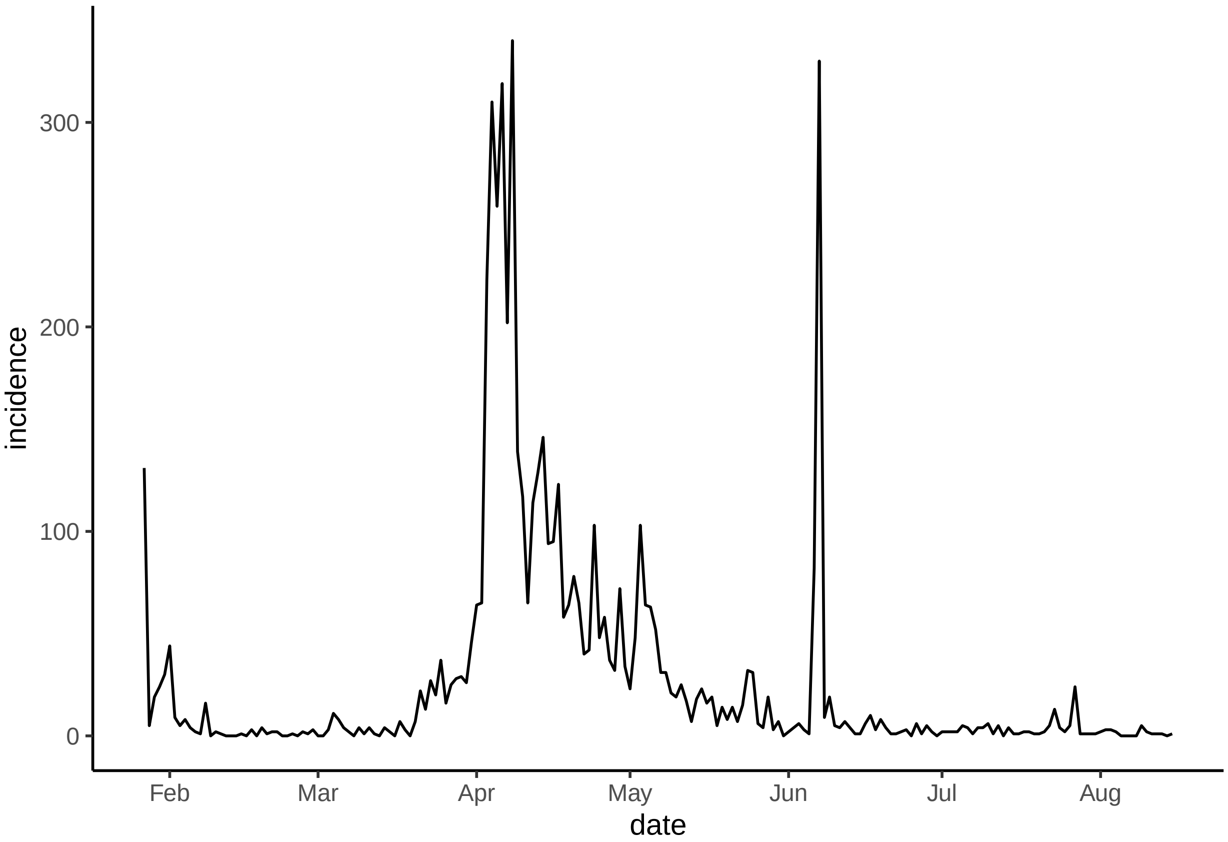
Project

Annie Cohen and Ryan McIntosh

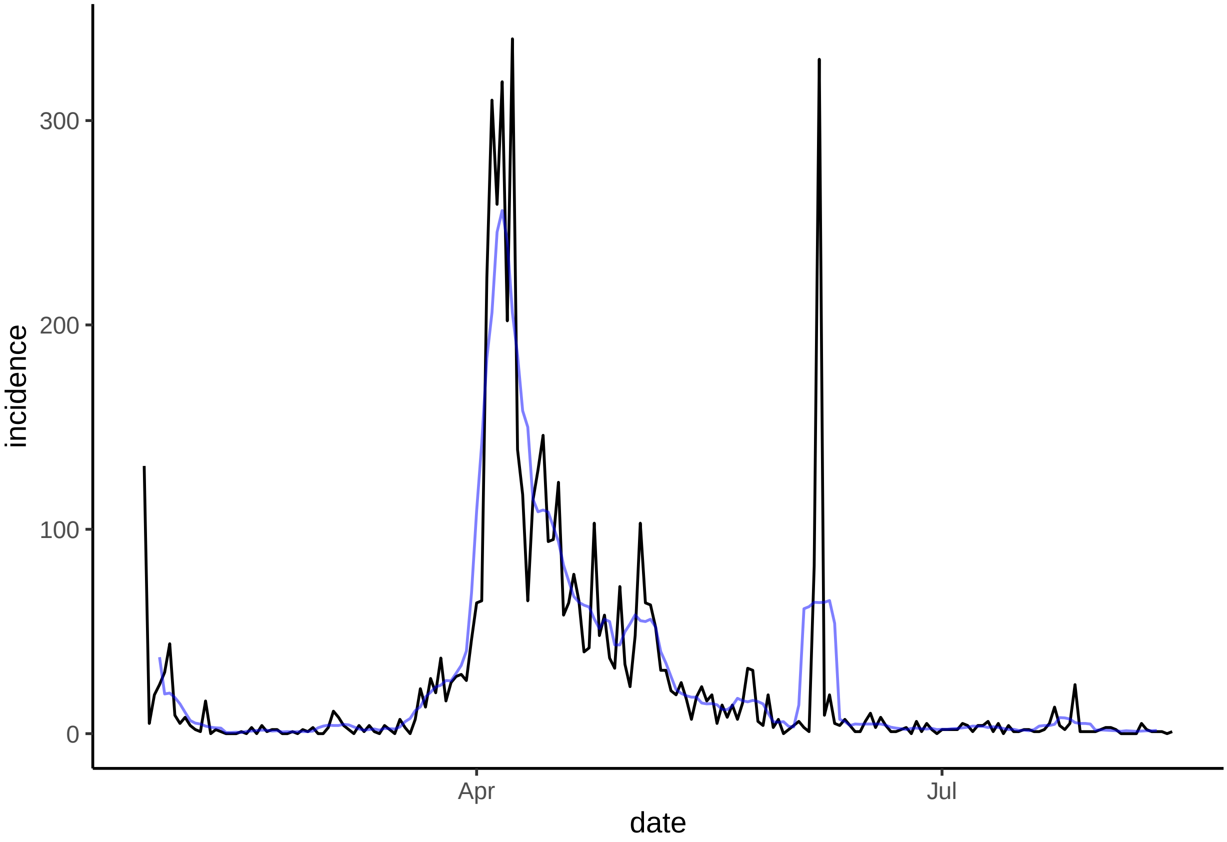
4/5/2022

```r
obsdata <- readRDS("incidence_hashtags.rds")

colnames(obsdata) <- c("date","incidence")

obsdata <- obsdata %>%
  filter(date > obsdata$date[18])

obsdata %>%
  ggplot(aes(date,incidence)) +
  geom_line() +
  theme_classic() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



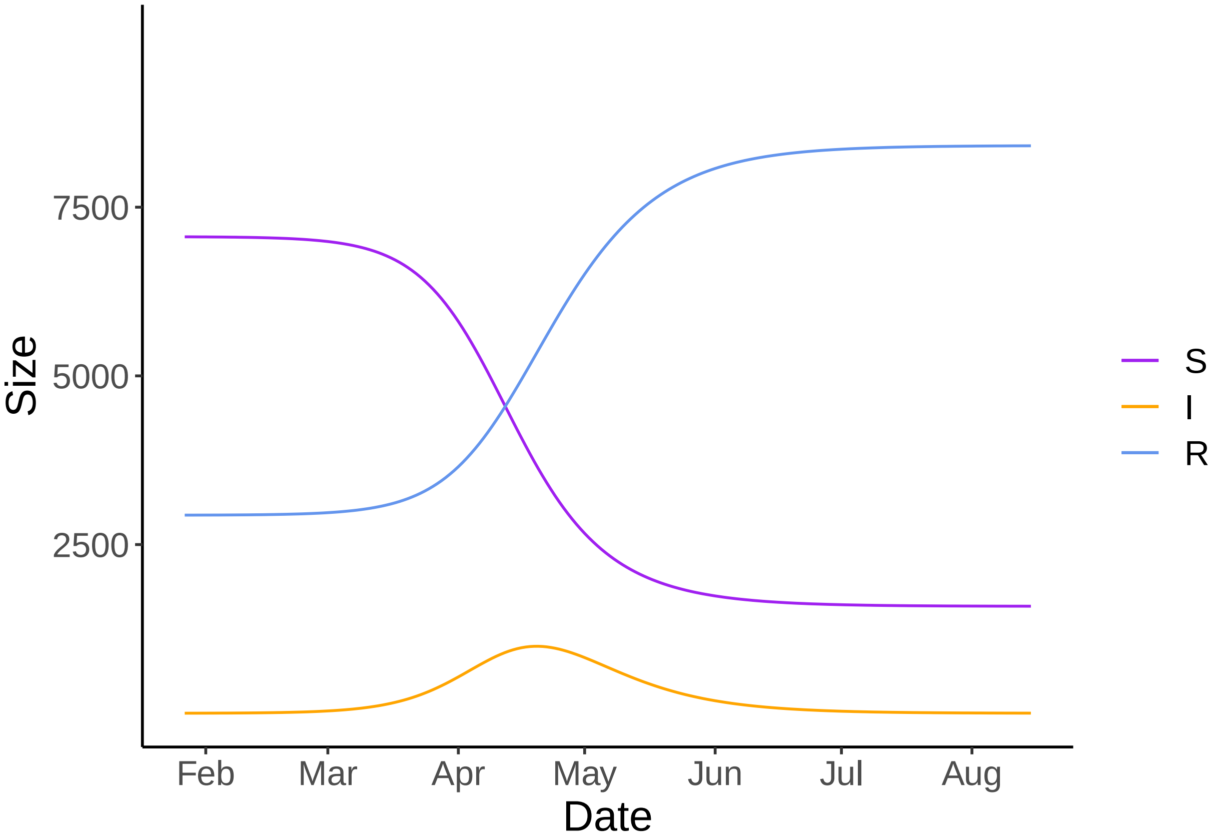
```
obsdata %>%
  mutate(runavg = zoo::rollmean(incidence, k = 7, fill = NA)) %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = incidence)) +
  geom_line(aes(y = runavg), col = "blue", alpha = 0.5) +
  theme_classic()
```

