covid_data_exploration

Nora Ghenciulescu

2025-05-12

Overall model

For K = 2,. . . ,6 breakpoints, we run model on entire data.

Data prep:

```
data <- read_csv("age_group_data.csv", show_col_types = FALSE)

summary(data['date'])
```

```
##       date
##  Min.   :2020-02-19
##  1st Qu.:2020-05-16
##  Median :2020-08-08
##  Mean   :2020-08-08
##  3rd Qu.:2020-11-01
##  Max.   :2021-01-24
```

```
data_full = data %>% group_by(date) %>%
  summarise(onsets=sum(onsets)) %>%
  right_join(tibble(date=seq(ymd("2020-02-19"), ymd("2020-05-15"), by = "1 day"))) %>%
  arrange(date) %>%
  mutate(onsets=replace_na(onsets,0))
```

```
## Joining with ‘by = join_by(date)‘
```

```
data_full
```

```
## # A tibble: 87 x 2
##    date       onsets
##    <date>      <dbl>
##  1 2020-02-19      1
##  2 2020-02-20      0
##  3 2020-02-21      0
##  4 2020-02-22      2
##  5 2020-02-23      1
##  6 2020-02-24      0
##  7 2020-02-25      1
##  8 2020-02-26      1
##  9 2020-02-27      1
## 10 2020-02-28      2
## # i 77 more rows
```

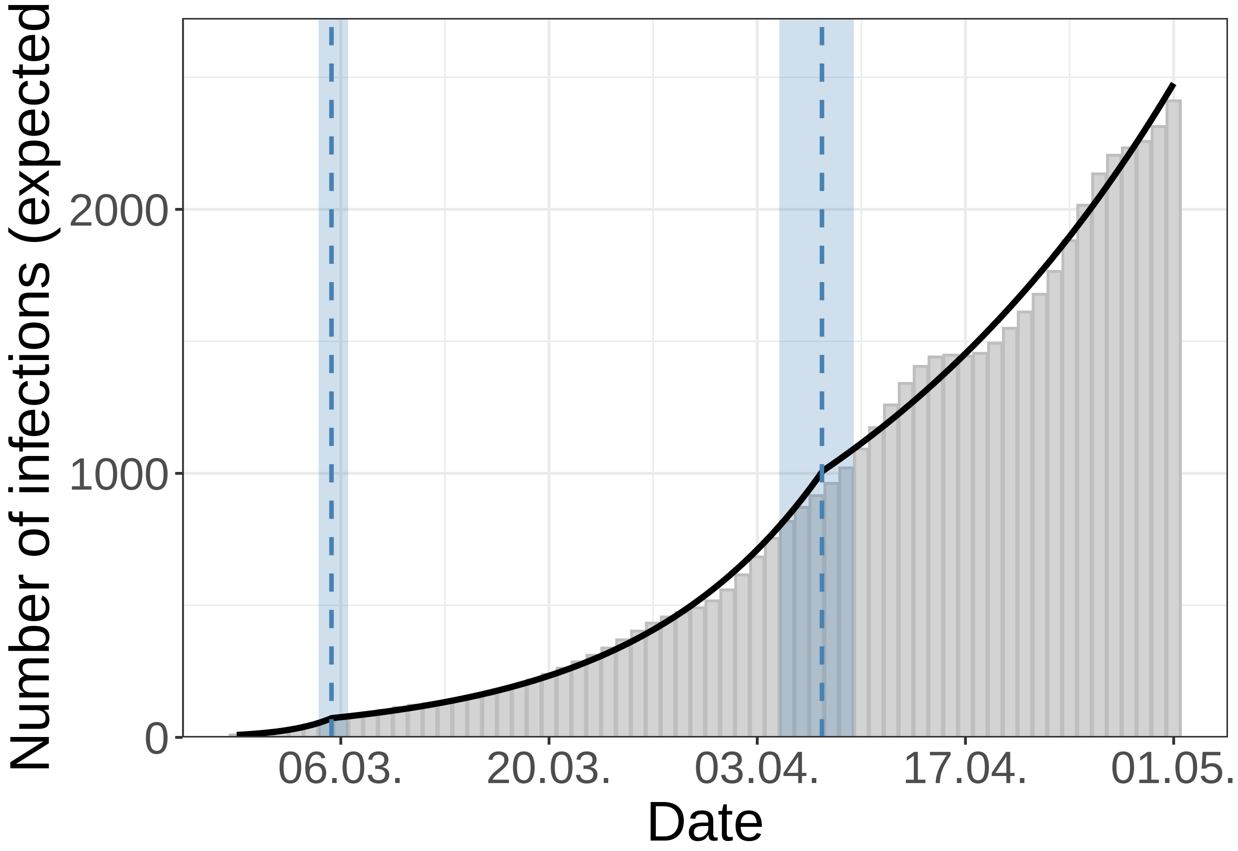
```r
cp_res_full = perform_cp_analysis(data = data_full,
                                  type = "both",
                                  cp_max_onset = 6,
                                  cp_max_backpro = 6,
                                  save_disc_optim_results = T,
                                  use_disc_optim_results = T,
                                  name_disc =  "saved_data")
```

```
## [1] "perform analysis of backprojected infections"
## [1] "estimate change point models based on segmented package infections"

## Warning: The returned fit is OK, but not of class ’segmented’.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## n bp:3

## Warning: The returned fit is OK, but not of class ’segmented’.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## n bp:4

## Warning: The returned fit is OK, but not of class ’segmented’.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## n bp:5

## Warning: The returned fit is OK, but not of class ’segmented’.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## n bp:6

## Warning: The returned fit is OK, but not of class ’segmented’.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## [1] "perform analysis of onsets"
## [1] "estimate change point models based on segmented package onset"
```

