Data Science FAIR - COVID-19

Carlos Utrilla Guerrero

3/19/2020

COVID-19 Pandemia. On going FAIR data science pipeline.

This is an R Markdown document. It is intented to publicy illustrate data from Johns Hopkins University (https://github.com/CSSEGISandData/COVID-19) and FAIR (https://www.go-fair.org/fair-principles/) data science.

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

```
## Warning: package 'lubridate' was built under R version 3.6.3
```

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

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

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

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages ----------------------------------------------------------------- tidyv
erse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'tibble' was built under R version 3.6.2
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------------------------------ tidyverse_c
onflicts() --
## 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
```

```
## Warning: package 'gridExtra' was built under R version 3.6.3
```

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

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

```
library(kableExtra) # build HTML and LaTeX tables
```

```
## Warning: package 'kableExtra' was built under R version 3.6.3
```

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

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

```
library(dplyr)
setwd("~/Github/COVID-19")
# Loading data
# At first, three CSV files, are downloaded and saved as local files
# and then 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 <- 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 <- 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')
# check dimension of data confirmed
dim(data.confirmed)
```

```
## [1] 460  60
```

```
# Table:
data.confirmed[1:10, 1:10] %>%
kable(booktabs = T, caption = 'Raw Data (Confirmed, First 10 Cols)') %>%
kable_styling(font_size = 6, latex_options = c('striped','hold_position','repeat_header'))
```

Raw Data (Confirmed, First 10 Cols)

Province.State	Country.Region	Lat	Long	X1.22.20	X1.23.20	X1.24.20	X1.25.20	X1.26.20	X1.27.20
	Thailand	15.0000	101.0000	2	3	5	7	8	8
	Japan	36.0000	138.0000	2	1	2	2	4	4
	Singapore	1.2833	103.8333	0	1	3	3	4	5
	Nepal	28.1667	84.2500	0	0	0	1	1	1
	Malaysia	2.5000	112.5000	0	0	0	3	4	4
British Columbia	Canada	49.2827	-123.1207	0	0	0	0	0	0
New South Wales	Australia	-33.8688	151.2093	0	0	0	0	3	4
Victoria	Australia	-37.8136	144.9631	0	0	0	0	1	1
Queensland	Australia	-28.0167	153.4000	0	0	0	0	0	0
	Cambodia	11.5500	104.9167	0	0	0	0	0	1

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

```
## [1] "2020-01-22" "2020-03-17"
```

```
min.date <- min(dates)
max.date <- max(dates)
# last update on 16 March 2020 max.date
# Data Preparation steps:
# 1.From wide to long format
# 2.Aggregate by country
# 3. merge into a signe 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 %<>% 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)
# 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)
```

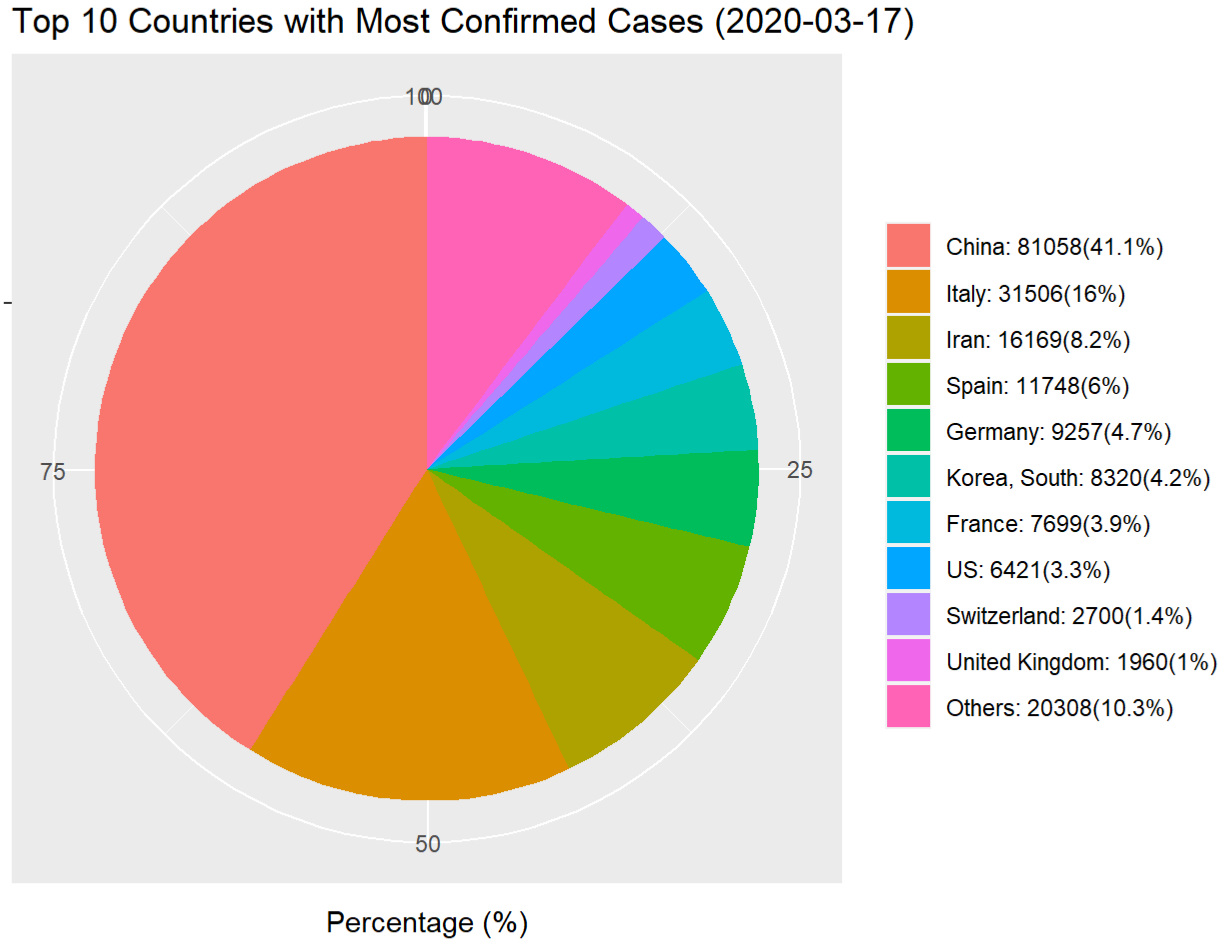
```
##    country       date confirmed deaths recovered
## 47   Spain 2020-03-08       673     17        30
## 48   Spain 2020-03-09      1073     28        32
## 49   Spain 2020-03-10      1695     35        32
## 50   Spain 2020-03-11      2277     54       183
## 51   Spain 2020-03-12      2277     55       183
## 52   Spain 2020-03-13      5232    133       193
## 53   Spain 2020-03-14      6391    195       517
## 54   Spain 2020-03-15      7798    289       517
## 55   Spain 2020-03-16      9942    342       530
## 56   Spain 2020-03-17     11748    533      1028
## Cases for the Whole World
# counts for worldwide
data.world <- data %>% group_by(date) %>%
  summarise(country='World',
            confirmed=sum(confirmed),
            deaths=sum(deaths),
            recovered=sum(recovered))
data %<>% rbind(data.world)
# remaining confirmed cases
data %<>% mutate(remaining.confirmed = confirmed - deaths - recovered)
# Daily Increases and Death Rates
# rate.upper = total deaths and recovered cases
# rate.lower = total deaths and confirmed cases
# expected death rate is to be between above rates
# rate.daily =daily deaths and recovered cases
## sort by country and date
data %<>% arrange(country,date)
# daily increases of deaths and recovered cases
# set NA to increase on day1
n <- nrow(data)
day1 <- min(data$date) # set NA day1
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 base 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 number f death/recovered on every single day
data %<>% mutate(rate.daily = (100 * deaths.inc / (deaths.inc + recovered.inc)) %>% round(1))
```

