COVID-19 Data Analysis with R - China

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

12 April 2020

Contents

1 Introduction

This is a simple analysis of data around the Novel Coronavirus (COVID-19) in China, to demonstrate data processing and visualisation with R, *tidyverse* and *ggplot2*.

I have also writen a similar report for COVID-19 worldwide. If you are interested, please find it at http://www.rdatamining.com/docs/Coronavirus-data-analysis-world.pdf.

1.1 Data Source

The data sourse used for this analysis is Ding Xiang Yuan[1], which provides the data around the Novel Coronavirus (COVID-19) in China. Specifically, the data was retrieved from the *COVID-19/2019-nCoV Time Series Infection Data Warehouse* repository on GitHub[2]. Detailed descriptions of the data can be found

[1]https://ncov.dxy.cn/ncovh5/view/pneumonia
[2]https://github.com/BlankerL/DXY-COVID-19-Data

at http://lab.isaaclin.cn/nCoV/en.

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

2 Loading Data

At first, the dataset, which is a CSV file, is downloaded and saved as a local file, and then it is loaded into R.

```
url <- 'https://raw.githubusercontent.com/BlankerL/DXY-COVID-19-Data/master/csv/DXYOverall.csv'
filename <- './data/DXYOverall.csv'
download.file(url, filename)
data.raw <- read.csv(filename)
# summary(data.raw)
# names(data.raw)
```

The data was last updated at 2020-04-12 02:00:42.

Then we select relevant columns and have a look at the first 30 rows.

```
## select columns
data.raw %<>% select(c(updateTime, curedCount, deadCount,
                       currentConfirmedCount, confirmedCount, suspectedCount,
                       #  seriousCount,
                       curedIncr, deadIncr, confirmedIncr, suspectedIncr
                       # currentConfirmedIncr,
                       # seriousIncr
                       ))
head(data.raw, 30) %>%
  kable('latex', booktabs=T, caption='Raw Data (with Selected Columns Only)') %>%
  kable_styling(font_size=4, latex_options = c('striped', 'hold_position', 'repeat_header'))
```

3 Data Preparation

3.1 Selecting Last Record of Each Day

There are many records with different timestamps for every single day. For this analysis, we focuse on daily numbers and therefore keep only the last record on each day. To acheieve that, we group dataset by date and then select the first record from each group (i.e., from each day).

```
## convert from character to date
data.raw %<>% mutate(date=date(updateTime))
## sort by timestamp
# data.raw %<>% arrange(updateTime)
## select the latest record on each day
```

Table 1: Raw Data (with Selected Columns Only)

updateTime	curedCount	deadCount	currentConfirmedCount	confirmedCount	suspectedCount	curedIncr	deadIncr	confirmedIncr	suspectedIncr
2020-04-12 02:00:42	77976	3349	2075	83400	1183				
2020-04-12 01:04:58	77976	3349	2075	83400	1183	138	4	95	42
2020-04-12 00:57:53	77976	3349	2075	83400	1183	138	4	95	42
2020-04-12 00:39:42	77976	3349	2075	83400	1183	138	4	95	42
2020-04-12 00:37:40	77976	3349	2075	83400	1183	138	4	95	42
2020-04-11 23:46:57	77976	3349	2075	83400	1183	138	4	95	42
2020-04-11 23:36:50	77976	3349	2075	83400	1183	138	4	95	42
2020-04-11 20:53:33	77976	3349	2075	83400	1183	138	4	95	42
2020-04-11 19:59:44	77973	3349	2078	83400	1183	135	4	95	42
2020-04-11 19:44:32	77973	3349	2078	83400	1183	135	4	95	42
2020-04-11 18:41:32	77973	3349	2078	83400	1183	135	4	95	42
2020-04-11 18:34:27	77973	3349	2078	83400	1183	135	4	95	42
2020-04-11 18:12:08	77973	3349	2078	83400	1183	135	4	95	42
2020-04-11 18:00:55	77973	3349	2078	83400	1183	135	4	95	42
2020-04-11 17:20:24	77973	3349	2078	83400	1183	135	4	95	42
2020-04-11 16:51:55	77973	3349	2078	83400	1183	135	4	95	42
2020-04-11 16:46:51	77946	3349	2094	83389	1183	108	4	84	42
2020-04-11 16:35:43	77946	3349	2094	83389	1183	108	4	84	42
2020-04-11 16:19:31	77943	3349	2097	83389	1183	105	4	84	42
2020-04-11 16:18:31	77943	3349	2097	83389	1183	105	4	84	42
2020-04-11 16:05:21	77943	3349	2097	83389	1183	105	4	84	42
2020-04-11 15:39:54	77935	3349	2102	83386	1183	97	4	81	42
2020-04-11 15:36:52	77935	3349	2102	83386	1183	97	4	81	42
2020-04-11 14:54:20	77935	3349	2102	83386	1183	97	4	81	42
2020-04-11 13:55:30	77935	3349	2102	83386	1183	97	4	81	42
2020-04-11 13:37:18	77935	3349	2102	83386	1183	97	4	81	42
2020-04-11 12:43:35	77935	3349	2102	83386	1183	97	4	81	42
2020-04-11 12:42:35	77935	3349	2102	83386	1183	97	4	81	42
2020-04-11 12:06:10	77935	3349	2102	83386	1183	97	4	81	42
2020-04-11 11:33:43	77935	3349	2102	83386	1183	97	4	81	42

```r
data <- tbl_df(data.raw) %>%
    group_by(date) %>%
    top_n(1, updateTime)
## sort by date ascendingly and remove updateTime
data %<>% arrange(date) %>% select(-updateTime)

min.date <- min(data.raw$date)
max.date <- max(data.raw$date)
min.date.txt <- min.date %>% format('%d %B %Y')
max.date.txt <- max.date %>% format('%d %B %Y')
```