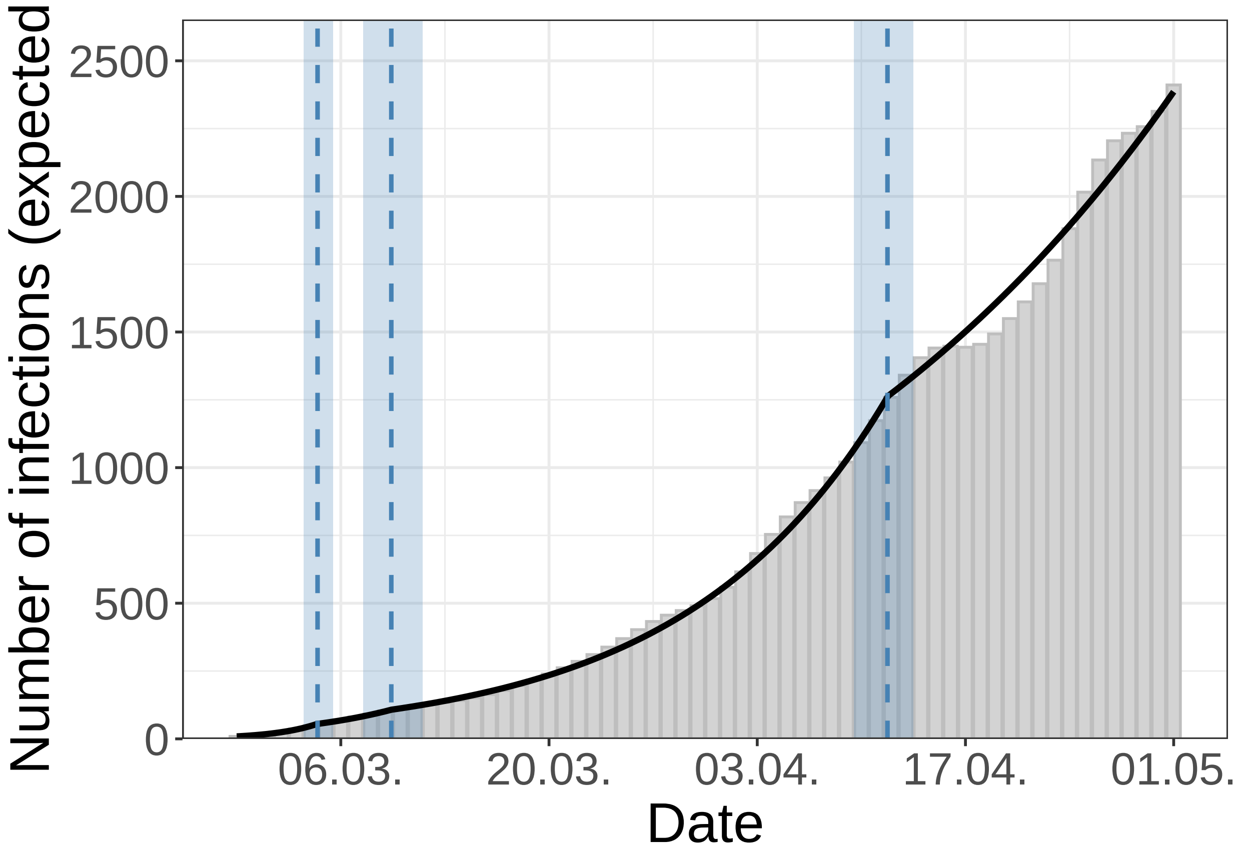
```r
cp_res_full
```

```
## $aic_backpro
##    two_bp  three_bp   four_bp   five_bp    six_bp
## -257.8450 -280.7983 -277.7172 -283.4957 -282.0102
##
```

```
## $bic_backpro
##    two_bp  three_bp   four_bp   five_bp    six_bp
## -240.5739 -259.2094 -251.8106 -253.2713 -247.4681
##
## $cp_segmented_list_backpro
## $cp_segmented_list_backpro$two_bp
## $cp_segmented_list_backpro$two_bp$segmented_model
## Generalized least squares fit by maximum likelihood
##   Model: NULL
##   Data: NULL
##   Log-likelihood: 136.9225
##
## Coefficients:
## (Intercept)           t        U1.t        U2.t      psi1.t      psi2.t
##  1.89961631  0.32315834 -0.24359003 -0.04151958  0.00000000  0.00000000
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
## 0.7947646
## Degrees of freedom: 64 total; 58 residual
## Residual standard error: 0.04657025
##
## $cp_segmented_list_backpro$two_bp$coef
## # A tibble: 3 x 3
##   mult_factor CI_lwr CI_upr
##         <dbl>  <dbl>  <dbl>
## 1        1.38   1.36   1.41
## 2        1.08   1.08   1.09
## 3        1.04   1.03   1.04
##
## $cp_segmented_list_backpro$two_bp$breakpoints
## # A tibble: 2 x 3
##   BP                BP_CI_lwr         BP_CI_upr
##   <chr>             <chr>             <chr>
## 1 7.4 (2020-03-05)  7.1 (2020-03-05)  7.6 (2020-03-06)
## 2 40.4 (2020-04-07) 38.9 (2020-04-05) 41.8 (2020-04-09)
##
## $cp_segmented_list_backpro$two_bp$plot
```

