

# Modeling Deaths

Safiya Sirota

2023-03-30

## Set up

### Load in Anja's and Ting-Hsuan's code

```
source("anja.R")

## Warning: package 'lubridate' was built under R version 4.2.2

##
## Attaching package: 'lubridate'

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

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.1    v purrr  1.0.1
## v tibble  3.1.8    v dplyr  1.1.0
## v tidyr   1.3.0    v stringr 1.5.0
## v readr   2.1.4    v forcats 1.0.0

## Warning: package 'ggplot2' was built under R version 4.2.2

## Warning: package 'tidyr' was built under R version 4.2.2

## Warning: package 'readr' was built under R version 4.2.2

## Warning: package 'purrr' was built under R version 4.2.2

## Warning: package 'dplyr' was built under R version 4.2.2

## Warning: package 'stringr' was built under R version 4.2.2

## Warning: package 'forcats' was built under R version 4.2.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date() masks base::date()
## x dplyr::filter() masks stats::filter()
## x lubridate::intersect() masks base::intersect()
## x dplyr::lag() masks stats::lag()
## x lubridate::setdiff() masks base::setdiff()
## x lubridate::union() masks base::union()
## Rows: 727 Columns: 19
## -- Column specification -----
## Delimiter: ","
## chr (1): date_of_interest
## dbl (18): CASE_COUNT, HOSPITALIZED_COUNT, DEATH_COUNT, BX_CASE_COUNT, BX_HOS...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
source("tinghsuan_wave1case.R")
```

```
## Rows: 727 Columns: 19
## -- Column specification -----
## Delimiter: ","
## chr (1): date_of_interest
## dbl (18): CASE_COUNT, HOSPITALIZED_COUNT, DEATH_COUNT, BX_CASE_COUNT, BX_HOS...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
## Warning: package 'ggpubr' was built under R version 4.2.2
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```

```
## Warning in log(1 + d * exp(-k * (t - t0))): NaNs produced
```











[illegible]































[illegible]



[illegible]















[illegible]

[illegible]



[illegible]



[illegible]



[illegible]

















[illegible]





[illegible]



[illegible]

[illegible]















[illegible]

[illegible]



[illegible]





[illegible]

[illegible]







[illegible]

[illegible]



[illegible]



[illegible]























[illegible]



[illegible]





[illegible]

[illegible]



[illegible]





[illegible]











[illegible]







[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]









[illegible]



[illegible]



[illegible]





[illegible]



[illegible]



[illegible]

[illegible]





[illegible]

[illegible]











[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]











[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]



[illegible]



[illegible]



[illegible]



[illegible]





[illegible]





[illegible]





[illegible]

[illegible]



[illegible]





[illegible]

[illegible]

[illegible]



[illegible]







[illegible]

[illegible]

[illegible]



[illegible]





[illegible]

[illegible]





[illegible]



[illegible]

[illegible]





[illegible]

[illegible]

[illegible]







[illegible]











[illegible]









[illegible]



[illegible]

[illegible]



















[illegible]



[illegible]







[illegible]





[illegible]





[illegible]



[illegible]









[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]









[illegible]



[illegible]





[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]





[illegible]











[illegible]

[illegible]





[illegible]





[illegible]

[illegible]

[illegible]



[illegible]



[illegible]







[illegible]















[illegible]







[illegible]





[illegible]





















[illegible]

[illegible]





[illegible]









[illegible]

[illegible]

















[illegible]

[illegible]









[illegible]





[illegible]

[illegible]

[illegible]



[illegible]







[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]







[illegible]







[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]









[illegible]

[illegible]









[illegible]

[illegible]



[illegible]

[illegible]







[illegible]

## Wave 1

Overwrite `df_wave` to consider death counts instead of death counts

Changing to deaths and setting up the data for each borough in wave 1

```

# All of NYC
df_wave1_nyc <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(date, death_count) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(death_count)
  ) %>%
  select(!death_count)

# Bronx
df_wave1_bx <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, bx_death_count)) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(bx_death_count)
  ) %>%
  select(!bx_death_count)

# Brooklyn
df_wave1_bk <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, bk_death_count)) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(bk_death_count)
  ) %>%
  select(!bk_death_count)

# Manhattan
df_wave1_mn <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, mn_death_count)) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(mn_death_count)
  ) %>%
  select(!mn_death_count)

# Queens
df_wave1_qn <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, qn_death_count)) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(qn_death_count)
  ) %>%
  select(!qn_death_count)