```
## Warning: Removed 6 row(s) containing missing values (geom_path).
```

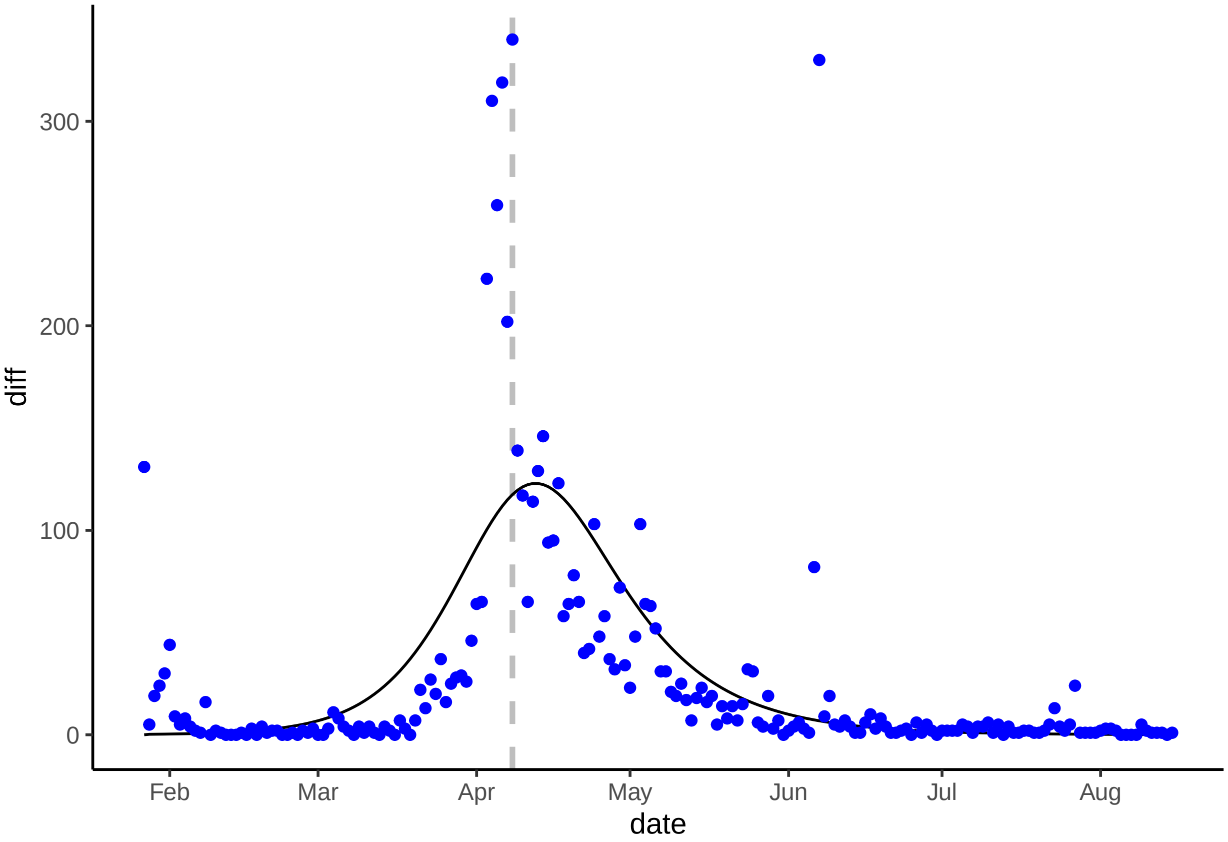


```
SIR_basic <- function(t,x,param){

  beta = param[1]
  delta = param[2]
  alpha = param[3]
  N = param[4]

  if (t > delta){
    gamma = param[5]
    zeta = param[6]
  }
  else{
    gamma = 0
    zeta = 0
  }
  S = x[1]
  inc = x[2]
  I = x[3]
  R = x[4]

  dxdt = numeric(length(x))
  dxdt[1] = -beta*S*I/N
  dxdt[2] = beta*S*I/N
  dxdt[3] = beta*S*I/N-alpha*I
  dxdt[4] = alpha*I

  return(list(dxdt))
}
```

```
n_days = length(obsdata$date)
timerange = seq(1,n_days)

beta = 0.3
delta = 14
alpha = 0.11
N = 9999

gamma = 0
zeta = 0

param = c(beta, delta, alpha, N, gamma, zeta)

x0 <- rep(0,4)
x0[1] = 7060.94
x0[2] = 0
x0[3] = 1.1
x0[4] = 2936.53

out = ode(y=x0,times=timerange, func=SIR_basic, parms = param, method = 'ode45')

out_df <- data.frame(out) %>%
  mutate(diff = c(0,diff(out[,3])))

names(out_df) = c('Time', 'S', 'inc', 'I', 'R','diff')
out_df$date = obsdata$date
order = c('S', 'I', 'R')

ggplot(data = out_df, aes(x = date)) +
  theme_classic() +
  geom_line(aes(y = S, col = 'S')) +
  geom_line(aes(y = I, col = 'I')) +
  geom_line(aes(y = R, col = 'R')) +
  scale_color_manual(breaks = order, values = c("purple","orange","cornflowerblue")) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  scale_y_continuous(limits = c(0, 10000), breaks = c(2500, 5000, 7500)) +
  labs(x = 'Date', y = 'Size', col = "") +
  theme(text = element_text(size = 16))
```



```
ggsave("fig7.png", width = 7, height = 7)


out_df %>%
  ggplot(aes(x = date, y = diff)) +
  geom_vline(xintercept = lubridate::as_date("2020-04-08"), lty = 2, col = "gray", size = 1) +
  geom_line() +
  geom_point(aes(y = obsdata$incidence), col = "blue") +
  theme_classic() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



```
max(out_df[,4])
```

```
## [1] 992.6237
```

```
# Calculating cumulative error for SIR Basic
n_days = length(obsdata$date)
timerange = seq(1,n_days)

beta = 0.3
delta = 14
alpha = 0.11
N = 9999
gamma = 0
zeta = 0
param = c(beta, delta, alpha, N, gamma, zeta)

x0 <- rep(0,4)
x0[1] = 7060.94
x0[2] = 0
x0[3] = 1.1
x0[4] = 2936.53

solved = as.data.frame(ode(y=x0,times=timerange, func=SIR_basic, parms = param, method = 'ode45'))
names(solved) = c('Time', 'S', 'inc', 'I', 'R')

obsdata$cumulative = cumsum(obsdata$incidence)
```

```r
SIR2norm = sqrt(sum((obsdata$cumulative - solved$inc)^2)) / sqrt(sum(obsdata$cumulative^2))

print(SIR2norm)
```

