Cases around the World (excl. China)

```
## if Australia in not in top 20, 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 - area plot
df %>% filter(country != 'World' & type != 'Total Confirmed') %>%
  ggplot(aes(x=date, y=count, fill=type)) +
  geom_area(alpha=0.5) +
  # xlab('') + ylab('') +
  labs(title=paste0('Numbers of COVID-19 Cases in Top 20 Countries - ',
                    max.date.txt)) +
  scale_fill_manual(values=c('red', 'green', 'black')) +
  theme(legend.title=element_blank(), legend.position='bottom',
```

```
      plot.title = element_text(size=12),
      axis.title.x=element_blank(),
      axis.title.y=element_blank(),
      legend.key.size=unit(0.4, 'cm'),
      # legend.text=element_text(size=7),
      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, scales='free_y')
```

Figure 10: COVID-19 Cases in Top 20 Countries. Ordered descendingly by number of confirmed cases.

```r
## cases by country - line plot - log scale
p <- df %>% filter(country != 'World') %>%
  ggplot(aes(x=date, y=count, color=type)) +
  geom_line() +
  labs(title=paste0('Numbers of COVID-19 Cases in Top 20 Countries (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=12),
        axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        legend.key.size=unit(0.4, 'cm'),
        # legend.text=element_text(size=7),
        strip.text.x=element_text(size=7),
        axis.text=element_text(size=7),
        axis.text.x=element_text(angle=45, hjust=1)) +
  scale_y_continuous(trans='log10')
p + facet_wrap(~country, ncol=4, scales='free_y')
```

