COVID-19

Ankush Jigyas Tanmay

10/09/2020

- LOADING LIBRARIES
- Data Preparation
  - Extracting first 10 data
  - Getting Information about dates from the data
  - Formatting the dates
  - DATA CLEANING
  - Getting first 10 cases record from China
  - Worldwide cases
  - Daily Increases and death rates
- Map
  - Number of cases
  - Current Confirmed Cases
  - Deaths and Recovered cases
  - Death rates
  - Top 20 countries
  - Confirmed Vs Deaths
  - Comparison Across Countries
  - Cases by country - line plot
  - Death rates
  - Countries with highest death rates

Dataset source for this project is https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series (https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series)

LOADING LIBRARIES

```
library(magrittr) # pipe operations
library(lubridate) # date operations
```

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

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tidyverse) # ggplot2, tidyr, dplyr...
```

```
## -- Attaching packages --------------------------------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.2     v dplyr   1.0.0
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

```
library(gridExtra) # multiple grid-based plots on a page
```

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

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

```
library(ggforce) # accelerating ggplot2
library(kableExtra) # complex tables
```

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

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

```
library(leaflet) # map
```

#LOADING DATA

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

```
## [1] 266 232
```

Data Preparation

Extracting first 10 data

```
# getting First 10 data
dt<-raw.data.confirmed[1:10, 1:10]
kable(dt, booktabs=T, caption='Raw Data (Confirmed, First 10 Columns only)')%>%
  kable_styling(latex_options = c('striped', 'hold_position', 'repeat_header'),full_width = F)
```

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	33.93911	67.70995	0	0	0	0	0	0
	Albania	41.15330	20.16830	0	0	0	0	0	0
	Algeria	28.03390	1.65960	0	0	0	0	0	0
	Andorra	42.50630	1.52180	0	0	0	0	0	0
	Angola	-11.20270	17.87390	0	0	0	0	0	0
	Antigua and Barbuda	17.06080	-61.79640	0	0	0	0	0	0
	Argentina	-38.41610	-63.61670	0	0	0	0	0	0
	Armenia	40.06910	45.03820	0	0	0	0	0	0
Australian Capital Territory	Australia	-35.47350	149.01240	0	0	0	0	0	0
New South Wales	Australia	-33.86880	151.20930	0	0	0	0	3	4

Getting Information about dates from 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" "2020-09-05"
```

Formatting the dates

```
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') %>% paste('UTC')
print(max.date.txt)
```

```
## [1] "05 Sep 2020 UTC"
```

DATA CLEANING

```
## 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)
```

```
## `summarise()` regrouping output by 'country' (override with `.groups` argument)
```

```
data.deaths <- raw.data.deaths %>% cleanData() %>% rename(deaths=count)
```

```
## `summarise()` regrouping output by 'country' (override with `.groups` argument)
```

```
data.recovered <- raw.data.recovered %>% cleanData() %>% rename(recovered=count)
```

```
## `summarise()` regrouping output by 'country' (override with `.groups` argument)
```

```
## 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')
```

Getting first 10 cases record from China

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

Worldwide cases

```
## 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))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

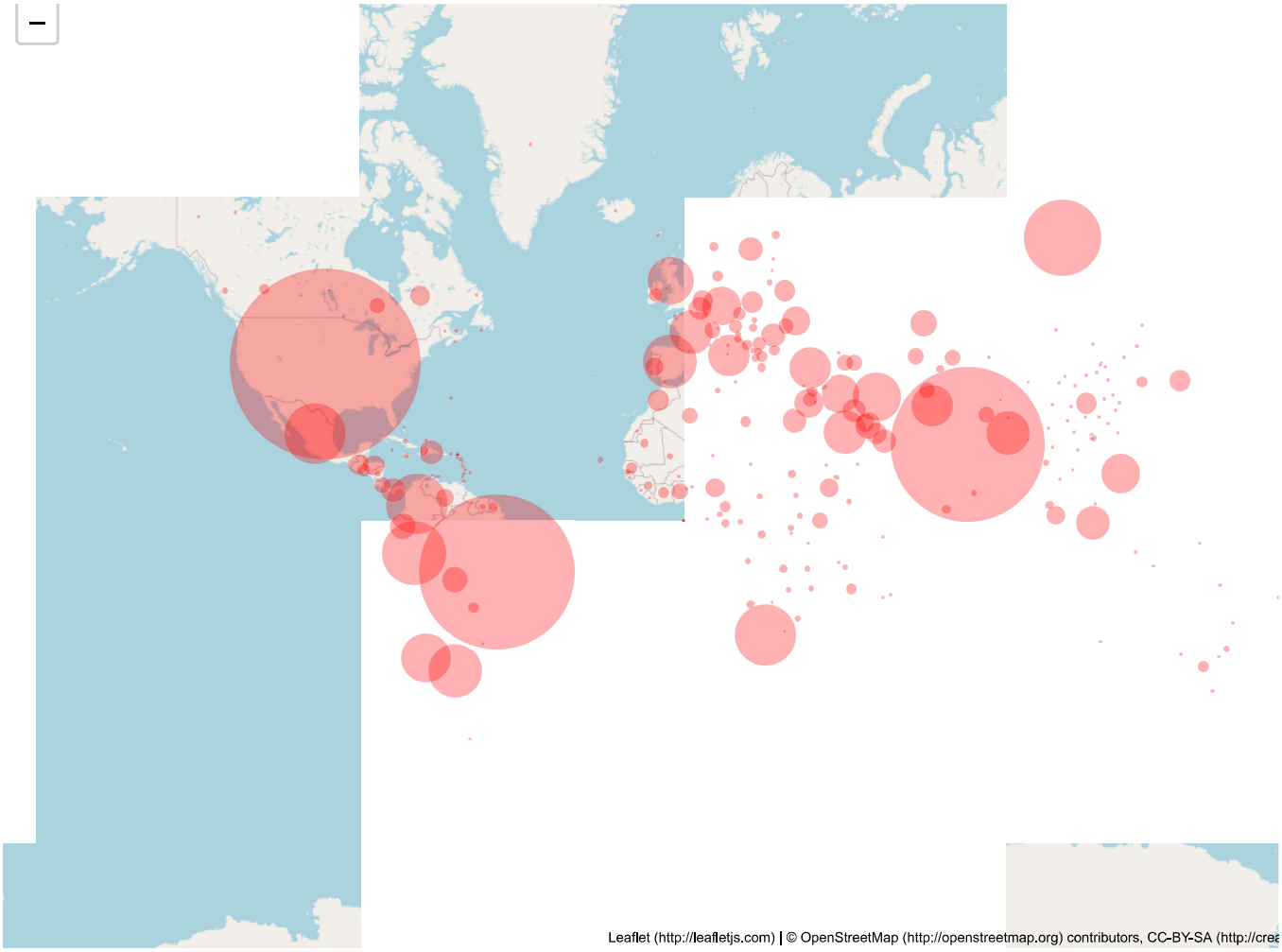
```
data %<>% rbind(data.world)
## current confirmed cases
data %<>% mutate(current.confirmed = confirmed - deaths - recovered)
```