3.2 Daily New Cases and Death Rates

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

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

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

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

```r
## daily new cases
n <- nrow(data)
data %<>% as.data.frame() %>%
  mutate(new.dead = deadCount - lag(deadCount, n=1),
```

```r
         new.cured = curedCount - lag(curedCount, n=1),
         new.confirmed = confirmedCount - lag(confirmedCount, n=1))

## death rate based on total deaths and cured cases
data %<>% mutate(rate.upper = (100 * deadCount / (deadCount + curedCount)) %>% round(1))
## lower bound: death rate based on total confirmed cases
data %<>% mutate(rate.lower = (100 * deadCount / confirmedCount) %>% round(1))
## death rate based on the number of death/cured on every single day
data %<>% mutate(rate.daily = (100 * new.dead / (new.dead + new.cured)) %>% round(1))
```

3.3 Data Imputation

Some rows of column *currentConfirmedCount* are not populated in the raw dataset and we impute it as below

```r
## impute missing currentConfirmedCount
data %<>% mutate(currentConfirmedCount =
                   ifelse(is.na(currentConfirmedCount),
                          confirmedCount - curedCount - deadCount,
                          currentConfirmedCount))
```

3.4 Data Discrepancy

There is discrepancy in the dataset, which is checked with code below. Please understand that some numbers are not 100% accurate.

```r
## check for data discrepancy
data %<>% mutate(total = currentConfirmedCount + curedCount + deadCount)
data %<>% mutate(error.dead = new.dead - deadIncr,
                 error.cured = new.cured - curedIncr,
                 error.total = total - confirmedCount)
data$error.dead %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    -108      -1       0      -2       0       1      11
```

```r
data$error.cured %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -569.0  -302.0  -101.0  -167.7   -19.0     0.0      11
```

```r
data$error.total %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

```r
# head(data %>% as.data.frame())
```

Since today's cured and death counts are subject to change and will not be finalised until end of today, we might want to exclude today's rates and new cases from some plots in next section.

```r
# data %<>% arrange(date)
# if(data$date[n] == today()) {
#   data$rate.daily[n] <- NA
#   data$new.dead[n] <- NA
#   data$new.cured[n] <- NA
#   data$new.confirmed[n] <- NA
# }
```

4 Visualisation

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

4.1 Number of Cases

Figure 1 shows the number of COVID-19 cases in China. The line arnd area plots show the numbers of dead, cured, current confirmed and suppected cases. Note that, in the area plot, the total number of confirmed cases is represented by the total areas of confirmed, cured and deaths.