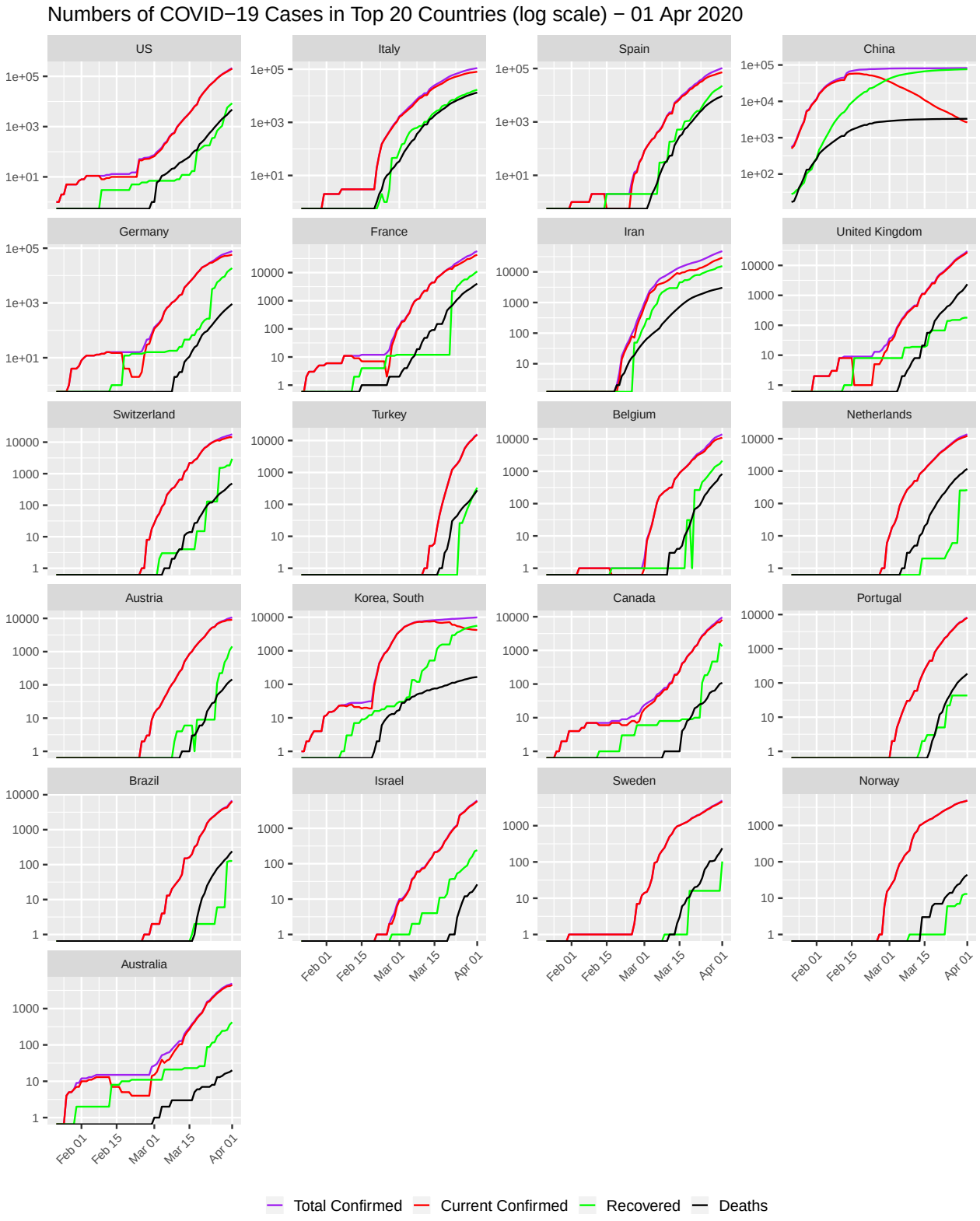


Figure 11: COVID-19 Cases Top 20 Countries (log scale). Ordered descendingly by number of confirmed cases.

```
## plot over multiple pages
# p + facet_wrap_paginate(~country, nrow=4, ncol=3, page=1, scales='free_y')
# p + facet_wrap_paginate(~country, nrow=4, ncol=3, page=2, scales='free_y')
```

Figures 10 and 11 show that the coronavirus seems to be under control in China, with an increase of recovered cases (in green) every day and a shrinking of the current 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.