Daily Increases and death rates

```
## 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'))
```

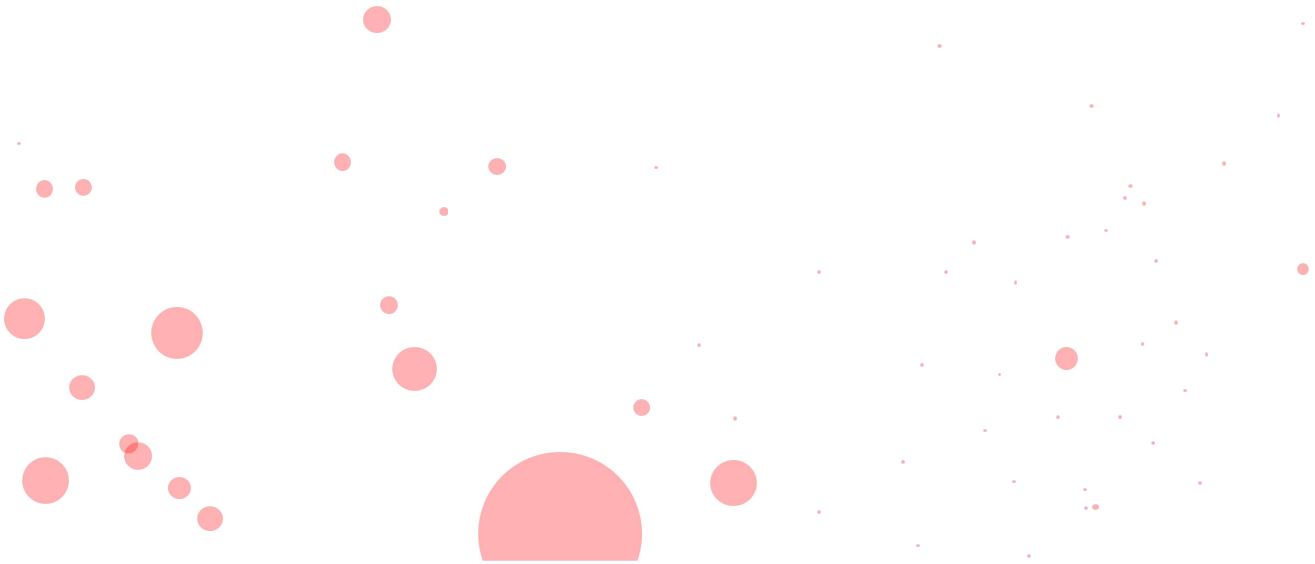
Map

```
## 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),
radius=0.03*sqrt(x$confirmed),
stroke=F,
color='red', fillOpacity=0.3,
popup=x$txt)
# world
m
```