```
# total/current confirmed cases
p <- ggplot(data, aes(x=date)) +
  geom_line(aes(y=suspectedCount, color='Suspected')) +
  geom_line(aes(y=confirmedCount, color='Total Confimed')) +
  geom_line(aes(y=currentConfirmedCount, color='Current Confimed')) +
  geom_line(aes(y=curedCount, color='Cured')) +
  geom_line(aes(y=deadCount, color='Deaths')) +
  xlab('') + ylab('Count') +
  theme(legend.title=element_blank(), axis.text.x = element_text(angle=45, hjust=1)) +
  scale_color_manual(values = c(
    'Suspected' = 'orange',
    'Total Confimed' = 'purple',
    'Current Confimed' = 'red',
    'Cured' = 'green',
    'Deaths' = 'black'))

plot1 <- p + labs(title=paste0('Number of Cases - ', max.date.txt))
plot2 <- p + scale_y_continuous(trans='log10') +
  labs(title=paste0('Number of Cases (log scale) - ', max.date.txt))

## convert from wide to long format, for purpose of drawing a area plot
data.long <- data %>% select(c(date, suspectedCount,
                               currentConfirmedCount, curedCount, deadCount)) %>%
  rename(Suspected=suspectedCount, Confimed=currentConfirmedCount,
         Cured=curedCount, Deaths=deadCount) %>%
  gather(key=type, value=count, -date)
## set factor levels to show them in a desirable order
data.long %<>% mutate(type = factor(type, c('Suspected', 'Confimed', 'Cured', 'Deaths')))
## area plot
plot3 <- ggplot(data.long, aes(x=date, y=count, fill=type)) +
  geom_area(alpha=0.5) + xlab('') + ylab('Count') +
  labs(title=paste0('COVID-19 in China - ', max.date.txt)) +
  theme(legend.title=element_blank(), axis.text.x = element_text(angle=45, hjust=1)) +
  scale_fill_manual(values=c('orange', 'red', 'green', 'black'))

## show three plots together
grid.arrange(plot1, plot2, plot3, ncol=1)
```

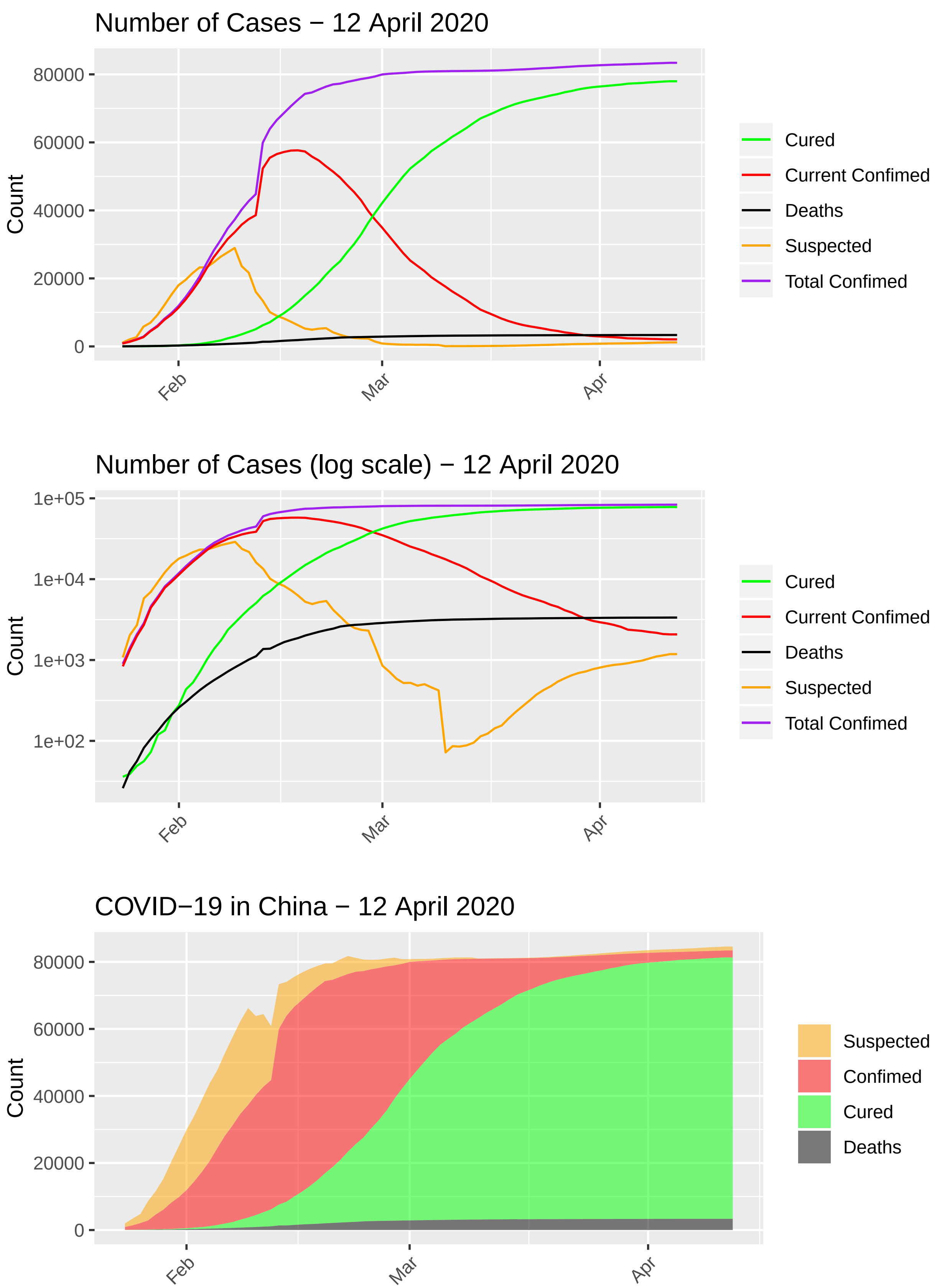

Figure 1: Numbers of COVID-19 Cases

Figure 1 (based on official stats) shows that the coronavirus seems to be contained in China, in that

- there are a lot of recovered cases (in green) every day,
- the remaining confrimed cases (in red) are shrinking significantly, and
- suspected cases (in orange) are almost gone.

4.2 Current (or Remaining) Confirmed Cases