```
# Visualisation
# After preparing the data, we portrait it in various graphs
# TOP Ten Countries
# ranking by confirmed cases
data.annex <- data
data.latest <- data %>% filter(date ==max(date)) %>%
                                select(country, date, confirmed, deaths, recovered, remaining.co
nfirmed) %>%
                                mutate(ranking = dense_rank(desc(confirmed)))
# top 10 countries incl 11 World
top.countries <- data.latest %>% filter(ranking <= 11) %>%
  arrange(ranking) %>% pull(country) %>% as.character()
top.countries %>% setdiff('World') %>% print()
```

```
##  [1] "China"          "Italy"          "Iran"           "Spain"
##  [5] "Germany"        "Korea, South"   "France"         "US"
##  [9] "Switzerland"    "United Kingdom"
## add 'Others'
top.countries %<>% c('Others')
## put all others in a single group of 'Others'
df <- data.latest %>% filter(!is.na(country) & country!= 'World')%>%
  mutate(country=ifelse(ranking <= 11, as.character(country), 'Others')) %>%
  mutate(country=country %>% factor(levels = c(top.countries)))
df %<>% group_by(country) %>% summarise(confirmed=sum(confirmed))
# percentage and label
df %<>% mutate(per = (100*confirmed/sum(confirmed)) %>% round(1)) %>%
                 mutate(txt = paste0(country, ': ', confirmed, '(', per, '%)'))
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) +
  theme(legend.title = element_blank(), legend.text = element_text((size=7)))
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family not
## found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```



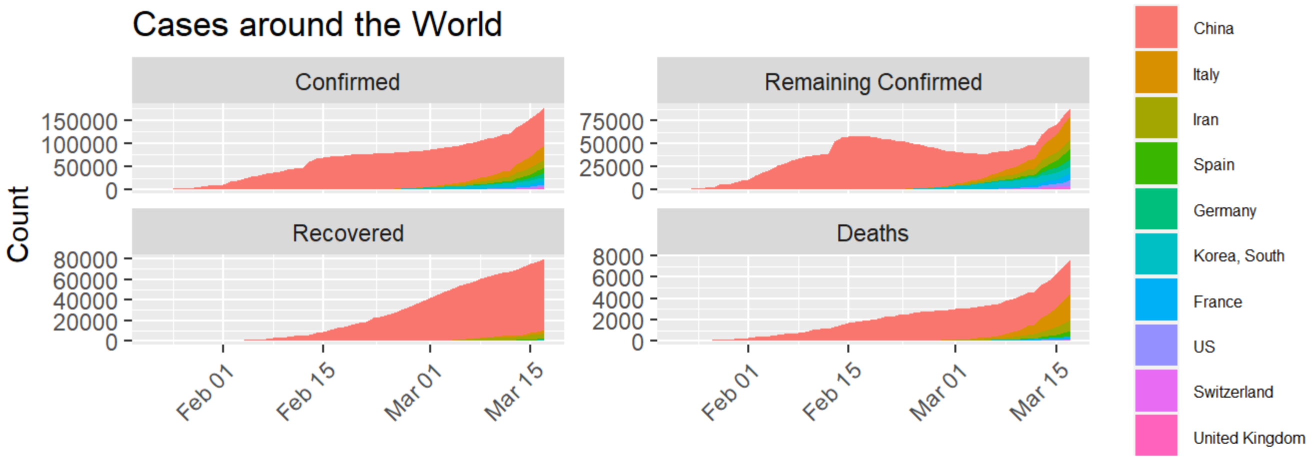
```
data.latest %>% filter(country %in% top.countries) %>% select(-c(date, ranking)) %>%
                                                                arrange(desc(confirmed)) %>%
                                                                kable(booktabs=T, row.names=T,
                                                                      caption = paste0('Cases in
Top Ten Countries (', max.date, '). See complete list of all infected countries at the annex A'
),
                                                                        format.args = list(big.m
ark = ',')) %>%
                                                                kable_styling(font_size = 7, lat
ex_options = c('striped', 'hold_position', 'repeat_header'))
```