Views of some countries

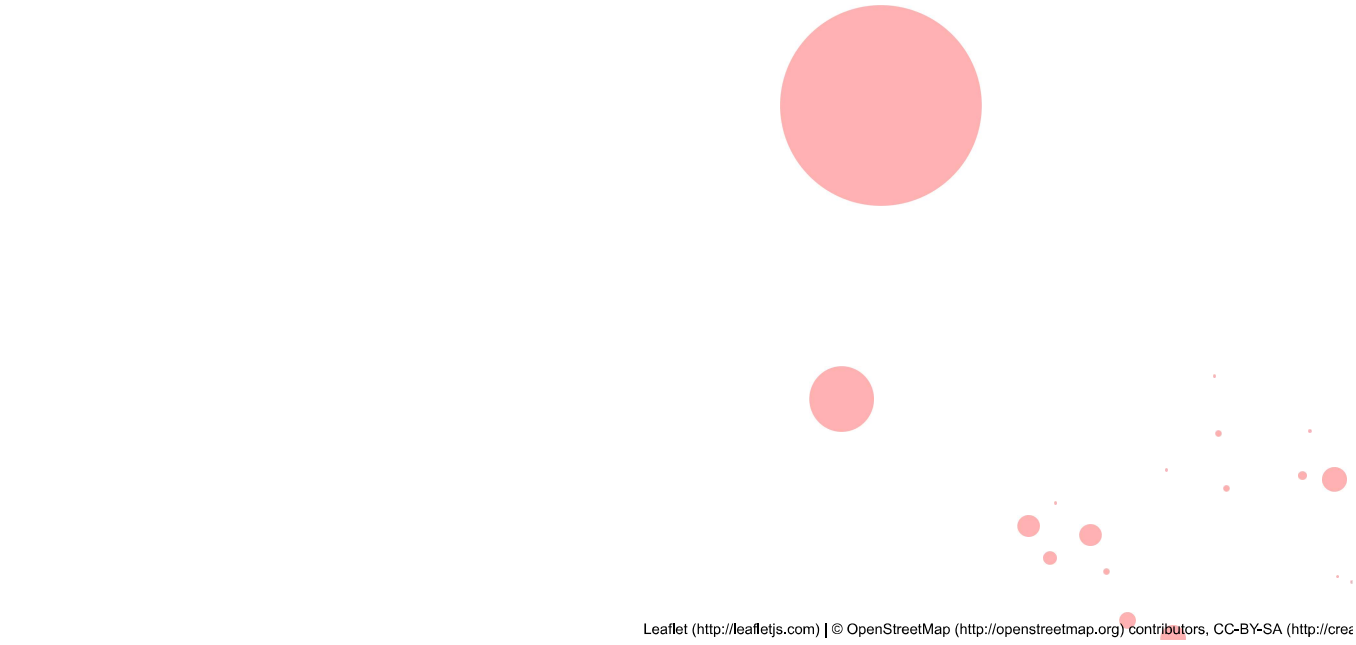
```
## China
m %>% setView(95, 35, zoom=4)
```




```
## Australia and New Zealand
m %>% setView(135, -27, zoom=4)
```



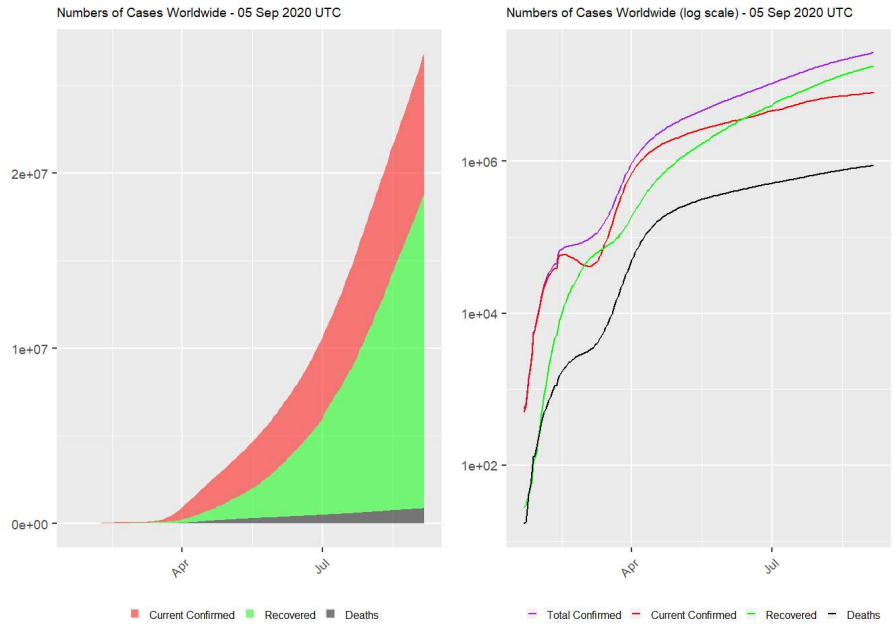
```
## US and Canada
m %>% setView(-105, 40, zoom=4)
```




```
## Europe
m %>% setView(10, 50, zoom=4)
```

Number of cases

```
# 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=7),
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=7),
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')
## show two plots side by side
grid.arrange(plot1, plot2, ncol=2)
```



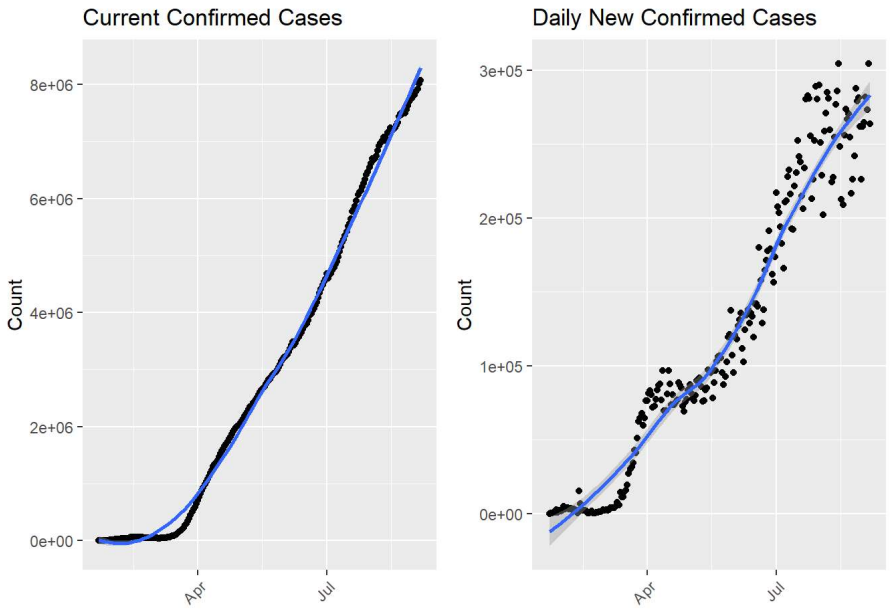
Current Confirmed Cases

```
data.world <- data %>% filter(country=='World')
n <- nrow(data.world)
## current confirmed and daily new confirmed
plot1 <- ggplot(data.world, aes(x=date, y=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)
```

```
## `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 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)
```