```r
## current confirmed and its increase
plot1 <- ggplot(data, aes(x=date, y=currentConfirmedCount)) +
  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, 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)) +
  ylim(0, 4500)
grid.arrange(plot1, plot2, ncol=2)
```

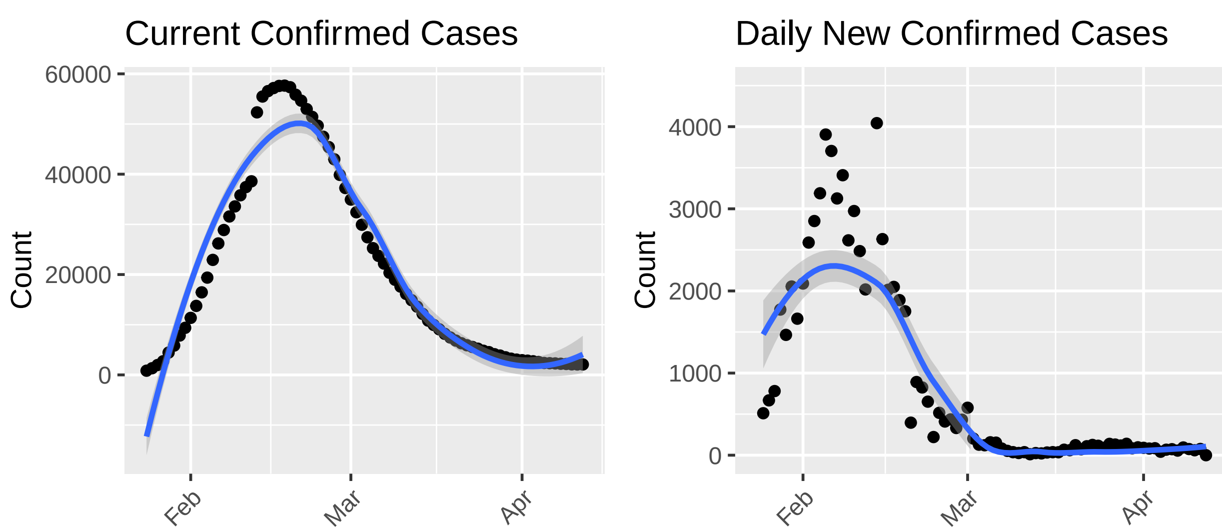

Figure 2: Current (or Remaining) Confirmed Cases

4.3 Deaths and Cured Cases

```r
## a scatter plot with a smoothed line and vertical x-axis labels
plot1 <- ggplot(data, aes(x=date, y=deadCount)) +
  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=curedCount)) +
  geom_point() + geom_smooth() +
  xlab('') + ylab('Count') + labs(title='Cured Cases') +
  theme(axis.text.x = element_text(angle=45, hjust=1))
plot3 <- ggplot(data, aes(x=date, y=new.dead))  +
  geom_point() + geom_smooth()  +
  xlab('') + ylab('Count') + labs(title='Daily New Deaths') +
```