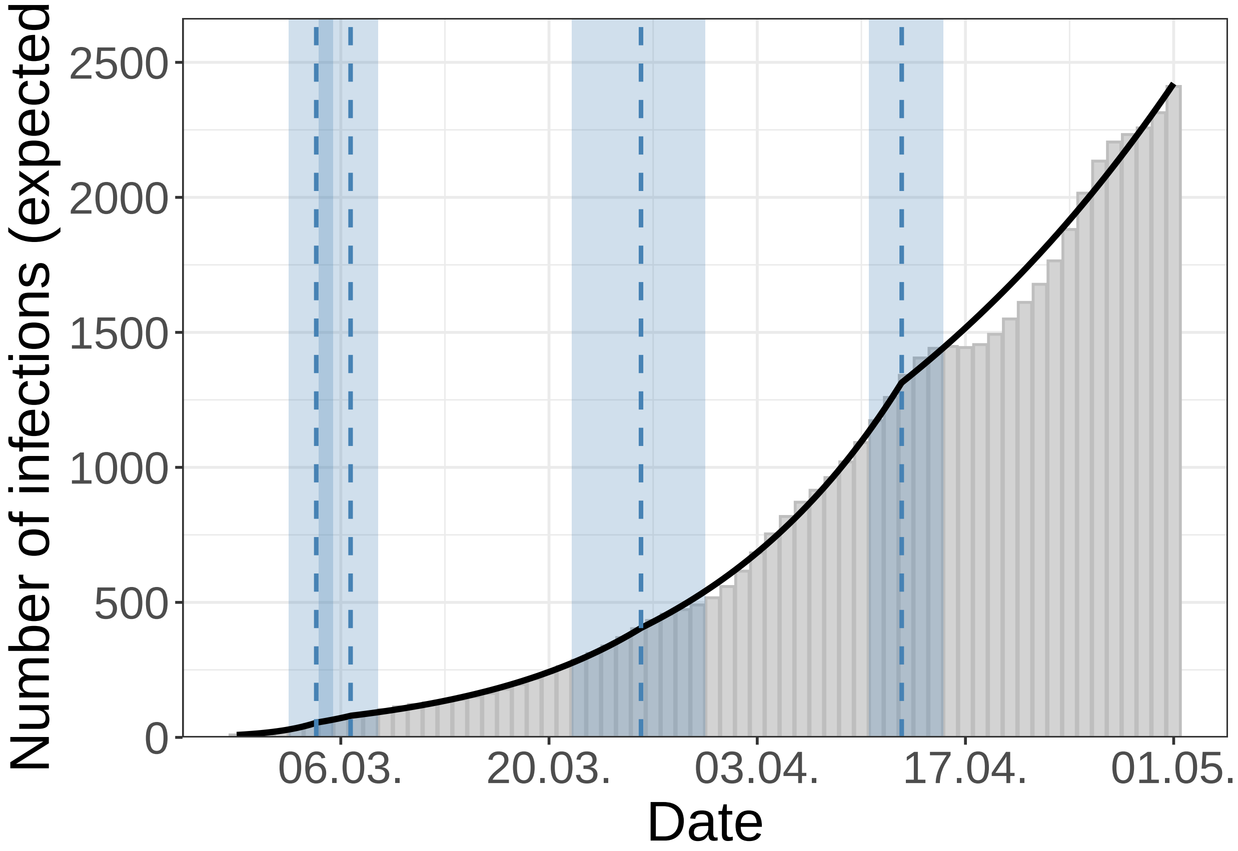


```
##
##
## $cp_segmented_list_backpro$three_bp
## $cp_segmented_list_backpro$three_bp$segmented_model
## Generalized least squares fit by maximum likelihood
##   Model: NULL
##   Data: NULL
##   Log-likelihood: 150.3991
##
## Coefficients:
## (Intercept)           t        U1.t        U2.t        U3.t      psi1.t
##  1.86442022  0.33425500 -0.19972404 -0.06070199 -0.04074813  0.00000000
##      psi2.t      psi3.t
##  0.00000000  0.00000000
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
## 0.8435632
## Degrees of freedom: 64 total; 56 residual
## Residual standard error: 0.04255586
##
## $cp_segmented_list_backpro$three_bp$coef
## # A tibble: 4 x 3
##   mult_factor CI_lwr CI_upr
##          <dbl>  <dbl>  <dbl>
## 1         1.40   1.37   1.42
## 2         1.14   1.12   1.17
## 3         1.08   1.07   1.08
## 4         1.03   1.03   1.04
##
## $cp_segmented_list_backpro$three_bp$breakpoints
## # A tibble: 3 x 3
##   BP                 BP_CI_lwr         BP_CI_upr
##   <chr>              <chr>             <chr>
## 1 6.4 (2020-03-04)  6.2 (2020-03-04)  6.7 (2020-03-05)
## 2 11.4 (2020-03-09) 10.5 (2020-03-08) 12.2 (2020-03-11)
## 3 44.8 (2020-04-12) 43.5 (2020-04-10) 46 (2020-04-13)
##
## $cp_segmented_list_backpro$three_bp$plot
```