5.3 Death Rates

```
## three death rates
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)
```

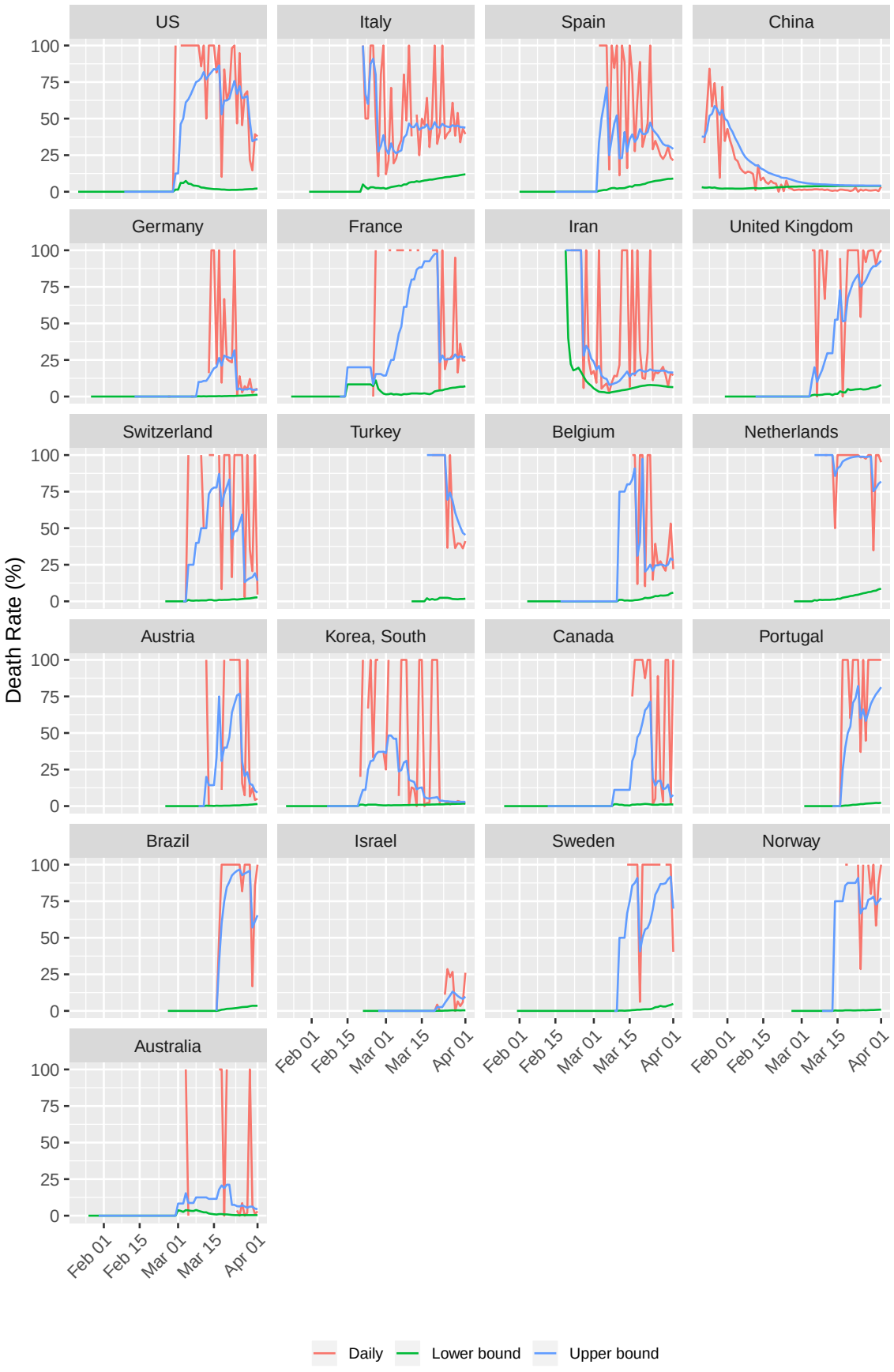


Figure 12: Death Rates

5.4 Countries with Highest Death Rates

Below are a list of top 20 countries with the highest death rates out of countires having 1000+ confirmed cases.

```
## sort the latest data by death rate, and if tie, by confirmed
df <- data %>% filter(date == max(date) & country != 'World' & confirmed >= 1000) %>%
  select(country, confirmed, new.confirmed, current.confirmed,
         recovered, deaths, new.deaths, 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.args=list(big.mark=',')) %>%
  kable_styling(font_size=7, latex_options=c('striped', 'hold_position', 'repeat_header'))
```

Table 4: Top 20 Countries with Highest Death Rates - 01 Apr 2020

	country	confirmed	new.confirmed	current.confirmed	recovered	deaths	new.deaths	death.rate
1	Italy	110,574	4,782	80,572	16,847	13,155	727	11.9%
2	Indonesia	1,677	149	1,417	103	157	21	9.4%
3	Spain	104,118	8,195	72,084	22,647	9,387	923	9.0%
4	Netherlands	13,696	1,029	12,261	260	1,175	135	8.6%
5	United Kingdom	29,865	4,384	27,329	179	2,357	564	7.9%
6	France	57,749	4,922	42,653	11,053	4,043	511	7.0%
7	Iran	47,593	2,988	29,084	15,473	3,036	138	6.4%
8	Belgium	13,964	1,189	11,004	2,132	828	123	5.9%
9	Sweden	4,947	512	4,605	103	239	59	4.8%
10	Dominican Republic	1,284	175	1,218	9	57	6	4.4%
11	Philippines	2,311	227	2,165	50	96	8	4.2%
12	China	82,361	82	2,640	76,405	3,316	7	4.0%
13	Romania	2,460	215	2,116	252	92	10	3.7%
14	Brazil	6,836	1,119	6,469	127	240	39	3.5%
15	Greece	1,415	101	1,313	52	50	1	3.5%
16	Ecuador	2,748	508	2,597	58	93	18	3.4%
17	Denmark	3,290	251	2,215	971	104	14	3.2%
18	India	1,998	601	1,792	148	58	23	2.9%
19	Peru	1,323	258	891	394	38	8	2.9%
20	Argentina	1,054	0	778	248	28	1	2.7%

6 Conclusions

As of 01 Apr 2020, there are 180 countries with confirmed COVID-19 cases. It seems to be contained in China, but starts to break out in rest of the world. The current death rate is in between 5% and 19.5%, but it is likely to change dramatically with the breakout in many countries, such as European countries.

Appendix A. Processed Data

Blow is the processed data for this analysis.