```r
  theme(axis.text.x = element_text(angle=45, hjust=1))
plot4 <- ggplot(data, aes(x=date, y=new.cured)) +
  geom_point() + geom_smooth()  +
  xlab('') + ylab('Count') + labs(title='Daily New Cured 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)
```

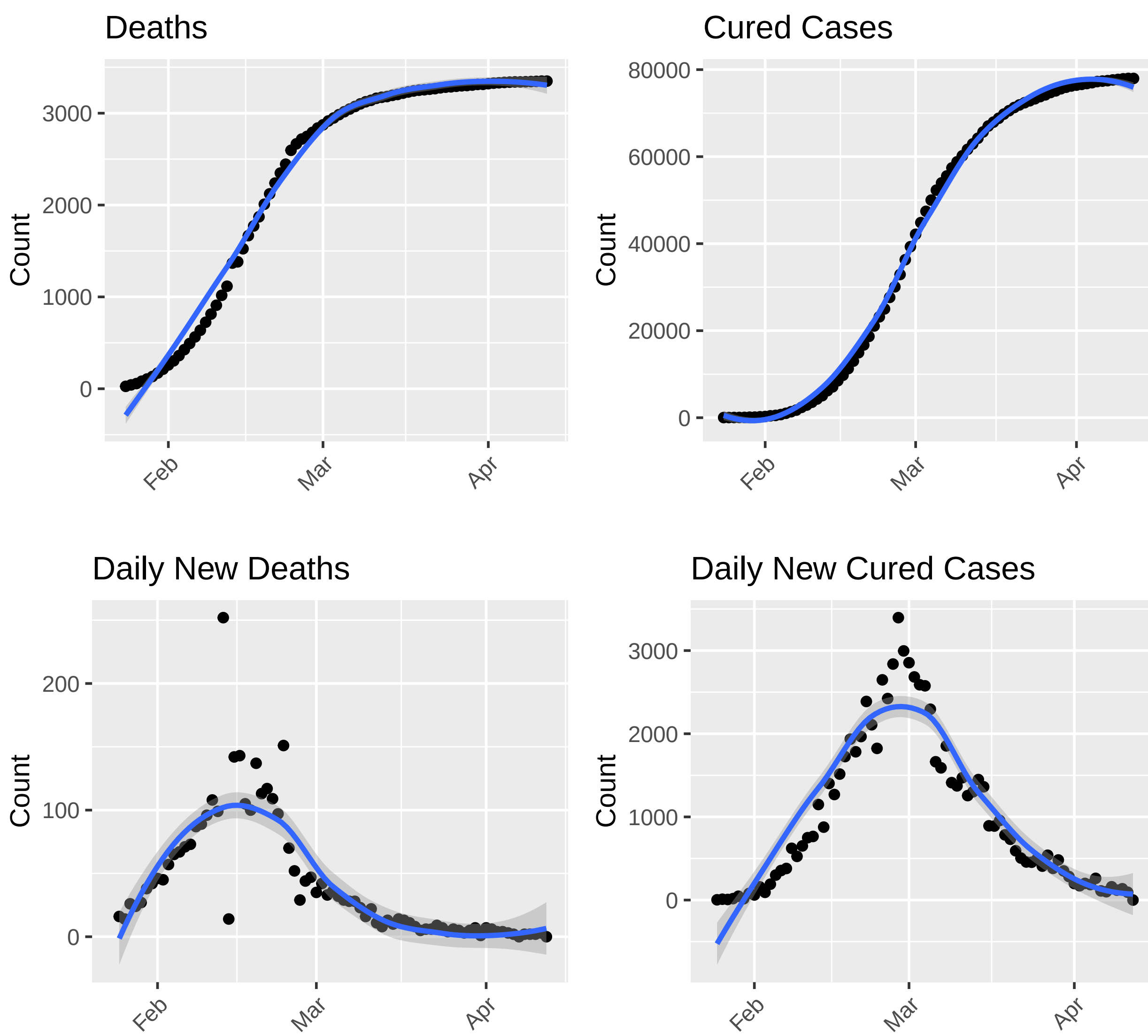

Figure 3: Deaths and Cured Cases

4.4 Death Rates

Figure 4 shows death rates caculated in three different ways (see Section 3.2 for details). The left chart shows the death rates from 24 January 2020 to 12 April 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 cured cases and fewer deaths 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 as of 12 April 2020, it will be between 4% and 4.1% (see the last row in the table at the end of this report).

```
## three death rates
plot1 <- ggplot(data, aes(x=date)) +
  geom_line(aes(y=rate.upper, colour='Upper bound')) +
  geom_line(aes(y=rate.lower, colour='Lower bound')) +
  geom_line(aes(y=rate.daily, colour='Daily')) +
  xlab('') + 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)
## focusing on last 2 weeks
plot2 <- ggplot(data[n-(14:0),], aes(x=date)) +
  geom_line(aes(y=rate.upper, colour='Upper bound')) +
  geom_line(aes(y=rate.lower, colour='Lower bound')) +
  geom_line(aes(y=rate.daily, colour='Daily')) +
  xlab('') + ylab('Death Rate (%)') + labs(title='Last two weeks') +
  theme(legend.position='bottom', legend.title=element_blank(),
        axis.text.x = element_text(angle=45, hjust=1)) +
  ylim(0, 8)
grid.arrange(plot1, plot2, ncol=2)
```

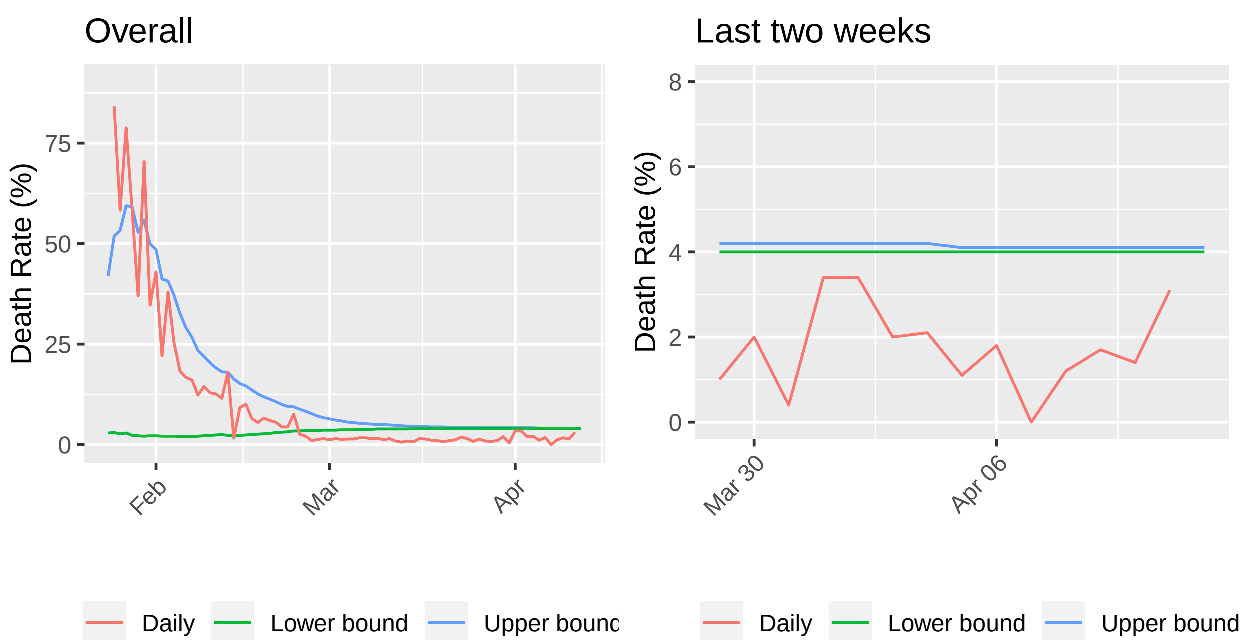

Figure 4: Death Rate

Appendix A. Processed Data

Blow is the processed data for this analysis and visualisation. Note that numbers in the first row of the table are subject to change, if they are about today (12 April 2020).

```
## sort by date descendingly and re-order columns
data %<>% arrange(desc(date)) %>%
  select(c(date, confirmedCount, deadCount, curedCount, currentConfirmedCount,
```