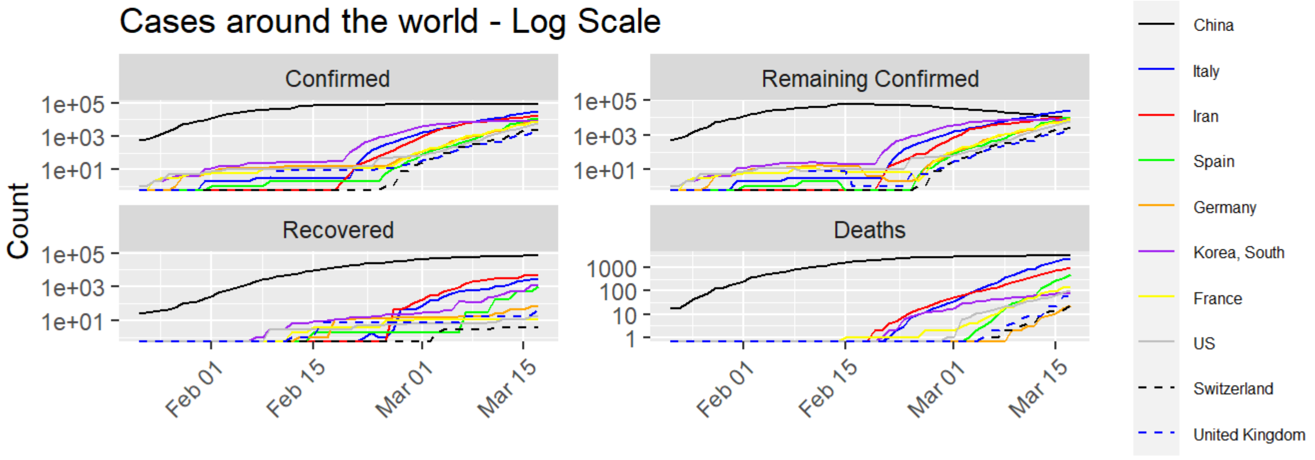
Cases in Top Ten Countries (2020-03-17). See complete list of all infected countries at the annex A

	country	confirmed	deaths	recovered	remaining.confirmed
1	World	197,146	7,905	80,840	108,401
2	China	81,058	3,230	68,798	9,030
3	Italy	31,506	2,503	2,941	26,062
4	Iran	16,169	988	5,389	9,792
5	Spain	11,748	533	1,028	10,187
6	Germany	9,257	24	67	9,166
7	Korea, South	8,320	81	1,407	6,832
8	France	7,699	148	12	7,539
9	US	6,421	108	17	6,296
10	Switzerland	2,700	27	4	2,669
11	United Kingdom	1,960	56	53	1,851

```
 # Comparison across Countries
# convert from wide to long format, for drawing area plot
data.long <- data %>%
  select(c(country, date, confirmed, remaining.confirmed, recovered, deaths)) %>%
  gather(key = type, value = count, -c(country,date))
# set for factor levels to show them in a desirable order
data.long %<>% mutate(type =recode_factor(type, confirmed= 'Confirmed',
                                          remaining.confirmed = 'Remaining Confirmed',
                                          recovered= 'Recovered',
                                          deaths='Deaths'))
# plot cases by type
df <- data.long %>% filter(country %in% top.countries) %<>%
  mutate(country=country %>% factor(levels=c(top.countries)))
### 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')
# 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 = 'Cases around the world - Log Scale') +
  scale_y_continuous(trans = 'log10')
# shows two plots together
grid.arrange(plot1, plot2, ncol=1)
```

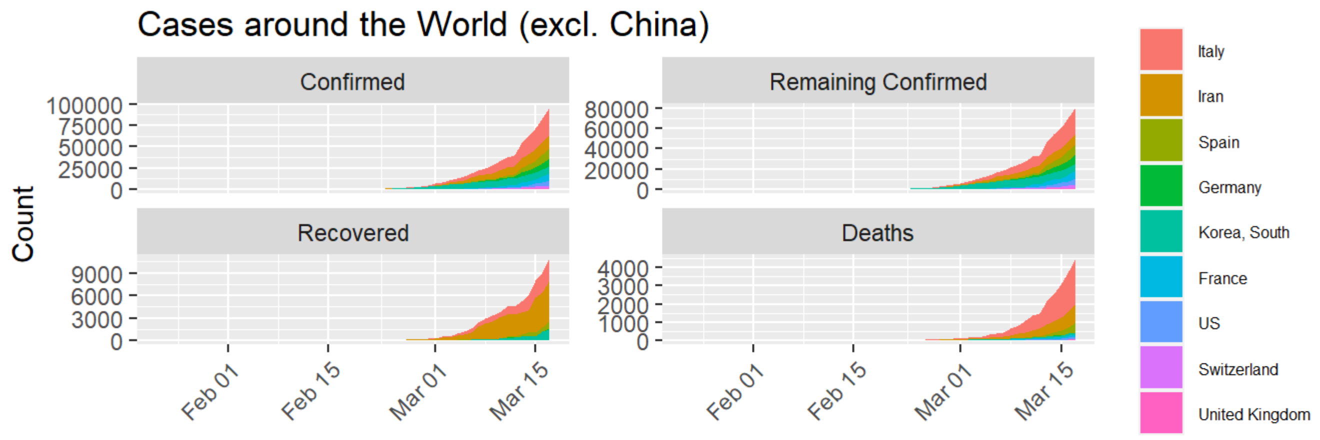
```
## Warning: Transformation introduced infinite values in continuous y-axis
```