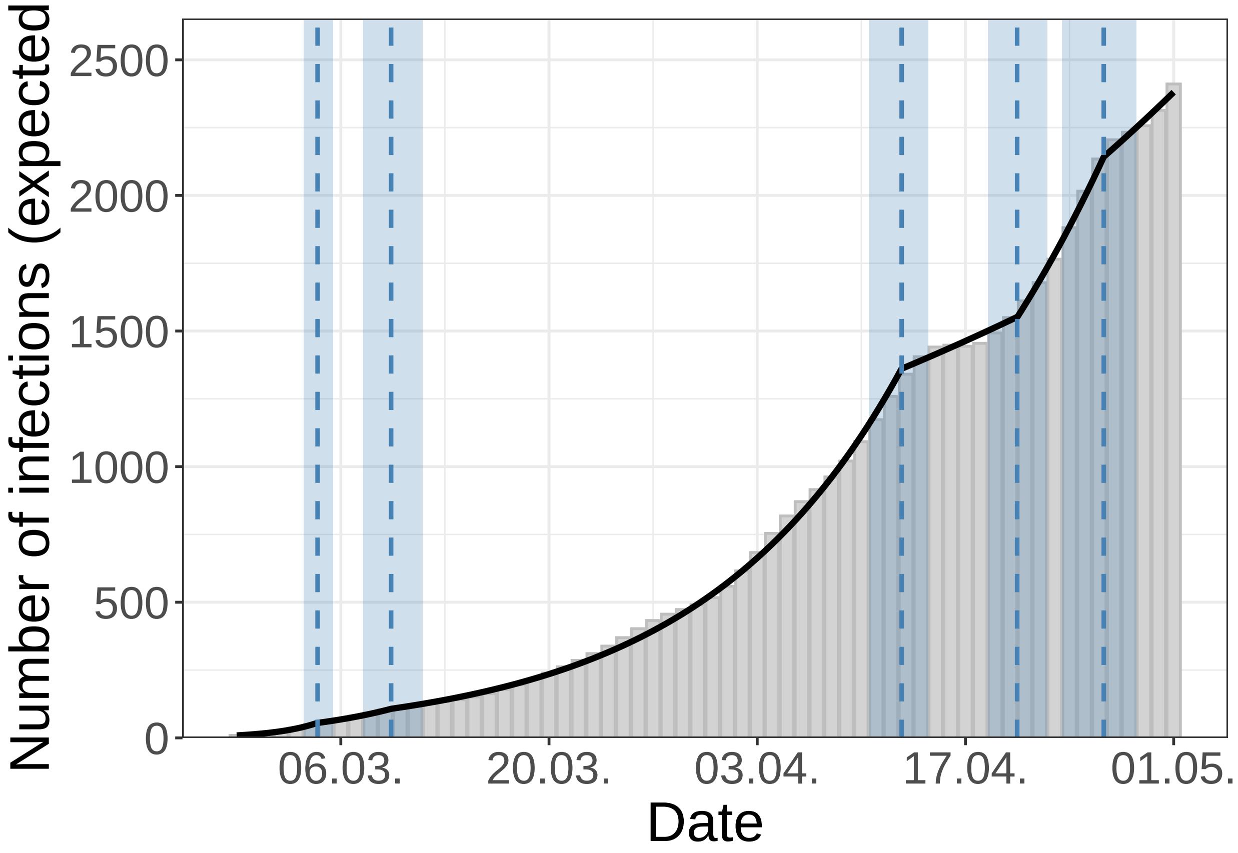


```
##
##
## $cp_segmented_list_backpro$four_bp
## $cp_segmented_list_backpro$four_bp$segmented_model
## Generalized least squares fit by maximum likelihood
##   Model: NULL
##   Data: NULL
##   Log-likelihood: 150.8586
##
```