```
            new.confirmed, new.dead, new.cured, rate.upper, rate.daily, rate.lower))
## to make column names shorter for output purpose only
names(data) %<>% gsub(pattern='Count', replacement='')
## output as a table
data %>% kable('latex', booktabs=T, longtable=T, caption='COVID-19 in China',
                format.args=list(big.mark=',')) %>%
    kable_styling(font_size=6, latex_options = c('striped', 'hold_position', 'repeat_header'))
```

Table 2: COVID-19 in China

date	confirmed	dead	cured	currentConfirmed	new.confirmed	new.dead	new.cured	rate.upper	rate.daily	rate.lower
2020-04-12	83,400	3,349	77,976	2,075	0	0	0	4.1		4.0
2020-04-11	83,400	3,349	77,976	2,075	76	3	94	4.1	3.1	4.0
2020-04-10	83,324	3,346	77,882	2,096	60	2	136	4.1	1.4	4.0
2020-04-09	83,264	3,344	77,746	2,174	75	2	119	4.1	1.7	4.0
2020-04-08	83,189	3,342	77,627	2,220	94	2	160	4.1	1.2	4.0
2020-04-07	83,095	3,340	77,467	2,288	56	0	100	4.1	0.0	4.0
2020-04-06	83,039	3,340	77,367	2,332	73	2	110	4.1	1.8	4.0
2020-04-05	82,966	3,338	77,257	2,371	67	3	261	4.1	1.1	4.0
2020-04-04	82,899	3,335	76,996	2,568	42	4	186	4.2	2.1	4.0
2020-04-03	82,857	3,331	76,810	2,716	85	4	200	4.2	2.0	4.0
2020-04-02	82,772	3,327	76,610	2,835	81	6	172	4.2	3.4	4.0
2020-04-01	82,691	3,321	76,438	2,932	90	7	199	4.2	3.4	4.0
2020-03-31	82,601	3,314	76,239	3,048	96	1	283	4.2	0.4	4.0
2020-03-30	82,505	3,313	75,956	3,236	84	7	350	4.2	2.0	4.0
2020-03-29	82,421	3,306	75,606	3,509	139	5	482	4.2	1.0	4.0
2020-03-28	82,282	3,301	75,124	3,857	118	3	381	4.2	0.8	4.0
2020-03-27	82,164	3,298	74,743	4,123	130	5	539	4.2	0.9	4.0
2020-03-26	82,034	3,293	74,204	4,537	138	6	408	4.2	1.4	4.0
2020-03-25	81,896	3,287	73,796	4,813	90	4	493	4.3	0.8	4.0
2020-03-24	81,806	3,283	73,303	5,220	115	7	455	4.3	1.5	4.0
2020-03-23	81,691	3,276	72,848	5,567	125	9	458	4.3	1.9	4.0
2020-03-22	81,566	3,267	72,390	5,909	109	6	505	4.3	1.2	4.0
2020-03-21	81,457	3,261	71,885	6,311	72	6	593	4.3	1.0	4.0
2020-03-20	81,385	3,255	71,292	6,838	122	5	731	4.4	0.7	4.0
2020-03-19	81,263	3,250	70,561	7,452	61	8	784	4.4	1.0	4.0
2020-03-18	81,202	3,242	69,777	8,183	67	11	957	4.4	1.1	4.0
2020-03-17	81,135	3,231	68,820	9,084	36	13	890	4.5	1.4	4.0
2020-03-16	81,099	3,218	67,930	9,951	37	14	893	4.5	1.5	4.0
2020-03-15	81,062	3,204	67,037	10,821	33	10	1,362	4.6	0.7	4.0
2020-03-14	81,029	3,194	65,675	12,160	22	13	1,449	4.6	0.9	3.9
2020-03-13	81,007	3,181	64,226	13,600	26	8	1,302	4.7	0.6	3.9
2020-03-12	80,981	3,173	62,924	14,884	12	11	1,256	4.8	0.9	3.9
2020-03-11	80,969	3,162	61,668	16,139	37	22	1,471	4.9	1.5	3.9
2020-03-10	80,932	3,140	60,197	17,595	27	16	1,373	5.0	1.2	3.9
2020-03-09	80,905	3,124	58,824	18,957	37	23	1,412	5.0	1.6	3.9
2020-03-08	80,868	3,101	57,412	20,355	53	28	1,854	5.1	1.5	3.8
2020-03-07	80,815	3,073	55,558	22,184	81	28	1,590	5.2	1.7	3.8
2020-03-06	80,734	3,045	53,968	23,721	153	29	1,663	5.3	1.7	3.8
2020-03-05	80,581	3,016	52,305	25,260	157	32	2,295	5.5	1.4	3.7
2020-03-04	80,424	2,984	50,010	27,430	121	36	2,576	5.6	1.4	3.7