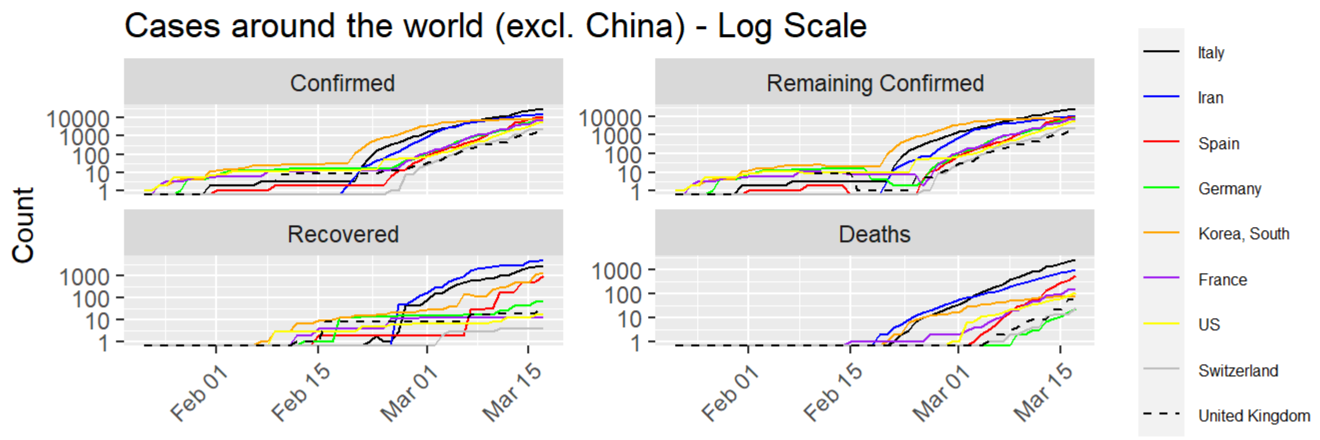




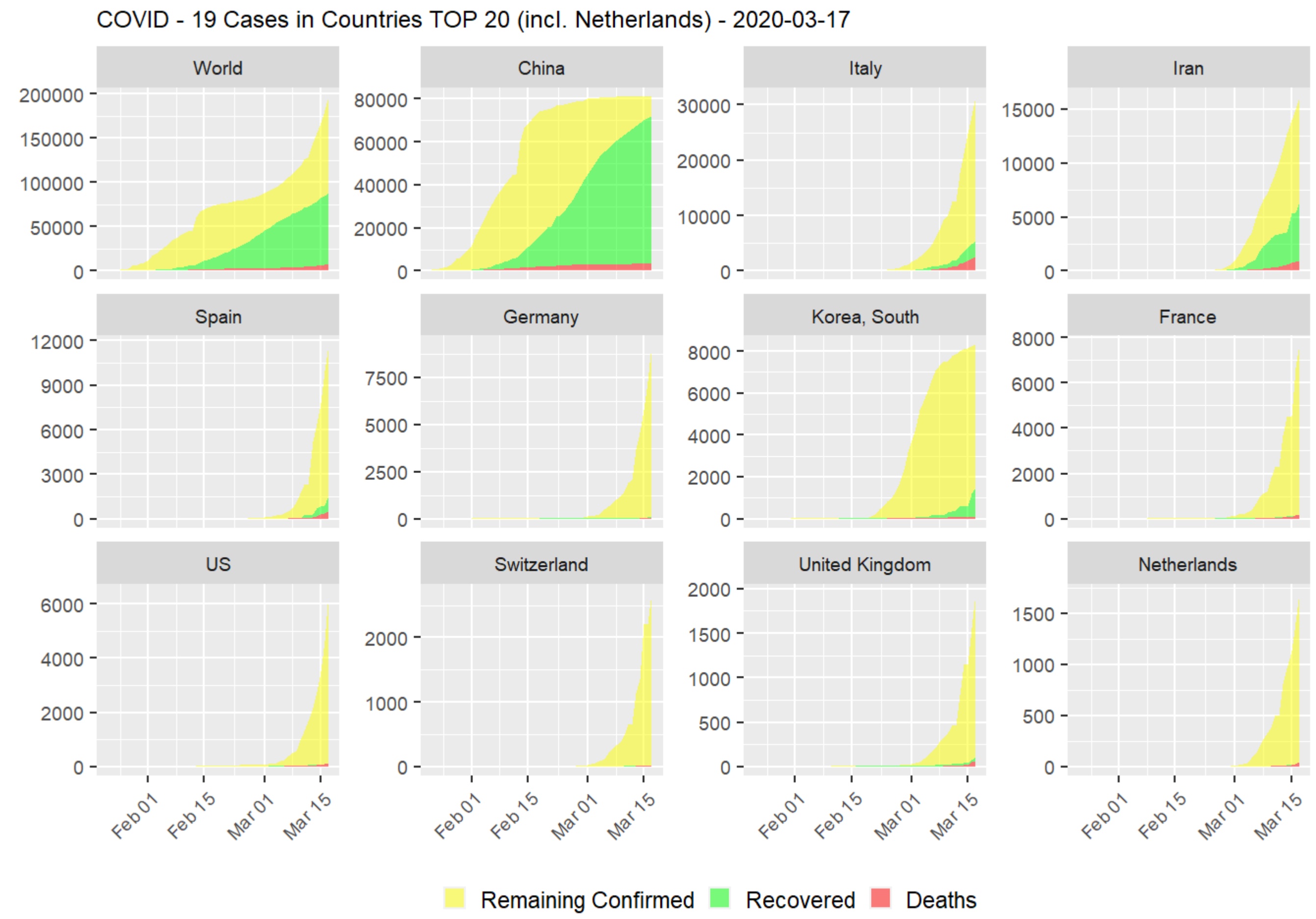
```
# Plot: excluding China
p <- df%>% filter(!(country %in% c('World', 'China'))) %>%
  ggplot(aes(x=date, y=count)) + xlab('') + ylab('Count') +
  theme(legend.title=element_blank(),
        legend.text = element_text(size=6),
        legend.key.size=unit(0.6, 'cm'),
        axis.text.x=element_text(angle = 45, hjust=1)) +
  facet_wrap(~type, ncol = 2, scale='free_y')
# area plot
plot1 <- p + geom_area(aes(fill=country)) +
  labs(title='Cases around the World (excl. China)')
# 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 = 'Cases around the world (excl. China) - Log Scale') +
  scale_y_continuous(trans = 'log10')
# shows two plots together
grid.arrange(plot1, plot2, ncol=1)
```