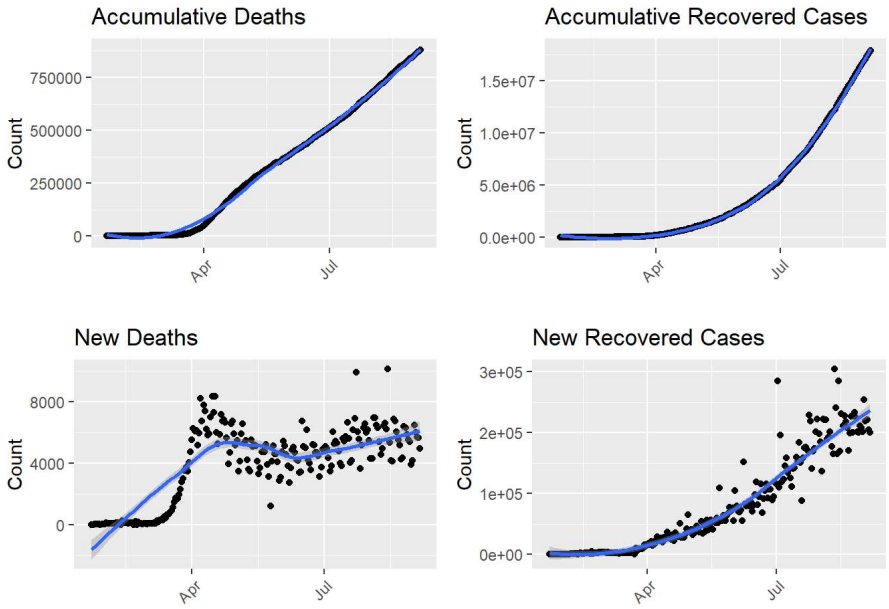
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## 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'
```

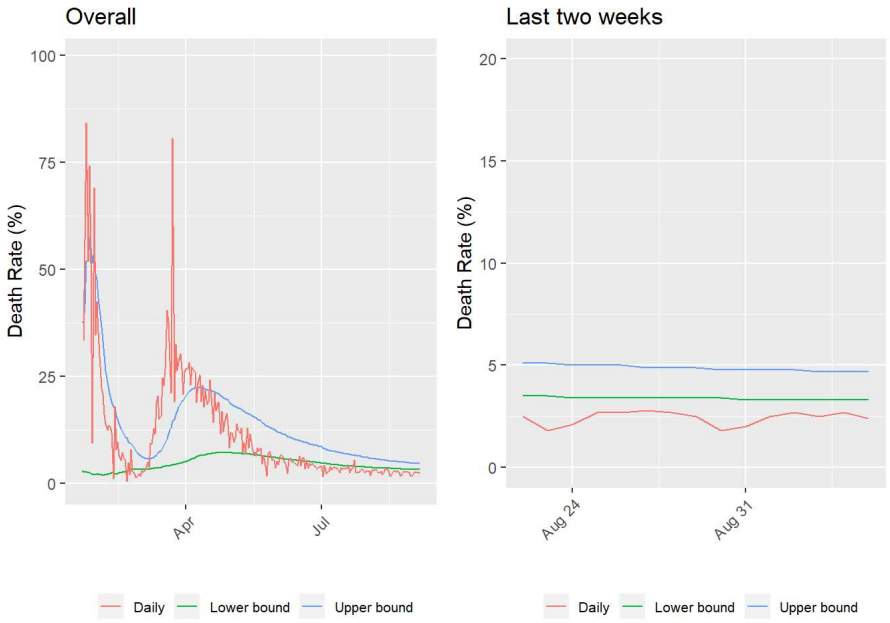
```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).

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



Death rates

```
## 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)) +
ylim(c(0, 99))
## 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, 20))
grid.arrange(plot1, plot2, ncol=2)
```

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



Top 20 countries

```
## 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)))
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()
```