2020-03-03	80,303	2,948	47,434	29,921	128	33	2,589	5.9	1.3	3.7
2020-03-02	80,175	2,915	44,845	32,415	203	42	2,683	6.1	1.5	3.6
2020-03-01	79,972	2,873	42,162	34,937	578	35	2,854	6.4	1.2	3.6
2020-02-29	79,394	2,838	39,308	37,248	432	47	2,996	6.7	1.5	3.6
2020-02-28	78,962	2,791	36,312	39,859	331	44	3,396	7.1	1.3	3.5
2020-02-27	78,631	2,747	32,916	42,968	436	29	2,838	7.7	1.0	3.5
2020-02-26	78,195	2,718	30,078	45,399	410	52	2,423	8.3	2.1	3.5
2020-02-25	77,785	2,666	27,655	47,464	516	70	2,648	8.8	2.6	3.4
2020-02-24	77,269	2,596	25,007	49,666	221	151	1,824	9.4	7.6	3.4
2020-02-23	77,048	2,445	23,183	51,420	652	97	2,108	9.5	4.4	3.2
2020-02-22	76,396	2,348	21,075	52,973	825	109	2,388	10.0	4.4	3.1
2020-02-21	75,571	2,239	18,687	54,645	891	117	1,966	10.7	5.6	3.0
2020-02-20	74,680	2,122	16,721	55,837	396	113	1,783	11.3	6.0	2.8
2020-02-19	74,284	2,009	14,938	57,337	1,752	137	1,935	11.9	6.6	2.7
2020-02-18	72,532	1,872	13,003	57,657	1,888	100	1,725	12.6	5.5	2.6
2020-02-17	70,644	1,772	11,278	57,594	2,049	105	1,515	13.6	6.5	2.5
2020-02-16	68,595	1,667	9,763	57,165	2,014	143	1,269	14.6	10.1	2.4

Table 2: COVID-19 in China *(continued)*

date	confirmed	dead	cured	currentConfirmed	new.confirmed	new.dead	new.cured	rate.upper	rate.daily	rate.lower
2020-02-15	66,581	1,524	8,494	56,563	2,631	142	1,402	15.2	9.2	2.3
2020-02-14	63,950	1,382	7,092	55,476	4,043	14	877	16.3	1.6	2.2
2020-02-13	59,907	1,368	6,215	52,324	15,142	252	1,149	18.0	18.0	2.3
2020-02-12	44,765	1,116	5,066	38,583	2,018	99	765	18.1	11.5	2.5
2020-02-11	42,747	1,017	4,301	37,429	2,485	108	750	19.1	12.6	2.4
2020-02-10	40,262	909	3,551	35,802	2,973	96	651	20.4	12.9	2.3
2020-02-09	37,289	813	2,900	33,576	2,616	89	525	21.9	14.5	2.2
2020-02-08	34,673	724	2,375	31,574	3,409	87	622	23.4	12.3	2.1
2020-02-07	31,264	637	1,753	28,874	3,126	73	380	26.7	16.1	2.0
2020-02-06	28,138	564	1,373	26,201	3,704	71	355	29.1	16.7	2.0
2020-02-05	24,434	493	1,018	22,923	3,904	67	300	32.6	18.3	2.0
2020-02-04	20,530	426	718	19,386	3,189	65	191	37.2	25.4	2.1
2020-02-03	17,341	361	527	16,453	2,851	57	93	40.7	38.0	2.1
2020-02-02	14,490	304	434	13,752	2,589	45	159	41.2	22.1	2.1
2020-02-01	11,901	259	275	11,367	2,090	46	61	48.5	43.0	2.2
2020-01-31	9,811	213	214	9,384	1,662	42	79	49.9	34.7	2.2
2020-01-30	8,149	171	135	7,843	2,054	38	16	55.9	70.4	2.1
2020-01-29	6,095	133	119	5,843	1,465	27	46	52.8	37.0	2.2
2020-01-28	4,630	106	73	4,451	1,773	24	17	59.2	58.5	2.3
2020-01-27	2,857	82	56	2,719	781	26	7	59.4	78.8	2.9
2020-01-26	2,076	56	49	1,971	668	14	10	53.3	58.3	2.7
2020-01-25	1,408	42	39	1,327	511	16	3	51.9	84.2	3.0
2020-01-24	897	26	36	835				41.9		2.9

Appendix B. How to Cite This Work

Citation

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

BibTex

@techreport{Zhao2020Covid19china,
Author = {Yanchang Zhao},
Institution = {RDataMining.com},
Title = {COVID-19 Data Analysis with R – China},
Url = {http://www.rdatamining.com/docs/Coronavirus-data-analysis-china.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!