Appendix A.1 COVID-19 Cases Worldwide

```
## sort by date descendingly and re-order columns
data.world %<>% arrange(desc(date)) %>%
```

```
  select(c(date, confirmed, deaths, recovered, current.confirmed,
           new.confirmed, new.deaths, new.recovered, rate.lower, rate.upper, rate.daily))
## output as a table
data.world %>% kable('latex', booktabs=T, longtable=T, caption='Cases in the Whole World',
               format.args=list(big.mark=',')) %>%
    kable_styling(font_size=4, latex_options=c('striped', 'hold_position', 'repeat_header'))
```

Table 5: Cases in the Whole World

date	confirmed	deaths	recovered	current.confirmed	new.confirmed	new.deaths	new.recovered	rate.lower	rate.upper	rate.daily
2020-04-01	932,605	46,809	193,177	692,619	75,118	4,702	15,143	5.0	19.5	23.7
2020-03-31	857,487	42,107	178,034	637,346	75,122	4,525	13,468	4.9	19.1	25.1
2020-03-30	782,365	37,582	164,566	580,217	62,248	3,657	15,484	4.8	18.6	19.1
2020-03-29	720,117	33,925	149,082	537,110	59,411	3,273	9,667	4.7	18.5	25.3
2020-03-28	660,706	30,652	139,415	490,639	67,415	3,454	8,500	4.6	18.0	28.9
2020-03-27	593,291	27,198	130,915	435,178	63,700	3,228	8,765	4.6	17.2	26.9
2020-03-26	529,591	23,970	122,150	383,471	61,938	2,789	8,363	4.5	16.4	25.0
2020-03-25	467,653	21,181	113,787	332,685	49,608	2,556	5,787	4.5	15.7	30.6
2020-03-24	418,045	18,625	108,000	291,420	39,810	2,120	9,649	4.5	14.7	18.0
2020-03-23	378,235	16,505	98,351	263,379	41,282	1,854	452	4.4	14.4	80.4
2020-03-22	336,953	14,651	97,899	224,403	32,557	1,678	6,207	4.3	13.0	21.3
2020-03-21	304,396	12,973	91,692	199,731	32,361	1,674	4,272	4.3	12.4	28.2
2020-03-20	272,035	11,299	87,420	173,316	29,535	1,432	2,445	4.2	11.4	36.9
2020-03-19	242,500	9,867	84,975	147,658	27,679	1,134	1,663	4.1	10.4	40.5
2020-03-18	214,821	8,733	83,312	122,776	17,719	828	2,472	4.1	9.5	25.1
2020-03-17	197,102	7,905	80,840	108,357	15,528	779	2,752	4.0	8.9	22.1
2020-03-16	181,574	7,126	78,088	96,360	14,120	686	2,054	3.9	8.4	25.0
2020-03-15	167,454	6,440	76,034	84,980	11,353	621	3,410	3.8	7.8	15.4
2020-03-14	156,101	5,819	72,624	77,658	10,896	415	2,373	3.7	7.4	14.9
2020-03-13	145,205	5,404	70,251	69,550	16,853	684	1,927	3.7	7.1	26.2
2020-03-12	128,352	4,720	68,324	55,308	2,477	105	1,321	3.7	6.5	7.4
2020-03-11	125,875	4,615	67,003	54,257	7,255	353	2,599	3.7	6.4	12.0
2020-03-10	118,620	4,262	64,404	49,954	5,030	274	1,910	3.6	6.2	12.5
2020-03-09	113,590	3,988	62,494	47,108	3,769	186	1,800	3.5	6.0	9.4
2020-03-08	109,821	3,802	60,694	45,325	3,974	244	2,336	3.5	5.9	9.5
2020-03-07	105,847	3,558	58,358	43,931	4,046	98	2,493	3.4	5.7	3.8
2020-03-06	101,801	3,460	55,865	42,476	3,915	112	2,069	3.4	5.8	5.1