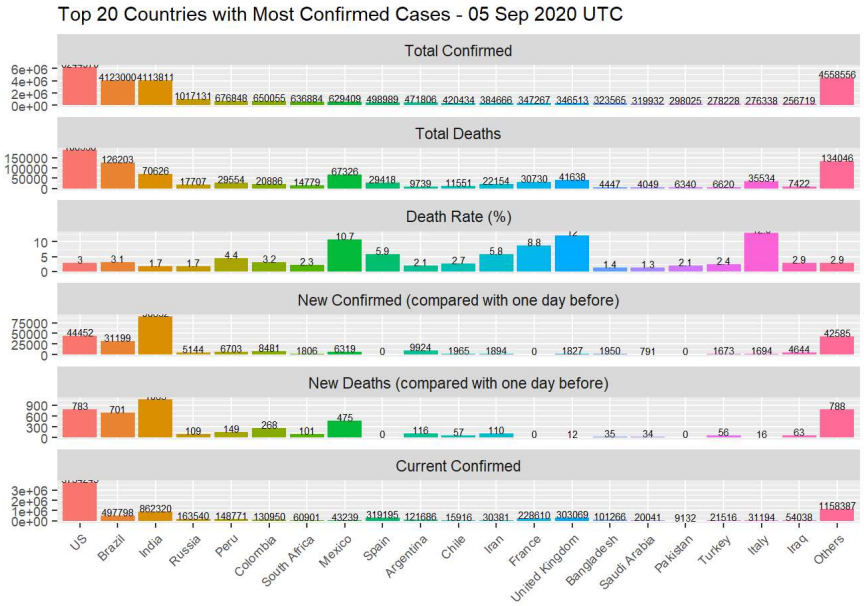
```
##  [1] "US"             "Brazil"         "India"          "Russia"
##  [5] "Peru"           "Colombia"       "South Africa"   "Mexico"
##  [9] "Spain"          "Argentina"      "Chile"          "Iran"
## [13] "France"         "United Kingdom" "Bangladesh"     "Saudi Arabia"
## [17] "Pakistan"       "Turkey"         "Italy"          "Iraq"
```

```
## [1] "US" "Brazil" "Russia" "India"
## [5] "United Kingdom" "Spain" "Italy" "Peru"
## [9] "France" "Iran" "Germany" "Turkey"
## [13] "Chile" "Mexico" "Pakistan" "Saudi Arabia"
## [17] "Canada" "Bangladesh" "China" "Qatar"
## 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, 'Others')))
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))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

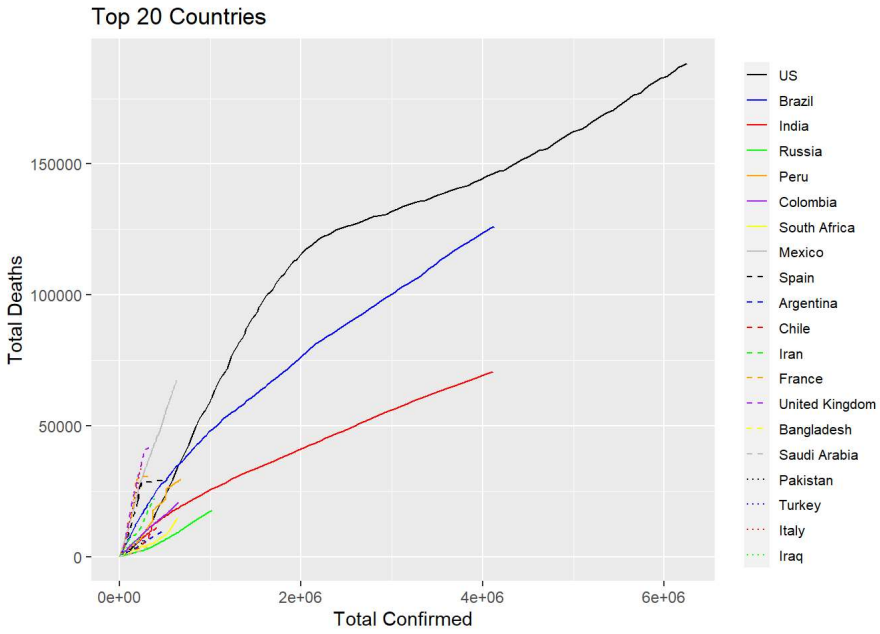
```
data.latest %<>% select(c(country, confirmed, deaths, death.rate,
new.confirmed, new.deaths, current.confirmed))
data.latest %>% mutate(death.rate=death.rate %>% format(nsmall=1) %>% paste0('%')) %>%
kable('latex', booktabs=T, row.names=T, align=c('l', rep('r', 6)),
caption=paste0('Cases in Top 20 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'))
```

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



Confirmed Vs Deaths

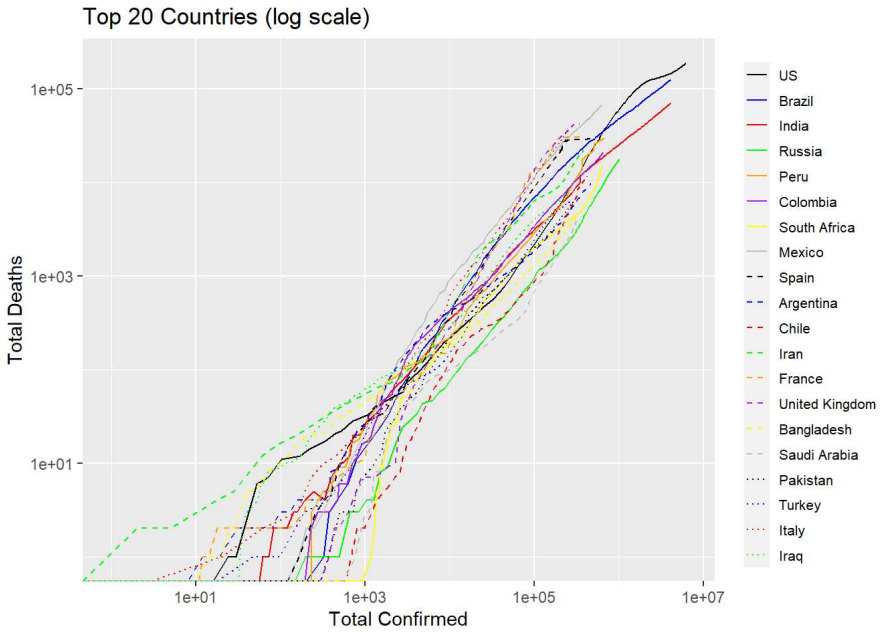
```
linetypes <- rep(c("dotted", "dashed", "solid"), each=8)
# colors <- rep(c('grey', 'yellow', 'purple', 'orange', 'green', 'red', 'blue', 'black'), 3)
linetypes <- rep(c("solid", "dashed", "dotted"), each=8)
colors <- rep(c('black', 'blue', 'red', 'green', 'orange', 'purple', 'yellow', 'grey'), 3)
df <- data %>% filter(country %in% setdiff(top.countries, c('World'))) %>%
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 + labs(title=paste0('Top 20 Countries'))
```




```
p + scale_x_log10() + scale_y_log10() +
labs(title=paste0('Top 20 Countries (log scale)'))
```

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

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



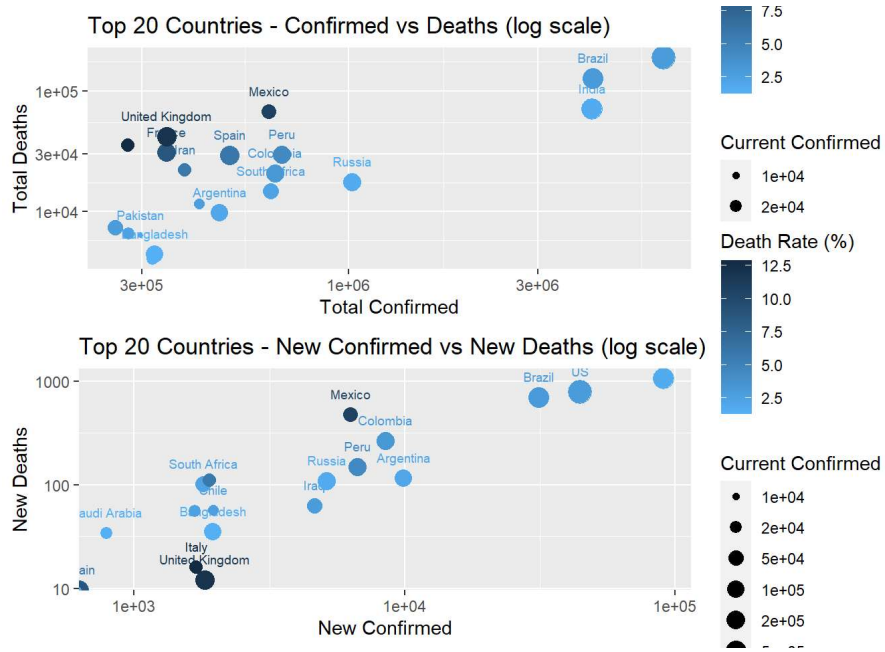
```
df <- data.latest %>% filter(country %in% setdiff(top.countries, 'World'))
## breaks for circle size in legend; needs to be adjusted accordingly when the number of total confirmed
breaks.confirmed <- c(5e3, 1e4, 2e4, 5e4, 1e5, 2e5, 5e5, 1e6, 2e6, 5e6, 1e7)
plot1 <- df %>% ggplot(aes(x=confirmed, y=deaths, col=death.rate, size=current.confirmed)) +
scale_size(name='Current Confirmed', trans='log2', breaks=breaks.confirmed) +
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() +
labs(title=paste0('Top 20 Countries - Confirmed vs Deaths (log scale)'))
plot2 <- df %>% ggplot(aes(x=new.confirmed, y=new.deaths, col=death.rate, size=current.confirmed)) +
scale_size(name='Current Confirmed', trans='log2', breaks=breaks.confirmed) +
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() +
labs(title=paste0('Top 20 Countries - New Confirmed vs New Deaths (log scale)'))
grid.arrange(plot1, plot2, ncol=1)
```

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

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

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

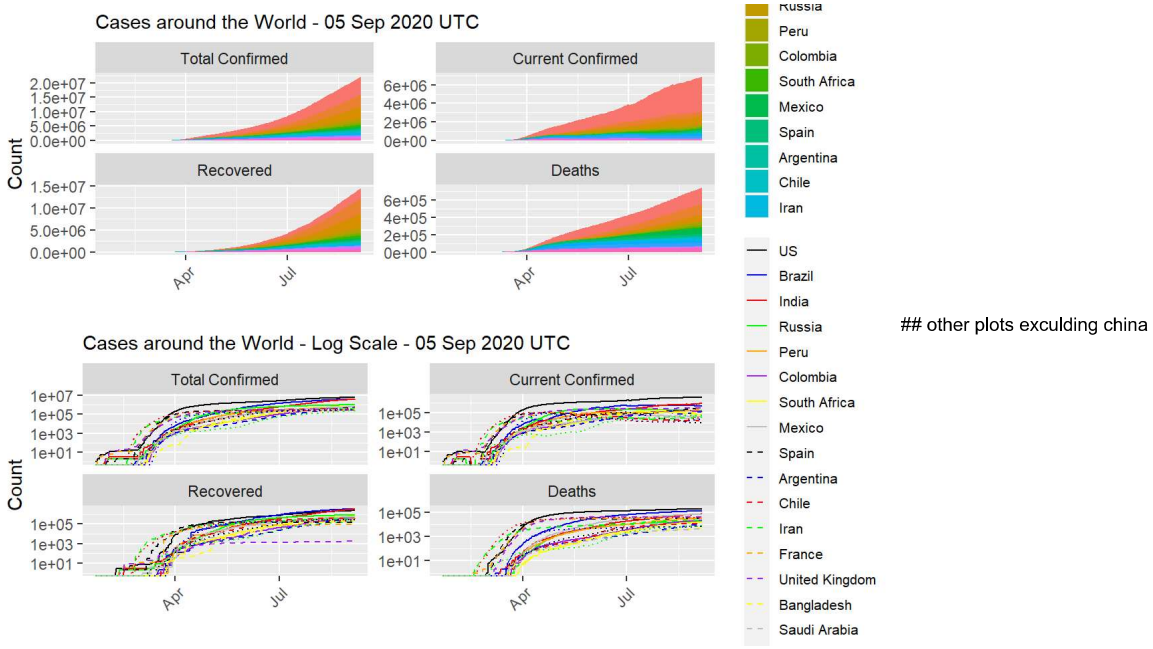
```
## Warning: Transformation introduced infinite values in continuous y-axis
```



Comparison Across Countries

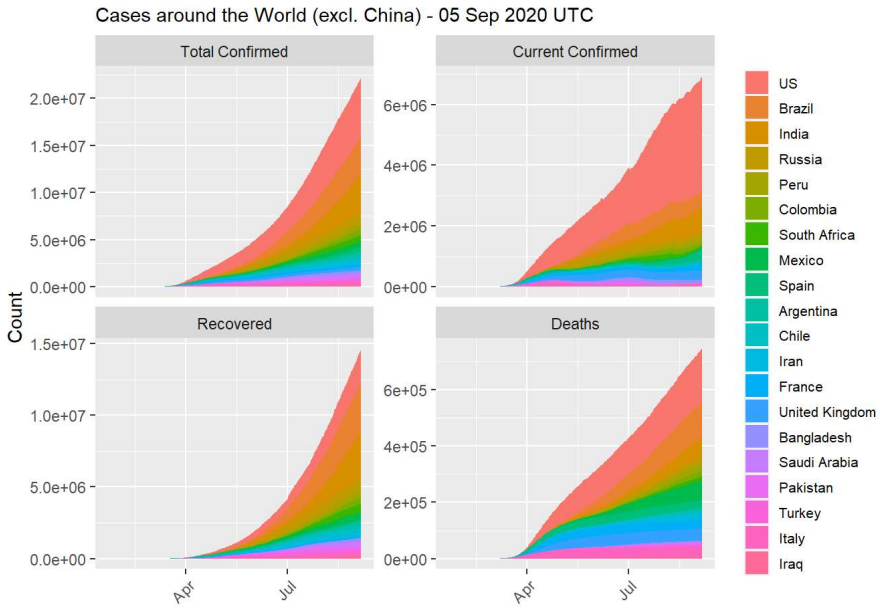
```
## plot: cases by type
df <- data.long %>% filter(country %in% top.countries) %<>%
mutate(country=country %>% factor(levels=c(top.countries)))
p <- df %>% filter(country != 'World') %>%
ggplot(aes(x=date, y=count)) + xlab('') + ylab('Count') +
theme(legend.title=element_blank(),
legend.text=element_text(size=8),
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)
```

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

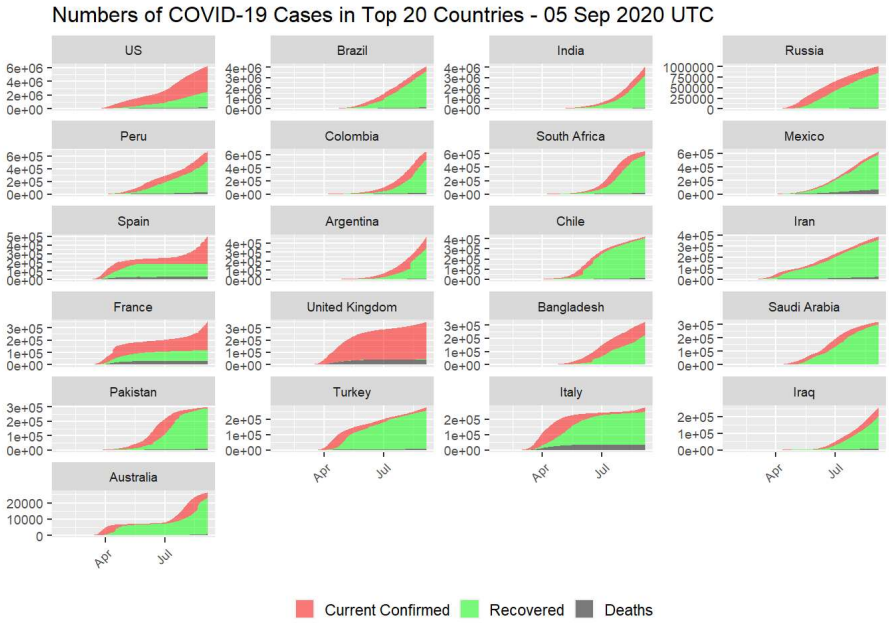


other plots exculding china

```
## 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))
```



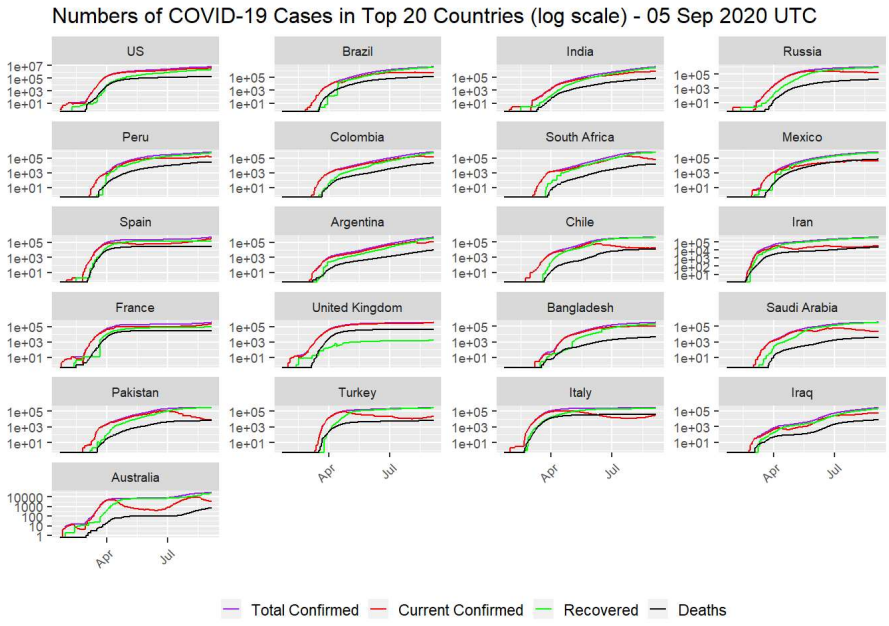