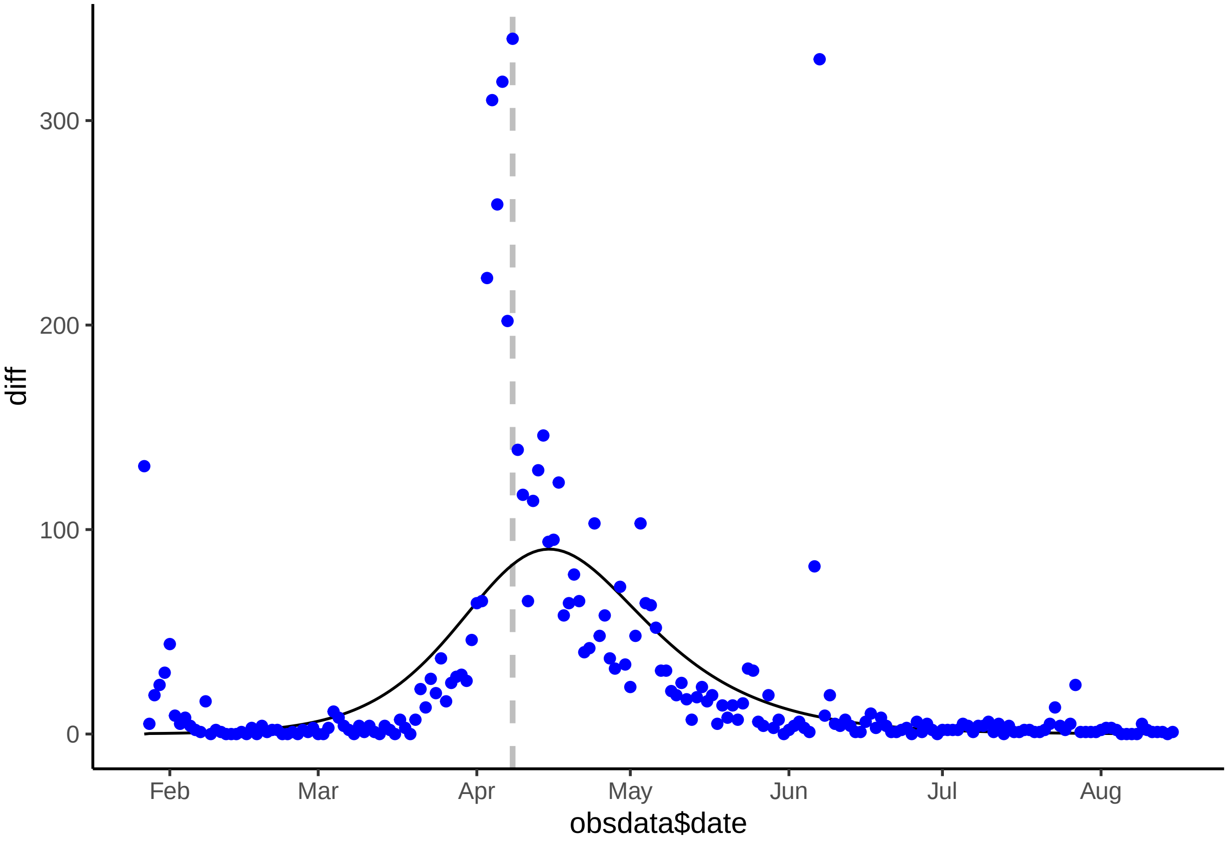
```
## [1] 0.05660188
```

```r
# That almost matches the paper- 0.0566 for ours, compared to 0.0561 for theirs
# Only if I set the initial incidence to 0, though. Setting x[2] to 131 gives a higher error
```

```r
SIR_factcheck <- function(t,x,param){

  beta = param[1]
  delta = param[2]
  alpha = param[3]
  N = param[4]

  if (t > delta){
    gamma = param[5]
  }
  else{
    gamma = 0
  }

  S = x[1]
  inc = x[2]
  I = x[3]
  R = x[4]

  dxdt = numeric(length(x))
  dxdt[1] = -beta*S*I/N-gamma*S
  dxdt[2] = beta*S*I/N
  dxdt[3] = beta*S*I/N-alpha*I
  dxdt[4] = alpha*I

  return(list(dxdt))
}
```

```r
n_days = length(obsdata$date)
timerange = seq(1,n_days)

beta = 0.3
delta = 14
alpha = 0.11
N = 9999

gamma = 0.01/7
zeta = 0.06/7

param = c(beta, delta, alpha, N, gamma, zeta)

x0 <- rep(0,4)
x0[1] = 7060.94
```

```
x0[2] = 1.1
x0[3] = 1.1
x0[4] = 2936.53

out = ode(y=x0,times=timerange, func=SIR_factcheck, parms = param, method = 'ode45')

out_df <- data.frame(out) %>%
  mutate(diff = c(0,diff(out[,3])))

out_df %>%
  ggplot(aes(x = obsdata$date, y = diff)) +
  geom_vline(xintercept = lubridate::as_date("2020-04-08"), lty = 2, col = "gray", size = 1) +
  geom_line() +
  geom_point(aes(y = obsdata$incidence), col = "blue") +
  theme_classic() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```

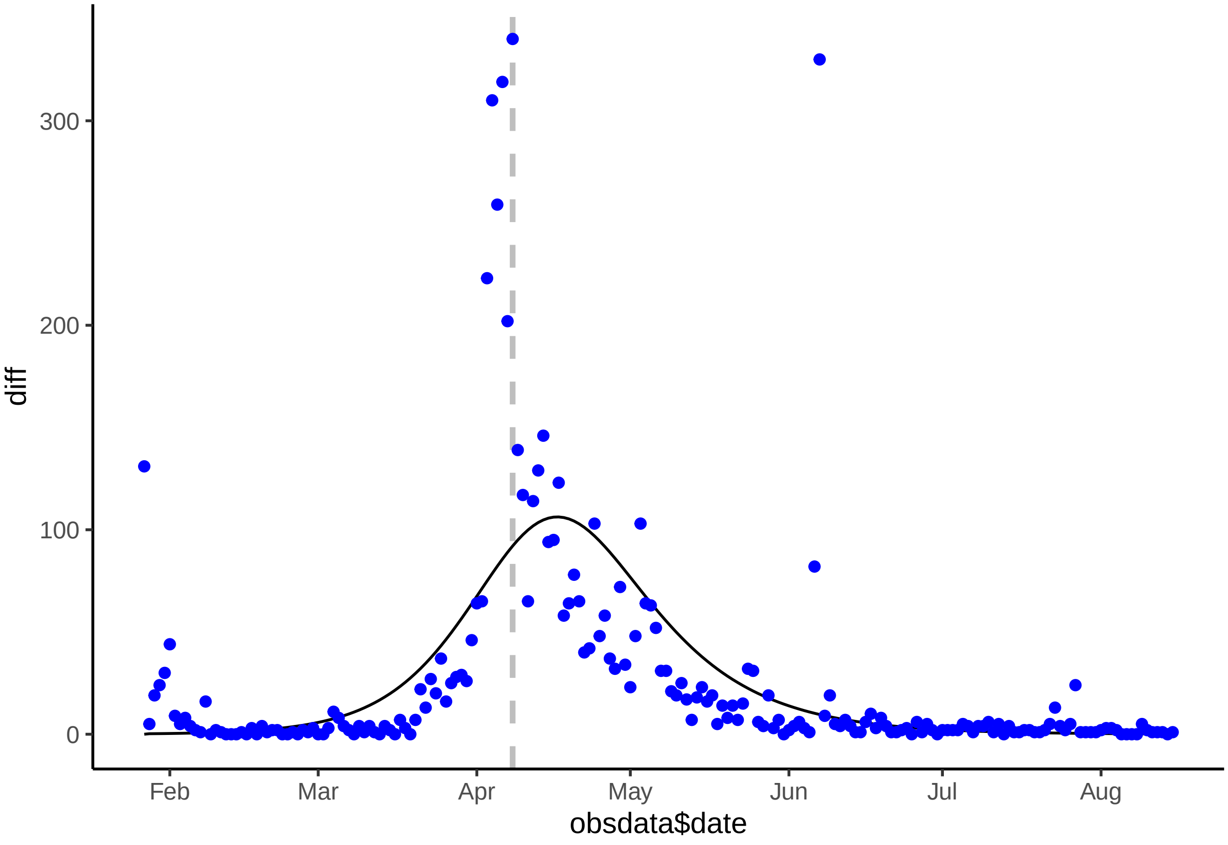