2020-03-05	97,886	3,348	53,796	40,742	2,766	94	2,626	3.4	5.9	3.5
2020-03-04	95,120	3,254	51,170	40,696	2,280	94	2,942	3.4	6.0	3.1
2020-03-03	92,840	3,160	48,228	41,452	2,534	75	2,626	3.4	6.1	2.8
2020-03-02	90,306	3,085	45,602	41,619	1,937	89	2,886	3.4	6.3	3.0
2020-03-01	88,369	2,996	42,716	42,657	2,358	55	2,934	3.4	6.6	1.8
2020-02-29	86,011	2,941	39,782	43,288	1,899	69	3,071	3.4	6.9	2.2
2020-02-28	84,112	2,872	36,711	44,529	1,366	58	3,434	3.4	7.3	1.7
2020-02-27	82,746	2,814	33,277	46,655	1,358	44	2,893	3.4	7.8	1.5
2020-02-26	81,388	2,770	30,384	48,234	982	62	2,479	3.4	8.4	2.4
2020-02-25	80,406	2,708	27,905	49,793	845	79	2,678	3.4	8.8	2.9
2020-02-24	79,561	2,629	25,227	51,705	603	160	1,833	3.3	9.4	8.0
2020-02-23	78,958	2,469	23,394	53,095	386	11	508	3.1	9.5	2.1
2020-02-22	78,572	2,458	22,886	53,228	1,753	207	3,996	3.1	9.7	4.9
2020-02-21	76,819	2,251	18,890	55,678	622	4	713	2.9	10.6	0.6
2020-02-20	76,197	2,247	18,177	55,773	558	125	2,056	2.9	11.0	5.7
2020-02-19	75,639	2,122	16,121	57,396	503	115	1,769	2.8	11.6	6.1
2020-02-18	75,136	2,007	14,352	58,777	1,878	139	1,769	2.7	12.3	7.3
2020-02-17	73,258	1,868	12,583	58,807	2,034	98	1,718	2.5	12.9	5.4
2020-02-16	71,224	1,770	10,865	58,589	2,194	104	1,470	2.5	14.0	6.6
2020-02-15	69,030	1,666	9,395	57,969	2,145	143	1,337	2.4	15.1	9.7
2020-02-14	66,885	1,523	8,058	57,304	6,517	152	1,763	2.3	15.9	7.9
2020-02-13	60,368	1,371	6,295	52,702	15,147	253	1,145	2.3	17.9	18.1
2020-02-12	45,221	1,118	5,150	38,953	419	5	467	2.5	17.8	1.1
2020-02-11	44,802	1,113	4,683	39,006	2,040	100	737	2.5	19.2	11.9
2020-02-10	42,762	1,013	3,946	37,803	2,612	107	702	2.4	20.4	13.2
2020-02-09	40,150	906	3,244	36,000	3,030	100	628	2.3	21.8	13.7
2020-02-08	37,120	806	2,616	33,698	2,729	87	605	2.2	23.6	12.6
2020-02-07	34,391	719	2,011	31,661	3,597	85	524	2.1	26.3	14.0
2020-02-06	30,794	634	1,487	28,673	3,159	70	363	2.1	29.9	16.2
2020-02-05	27,635	564	1,124	25,947	3,743	72	272	2.0	33.4	20.9
2020-02-04	23,892	492	852	22,548	4,011	66	229	2.1	36.6	22.4
2020-02-03	19,881	426	623	18,832	3,094	64	151	2.1	40.6	29.8
2020-02-02	16,787	362	472	15,953	4,749	103	188	2.2	43.4	35.4
2020-02-01	12,038	259	284	11,495	2,111	46	62	2.2	47.7	42.6
2020-01-31	9,927	213	222	9,492	1,693	42	79	2.1	49.0	34.7
2020-01-30	8,234	171	143	7,920	2,068	38	17	2.1	54.5	69.1
2020-01-29	6,166	133	126	5,907	588	2	19	2.2	51.4	9.5
2020-01-28	5,578	131	107	5,340	2,651	49	46	2.3	55.0	51.6
2020-01-27	2,927	82	61	2,784	809	26	9	2.8	57.3	74.3
2020-01-26	2,118	56	52	2,010	684	14	13	2.6	51.9	51.9
2020-01-25	1,434	42	39	1,353	493	16	3	2.9	51.9	84.2
2020-01-24	941	26	36	879	287	8	6	2.8	41.9	57.1
2020-01-23	654	18	30	606	99	1	2	2.8	37.5	33.3
2020-01-22	555	17	28	510				3.1	37.8	