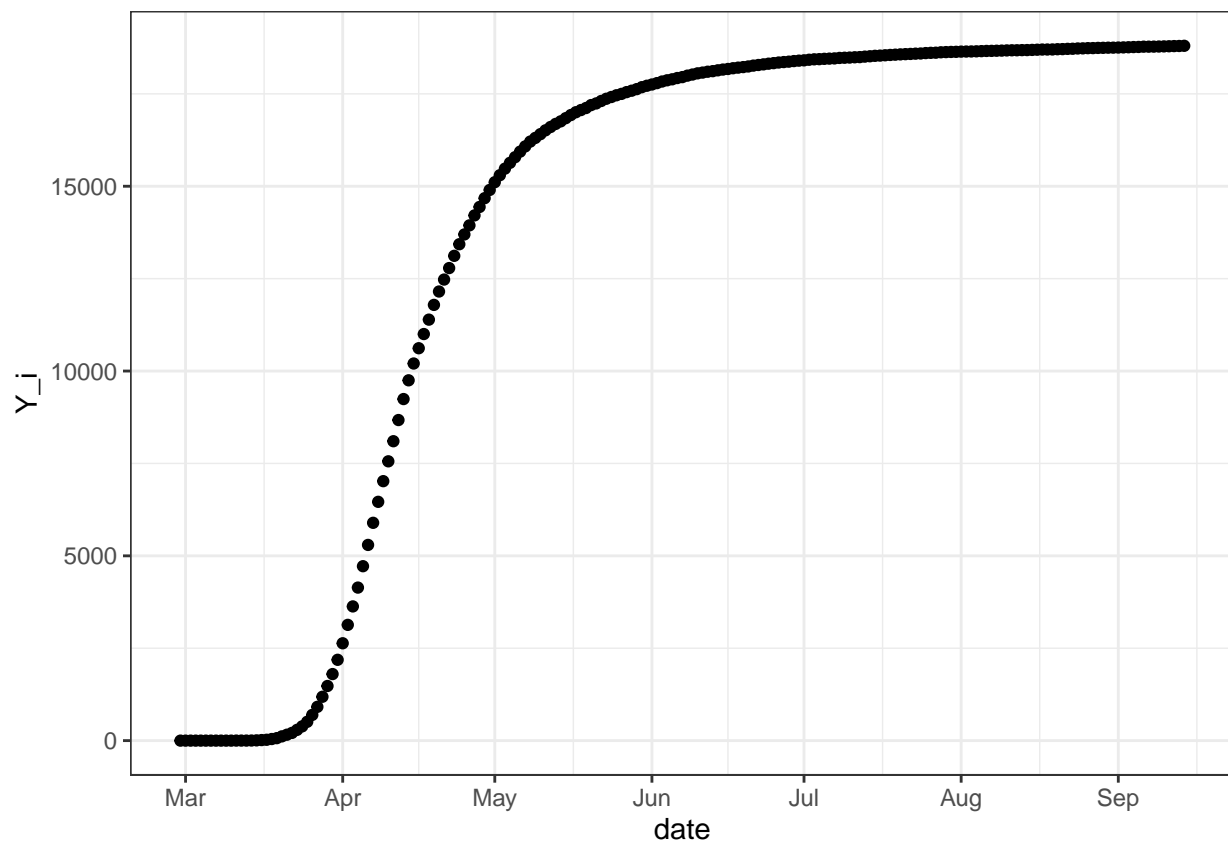
# Staten Island
df_wave1_si <- df %>%
  filter(date >= start_date & date <= end_date) %>%

```

```
select(c(date, si_death_count)) %>%
mutate(
  t_i = get_diff(start_date, date),
  Y_i = cumsum(si_death_count)
) %>%
select(!si_death_count)
```

Check cumulative deaths over wave 1 for all of NYC

```
df_wave1_nyc %>%
  ggplot(aes(x = date, y = Y_i)) +
  geom_point() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



## Run N-R

Running N-R for each borough and all of NYC using code from loaded R files

```
# All of NYC
start_nyc <- get_start(df_wave1_nyc, infl_point = "2020-04-10")
res_nyc <- NewtonRaphson(df_wave1_nyc, start_nyc)
final_nyc <- res_nyc[nrow(res_nyc), 4:7]
```

```

# Bronx
start_bx <- get_start(df_wave1_bx, infl_point = "2020-04-10")
res_bx <- NewtonRaphson(df_wave1_bx, start_bx)
final_bx <- res_bx[nrow(res_bx), 4:7]

# Brooklyn
start_bk <- get_start(df_wave1_bk, infl_point = "2020-04-10")
res_bk <- NewtonRaphson(df_wave1_bk, start_bk)
final_bk <- res_bk[nrow(res_bk), 4:7]

# Manhattan
start_mn <- get_start(df_wave1_mn, infl_point = "2020-04-10")
res_mn <- NewtonRaphson(df_wave1_mn, start_mn)
final_mn <- res_mn[nrow(res_mn), 4:7]

# Queens
start_qn <- get_start(df_wave1_qn, infl_point = "2020-04-10")
res_qn <- NewtonRaphson(df_wave1_qn, start_qn)
final_qn <- res_qn[nrow(res_qn), 4:7]

# Staten Island
start_si <- get_start(df_wave1_si, infl_point = "2020-04-10")
res_si <- NewtonRaphson(df_wave1_si, start_si)
final_si <- res_si[nrow(res_si), 4:7]