```
max(out_df[,4])
```

```
## [1] 745.6516
```

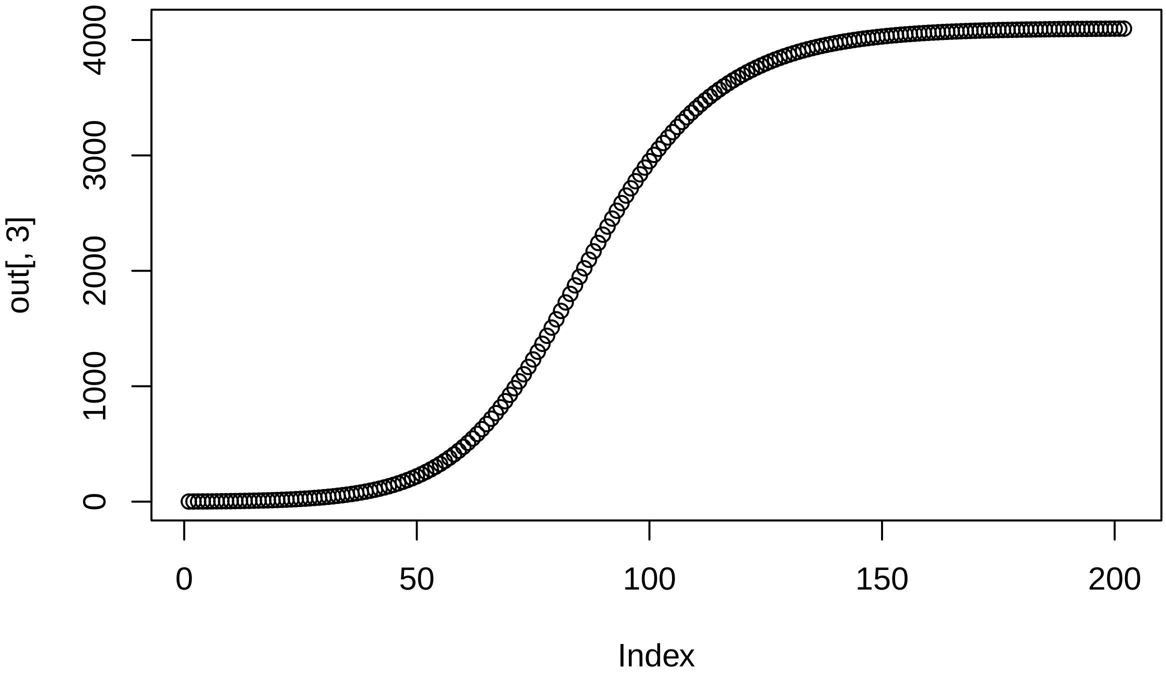
```
SIR_deletion <- function(t,x,param){

  beta = param[1]
  delta = param[2]
  alpha = param[3]
  N = param[4]

  if (t > delta){
    zeta = param[6]
  }
  else{
    zeta = 0
  }

  S = x[1]
  inc = x[2]
  I = x[3]
  R = x[4]

  dxdt = numeric(length(x))
  dxdt[1] = -beta*S*I/N
  dxdt[2] = beta*S*I/N
  dxdt[3] = beta*S*I/N-(alpha+zeta)*I
  dxdt[4] = (alpha+zeta)*I

  return(list(dxdt))
}
```

```
n_days = length(obsdata$date)
timerange = seq(1,n_days)

beta = 0.3
delta = 14
alpha = 0.11
N = 9999

gamma = 0.01/7
zeta = 0.06/7

param = c(beta, delta, alpha, N, gamma, zeta)

x0 <- rep(0,4)
x0[1] = 7060.94
x0[2] = 131
x0[3] = 1.1
x0[4] = 2936.53

out = ode(y=x0,times=timerange, func=SIR_deletion, parms = param, method = 'ode45')

out_df <- data.frame(out) %>%
  mutate(diff = c(0,diff(out[,3])))

out_df %>%
  ggplot(aes(x = obsdata$date, y = diff)) +
  geom_vline(xintercept = lubridate::as_date("2020-04-08"), lty = 2, col = "gray", size = 1) +
  geom_line() +
  geom_point(aes(y = obsdata$incidence), col = "blue") +
  theme_classic() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



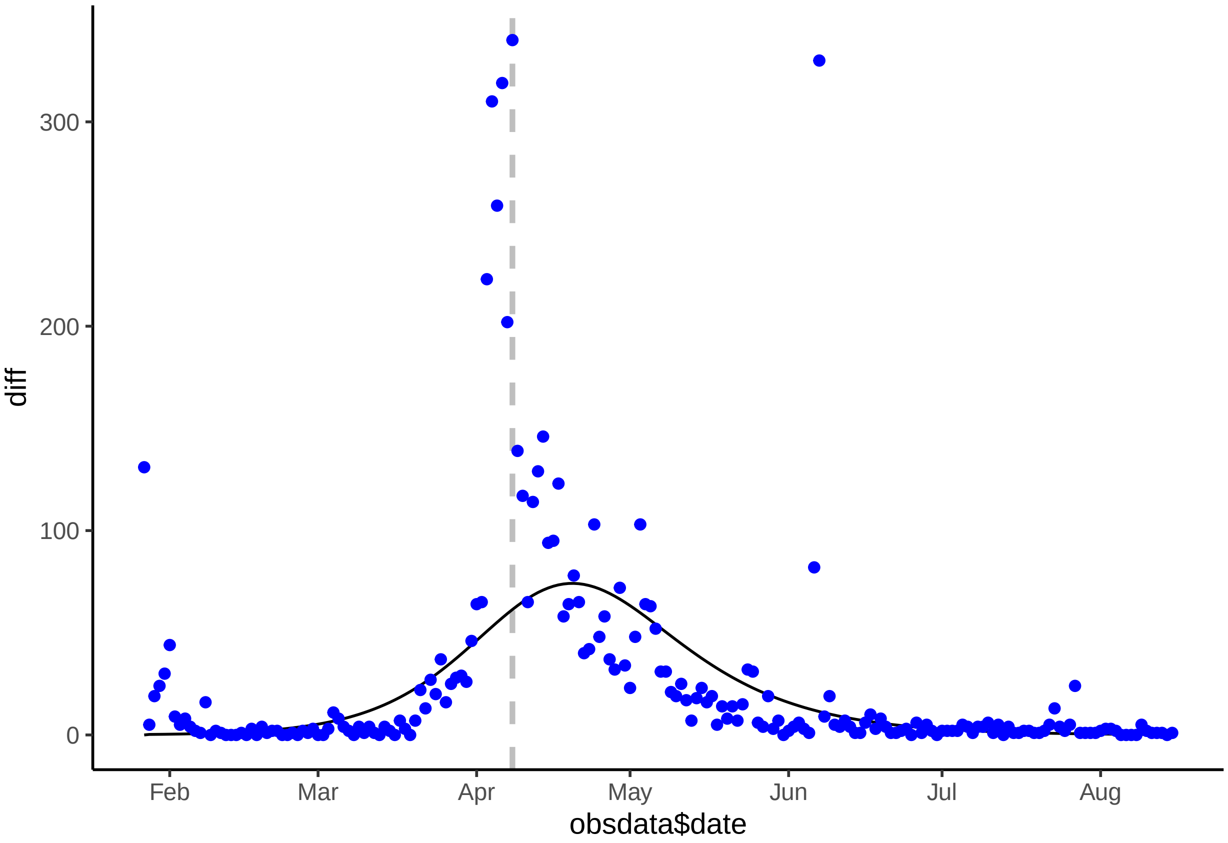
```
SIR_mixed <- function(t,x,param){

  beta = param[1]
  delta = param[2]
  alpha = param[3]
  N = param[4]

  if (t > delta){
    gamma = param[5]
    zeta = param[6]
  }
  else{
    gamma = 0
    zeta = 0
  }

  S = x[1]
  inc = x[2]
  I = x[3]
  R = x[4]

  dxdt = numeric(length(x))
  dxdt[1] = -beta*S*I/N-gamma*S
  dxdt[2] = beta*S*I/N
  dxdt[3] = beta*S*I/N-(alpha+zeta)*I
  dxdt[4] = (alpha+zeta)*I+gamma*S

  return(list(dxdt))
}
```

```
n_days = length(obsdata$date)
timerange = seq(1,n_days)

beta = 0.3
delta = 14
alpha = 0.11
N = 9999

gamma = 0.01/7
zeta = 0.06/7

param = c(beta, delta, alpha, N, gamma, zeta)

x0 <- rep(0,4)
x0[1] = 7060.94
x0[2] = 1.1
x0[3] = 1.1
x0[4] = 2936.53

out = ode(y=x0,times=timerange, func=SIR_mixed, parms = param, method = 'ode45')
plot(out[,3])
```



```
out_df <- data.frame(out) %>%
  mutate(diff = c(0,diff(out[,3])))
```

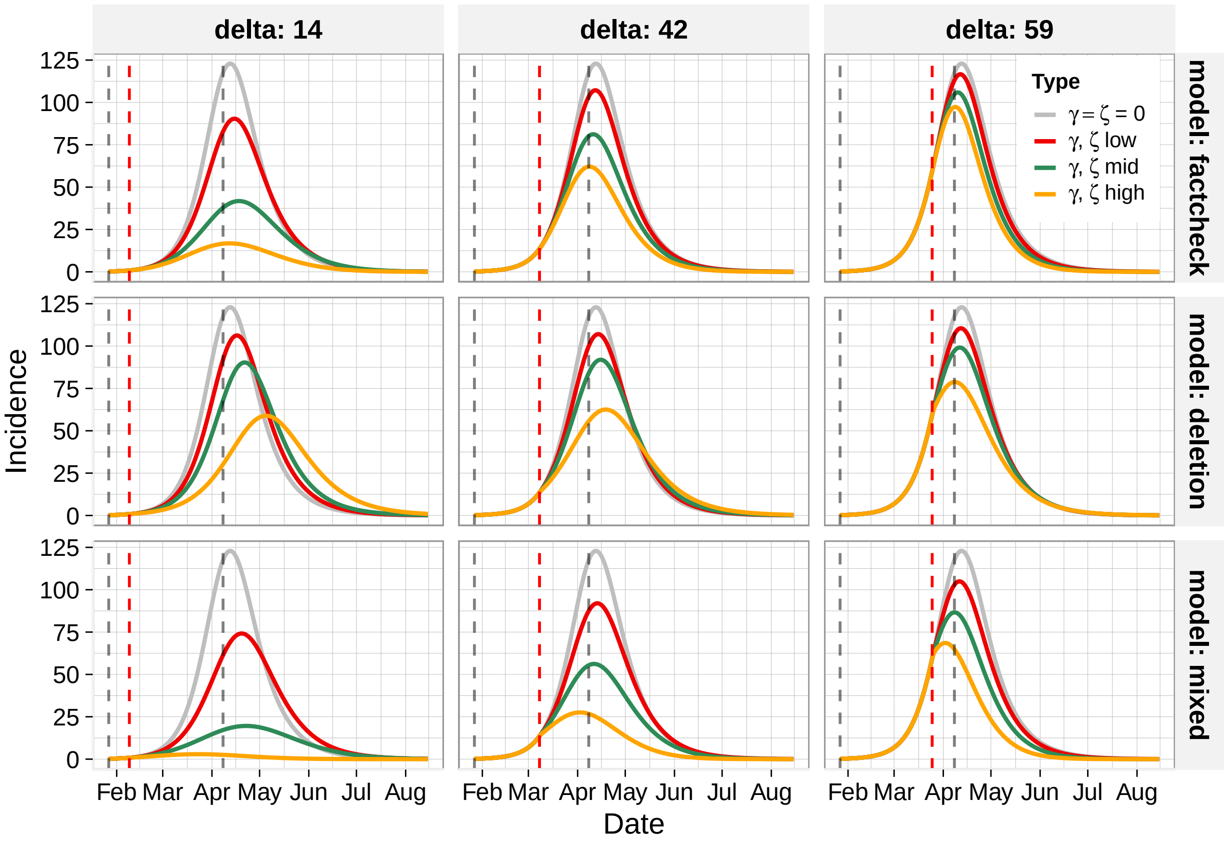
```
out_df %>%
  ggplot(aes(x = obsdata$date, y = diff)) +
  geom_vline(xintercept = lubridate::as_date("2020-04-08"), lty = 2, col = "gray", size = 1) +
  geom_line() +
  geom_point(aes(y = obsdata$incidence), col = "blue") +
  theme_classic() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



```
n_days = length(obsdata$date)
timerange = seq(1,n_days)

incidence_out <- data.frame(delta = c(rep(14,16),rep(42,16),rep(59,16)),
           gamma = rep(c(rep(0,4),rep(.01/7,4),rep(.03/7,4),rep(.05/7,4)),3),
           zeta = c(rep(c(0,0.06/7,0.12/7,0.25/7))), incidence = rep(NA,48))

for (i in 1:48){
  param = c(0.3, incidence_out$delta[i], 0.11, 9999, incidence_out$gamma[i], incidence_out$zeta[i])

  x0 <- rep(0,4)
  x0[1] = 7060.94
  x0[2] = 131
  x0[3] = 1.1
  x0[4] = 2936.53

  out = ode(y=x0,times=timerange, func=SIR_mixed, parms = param, method = 'ode45')
  out_df <- data.frame(out) %>%
    mutate(diff = c(0,diff(out[,3])))

  incidence_out$incidence[i] <- list(out_df$diff)
}

vars <- incidence_out[,1:3]
lst <- unlist(lapply(incidence_out$incidence, unlist))

df <- vars %>%
  slice(rep(1:48,each = 202)) %>%
  mutate(incidence = lst,
         time = rep(timerange, 48))

basic<-df[which(df$zeta==0&df$gamma==0),] %>%
  select(incidence) %>%
  unique()

new_df <- df[-which(df$zeta==0&df$gamma==0),]

new_df <- new_df %>%
  mutate(basic = rep(basic$incidence,45),
         model = factor(ifelse(zeta == 0, "factcheck",
                               ifelse(gamma == 0, "deletion",
                               ifelse(gamma == unique(new_df$gamma)[2] & zeta == unique(new_df$zeta)[1]
           gamma == unique(new_df$gamma)[3] & zeta == unique(new_df$zeta)[2] |
           gamma == unique(new_df$gamma)[4] & zeta == unique(new_df$zeta)[3],"mixed",NA))),levels = c("
         type = ifelse((gamma == unique(new_df$gamma)[2] & zeta == 0)|(zeta == unique(new_df$zeta)[1] &
                              ifelse((gamma == unique(new_df$gamma)[3] & zeta == 0)|(zeta == unique(new_
                                     ifelse((gamma == unique(new_df$gamma)[4] & zeta == 0)|(zeta == uni
         date = rep(obsdata$date,45)) %>%
  filter(!is.na(model))

new_df$date[delta]
```

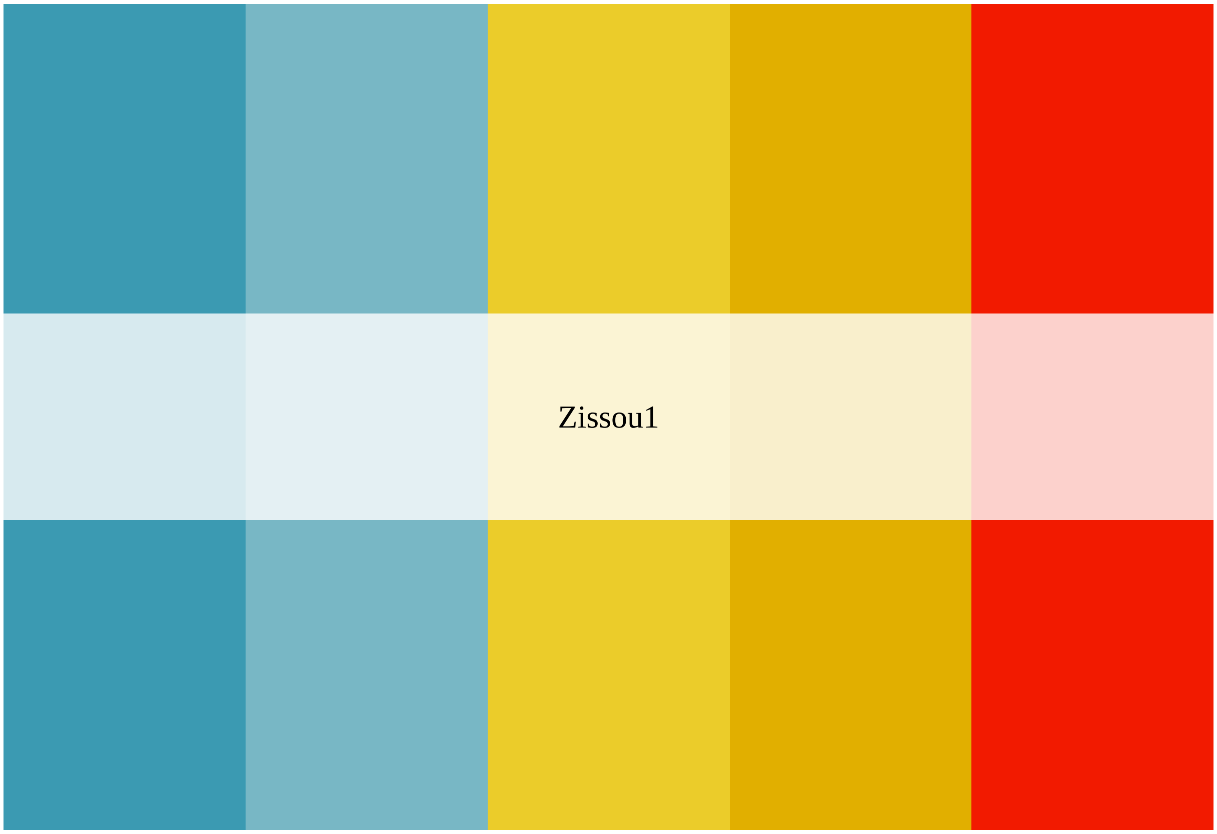
```
## [1] "2020-02-09"
```