Appendix A.2 Latest Cases by Country

```
## hightlight high death rates (if >= 5%) for those countries with 1000+ confirmed cases
data.latest.all %>% arrange(desc(confirmed)) %>% select(-c(date, ranking)) %>%
  mutate(death.rate = cell_spec(death.rate, "latex",
                                color = ifelse(confirmed >= 1000 & death.rate >= 5, "red", "black"),
                                bold = ifelse(confirmed >= 1000 & death.rate >= 5, T, F))) %>%
  kable(format='latex', escape=F, booktabs=T, longtable=T, row.names=T,
        caption=paste0('Cases by Country (', max.date.txt, ')'),
        format.args=list(big.mark=','), align=c('l', rep('r', 7))) %>%
  kable_styling(font_size=6, latex_options=c('striped', 'hold_position', 'repeat_header'))
```

Table 6: Cases by Country (01 Apr 2020)

	country	confirmed	new.confirmed	current.confirmed	recovered	deaths	new.deaths	death.rate
1	World	932,605	75,118	692,619	193,177	46,809	4,702	**5**
2	US	213,372	25,200	200,141	8,474	4,757	884	2.2
3	Italy	110,574	4,782	80,572	16,847	13,155	727	**11.9**
4	Spain	104,118	8,195	72,084	22,647	9,387	923	**9**
5	China	82,361	82	2,640	76,405	3,316	7	4
6	Germany	77,872	6,064	58,252	18,700	920	145	1.2
7	France	57,749	4,922	42,653	11,053	4,043	511	**7**
8	Iran	47,593	2,988	29,084	15,473	3,036	138	**6.4**
9	United Kingdom	29,865	4,384	27,329	179	2,357	564	**7.9**
10	Switzerland	17,768	1,163	14,313	2,967	488	55	2.7
11	Turkey	15,679	2,148	15,069	333	277	63	1.8
12	Belgium	13,964	1,189	11,004	2,132	828	123	**5.9**
13	Netherlands	13,696	1,029	12,261	260	1,175	135	**8.6**
14	Austria	10,711	531	9,129	1,436	146	18	1.4
15	Korea, South	9,887	101	4,155	5,567	165	3	1.7
16	Canada	9,560	1,033	8,127	1,324	109	8	1.1
17	Portugal	8,251	808	8,021	43	187	27	2.3
18	Brazil	6,836	1,119	6,469	127	240	39	3.5
19	Israel	6,092	734	5,825	241	26	6	0.4
20	Sweden	4,947	512	4,605	103	239	59	4.8
21	Norway	4,863	222	4,806	13	44	5	0.9
22	Australia	4,862	303	4,420	422	20	2	0.4
23	Czechia	3,508	200	3,408	61	39	8	1.1
24	Ireland	3,447	212	3,357	5	85	14	2.5
25	Denmark	3,290	251	2,215	971	104	14	3.2
26	Chile	3,031	293	2,781	234	16	4	0.5
27	Malaysia	2,908	142	2,218	645	45	2	1.5
28	Russia	2,777	440	2,563	190	24	7	0.9
29	Ecuador	2,748	508	2,597	58	93	18	3.4
30	Poland	2,554	243	2,464	47	43	10	1.7
31	Romania	2,460	215	2,116	252	92	10	3.7
32	Luxembourg	2,319	141	2,210	80	29	6	1.3
33	Philippines	2,311	227	2,165	50	96	8	4.2
34	Japan	2,178	225	1,649	472	57	1	2.6
35	Pakistan	2,118	180	1,997	94	27	1	1.3
36	India	1,998	601	1,792	148	58	23	2.9
37	Thailand	1,771	120	1,254	505	12	2	0.7
38	Saudi Arabia	1,720	157	1,440	264	16	6	0.9
39	Indonesia	1,677	149	1,417	103	157	21	**9.4**
40	Finland	1,446	28	1,419	10	17	0	1.2
41	Greece	1,415	101	1,313	52	50	1	3.5
42	South Africa	1,380	27	1,325	50	5	0	0.4
43	Peru	1,323	258	891	394	38	8	2.9
44	Dominican Republic	1,284	175	1,218	9	57	6	4.4
45	Iceland	1,220	85	993	225	2	0	0.2
46	Mexico	1,215	121	1,151	35	29	1	2.4
47	Panama	1,181	0	1,142	9	30	0	2.5
48	Colombia	1,065	159	1,009	39	17	1	1.6
49	Serbia	1,060	160	1,032	0	28	12	2.6
50	Argentina	1,054	0	778	248	28	1	2.7
51	Singapore	1,000	74	752	245	3	0	0.3
52	Croatia	963	96	884	73	6	0	0.6

Table 6: Cases by Country (01 Apr 2020) *(continued)*

	country	confirmed	new.confirmed	current.confirmed	recovered	deaths	new.deaths	death.rate
53	Algeria	847	131	728	61	58	14	6.8
54	Slovenia	841	39	816	10	15	0	1.8
55	Qatar	835	54	762	71	2	0	0.2
56	United Arab Emirates	814	150	745	61	8	2	1
57	Ukraine	794	149	761	13	20	3	2.5
58	Egypt	779	69	548	179	52	6	6.7
59	Estonia	779	34	741	33	5	1	0.6
60	Iraq	728	34	494	182	52	2	7.1
61	Diamond Princess	712	0	98	603	11	1	1.5
62	New Zealand	708	61	624	83	1	0	0.1
63	Morocco	654	37	586	29	39	3	6
64	Lithuania	581	44	566	7	8	0	1.4
65	Armenia	571	39	536	31	4	1	0.7
66	Bahrain	569	2	228	337	4	0	0.7
67	Hungary	525	33	465	40	20	4	3.8
68	Lebanon	479	9	422	43	14	2	2.9
69	Bosnia and Herzegovina	459	39	427	19	13	0	2.8
70	Latvia	446	48	445	1	0	0	0
71	Moldova	423	70	395	23	5	1	1.2
72	Tunisia	423	29	406	5	12	2	2.8
73	Bulgaria	422	23	392	20	10	2	2.4
74	Slovakia	400	37	396	3	1	1	0.2
75	Andorra	390	14	366	10	14	2	3.6
76	Kazakhstan	380	37	351	26	3	1	0.8
77	Costa Rica	375	28	369	4	2	0	0.5
78	Azerbaijan	359	61	328	26	5	0	1.4
79	North Macedonia	354	25	326	17	11	2	3.1
80	Uruguay	338	0	295	41	2	1	0.6
81	Taiwan*	329	7	285	39	5	0	1.5
82	Cyprus	320	58	283	28	9	1	2.8
83	Kuwait	317	28	237	80	0	0	0
84	Burkina Faso	282	21	220	46	16	2	5.7
85	Jordan	278	4	237	36	5	0	1.8
86	Albania	259	16	177	67	15	0	5.8
87	Afghanistan	237	63	228	5	4	0	1.7
88	San Marino	236	0	197	13	26	0	11
89	Cameroon	233	40	217	10	6	0	2.6
90	Vietnam	218	6	155	63	0	0	0
91	Cuba	212	26	194	12	6	0	2.8
92	Oman	210	18	175	34	1	0	0.5
93	Ghana	195	34	159	31	5	0	2.6
94	Cote d'Ivoire	190	11	180	9	1	0	0.5
95	Senegal	190	15	144	45	1	1	0.5
96	Malta	188	19	186	2	0	0	0
97	Uzbekistan	181	9	167	12	2	0	1.1
98	Nigeria	174	39	163	9	2	0	1.1
99	Honduras	172	31	159	3	10	3	5.8
100	Belarus	163	11	108	53	2	1	1.2
101	Mauritius	161	18	155	0	6	1	3.7
102	Sri Lanka	146	3	122	21	3	1	2.1
103	Venezuela	143	8	99	41	3	0	2.1
104	West Bank and Gaza	134	15	115	18	1	0	0.7
105	Brunei	131	2	78	52	1	0	0.8
106	Kosovo	125	13	114	10	1	0	0.8
107	Montenegro	123	14	121	0	2	0	1.6
108	Georgia	117	7	94	23	0	0	0
109	Bolivia	115	8	107	1	7	1	6.1
110	Kyrgyzstan	111	4	108	3	0	0	0
111	Cambodia	109	0	84	25	0	0	0
112	Congo (Kinshasa)	109	11	97	3	9	1	8.3
113	Trinidad and Tobago	90	3	84	1	5	2	5.6
114	Rwanda	82	7	82	0	0	0	0
115	Kenya	81	22	77	3	1	0	1.2
116	Niger	74	47	69	0	5	2	6.8
117	Paraguay	69	4	65	1	3	0	4.3
118	Liechtenstein	68	0	68	0	0	0	0
119	Madagascar	57	0	57	0	0	0	0
120	Monaco	55	3	52	2	1	0	1.8

Table 6: Cases by Country (01 Apr 2020) *(continued)*

	country	confirmed	new.confirmed	current.confirmed	recovered	deaths	new.deaths	death.rate
121	Bangladesh	54	3	23	25	6	1	11.1
122	Jamaica	44	8	39	2	3	2	6.8
123	Uganda	44	0	44	0	0	0	0
124	Guatemala	39	1	26	12	1	0	2.6
125	Togo	36	2	24	10	2	1	5.6
126	Zambia	36	1	36	0	0	0	0
127	Barbados	34	0	34	0	0	0	0
128	Djibouti	33	3	33	0	0	0	0
129	El Salvador	32	0	31	0	1	0	3.1
130	Mali	31	3	28	0	3	1	9.7
131	Guinea	30	8	30	0	0	0	0
132	Ethiopia	29	3	27	2	0	0	0
133	Bahamas	21	7	19	1	1	1	4.8
134	Tanzania	20	1	18	1	1	0	5
135	Congo (Brazzaville)	19	0	19	0	0	0	0
136	Guyana	19	7	17	0	2	0	10.5
137	Maldives	19	1	6	13	0	0	0
138	Gabon	18	2	17	0	1	0	5.6
139	Haiti	16	1	15	1	0	0	0
140	Burma	15	0	14	0	1	0	6.7
141	Equatorial Guinea	15	3	14	1	0	0	0
142	Eritrea	15	0	15	0	0	0	0
143	Mongolia	14	2	12	2	0	0	0
144	Namibia	14	3	12	2	0	0	0
145	Benin	13	4	12	1	0	0	0
146	Saint Lucia	13	0	12	1	0	0	0
147	Dominica	12	0	12	0	0	0	0
148	Laos	10	1	10	0	0	0	0
149	Libya	10	0	10	0	0	0	0
150	Mozambique	10	2	10	0	0	0	0
151	Seychelles	10	0	10	0	0	0	0
152	Suriname	10	1	10	0	0	0	0
153	Syria	10	0	8	0	2	0	20
154	Eswatini	9	0	9	0	0	0	0
155	Grenada	9	0	9	0	0	0	0
156	Guinea-Bissau	9	1	9	0	0	0	0
157	MS Zaandam	9	7	7	0	2	2	22.2
158	Angola	8	1	5	1	2	0	25
159	Saint Kitts and Nevis	8	0	8	0	0	0	0
160	Zimbabwe	8	0	7	0	1	0	12.5
161	Antigua and Barbuda	7	0	7	0	0	0	0
162	Chad	7	0	7	0	0	0	0
163	Sudan	7	0	3	2	2	0	28.6
164	Cabo Verde	6	0	5	0	1	0	16.7
165	Holy See	6	0	6	0	0	0	0
166	Liberia	6	3	6	0	0	0	0
167	Mauritania	6	0	3	2	1	0	16.7
168	Fiji	5	0	5	0	0	0	0
169	Nepal	5	0	4	1	0	0	0
170	Nicaragua	5	0	4	0	1	0	20
171	Somalia	5	0	4	1	0	0	0
172	Bhutan	4	0	4	0	0	0	0
173	Botswana	4	0	3	0	1	0	25
174	Gambia	4	0	1	2	1	0	25
175	Belize	3	0	3	0	0	0	0
176	Central African Republic	3	0	3	0	0	0	0
177	Burundi	2	0	2	0	0	0	0
178	Sierra Leone	2	1	2	0	0	0	0
179	Papua New Guinea	1	0	1	0	0	0	0
180	Saint Vincent and the Grenadines	1	0	0	1	0	0	0
181	Timor-Leste	1	0	1	0	0	0	0

Appendix B. How to Cite This Work

Citation

Yanchang Zhao, COVID-19 Data Analysis with R – Worldwide. RDataMining.com, 2020. URL: http://www.rdatamining.com/docs/Coronavirus-data-analysis-world.pdf.

BibTex

```
@techreport{Zhao2020Covid19world,
Author = {Yanchang Zhao},
Institution = {RDataMining.com},
Title = {COVID-19 Data Analysis with R – Worldwide},
Url = {http://www.rdatamining.com/docs/Coronavirus-data-analysis-world.pdf},
Year = {2020}}
```

Appendix C. Contact

Contact:

Dr. Yanchang Zhao

Email: yanchang@RDataMining.com

Twitter: @RDataMining

LinkedIn: http://group.rdatamining.com

Comments and suggestions and welcome. Thanks!