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



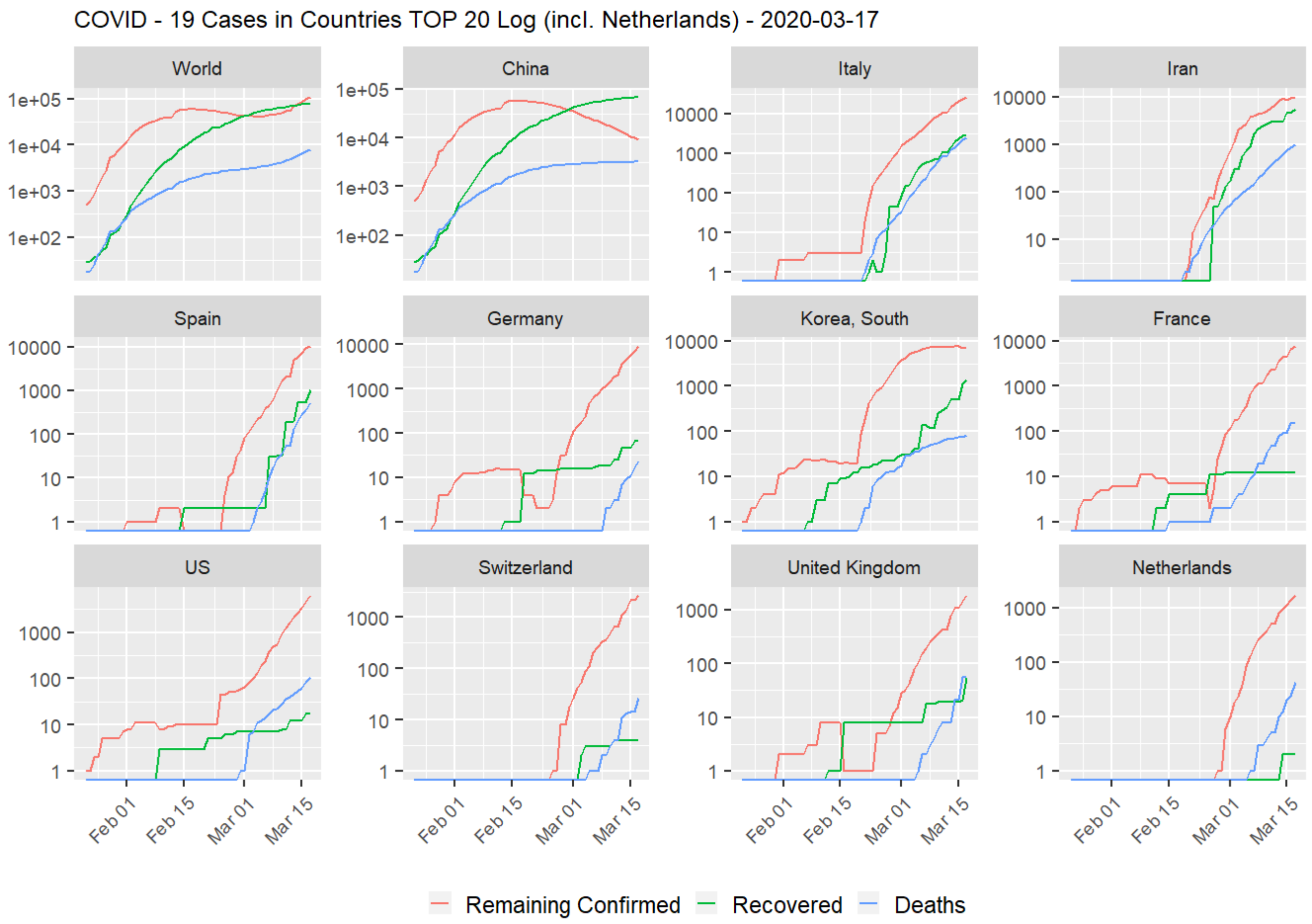


```
# # # list(countries) == 'Netherlands'
## If The Netherland is not top 20, add it in and remove 'Others'
if(!('Netherlands' %in% top.countries)) {
  top.countries %<>% setdiff('Others') %>% c('Netherlands')
  df <- data.long %>% filter(country %in% top.countries) %>%
    mutate(country=country %>% factor(levels = c(top.countries)))
}
# cases by country - area plot
df %>% filter(type != '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') #+ scale_y_continuous(trans = 'log10')
```



```
# cases by country - log case
df %>% filter(type != 'Confirmed') %>%
  ggplot(aes(x=date, y=count, color=type)) +
  geom_line() +
  labs(title = paste0('COVID - 19 Cases in Countries TOP 20 Log (incl. Netherlands) - ', max.dat
e)) +
  scale_fill_manual(values=c('red','green','blue')) +
  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') + scale_y_continuous(trans = 'log10')
```

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

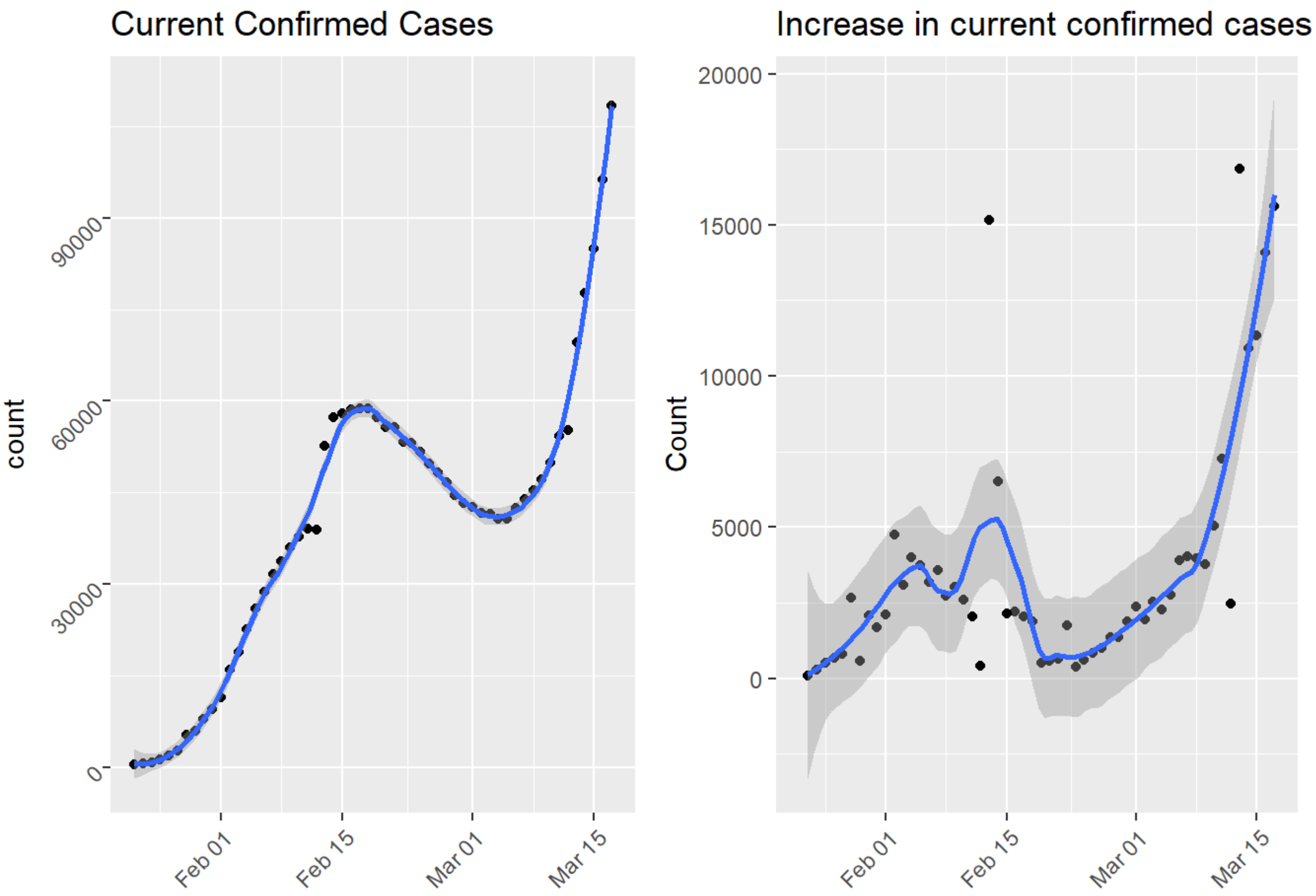


```
### Current confirmed cases:
#data.test <- data %>% filter(country %in% c('Italy', 'Spain', 'Netherlands'))
data %<>% filter(country=='World')
data.annexf <- data.annex
n <- nrow(data)
# current confirmed and it is increase with worldwide case
plot1 <- ggplot(data, aes(x = date, y=remaining.confirmed)) +
  geom_point() + geom_smooth(span=0.3) +
  xlab('') + ylab('count') + labs(title= 'Current Confirmed Cases') +
  theme(axis.text = element_text(angle = 45, hjust=1))
plot2 <- ggplot(data, aes(x =date, y=confirmed.inc)) +
  geom_point() + geom_smooth(span=0.3) +
  xlab('') + ylab('Count') + labs(title= 'Increase in current confirmed cases') +
  theme (axis.text.x = element_text(angle=45, hjust =1))
# show plot 1 and 2 side by side
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).
```



```
# Deaths and recovery cases
plot1 <- ggplot(data,aes(x=date, y=deaths)) +
  geom_point() + geom_smooth() +
  xlab('') + ylab('Count') + labs(title = 'Deaths') +
  theme(axis.text.x = element_text(angle = 45, hjust=1))
plot2 <- ggplot(data,aes(x=date, y=recovered)) +
  geom_point() + geom_smooth() +
  xlab('') + ylab('Count') + labs(title = 'Recovered Cases') +
  theme(axis.text.x = element_text(angle = 45, hjust=1))
plot3 <- ggplot(data,aes(x=date, y=deaths.inc)) +
  geom_point() + geom_smooth() +
  xlab('') + ylab('Count') + labs(title = 'Increase in Deaths') +
  theme(axis.text.x = element_text(angle = 45, hjust=1))
plot4 <- ggplot(data,aes(x=date, y=recovered.inc)) +
  geom_point() + geom_smooth() +
  xlab('') + ylab('Count') + labs(title = 'Increase Recovered cases') +
  theme(axis.text.x = element_text(angle = 45, hjust=1))
# shows plots together
grid.arrange(plot1, plot2, plot3, plot4, nrow=2)
```

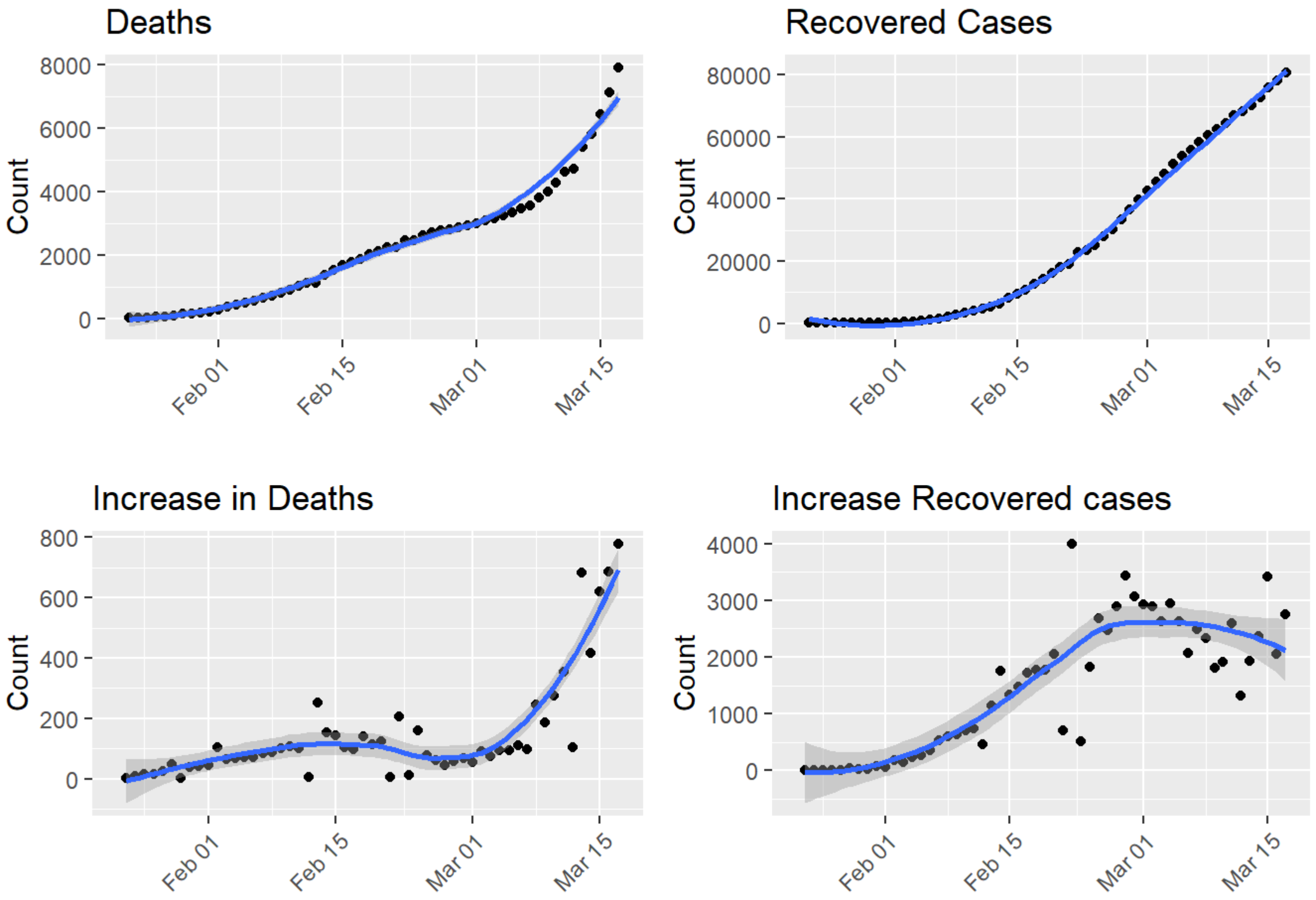
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## Warning: Removed 1 rows containing non-finite values (stat_smooth).

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

```
## `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).
```



Deaths

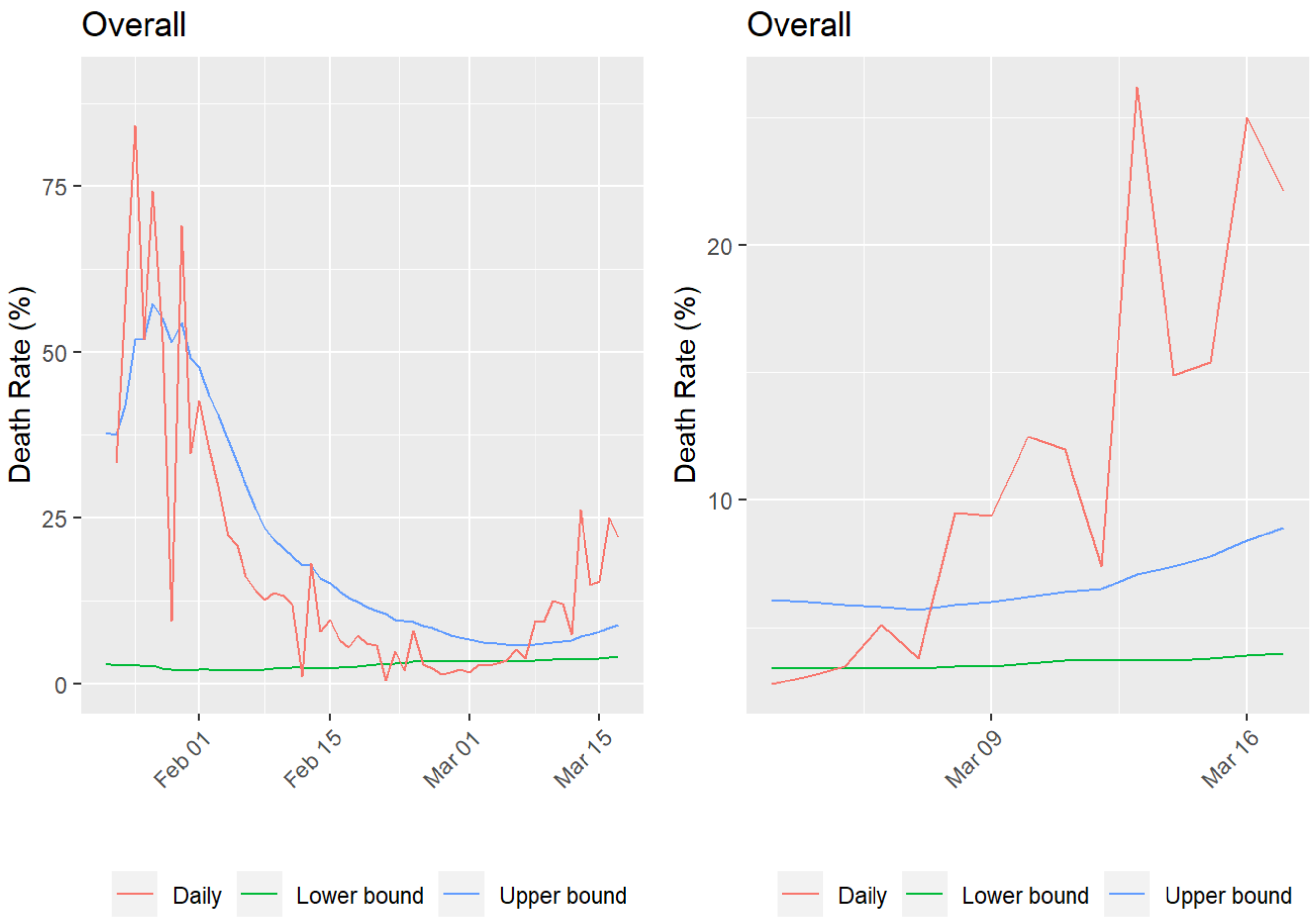
Recovered Cases

Increase in Deaths

Increase Recovered cases

```
## Deaths 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('') + ylab('Death Rate (%)') + labs(title='Overall') +
  theme(legend.position='bottom', legend.title=element_blank(),
        axis.text.x=element_text(angle=45, hjust=1)) +
  ylim(0,90)
## insert las two 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('') + ylab('Death Rate (%)') + labs(title = 'Overall') +
  theme(legend.position='bottom', legend.title =element_blank(),
        axis.text.x=element_text(angle=45, hjust=1))
grid.arrange(plot1, plot2, nrow=1)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



```
# sort by date, descending order
datatable <- data %>% arrange(desc(date)) %>%
  select(c(date, confirmed, deaths, recovered, remaining.confirmed,
           confirmed.inc, deaths.inc, recovered.inc, rate.lower, rate.upper, rate.daily))
kable(datatable,booktabs =T, longtable=T, caption = 'Appendix A. COVID-19 Cases Worldwide', form
at.args=list(big.mark=',')) %>% kable_styling(font_size=5, latex_options=c('striped', 'hold_posi
tion', 'repeat_header'))
```

Appendix A. COVID-19 Cases Worldwide

date	confirmed	deaths	recovered	remaining.confirmed	confirmed.inc	deaths.inc	recovered.inc	rate.lower	rate.upper	rate.daily
2020-03-17	197,146	7,905	80,840	108,401	15,615	779	2,752	4.0	8.9	22.1
2020-03-16	181,531	7,126	78,088	96,317	14,082	686	2,054	3.9	8.4	25.0
2020-03-15	167,449	6,440	76,034	84,975	11,352	621	3,410	3.8	7.8	15.4
2020-03-14	156,097	5,819	72,624	77,654	10,904	415	2,373	3.7	7.4	14.9
2020-03-13	145,193	5,404	70,251	69,538	16,850	684	1,927	3.7	7.1	26.2
2020-03-12	128,343	4,720	68,324	55,299	2,478	105	1,321	3.7	6.5	7.4
2020-03-11	125,865	4,615	67,003	54,247	7,273	353	2,599	3.7	6.4	12.0
2020-03-10	118,592	4,262	64,404	49,926	5,031	274	1,910	3.6	6.2	12.5
2020-03-09	113,561	3,988	62,494	47,079	3,766	186	1,800	3.5	6.0	9.4
2020-03-08	109,795	3,802	60,694	45,299	3,974	244	2,336	3.5	5.9	9.5
2020-03-07	105,821	3,558	58,358	43,905	4,037	98	2,493	3.4	5.7	3.8
2020-03-06	101,784	3,460	55,865	42,459	3,902	112	2,069	3.4	5.8	5.1
2020-03-05	97,882	3,348	53,796	40,738	2,762	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	1,891	69	3,071	3.4	6.9	2.2
2020-02-28	84,120	2,872	36,711	44,537	1,366	58	3,434	3.4	7.3	1.7
2020-02-27	82,754	2,814	33,277	46,663	1,359	44	2,893	3.4	7.8	1.5
2020-02-26	81,395	2,770	30,384	48,241	982	62	2,479	3.4	8.4	2.4
2020-02-25	80,413	2,708	27,905	49,800	845	79	2,678	3.4	8.8	2.9
2020-02-24	79,568	2,629	25,227	51,712	603	160	1,833	3.3	9.4	8.0
2020-02-23	78,965	2,469	23,394	53,102	386	11	508	3.1	9.5	2.1
2020-02-22	78,579	2,458	22,886	53,235	1,756	207	3,996	3.1	9.7	4.9
2020-02-21	76,823	2,251	18,890	55,682	626	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

date	confirmed	deaths	recovered	remaining.confirmed	confirmed.inc	deaths.inc	recovered.inc	rate.lower	rate.upper	rate.daily
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	3,574	85	524	2.1	26.3	14.0
2020-02-06	30,817	634	1,487	28,696	3,182	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	288	8	6	2.8	41.9	57.1
2020-01-23	653	18	30	605	98	1	2	2.8	37.5	33.3
2020-01-22	555	17	28	510	NA	NA	NA	3.1	37.8	NA

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