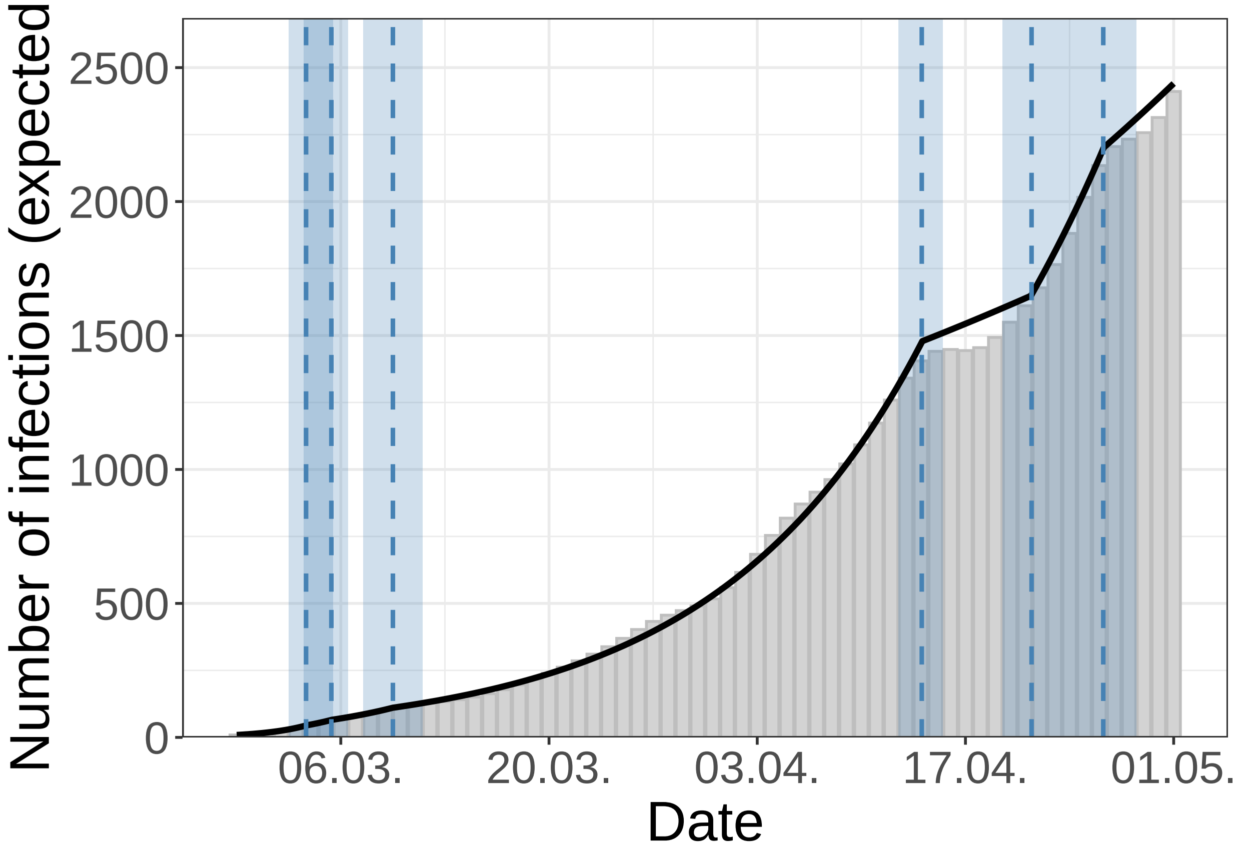
```
## Coefficients:
## (Intercept)           t        U1.t        U2.t        U3.t        U4.t
##  1.86181393  0.33556015 -0.17172066 -0.08079637 -0.01591590 -0.03368810
##      psi1.t      psi2.t      psi3.t      psi4.t
##  0.00000000  0.00000000  0.00000000  0.00000000
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
## 0.7973277
## Degrees of freedom: 64 total; 54 residual
## Residual standard error: 0.03766359
##
## $cp_segmented_list_backpro$four_bp$coef
## # A tibble: 5 x 3
##   mult_factor CI_lwr CI_upr
##         <dbl>  <dbl>  <dbl>
## 1        1.40   1.37   1.43
## 2        1.18   1.12   1.24
## 3        1.09   1.08   1.09
## 4        1.07   1.06   1.08
## 5        1.03   1.03   1.04
##
## $cp_segmented_list_backpro$four_bp$breakpoints
## # A tibble: 4 x 3
##   BP                 BP_CI_lwr          BP_CI_upr
##   <chr>              <chr>              <chr>
## 1 6.4 (2020-03-04)  6 (2020-03-03)    6.7 (2020-03-05)
## 2 8.7 (2020-03-07)  7.9 (2020-03-05)  9.5 (2020-03-08)
## 3 28.2 (2020-03-26) 24.9 (2020-03-22) 31.4 (2020-03-30)
## 4 45.7 (2020-04-13) 44.2 (2020-04-11) 47.3 (2020-04-15)
##
## $cp_segmented_list_backpro$four_bp$plot
```



```
##
##
## $cp_segmented_list_backpro$five_bp
## $cp_segmented_list_backpro$five_bp$segmented_model
## Generalized least squares fit by maximum likelihood
##   Model: NULL
##   Data: NULL
##   Log-likelihood: 155.7478
##
## Coefficients:
## (Intercept)           t        U1.t        U2.t        U3.t        U4.t
##  1.86234237  0.33453917 -0.20022504 -0.06023710 -0.05718015  0.03862937
##        U5.t      psi1.t      psi2.t      psi3.t      psi4.t      psi5.t
## -0.03308911  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
## 0.8168611
## Degrees of freedom: 64 total; 52 residual
## Residual standard error: 0.03648349
##
## $cp_segmented_list_backpro$five_bp$coef
## # A tibble: 6 x 3
##   mult_factor CI_lwr CI_upr
##         <dbl>  <dbl>  <dbl>
## 1        1.40   1.37   1.42
## 2        1.14   1.12   1.17
## 3        1.08   1.07   1.08
## 4        1.02   1.00   1.03
## 5        1.06   1.04   1.08
## 6        1.02   1.00   1.04
##
## $cp_segmented_list_backpro$five_bp$breakpoints
## # A tibble: 5 x 3
##   BP                 BP_CI_lwr          BP_CI_upr
##   <chr>              <chr>              <chr>
## 1 6.4 (2020-03-04)   6.2 (2020-03-04)   6.7 (2020-03-05)
## 2 11.4 (2020-03-09)  10.6 (2020-03-08)  12.2 (2020-03-11)
## 3 45.7 (2020-04-13)  44.9 (2020-04-11)  46.6 (2020-04-14)
## 4 53.5 (2020-04-20)  52.1 (2020-04-19)  54.8 (2020-04-22)
## 5 59.3 (2020-04-26)  57.7 (2020-04-24)  60.9 (2020-04-28)
##
## $cp_segmented_list_backpro$five_bp$plot
```



```