```
new_df %>%
  ggplot(aes(x = date,y = incidence)) +
  geom_line(aes(y = basic, col = "hi"), size = 0.75) +
  geom_path(aes(col = type), size = 0.75) +
  geom_vline(aes(xintercept = date[delta]), col = "red", lty = 2, size = 0.5)+
  facet_grid(model~delta, labeller = label_both) +
  geom_vline(xintercept = as.Date("2020-04-08"), lty = 2, alpha = 0.5, size = 0.5)+
  geom_vline(xintercept = as.Date("2020-01-27"), lty = 2, alpha = 0.5, size = 0.5)+
  theme_linedraw() +
  theme(axis.line = element_line(color = "grey95"), strip.background = element_rect(fill = "grey95", co
scale_color_manual(values = c("gray","red2","seagreen","orange"), labels = c(expression(gamma==zeta~"= 
  labs(x = "Date",y = "Incidence", col = "Type")+
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



```
ggsave("fig2.png", width = 7, height = 7)
```

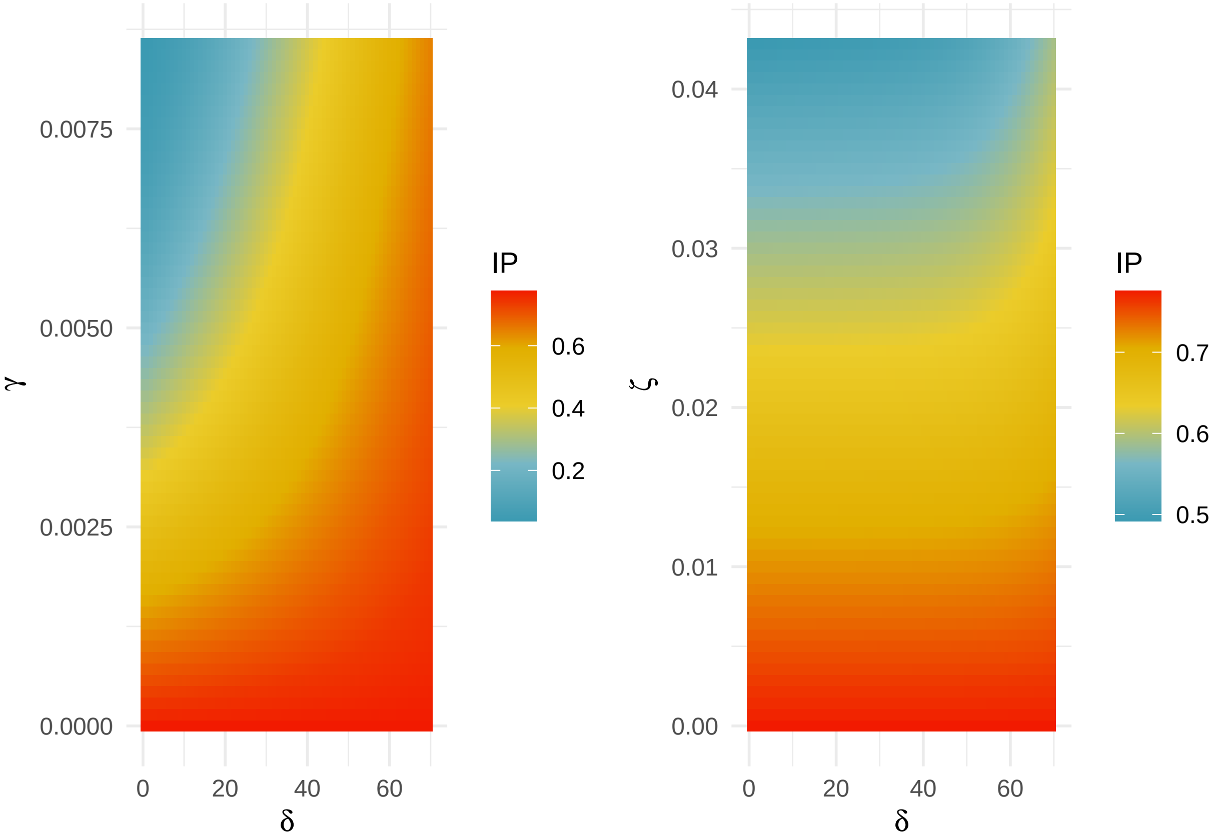
```
factcheck <- data.frame("delta"=c(rep(0:70,each=61)),"gamma" = c(rep(seq(0,0.06/7,by=0.001/7),71)), "IP

for (i in 1:4331){
  param = c(0.3, factcheck$delta[i], 0.11, 9999, factcheck$gamma[i], 0)

  x0 <- rep(0,4)
  x0[1] = 7060.94
  x0[2] = 1.1
  x0[3] = 1.1
  x0[4] = 2936.53

  out = ode(y=x0,times=timerange, func=SIR_factcheck, parms = param, method = 'ode45')

  factcheck$IP[i] <- out[202,3]/7060.94
}

wes_palette("Zissou1")
```



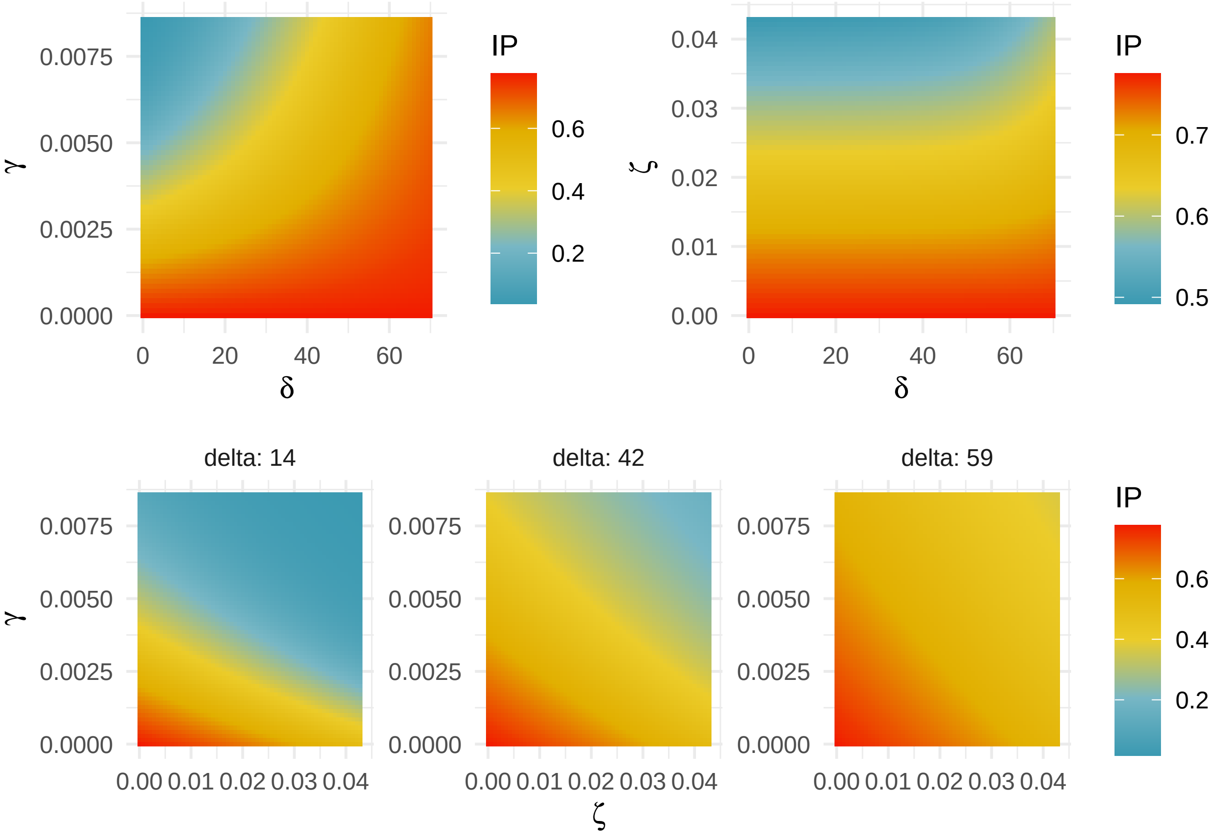
```
pal <- wes_palette("Zissou1", 21, type = "continuous")

fc <- factcheck %>%
  ggplot(aes(x = delta, y = gamma)) +
  geom_tile(aes(fill = IP)) +
  scale_fill_gradientn(colors = pal) +
  labs(x = expression(delta), y = expression(gamma)) +
  theme_minimal()
```

```
tweet_delete <- data.frame("delta"=c(rep(0:70,each=61)),"zeta" = c(rep(seq(0,0.3/7,by=0.005/7),71)), "I

for (i in 1:4331){
  param = c(0.3, tweet_delete$delta[i], 0.11, 9999, 0, tweet_delete$zeta[i])

  x0 <- rep(0,4)
  x0[1] = 7060.94
  x0[2] = 1.1
  x0[3] = 1.1
  x0[4] = 2936.53

  out = ode(y=x0,times=timerange, func=SIR_deletion, parms = param, method = 'ode45')

  tweet_delete$IP[i] <- out[202,3]/7060.94
}

pal <- wes_palette("Zissou1", 21, type = "continuous")
```

```
td <- tweet_delete %>%
  ggplot(aes(x = delta, y = zeta)) +
  geom_tile(aes(fill = IP)) +
  scale_fill_gradientn(colors = pal) +
  labs(x = expression(delta), y = expression(zeta)) +
  theme_minimal()

gridExtra::grid.arrange(fc, td, nrow = 1)
```



```
mixed <- data.frame("delta" = rep(c(14,42,59),each = 3721), "gamma" = rep(c(rep(seq(0,0.06/7,by=0.001/7

for (i in 1:11163){
  param = c(0.3, mixed$delta[i], 0.11, 9999, mixed$gamma[i], mixed$zeta[i])

  x0 <- rep(0,4)
  x0[1] = 7060.94
  x0[2] = 1.1
  x0[3] = 1.1
  x0[4] = 2936.53

  out = ode(y=x0,times=timerange, func=SIR_mixed, parms = param, method = 'ode45')

  mixed$IP[i] <- out[202,3]/7060.94
}

m <- mixed %>%
  ggplot(aes(x = zeta, y = gamma)) +
  geom_tile(aes(fill = IP)) +
  scale_fill_gradientn(colors = pal) +
  facet_wrap(~delta, scales = "free", labeller = label_both) +
  labs(y = expression(gamma), x = expression(zeta)) +
  theme_minimal()

gridExtra::grid.arrange(fc, td, m, nrow = 2, layout_matrix = rbind(c(1,2),c(3,3)))
```



```
g<-gridExtra::arrangeGrob(fc, td, m, nrow = 2, layout_matrix = rbind(c(1,2),c(3,3)))

ggsave("fig3.png", g, width = 12, height = 7)
```

```
SEIR_mixed <- function(t,x,param){
  beta = param[1]
  delta = param[2]
  alpha = param[3]
  N = 9999
  sigma = param[7]

  if (t > delta){
    gamma = param[5]
    zeta = param[6]
  }
  else{
    gamma = 0
    zeta = 0
  }

  S = x[1]
  E = x[2]
  inc = x[3]
  I = x[4]
  R = x[5]

  dxdt = numeric(length(x))
  dxdt[1] = -beta*S*I/N
  dxdt[2] = beta*S*I/N-(sigma+gamma)*E
  dxdt[3] = sigma*E
  dxdt[4] = sigma*E-(alpha+zeta)*I
  dxdt[5] = (alpha+zeta)*I+gamma*E

  return(list(dxdt))
}
```

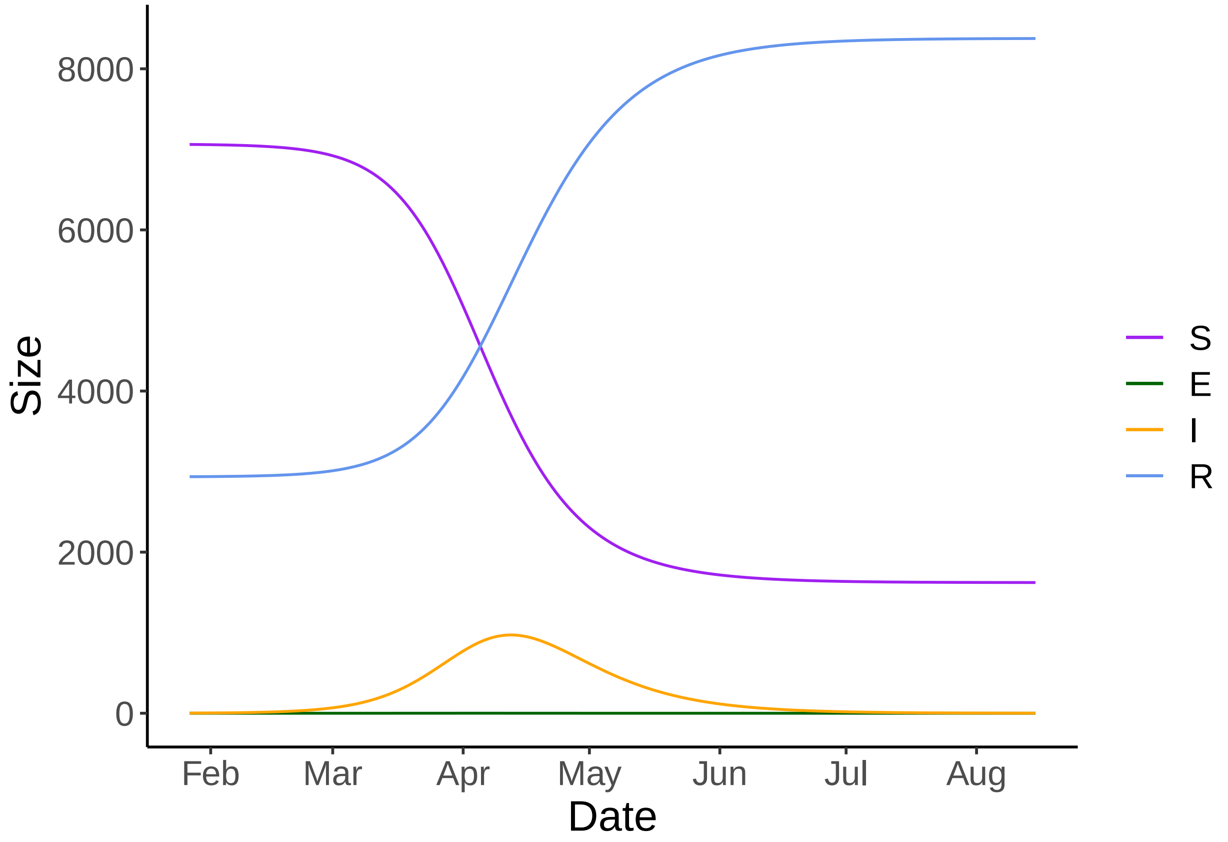
```
# Adding in the relative error 2-norm function to optimize sigma while keeping beta and alpha the same
re2norm = function(sigma){
  parms = c(0.3, 42, 0.11, 9999, 0, 0, sigma)
  x0 = c(7060.94, 0, 0, 1.1, 2936.53)

  solved = as.data.frame(ode(y = x0, times = timerange, func = SEIR_mixed,
                             parms = parms, method = 'ode45'))
  names(solved) = c('Time', 'S', 'E', 'inc', 'I', 'R')

  out = sqrt(sum((obsdata$cumulative - solved$inc)^2)) / sqrt(sum(obsdata$cumulative^2))

  return(out)
}

fulloptim = function(parms){
  beta = parms[1]
  delta = 42
  alpha = parms[2]
  N = 9999
  gamma = 0
  zeta = 0
  sigma = parms[3]

  params = c(beta, delta, alpha, N, gamma, zeta, sigma)
  x0 = c(7060.94, 0, 0, 1.1, 2936.53)
  n_days = length(obsdata$date)
  timerange = seq(1,n_days)

  solved = as.data.frame(ode(x0, timerange, SEIR_mixed, params, method = 'ode45'))

  names(solved) = c('Time', 'S', 'E', 'inc', 'I', 'R')

  out = sqrt(sum((obsdata$cumulative - solved$inc)^2)) / sqrt(sum(obsdata$cumulative^2))
  return(out)
}

# Running the optimizer
# Just for sigma
range = c(0, 100)
opt2norm = optimize(f = re2norm, interval = range)
opt_sigma = as.numeric(opt2norm[1])

# Calculating cumulative error with optimized sigma, same beta and alpha as SIR
print(re2norm(opt_sigma))
```

```
## [1] 0.05670288
```