```

## Plot final curves

```

# Bronx
death_wave1_bx <- df_wave1_bx %>%
  mutate(
    Y_i_hat_final = N(t_i, final_bx)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Bronx", x = "Days (since Feb 29, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

# Brooklyn
death_wave1_bk <- df_wave1_bk %>%
  mutate(
    Y_i_hat_final = N(t_i, final_bk)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Brooklyn", x = "Days (since Feb 29, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

## Manhattan
death_wave1_mn <- df_wave1_mn %>%

```

```

mutate(
  Y_i_hat_final = N(t_i, final_mn)
) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Manhattan", x = "Days (since Feb 29, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

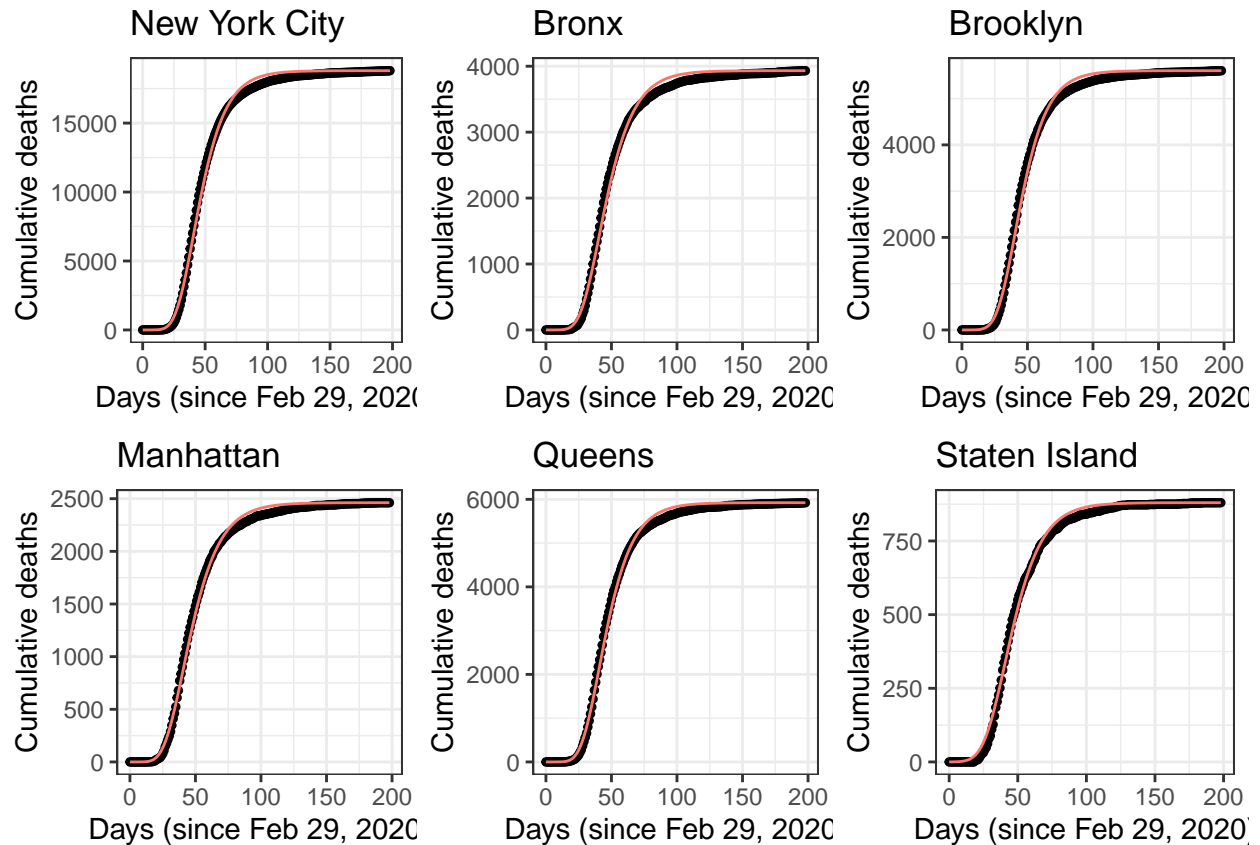
## Queens
death_wave1_qn <- df_wave1_qn %>%
  mutate(
    Y_i_hat_final = N(t_i, final_qn)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Queens", x = "Days (since Feb 29, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

## Staten Island
death_wave1_si <- df_wave1_si %>%
  mutate(
    Y_i_hat_final = N(t_i, final_si)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Staten Island", x = "Days (since Feb 29, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

## NYC
death_wave1_nyc <- df_wave1_nyc %>%
  mutate(
    Y_i_hat_final = N(t_i, final_nyc)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "New York City", x = "Days (since Feb 29, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

ggarrange(death_wave1_nyc, death_wave1_bx, death_wave1_bk, death_wave1_mn, death_wave1_qn, death_wave1_si)

```



## Wave 1 final parameters

```
final_nyc_parameter <-
  tibble(
    a = final_nyc[["a"]],
    k = final_nyc[["k"]],
    d = final_nyc[["d"]],
    t0 = final_nyc[["t0"]]) %>%
  mutate(location = "New York City")

final_bk_parameter <-
  tibble(
    a = final_bk[["a"]],
    k = final_bk[["k"]],
    d = final_bk[["d"]],
    t0 = final_bk[["t0"]]) %>%
  mutate(location = "Brooklyn")

final_bx_parameter <-
  tibble(
    a = final_bx[["a"]],
    k = final_bx[["k"]],
    d = final_bx[["d"]],
    t0 = final_bx[["t0"]]) %>%
```



```

mutate(location = "Bronx")

final_mn_parameter <-
  tibble(
    a = final_mn[["a"]],
    k = final_mn[["k"]],
    d = final_mn[["d"]],
    t0 = final_mn[["t0"]]) %>%
  mutate(location = "Manhattan")

final_qn_parameter <-
  tibble(
    a = final_qn[["a"]],
    k = final_qn[["k"]],
    d = final_qn[["d"]],
    t0 = final_qn[["t0"]]) %>%
  mutate(location = "Queens")

final_si_parameter <-
  tibble(
    a = final_si[["a"]],
    k = final_si[["k"]],
    d = final_si[["d"]],
    t0 = final_si[["t0"]]) %>%
  mutate(location = "Staten Island")

bind_rows(final_nyc_parameter, final_bk_parameter, final_bx_parameter, final_mn_parameter,
          final_qn_parameter, final_si_parameter) %>%
  knitr::kable()

```

	a	k	d	t0	location
	18799.9975	0.0705570	0.0167689	40.85405	New York City
	5602.9895	0.0747471	0.0592430	40.78483	Brooklyn
	3931.9844	0.0680085	-0.0037896	40.80752	Bronx
	2461.9921	0.0653303	-0.0179362	40.98036	Manhattan
	5922.9928	0.0711746	0.0033253	40.88255	Queens
	879.9963	0.0690191	0.1212710	41.03465	Staten Island

## Wave 2

### Reset start and end dates

Loading in wave2death.R to reset start and end dates and to replace the `get_start` function with the proper method/parameters for Wave 2

```

source("wave2case.R")

## Rows: 727 Columns: 19
## -- Column specification -----
## Delimiter: ","

```

```
## chr (1): date_of_interest
## dbl (18): CASE_COUNT, HOSPITALIZED_COUNT, DEATH_COUNT, BX_CASE_COUNT, BX_HOS...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

## Set up data

```
# All of NYC
df_wave2_nyc <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(date, death_count) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(death_count)
  ) %>%
  select(!death_count)

# Bronx
df_wave2_bx <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, bx_death_count)) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(bx_death_count)
  ) %>%
  select(!bx_death_count)

# Brooklyn
df_wave2_bk <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, bk_death_count)) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(bk_death_count)
  ) %>%
  select(!bk_death_count)

# Manhattan
df_wave2_mn <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, mn_death_count)) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(mn_death_count)
  ) %>%
  select(!mn_death_count)

# Queens
df_wave2_qn <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, qn_death_count)) %>%
```

```

mutate(
  t_i = get_diff(start_date, date),
  Y_i = cumsum(qn_death_count)
) %>%
select(!qn_death_count)

# Staten Island
df_wave2_si <- df %>%
  filter(date >= start_date & date <= end_date) %>%
  select(c(date, si_death_count)) %>%
  mutate(
    t_i = get_diff(start_date, date),
    Y_i = cumsum(si_death_count)
  ) %>%
  select(!si_death_count)

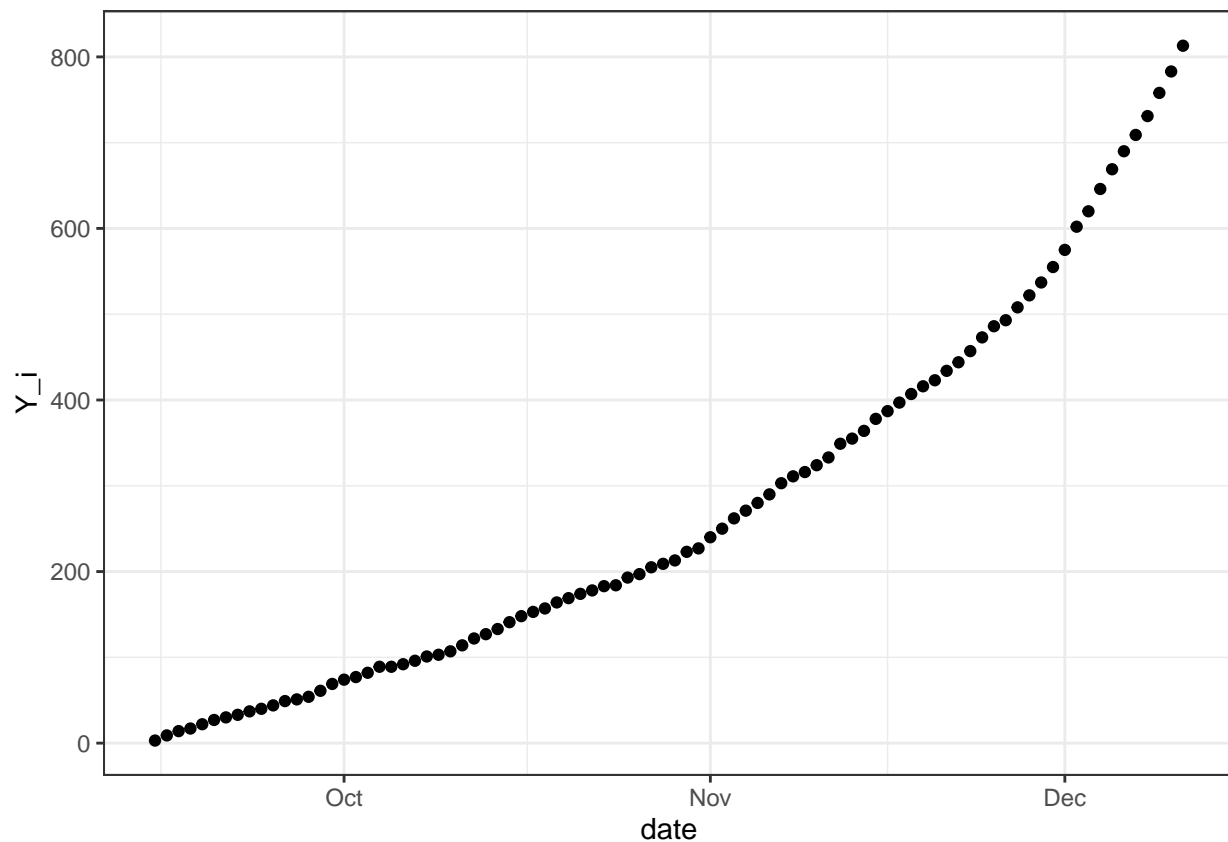
```

Check cumulative deaths over wave 2 for all of NYC

```

df_wave2_nyc %>%
  ggplot(aes(x = date, y = Y_i)) +
  geom_point() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")

```



## Run N-R

Running N-R for each borough and all of NYC using code from loaded R files

```
# All of NYC
start_nyc2 <- get_start(df_wave2_nyc)
res_nyc2 <- NewtonRaphson(df_wave2_nyc, start_nyc2)
final_nyc2 <- res_nyc2[nrow(res_nyc2), 4:7]

# Bronx
start_bx2 <- get_start(df_wave2_bx)
res_bx2 <- NewtonRaphson(df_wave2_bx, start_bx2)
final_bx2 <- res_bx2[nrow(res_bx2), 4:7]

# Brooklyn
start_bk2 <- get_start(df_wave2_bk)
res_bk2 <- NewtonRaphson(df_wave2_bk, start_bk2)
final_bk2 <- res_bk2[nrow(res_bk2), 4:7]

# Manhattan
start_mn2 <- get_start(df_wave2_mn)
res_mn2 <- NewtonRaphson(df_wave2_mn, start_mn2)
final_mn2 <- res_mn2[nrow(res_mn2), 4:7]

# Queens
start_qn2 <- get_start(df_wave2_qn)
res_qn2 <- NewtonRaphson(df_wave2_qn, start_qn2)
final_qn2 <- res_qn2[nrow(res_qn2), 4:7]

# Staten Island
start_si2 <- get_start(df_wave2_si)
res_si2 <- NewtonRaphson(df_wave2_si, start_si2)
final_si2 <- res_si2[nrow(res_si2), 4:7]
```

## Plot final curves

```
# Bronx
death_wave2_bx <- df_wave2_bx %>%
  mutate(
    Y_i_hat_final = N(t_i, final_bx2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Bronx", x = "Days (since Sep 15, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

# Brooklyn
death_wave2_bk <- df_wave2_bk %>%
  mutate(
    Y_i_hat_final = N(t_i, final_bk2)
  ) %>%
```

```

ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Brooklyn", x = "Days (since Sep 15, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

## Manhattan
death_wave2_mn <- df_wave2_mn %>%
  mutate(
    Y_i_hat_final = N(t_i, final_mn2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Manhattan", x = "Days (since Sep 15, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

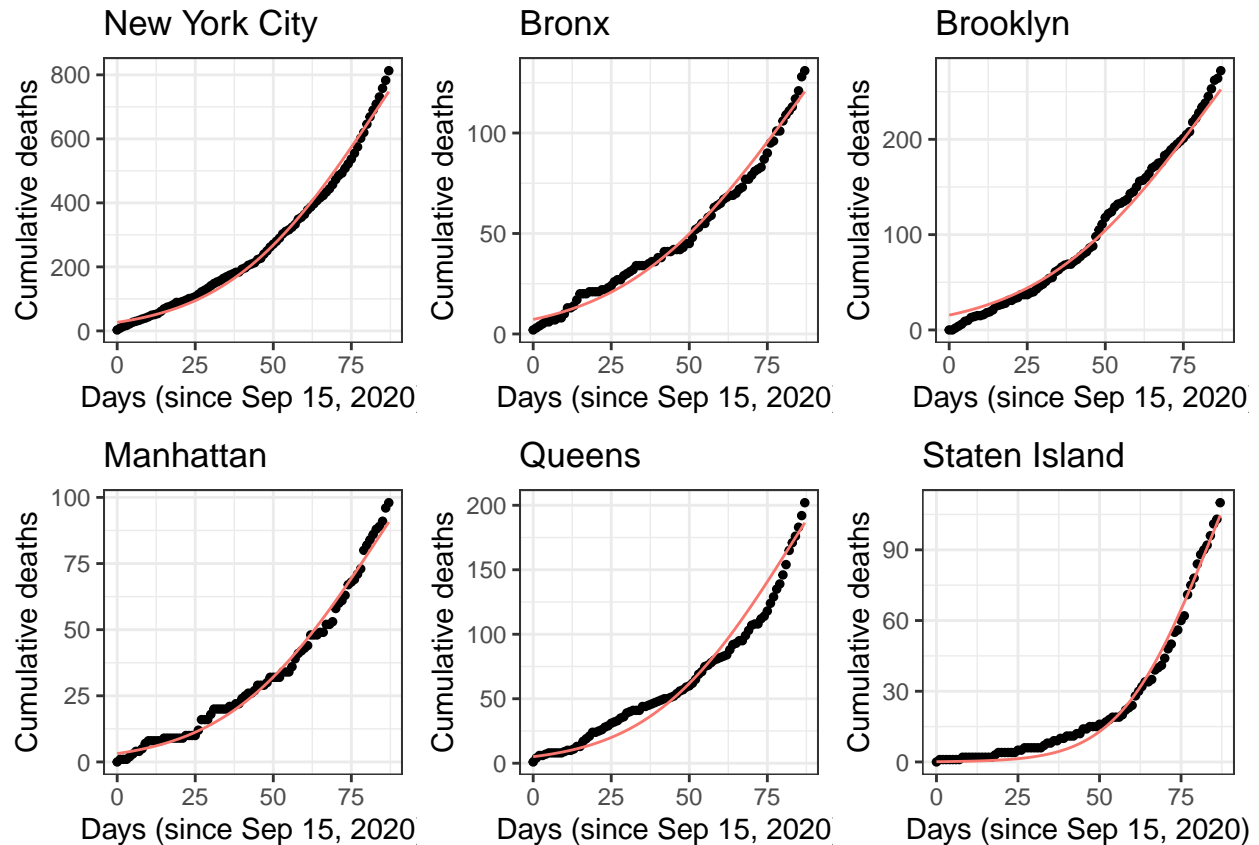
## Queens
death_wave2_qn <- df_wave2_qn %>%
  mutate(
    Y_i_hat_final = N(t_i, final_qn2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Queens", x = "Days (since Sep 15, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

## Staten Island
death_wave2_si <- df_wave2_si %>%
  mutate(
    Y_i_hat_final = N(t_i, final_si2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "Staten Island", x = "Days (since Sep 15, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

## NYC
death_wave2_nyc <- df_wave2_nyc %>%
  mutate(
    Y_i_hat_final = N(t_i, final_nyc2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
  geom_line(aes(y = Y_i_hat_final, color = "red")) +
  labs(title = "New York City", x = "Days (since Sep 15, 2020)", y = "Cumulative deaths") +
  theme(legend.position = "none")

ggarrange(death_wave2_nyc, death_wave2_bx, death_wave2_bk, death_wave2_mn, death_wave2_qn, death_wave2_si)

```



## Task 1.3

### NYC extended curve

```
# Constructing a vector of future dates
more_days <- seq(from = as.Date("2020-12-12"), to = as.Date("2021-06-12"),
  by = 'day')
more_days <- ymd(more_days)
currow <- nrow(df_wave2_nyc)
more_times <- seq(from = currow + 1,
  to = length(more_days) + currow)
Ys <- rep(NA, length(more_days))
more_data <- tibble(date = more_days, t_i = more_times, Y_i = Ys)

# Adding data to current data set
df_wave2_nyc_ext <- rbind(df_wave2_nyc, more_data)

death_wave2_nyc_ext <- df_wave2_nyc_ext %>%
  mutate(
    Y_i_hat_final = N(t_i, final_nyc2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = Y_i), size = 1) +
```

```
geom_line(aes(y = Y_i_hat_final, color = "red")) +
labs(title = "New York City", x = "Days (since Sep 15, 2020)", y = "Cumulative deaths") +
theme(legend.position = "none") +
coord_cartesian(xlim = c(0, 300))
```

## BX extended curve

```
# Constructing a vector of future dates
more_days <- seq(from = as.Date("2020-12-12"), to = as.Date("2021-06-12"),
                 by = 'day')
more_days <- ymd(more_days)
currow <- nrow(df_wave2_bx)
more_times <- seq(from = currow + 1,
                 to = length(more_days) + currow)
Ys <- rep(NA, length(more_days))
more_data <- tibble(date = more_days, t_i = more_times, Y_i = Ys)

# Adding data to current data set
df_wave2_bx_ext <- rbind(df_wave2_bx, more_data)

# Saving population value
pop_bx <- 1472654

# Making new plot
death_wave2_bx_ext <- df_wave2_bx_ext %>%
  mutate(
    Y_i_hat_final = N(t_i, final_bx2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = (Y_i/pop_bx)*100000), size = 1) +
  geom_line(aes(y = (Y_i_hat_final/pop_bx)*100000, color = "red")) +
  labs(title = "Bronx", x = "Days (since Sep 15, 2020)",
       y = "Cumulative deaths per 100,000 capita") +
  theme(legend.position = "none") +
  coord_cartesian(xlim = c(0, 300), ylim = c(0, 45))
```

## BK extended curve

```
# Constructing a vector of future dates
more_days <- seq(from = as.Date("2020-12-12"), to = as.Date("2021-06-12"),
                 by = 'day')
more_days <- ymd(more_days)
currow <- nrow(df_wave2_bk)
more_times <- seq(from = currow + 1,
                 to = length(more_days) + currow)
Ys <- rep(NA, length(more_days))
more_data <- tibble(date = more_days, t_i = more_times, Y_i = Ys)

# Adding data to current data set
```

```

df_wave2_bk_ext <- rbind(df_wave2_bk, more_data)

# Saving population value
pop_bk <- 2736074

# Making new plot
death_wave2_bk_ext <- df_wave2_bk_ext %>%
  mutate(
    Y_i_hat_final = N(t_i, final_bk2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = (Y_i/pop_bk)*100000), size = 1) +
  geom_line(aes(y = (Y_i_hat_final/pop_bk)*100000, color = "red") +
  labs(title = "Brooklyn", x = "Days (since Sep 15, 2020)",
    y = "Cumulative deaths per 100,000 capita") +
  theme(legend.position = "none") +
  coord_cartesian(xlim = c(0, 300), ylim = c(0, 45))

```

## MN extended curve

```

# Constructing a vector of future dates
more_days <- seq(from = as.Date("2020-12-12"), to = as.Date("2021-06-12"),
  by = 'day')
more_days <- ymd(more_days)
currow <- nrow(df_wave2_mn)
more_times <- seq(from = currow + 1,
  to = length(more_days) + currow)
Ys <- rep(NA, length(more_days))
more_data <- tibble(date = more_days, t_i = more_times, Y_i = Ys)

# Adding data to current data set
df_wave2_mn_ext <- rbind(df_wave2_mn, more_data)

# Saving population value
pop_mn <- 1694251

# Making new plot
death_wave2_mn_ext <- df_wave2_mn_ext %>%
  mutate(
    Y_i_hat_final = N(t_i, final_mn2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = (Y_i/pop_mn)*100000), size = 1) +
  geom_line(aes(y = (Y_i_hat_final/pop_mn)*100000, color = "red") +
  labs(title = "Manhattan", x = "Days (since Sep 15, 2020)",
    y = "Cumulative deaths per 100,000 capita") +
  theme(legend.position = "none") +
  coord_cartesian(xlim = c(0, 300), ylim = c(0, 45))

```



## QN extended curve

```
# Constructing a vector of future dates
more_days <- seq(from = as.Date("2020-12-12"), to = as.Date("2021-06-12"),
  by = 'day')
more_days <- ymd(more_days)
currow <- nrow(df_wave2_qn)
more_times <- seq(from = currow + 1,
  to = length(more_days) + currow)
Ys <- rep(NA, length(more_days))
more_data <- tibble(date = more_days, t_i = more_times, Y_i = Ys)

# Adding data to current data set
df_wave2_qn_ext <- rbind(df_wave2_qn, more_data)

# Saving population value
pop_qn <- 2405464

# Making new plot
death_wave2_qn_ext <- df_wave2_qn_ext %>%
  mutate(
    Y_i_hat_final = N(t_i, final_qn2)
  ) %>%
  ggplot(aes(x = t_i)) +
  geom_point(aes(y = (Y_i/pop_qn)*100000), size = 1) +
  geom_line(aes(y = (Y_i_hat_final/pop_qn)*100000), color = "red") +
  labs(title = "Queens", x = "Days (since Sep 15, 2020)",
    y = "Cumulative deaths per 100,000 capita") +
  theme(legend.position = "none") +
  coord_cartesian(xlim = c(0, 300), ylim = c(0, 45))
```

## SI extended curve

```
# Constructing a vector of future dates
more_days <- seq(from = as.Date("2020-12-12"), to = as.Date("2021-06-12"),
  by = 'day')
more_days <- ymd(more_days)
currow <- nrow(df_wave2_si)
more_times <- seq(from = currow + 1,
  to = length(more_days) + currow)
Ys <- rep(NA, length(more_days))
more_data <- tibble(date = more_days, t_i = more_times, Y_i = Ys)

# Adding data to current data set
df_wave2_si_ext <- rbind(df_wave2_si, more_data)

# Saving population value
pop_si <- 495747

# Making new plot
death_wave2_si_ext <- df_wave2_si_ext %>%
```

```

mutate(
  Y_i_hat_final = N(t_i, final_si2)
) %>%
ggplot(aes(x = t_i)) +
  geom_point(aes(y = (Y_i/pop_si)*100000), size = 1) +
  geom_line(aes(y = (Y_i_hat_final/pop_si)*100000), color = "red") +
  labs(title = "Staten Island", x = "Days (since Sep 15, 2020)",
        y = "Cumulative deaths per 100,000 capita") +
  theme(legend.position = "none") +
  coord_cartesian(xlim = c(0, 300), ylim = c(0, 45))

```

## Showing all extended curves

```

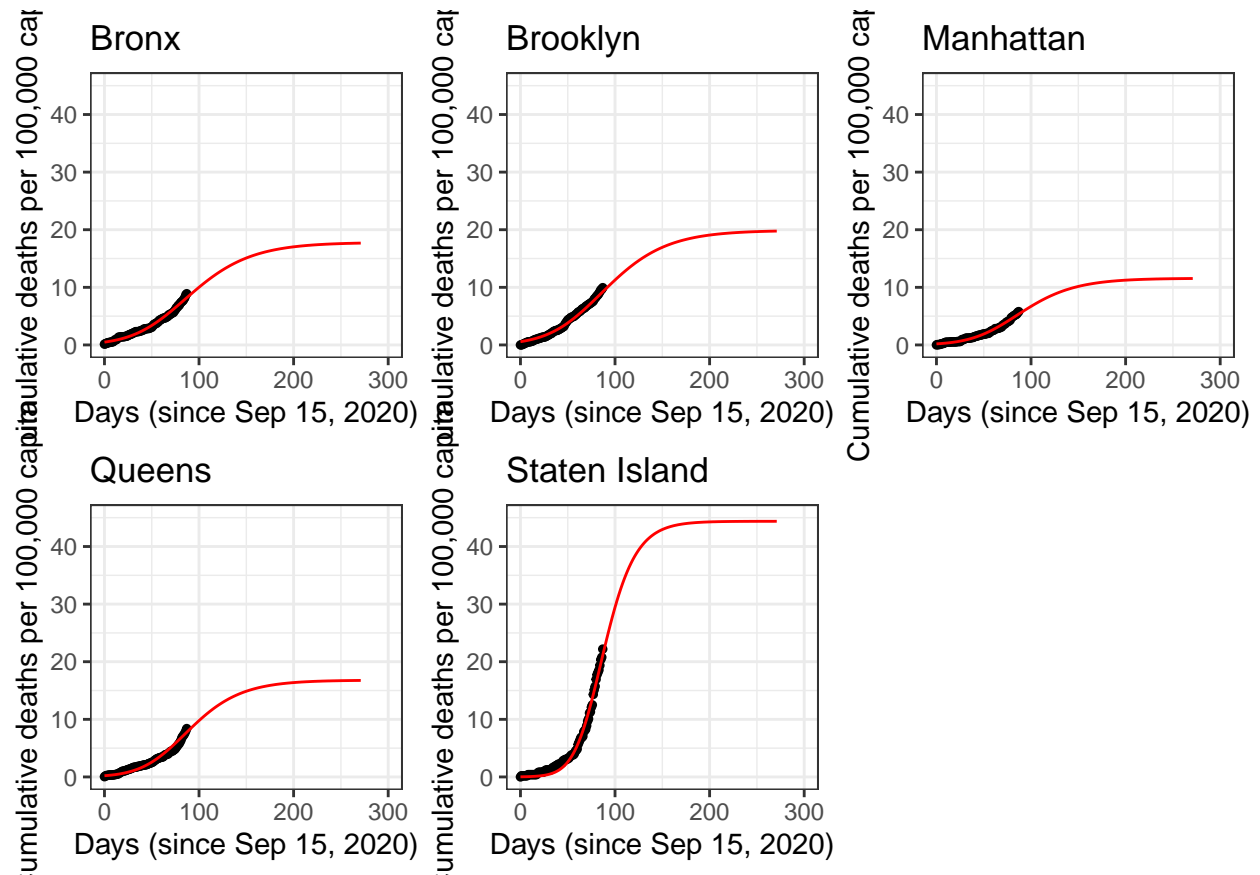
ggarrange(death_wave2_bx_ext,
           death_wave2_bk_ext, death_wave2_mn_ext,
           death_wave2_qn_ext, death_wave2_si_ext,
           nrow = 2, ncol = 3, common.legend = TRUE, legend = "bottom")

```

```

## Warning: Removed 183 rows containing missing values ('geom_point()').
## Removed 183 rows containing missing values ('geom_point()').
## Removed 183 rows containing missing values ('geom_point()').
## Removed 183 rows containing missing values ('geom_point()').
## Removed 183 rows containing missing values ('geom_point()').
## Removed 183 rows containing missing values ('geom_point()').
## Removed 183 rows containing missing values ('geom_point()').
## Removed 183 rows containing missing values ('geom_point()').
## Removed 183 rows containing missing values ('geom_point()').

```



```
death_wave2_nyc_ext
```

```
## Warning: Removed 183 rows containing missing values (‘geom_point()’).
```

# New York City