##
##
## $cp_segmented_list_backpro$six_bp
## $cp_segmented_list_backpro$six_bp$segmented_model
## Generalized least squares fit by maximum likelihood
##   Model: NULL
##   Data: NULL
##   Log-likelihood: 157.0051
##
## Coefficients:
## (Intercept)           t        U1.t        U2.t        U3.t        U4.t
##  1.85575713  0.34208645 -0.11959709 -0.09511072 -0.05454310 -0.05799184
##        U5.t        U6.t      psi1.t      psi2.t      psi3.t      psi4.t
##  0.04425727 -0.03718547  0.00000000  0.00000000  0.00000000  0.00000000
##      psi5.t      psi6.t
##  0.00000000  0.00000000
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##      Phi
## 0.872648
## Degrees of freedom: 64 total; 50 residual
## Residual standard error: 0.04214585
##
## $cp_segmented_list_backpro$six_bp$coef
## # A tibble: 7 x 3
##   mult_factor CI_lwr CI_upr
##         <dbl>  <dbl>  <dbl>
## 1        1.41  1.38    1.44
## 2        1.25  1.19    1.31
## 3        1.14  1.11    1.17
## 4        1.08  1.07    1.08
## 5        1.01  0.999   1.03
## 6        1.06  1.04    1.08
## 7        1.02  1.00    1.04
##
## $cp_segmented_list_backpro$six_bp$breakpoints
## # A tibble: 6 x 3
##   BP                BP_CI_lwr         BP_CI_upr
##   <chr>             <chr>             <chr>
## 1 5.7 (2020-03-04)  5.2 (2020-03-03)  6.1 (2020-03-05)
## 2 7.4 (2020-03-05)  6.8 (2020-03-04)  7.9 (2020-03-06)
## 3 11.5 (2020-03-10) 10.6 (2020-03-08) 12.4 (2020-03-11)
## 4 47.1 (2020-04-14) 46.2 (2020-04-13) 47.9 (2020-04-15)
## 5 54.4 (2020-04-21) 53.3 (2020-04-20) 55.6 (2020-04-23)
## 6 59.3 (2020-04-26) 57.9 (2020-04-24) 60.7 (2020-04-28)
##
## $cp_segmented_list_backpro$six_bp$plot
```



```
##
##
##
## $aic_onset
##   two_bp three_bp  four_bp  five_bp   six_bp
##       NA       NA       NA       NA       NA
##
## $bic_onset
##   two_bp three_bp  four_bp  five_bp   six_bp
##       NA       NA       NA       NA       NA
##
## $cp_segmented_list_onset
## $cp_segmented_list_onset$two_bp
## $cp_segmented_list_onset$two_bp$segmented_model
## [1] NA
##
## $cp_segmented_list_onset$two_bp$coef
## [1] NA
##
## $cp_segmented_list_onset$two_bp$breakpoints
## [1] NA
##
## $cp_segmented_list_onset$two_bp$plot
## [1] NA
##
##
```

```
## $cp_segmented_list_onset$three_bp
## $cp_segmented_list_onset$three_bp$segmented_model
## [1] NA
##
## $cp_segmented_list_onset$three_bp$coef
## [1] NA
##
## $cp_segmented_list_onset$three_bp$breakpoints
## [1] NA
##
## $cp_segmented_list_onset$three_bp$plot
## [1] NA
##
##
## $cp_segmented_list_onset$four_bp
## $cp_segmented_list_onset$four_bp$segmented_model
## [1] NA
##
## $cp_segmented_list_onset$four_bp$coef
## [1] NA
##
## $cp_segmented_list_onset$four_bp$breakpoints
## [1] NA
##
## $cp_segmented_list_onset$four_bp$plot
## [1] NA
##
##
## $cp_segmented_list_onset$five_bp
## $cp_segmented_list_onset$five_bp$segmented_model
## [1] NA
##
## $cp_segmented_list_onset$five_bp$coef
## [1] NA
##
## $cp_segmented_list_onset$five_bp$breakpoints
## [1] NA
##
## $cp_segmented_list_onset$five_bp$plot
## [1] NA
##
##
## $cp_segmented_list_onset$six_bp
## $cp_segmented_list_onset$six_bp$segmented_model
## [1] NA
##
## $cp_segmented_list_onset$six_bp$coef
## [1] NA
##
## $cp_segmented_list_onset$six_bp$breakpoints
## [1] NA
##
## $cp_segmented_list_onset$six_bp$plot
## [1] NA
```