```
## if Australia in not in top 20, add it in and remove 'Others'
if(!('Australia' %in% top.countries)) {
top.countries %<>% setdiff('Others') %>% c('Australia')
df <- data.long %>% filter(country %in% top.countries) %<>%
mutate(country=country %>% factor(levels=c(top.countries)))
}
## cases by country - 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')
```



Cases by country - line plot

```
## 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')
```

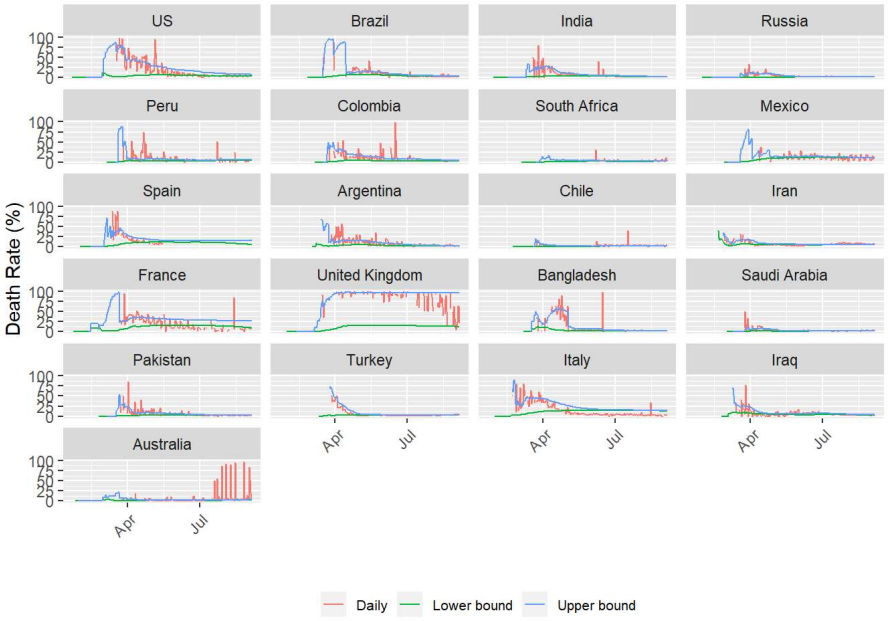
```
## Warning: Transformation introduced infinite values in continuous y-axis
```



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, 99)) +
facet_wrap(~country, ncol=4)
```

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



Countries with highest death rates

```
## sort the latest data by death rate, and if tie, by confirmed
df <- data %>% filter(date == max(date) & country != 'World' & confirmed >= 2000) %>%
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(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'))
```

Top 20 Countries with Highest Death Rates - 05 Sep 2020 UTC

	country	confirmed	new.confirmed	current.confirmed	recovered	deaths	new.deaths	death.rate
1	Italy	276,338	1,694	31,194	209,610	35,534	16	12.9%
2	United Kingdom	346,513	1,827	303,069	1,806	41,638	12	12.0%
3	Belgium	87,825	651	59,364	18,555	9,906	5	11.3%
4	Mexico	629,409	6,319	43,239	518,844	67,326	475	10.7%
5	France	347,267	0	228,610	87,927	30,730	0	8.8%
6	Netherlands	76,907	734	69,078	1,554	6,275	5	8.2%
7	Hungary	7,892	510	3,316	3,952	624	3	7.9%
8	Canada	133,511	370	6,351	117,968	9,192	2	6.9%
9	Sweden	84,985	0	79,150	0	5,835	0	6.9%
10	Sudan	13,407	218	5,850	6,725	832	9	6.2%
11	Ireland	29,534	231	4,393	23,364	1,777	0	6.0%
12	Spain	498,989	0	319,195	150,376	29,418	0	5.9%
13	Iran	384,666	1,894	30,381	332,131	22,154	110	5.8%
14	Ecuador	118,045	870	9,017	102,304	6,724	50	5.7%
15	Egypt	99,712	130	16,993	77,208	5,511	16	5.5%
16	China	90,025	17	460	84,837	4,728	0	5.3%
17	Switzerland	43,957	425	4,844	37,100	2,013	0	4.6%
18	Bolivia	120,241	661	45,277	69,566	5,398	55	4.5%
19	Mali	2,833	19	474	2,233	126	0	4.4%
20	Peru	676,848	6,703	148,771	498,523	29,554	149	4.4%