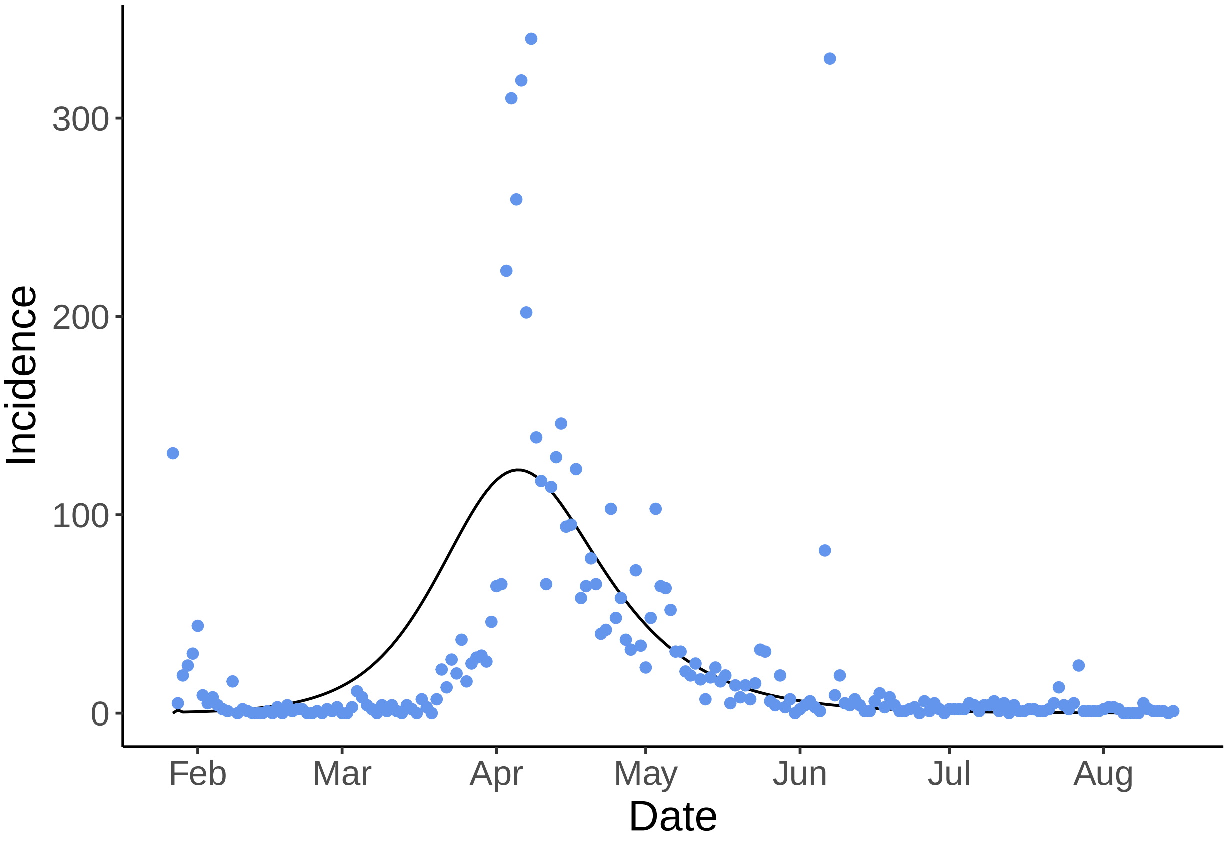
```
# For all parameters
guesses = c(0.3, 0.11, 0.2)
opt2norm_all = optim(guesses, fulloptim)
optimized3 = opt2norm_all$par[1:3]

# Checking cumulative error with optimized beta, alpha, and sigma
print(fulloptim(optimized3))
```

```
## [1] 0.05625518
```

```
param = c(optimized3[1], 42, optimized3[2], 9999, 0, 0, optimized3[3])

x0 <- rep(0,5)
x0[1] = 7060.94
x0[2] = 1.1
x0[3] = 1.1
x0[4] = 1.1
x0[5] = 2936.53

out <- ode(y=x0,times=timerange, func=SEIR_mixed, parms = param, method = 'ode45')
leg_ord <- c("S","E","I","R")

data.frame(out) %>%
  mutate(date = obsdata$date) %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = X2, col = "E")) +
  geom_line(aes(y = X4, col = "I")) +
  geom_line(aes(y = X1, col = "S")) +
  geom_line(aes(y = X5, col = "R")) +
  scale_color_manual(breaks = leg_ord, values = c("purple","darkgreen","orange","cornflowerblue")) +
  scale_x_date(breaks = "1 month", date_labels = "%b") +
  labs(y = "Size",x = "Date", col = "") +
  theme_classic() +
  theme(text = element_text(size = 16))
```

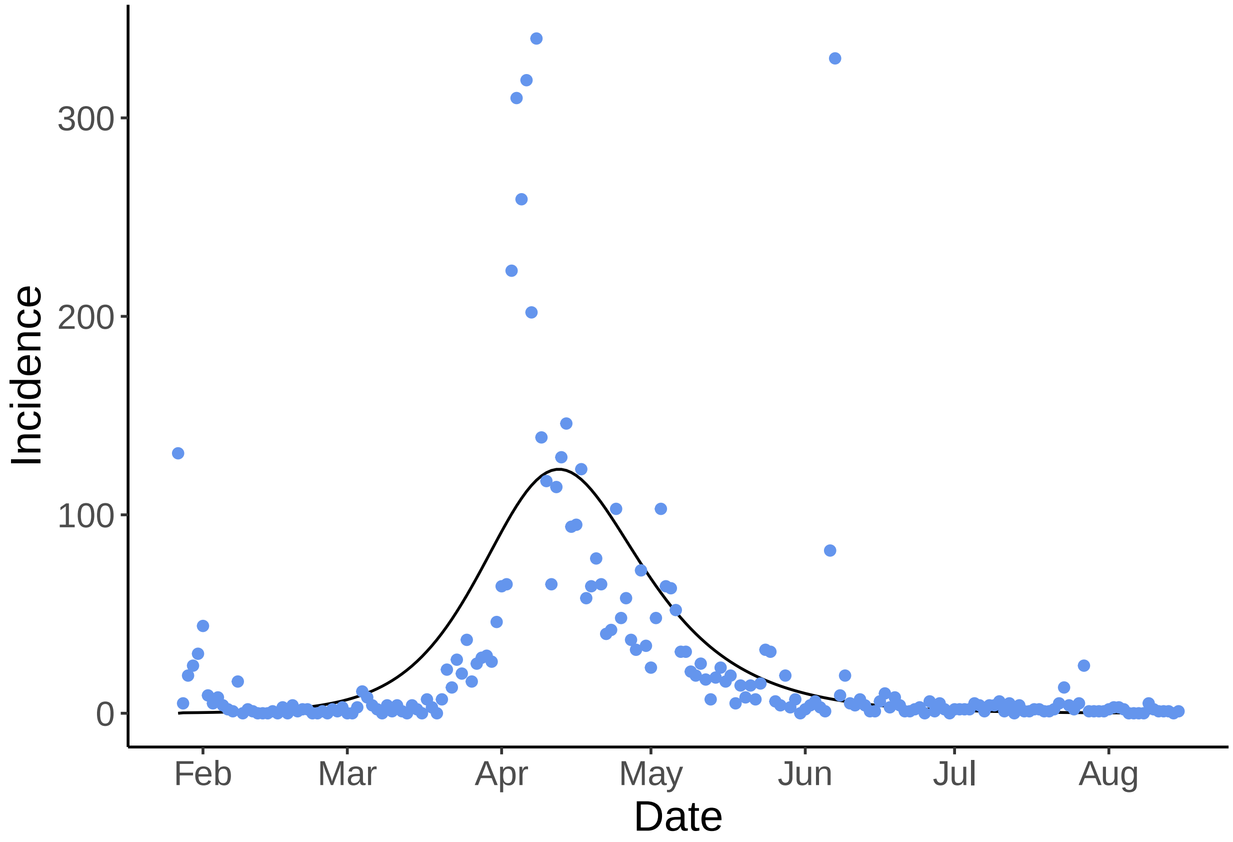


```
ggsave("fig5.png", width = 7, height = 7)

data.frame(out) %>%
  mutate(date = obsdata$date) %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = c(0,diff(X3)))) +
  geom_point(aes(y = obsdata$incidence), col = "cornflowerblue") +
  scale_x_date(breaks = "1 month", date_labels = "%b") +
  labs(y = "Incidence", x = "Date") +
  theme_classic() +
  theme(text = element_text(size = 16))
```



```
ggsave("fig4.png", width = 7, height = 7)
```

```
parms = c(0.3, 11, 0.11, 9999, 0, 0)
x0 = c(7060.94, 0, 1.1, 2936.53)
n_days = length(obsdata$date)
timerange = seq(1,n_days)

SIRincidence = as.data.frame(ode(x0, timerange, SIR_basic, parms, method = 'ode45'))
names(SIRincidence) = c('Time', 'S', 'inc', 'I', 'R')
SIRincidence$date = obsdata$date

ggplot(data = SIRincidence, aes(x = date)) +
  geom_line(aes(y = c(0,diff(inc)))) +
  geom_point(aes(y = obsdata$incidence), col = "cornflowerblue") +
  scale_x_date(breaks = "1 month", date_labels = "%b") +
  labs(y = "Incidence", x = "Date") +
  theme_classic() +
  theme(text = element_text(size = 16))
```



```r
ggsave("fig6.png", width = 7, height = 7)
```