BIC:

```
cp_res_full['bic_backpro']
```

```
## $bic_backpro
##    two_bp  three_bp   four_bp   five_bp    six_bp
## -240.5739 -259.2094 -251.8106 -253.2713 -247.4681
```

Optimal number of breakpoints is 3 break points!

Models per age group

```
# Split data:
data_014 = data %>% dplyr::filter(age_group=="0-14") %>%
  group_by(date) %>%
  summarise(onsets=sum(onsets)) %>%
  right_join(tibble(date=seq(ymd("2020-02-24"), ymd("2020-05-15"), by = "1 day"))) %>%
  arrange(date) %>%
  mutate(onsets=replace_na(onsets,0))
```

```
## Joining with ‘by = join_by(date)‘
```

```
data_1559 = data %>% dplyr::filter(age_group=="15-59") %>%
  group_by(date) %>%
  summarise(onsets=sum(onsets)) %>%
  right_join(tibble(date=seq(ymd("2020-02-24"), ymd("2020-05-15"), by = "1 day"))) %>%
  arrange(date) %>%
  mutate(onsets=replace_na(onsets,0))
```

```
## Joining with ‘by = join_by(date)‘
```

```
data_6079 = data %>% dplyr::filter(age_group=="60-79") %>%
  group_by(date) %>%
  summarise(onsets=sum(onsets)) %>%
  right_join(tibble(date=seq(ymd("2020-02-24"), ymd("2020-05-15"), by = "1 day"))) %>%
  arrange(date) %>%
  mutate(onsets=replace_na(onsets,0))
```

```
## Joining with ‘by = join_by(date)‘
```

```
data_80 = data %>% dplyr::filter(age_group=="80+") %>%
  group_by(date) %>%
  summarise(onsets=sum(onsets)) %>%
  right_join(tibble(date=seq(ymd("2020-02-24"), ymd("2020-05-15"), by = "1 day"))) %>%
  arrange(date) %>%
  mutate(onsets=replace_na(onsets,0))
```

```
## Joining with ‘by = join_by(date)‘
```

```r
# Run model for each:
cp_res_014 = perform_cp_analysis(data = data_014,
                                      type = "backpro",
                                      cp_max_onset = 6,
                                      cp_max_backpro = 6,
                                      save_disc_optim_results = T,
                                      use_disc_optim_results = T,
                                      name_disc =  "saved_data_014")
```

```
## [1] "perform analysis of backprojected infections"
## [1] "estimate change point models based on segmented package infections"

## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## n bp:3

## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## n bp:4

## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## n bp:5

## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)

## n bp:6

## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```r
res_014_selected_backpro = cp_res_014$cp_segmented_list_backpro[[which.min(cp_res_014$bic_backpro)]]
```

```r
cp_res_1559 = perform_cp_analysis(data = data_1559,
                                      type = "backpro",
                                      cp_max_onset = 6,
                                      cp_max_backpro = 6,
                                      save_disc_optim_results = T,
                                      use_disc_optim_results = T,
                                      name_disc =  "saved_data_1559")
```

```
## [1] "perform analysis of backprojected infections"
## [1] "estimate change point models based on segmented package infections"

## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
## n bp:3
```

```
## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```
## n bp:4
```

```
## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```
## n bp:5
```

```
## Warning: max number of iterations (1008) attained
## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```
## n bp:6
```

```
res_1559_selected_backpro = cp_res_1559$cp_segmented_list_backpro[[which.min(cp_res_1559$bic_backpro)]]
```

```
cp_res_6079 = perform_cp_analysis(data = data_6079,
                                     type = "backpro",
                                     cp_max_onset = 6,
                                     cp_max_backpro = 6,
                                     save_disc_optim_results = T,
                                     use_disc_optim_results = T,
                                     name_disc =  "saved_data_6079")
```

```
## [1] "perform analysis of backprojected infections"
## [1] "estimate change point models based on segmented package infections"
```

```
## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```
## n bp:3
```

```
## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```
## n bp:4
```

```
## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```
## n bp:5
```

```
## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```
## n bp:6
```

```
## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

```r
res_6079_selected_backpro = cp_res_6079$cp_segmented_list_backpro[[which.min(cp_res_6079$bic_backpro)]]
```

```r
cp_res_80 = perform_cp_analysis(data = data_80,
                                     type = "backpro",
                                     cp_max_onset = 2,
                                     cp_max_backpro = 2,
                                     save_disc_optim_results = T,
                                     use_disc_optim_results = T,
                                     name_disc =  "saved_data_80")
```

```
## [1] "perform analysis of backprojected infections"
## [1] "estimate change point models based on segmented package infections"

## Warning: The returned fit is OK, but not of class 'segmented'.
##  If interested, call explicitly the segmented methods (plot.segmented, confint.segmented,..)
```

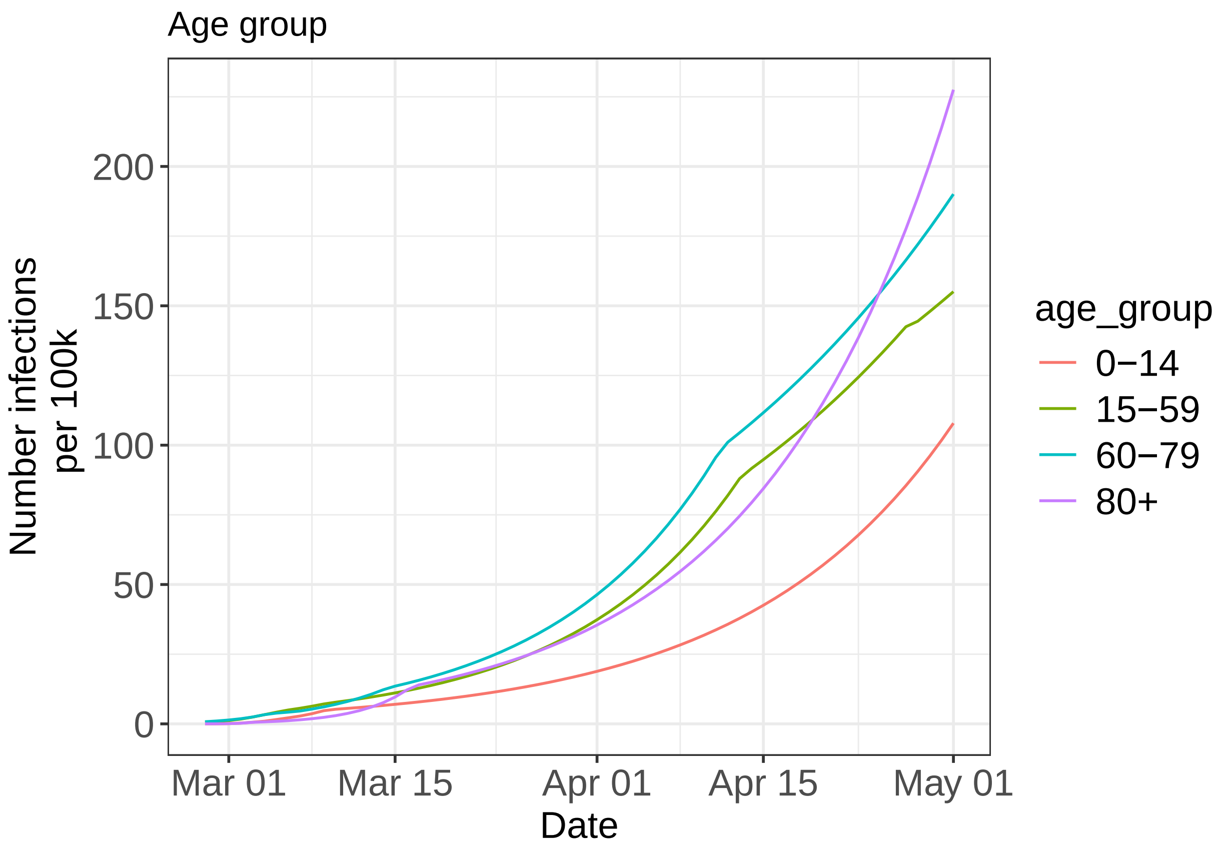
```r
res_80_selected_backpro = cp_res_80$cp_segmented_list_backpro[[which.min(cp_res_80$bic_backpro)]]
```

!!!! the 80+ group only works for 2 break points (cp__max__onset = 2, cp__max__backpro = 2)

```r
age = read.csv("total_age_distribution.csv")
age_file = age %>% mutate(age_group=cut(AGE, c(-1,14,59,79,120), labels = c("0-14", "15-59", "60-79", "8
  group_by(age_group) %>%
  summarise(n=sum(num))


# Plot:
plot_age_group = res_014_selected_backpro$segmented_model$model %>%
  dplyr::select(t, logbackpro) %>%
  mutate(age_group="0-14") %>%
  cbind(pred=res_014_selected_backpro$segmented_model$fitted) %>%
  rbind(res_1559_selected_backpro$segmented_model$model %>%
          dplyr::select(t, logbackpro) %>%
          mutate(age_group="15-59") %>%
          cbind(pred=res_1559_selected_backpro$segmented_model$fitted)) %>%
  rbind(res_6079_selected_backpro$segmented_model$model %>%
          dplyr::select(t, logbackpro) %>%
          mutate(age_group="60-79") %>%
          cbind(pred=res_6079_selected_backpro$segmented_model$fitted)) %>%
  rbind(res_80_selected_backpro$segmented_model$model %>%
          dplyr::select(t, logbackpro) %>%
          mutate(age_group="80+") %>%
          cbind(pred=res_80_selected_backpro$segmented_model$fitted)) %>%
  mutate(t=ymd("2020-02-27")+t) %>%
  right_join(age_file) %>%
  mutate(pred_per_100k = (exp(pred)/n)*100000) %>%
  ggplot() +
  geom_line(aes(t, pred_per_100k, color=age_group)) +
  ylab("Number infections\n per 100k") + xlab("Date") +
  theme(
    axis.text=element_text(size = rel(1.3)),
    axis.title=element_text(size = rel(1.3)),
    legend.text = element_text(size = rel(1.3)),
    legend.title =element_text(size = rel(1.3)))+
  guides(lty=guide_legend(title="Age group"))
```

```
## Joining with ‘by = join_by(age_group)‘
```

```
plot_age_group + ggtitle("Age group") + theme()
```



Models for 3 cities

```
# data_mty <- read_csv("age_group_data_mty.csv", show_col_types = FALSE)
# data_mty <- data_mty[ , !(names(data_mty) %in% c('state'))]
# data_mty
#
# data_mty_full = data_mty %>% group_by(date) %>%
#   summarise(onsets=sum(onsets)) %>%
#   right_join(tibble(date=seq(ymd("2020-02-24"), ymd("2020-05-15"), by = "1 day"))) %>%
#   arrange(date) %>%
#   mutate(onsets=replace_na(onsets,0))
#
# data_mty_full
# cp_res_mty = perform_cp_analysis(data = data_mty_full,
#                                       type = "both",
#                                       cp_max_onset = 6,
#                                       cp_max_backpro = 6,
#                                       save_disc_optim_results = T,
#                                       use_disc_optim_results = T,
#                                       name_disc =  "saved_data_mty")
# cp_res_mty
```

```
# cp_res_mty['bic_backpro']
```

Mty:

```
# data_jal <- read_csv("age_group_data_jal.csv", show_col_types = FALSE)
#
# data_jal_full = data_jal %>% group_by(date) %>%
#   summarise(onsets=sum(onsets)) %>%
#   right_join(tibble(date=seq(ymd("2020-02-24"), ymd("2020-05-15"), by = "1 day"))) %>%
#   arrange(date) %>%
#   mutate(onsets=replace_na(onsets,0))
#
# data_jal_full
```

```
# cp_res_jal = perform_cp_analysis(data = data_jal_full,
#                                       type = "both",
#                                       cp_max_onset = 6,
#                                       cp_max_backpro = 6,
#                                       save_disc_optim_results = T,
#                                       use_disc_optim_results = T,
#                                       name_disc =  "saved_data_jal")
# cp_res_jal
```

```
# cp_res_jal['bic_backpro']
```

Jal:

```
# data_mx <- read_csv("age_group_data_mx.csv", show_col_types = FALSE)
#
# data_mx_full = data_mx %>% group_by(date) %>%
#   summarise(onsets=sum(onsets)) %>%
#   right_join(tibble(date=seq(ymd("2020-02-24"), ymd("2020-05-15"), by = "1 day"))) %>%
#   arrange(date) %>%
#   mutate(onsets=replace_na(onsets,0))
#
# data_mx_full
```

```

# cp_res_mx = perform_cp_analysis(data = data_mx_full,
#                                       type = "both",
#                                       cp_max_onset = 6,
#                                       cp_max_backpro = 6,
#                                       save_disc_optim_results = T,
#                                       use_disc_optim_results = T,
#                                       name_disc =  "saved_data_mx")
# cp_res_mx
```

```

#cp_res_mx['bic_backpro']
```

Mx: