MacPan Example

Installing MacPan

Clone/download the repository (from here) and install locally or use:

`remotes::install_github("bbolker/McMasterPandemic")` to install the package. You will need to first install the developer version of `bbmle` (`remotes::install_github("bbolker/bbmle")`) before installing `McMasterPandemic`.

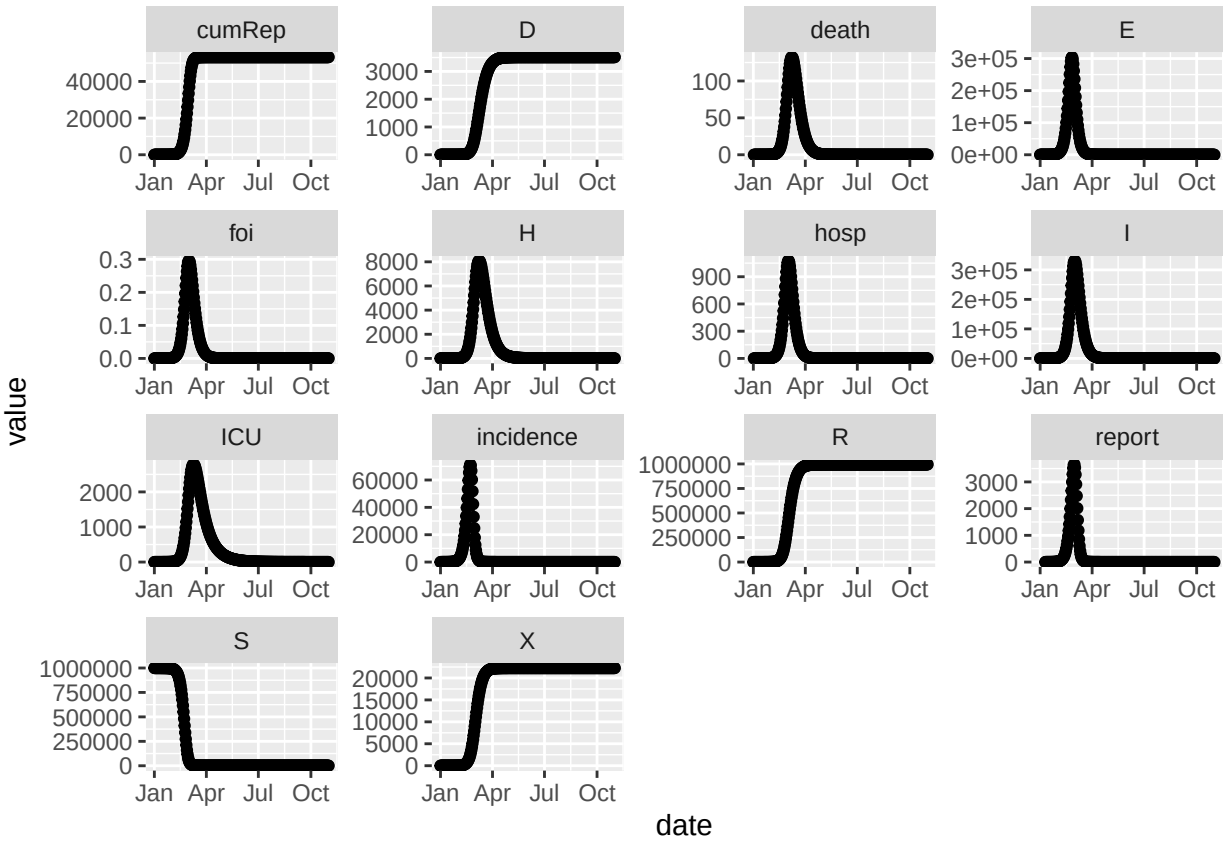
Simulating data time series

MLi: Do we have a document of the basic model (e.g. the flow diagram, and what the states/compartments mean?)

```
params <- read_params("ICU1.csv")

## Need to set up opt_pars because you need this for forecast_sim
opt_pars <- list(params=c(beta0=params[["beta0"]]))
simdat <- forecast_sim(p = unlist(opt_pars)
    , opt_pars = opt_pars
    , base_params = params
    , start_date = "2020-01-01"
    , end_date = "2020-11-01"
)
```

Plotting simulated time series

```
gg <- (ggplot(simdat, aes(x=date,y=value))
    + geom_point()
    + geom_line()
    + facet_wrap(~var,scale="free")
)

print(gg)
```



Changing parameters

MLi: Maybe use Zach's shinny app to play around with different parameters combinations. This is the way to manually change it via code.

```
print(summary(params))
```

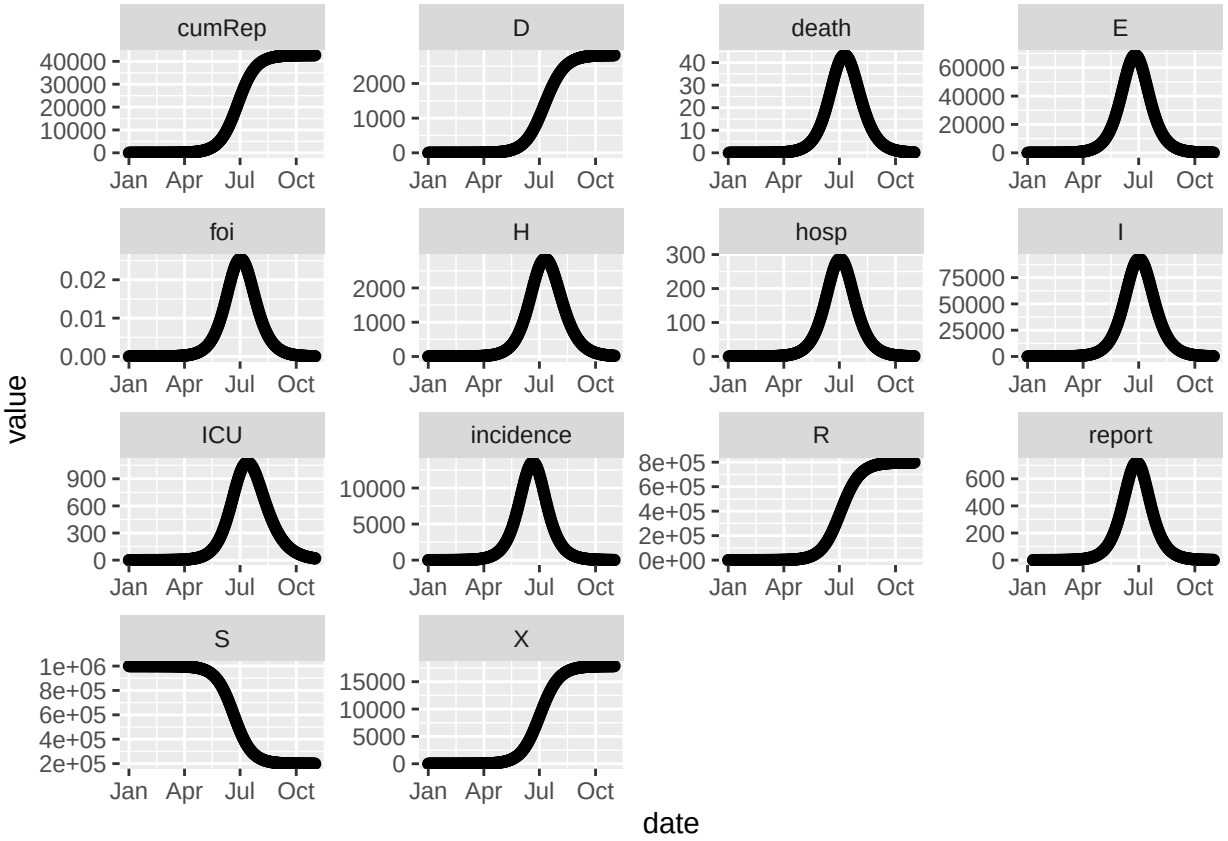
```
##         r0         R0       Gbar    CFR_gen   dbl_time
##  0.2278149  6.5180089 12.1897402  0.0352000  3.0425898
```

```
## Change R0
newparams <- fix_pars(params, target=c(R0=2))

print(summary(newparams))
```

```
##          r0          R0        Gbar     CFR_gen    dbl_time
##  0.06649208  2.00002038 12.18974018  0.03520000 10.42450796
```

```
new_opt_pars <- list(params=c(beta0=newparams[["beta0"]]))

simdat2 <- forecast_sim(p = unlist(new_opt_pars)
    , opt_pars = new_opt_pars
    , base_params = newparams    ## change parameter set here!
    , start_date = "2020-01-01"
    , end_date = "2020-11-01"
)

print(gg %+% simdat2)
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```

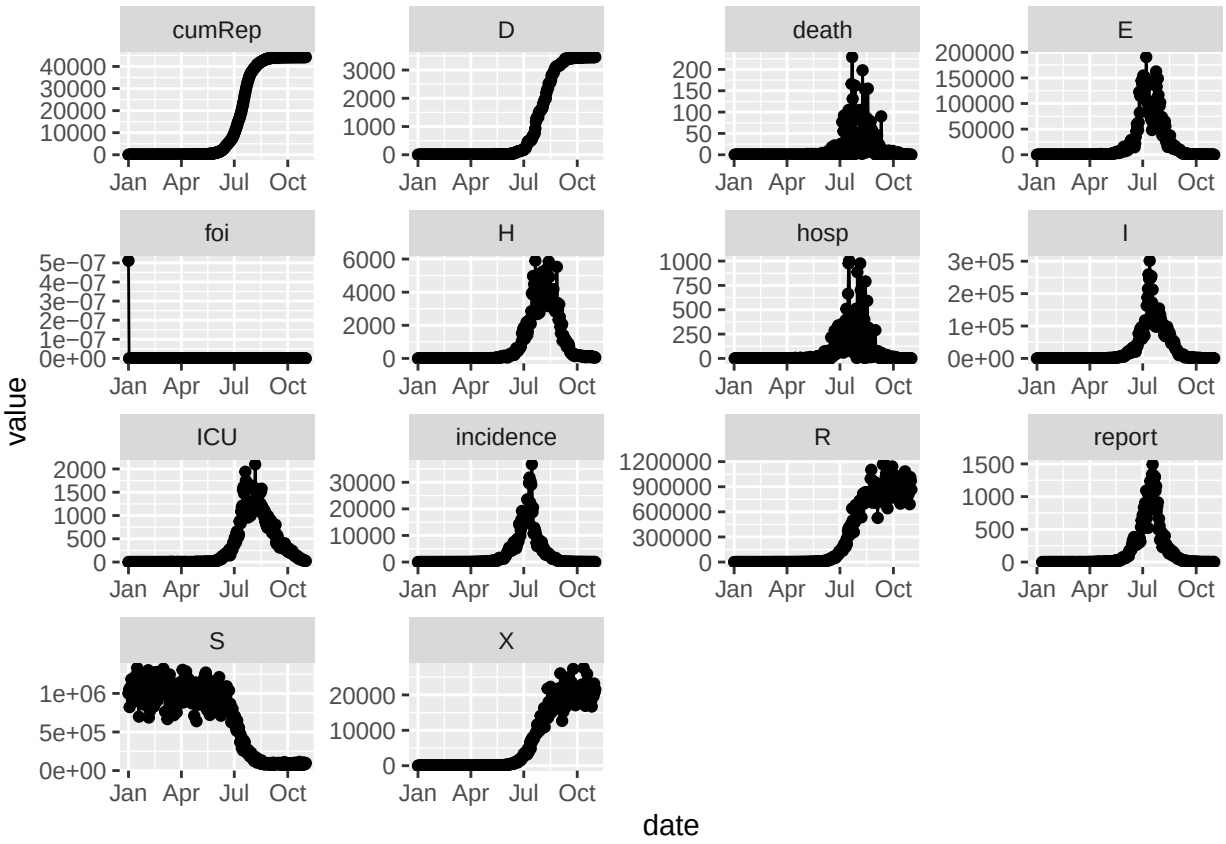


Question: Extract the reported cases time series and use epigrowthfit to estimate little r. Double check if it is the same using the summary function in macpan.

Adding Stochastic Noise

```
newparams2 <- update(newparams, obs_disp = 50, proc_disp=1)

simdat3 <- forecast_sim(p = unlist(new_opt_pars)
    , opt_pars = new_opt_pars
    , base_params = newparams2
    , stoch = c(proc=TRUE,obs=TRUE)
    , stoch_start = c(proc="2020-01-01",obs="2020-01-01")
    , start_date = "2020-01-01"
    , end_date = "2020-11-01"
)

print(gg %+% simdat3)
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```



Calibrating to simuated data

```
report_dat <- (simdat3
    %>% filter(var == "report")
)

## I am estimating beta0 only, you need to specify what parameters you want to estimate

opt_pars <- list(params = c(beta0=0.1))

fitmod <- calibrate_comb(data = report_dat
    , params = newparams2
    , opt_pars = opt_pars
    , use_DEoptim = FALSE ## We don't want to wait that long
    , debug_plot = FALSE ## TRUE to watch fitting process, don't do it in rmd
)

print(summary(fitmod))
```

```
##   start_date         r0       R0     Gbar CFR_gen dbl_time
## 1 2019-12-17 0.05846677 1.841338 12.18974  0.0352  11.8554
```

```
print(summary(newparams2))
```

```
##          r0          R0        Gbar     CFR_gen    dbl_time
##  0.06649208  2.00002038 12.18974018  0.03520000 10.42450796
```

Ontario, Canada

Reading in data from MLi's github page

```
tsdat_url <- "https://wzmli.github.io/COVID19-Canada/git_push/clean.Rout.csv"

tsdat <- read_csv(tsdat_url)
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

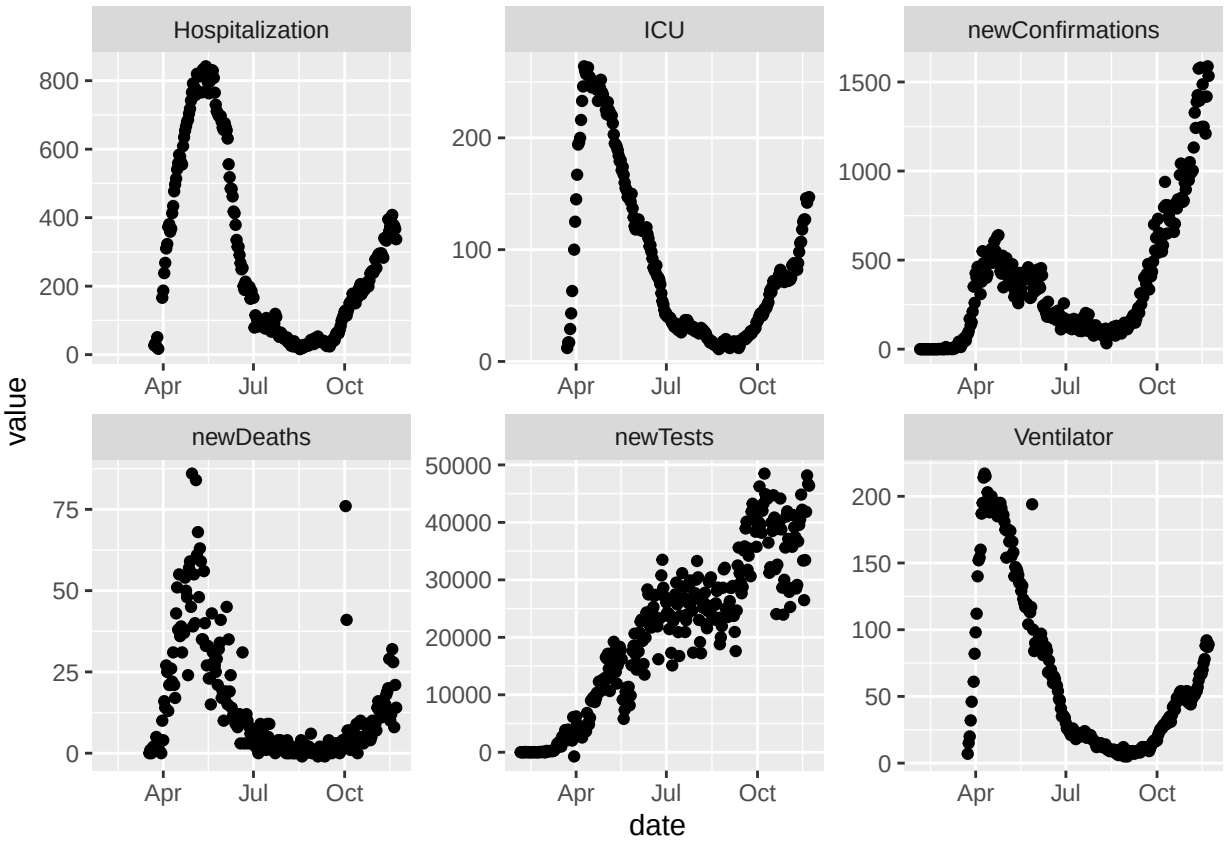
```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Date = col_date(format = ""),
##   Province = col_character(),
##   source = col_character(),
##   Note = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
## Section 2: Clean data
### Clean ts data
Ontario_dat <- (tsdat
    %>% filter(Province=="ON")
   %>% select(Province,Date,Hospitalization,ICU,Ventilator,deceased,newConfirmations,newTests)
    %>% mutate(newDeaths=c(NA,diff(deceased))
    ## ON hosp includes ICU, our model compartment is just acute care
    , Hospitalization=Hospitalization-ICU)
   %>% select(-deceased)
   %>% pivot_longer(names_to="var",-c(Date,Province))
   %>% setNames(tolower(names(.)))
    %>% ungroup()
)
```

Question: Make some time series plots using the data and describe what is going on. Adding important dates!

```
ggont <- (ggplot(data=Ontario_dat, aes(x=date,y=value))
    + geom_point()
    + facet_wrap(~var, scale="free")
)

print(ggont)
```



Ontario MacPan setup

```
## translate variable names to internally used values
## drop unused variables
keep_vars <- c("H","ICU","death","report","newTests")

## Maybe keep reports only for simplicity

keep_vars <- c("report")

clean_tsdata <- (Ontario_dat
    %>% mutate_at("var",trans_state_vars)
    %>% filter(var %in% keep_vars)
)

date_vec <- as.Date(min(clean_tsdata$date):max(clean_tsdata$date))

date_df <- data.frame(date = rep(date_vec,1)
   , var = rep(c("report"),each=length(date_vec))
   )

calibrate_dat <- (left_join(date_df,clean_tsdata))

## Joining, by = c("date", "var")
```

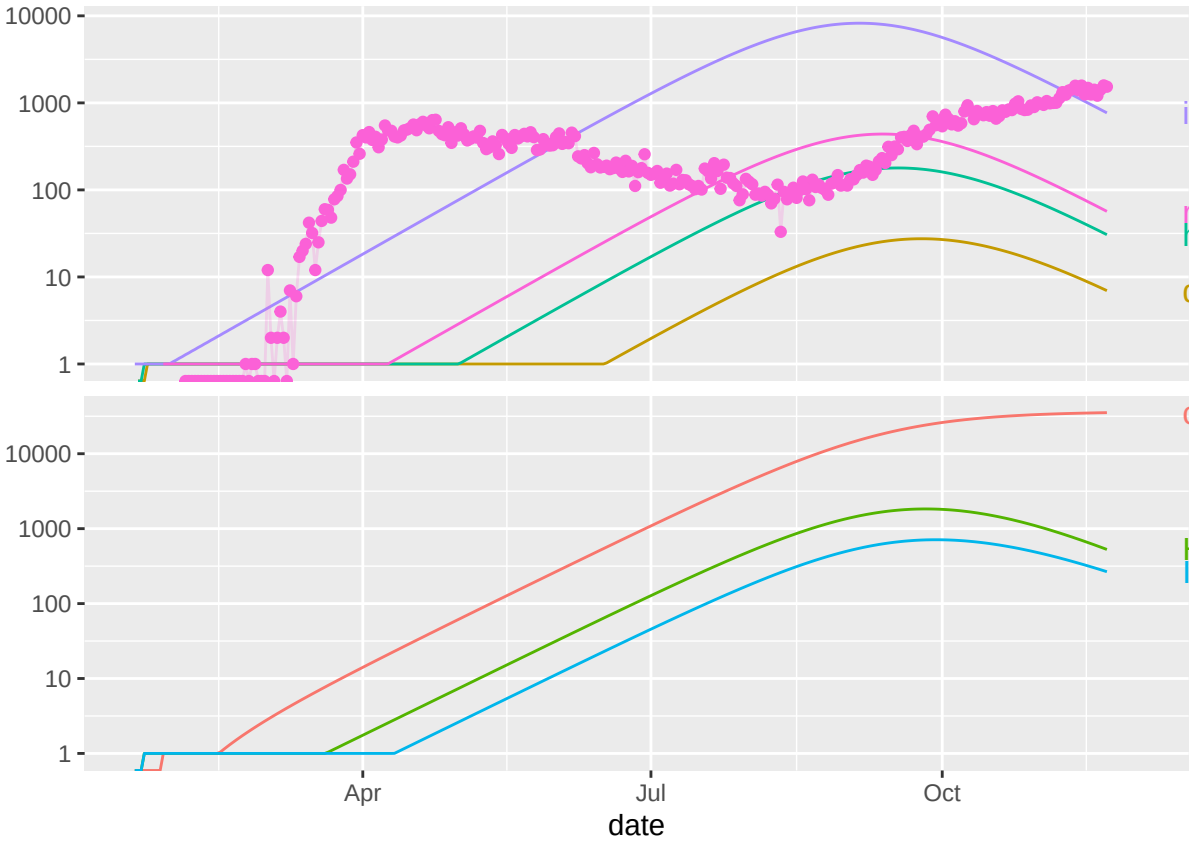
Fitting basic MacPan model

```
ontmod0 <- calibrate_comb(data = calibrate_dat
    , params = newparams2
    , opt_pars = opt_pars
    , use_DEoptim = FALSE ## We don't want to wait that long
    , debug_plot = FALSE ## TRUE to watch fitting process, don't do it in rmd
)


print(summary(ontmod0))
```

```
##   start_date         r0       R0     Gbar CFR_gen dbl_time
## 1 2020-01-20 0.05127171 1.680297 12.18974  0.0352  13.5191
```

```
print(plot(ontmod0,data=calibrate_dat))
```

```
## Loading required namespace: directlabels
```



Question: Why are the fits so bad? Ans: - Model is too simple - strong assumptions - interventions and lockdown - two distinct waves

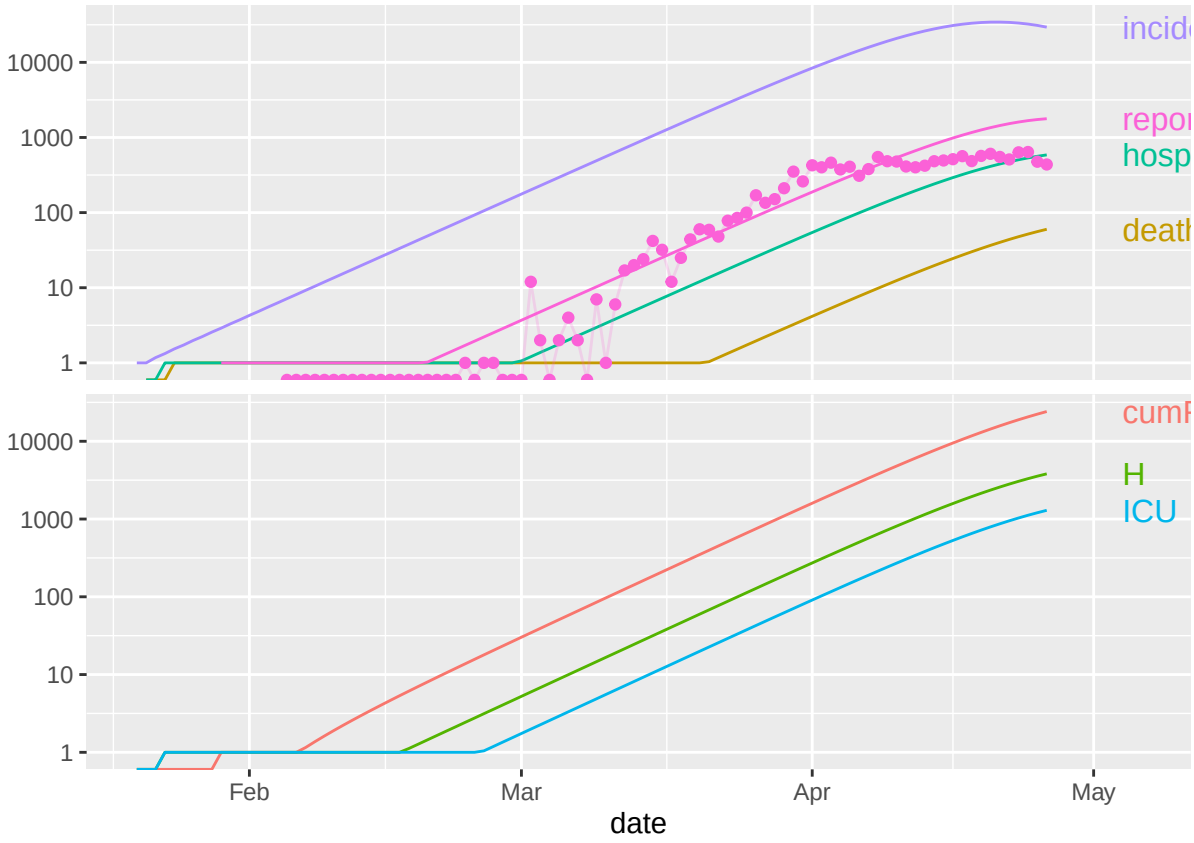
Initial wave

```
ont1stwave <- calibrate_dat %>% filter(date <= as.Date("2020-04-26"))

ontmod1 <- calibrate_comb(data = ont1stwave
```

```
    , params = newparams2
    , opt_pars = opt_pars
    , use_DEoptim = FALSE ## We don't want to wait that long
    , debug_plot = FALSE ## TRUE to watch fitting process, don't do it in rmd
)


print(summary(ontmod1))
```

```
##   start_date        r0       R0     Gbar CFR_gen dbl_time
## 1 2020-01-20 0.1283672 3.355756 12.18974  0.0352 5.399721
```

```
print(plot(ontmod1,data=ont1stwave))
```



Question: What is the growth rate (little r)? Do we get similar estimates as epigrowthfit?

MLi: Maybe do the same thing for second wave?

Mobility

We can use mobility as a proxy for change in transmission rate.

```
## seems this can be very slow, so we cache ...
## mobility CSV *won't* get redownloaded after the first time you run this chunk ...
## we should consider saving a CSV file ... could also use the 'pins' package, which
## does smart URL caching
google_url <- "https://www.gstatic.com/covid19/mobility/Global_Mobility_Report.csv"
```

```
google <- read_csv(google_url)
```

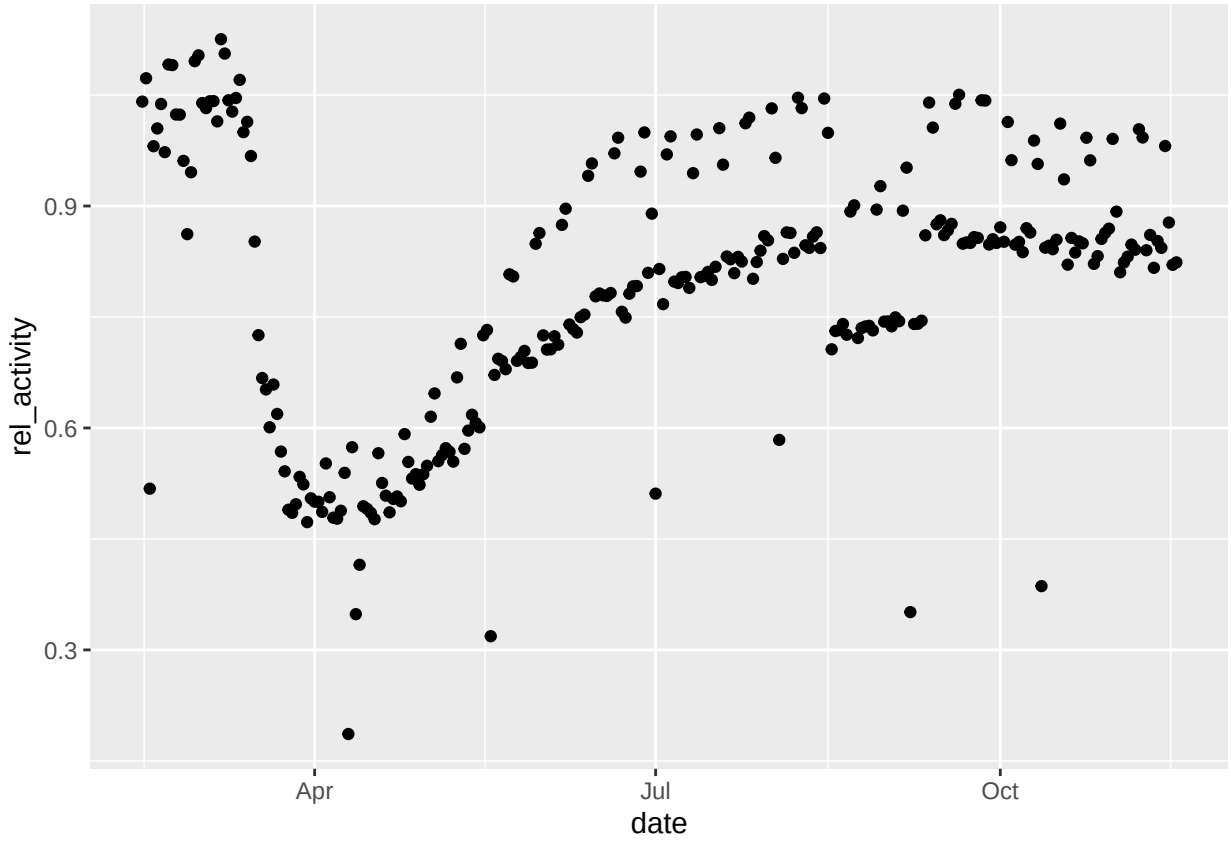
```
## Parsed with column specification:
## cols(
##   country_region_code = col_character(),
##   country_region = col_character(),
##   sub_region_1 = col_character(),
##   sub_region_2 = col_logical(),
##   metro_area = col_logical(),
##   iso_3166_2_code = col_character(),
##   census_fips_code = col_logical(),
##   date = col_date(format = ""),
##   retail_and_recreation_percent_change_from_baseline = col_double(),
##   grocery_and_pharmacy_percent_change_from_baseline = col_double(),
##   parks_percent_change_from_baseline = col_double(),
##   transit_stations_percent_change_from_baseline = col_double(),
##   workplaces_percent_change_from_baseline = col_double(),
##   residential_percent_change_from_baseline = col_double()
## )
```

```
## Warning: 3332938 parsing failures.
##  row        col           expected                   actual
## 2442 metro_area 1/0/T/F/TRUE/FALSE Kabul Metropolitan Area 'https://www.gstatic.com/covid19/mobility/
## 2443 metro_area 1/0/T/F/TRUE/FALSE Kabul Metropolitan Area 'https://www.gstatic.com/covid19/mobility/
## 2444 metro_area 1/0/T/F/TRUE/FALSE Kabul Metropolitan Area 'https://www.gstatic.com/covid19/mobility/
## 2445 metro_area 1/0/T/F/TRUE/FALSE Kabul Metropolitan Area 'https://www.gstatic.com/covid19/mobility/
## 2446 metro_area 1/0/T/F/TRUE/FALSE Kabul Metropolitan Area 'https://www.gstatic.com/covid19/mobility/
## .... ......... .................. ...................... ....................................
## See problems(...) for more details.
```

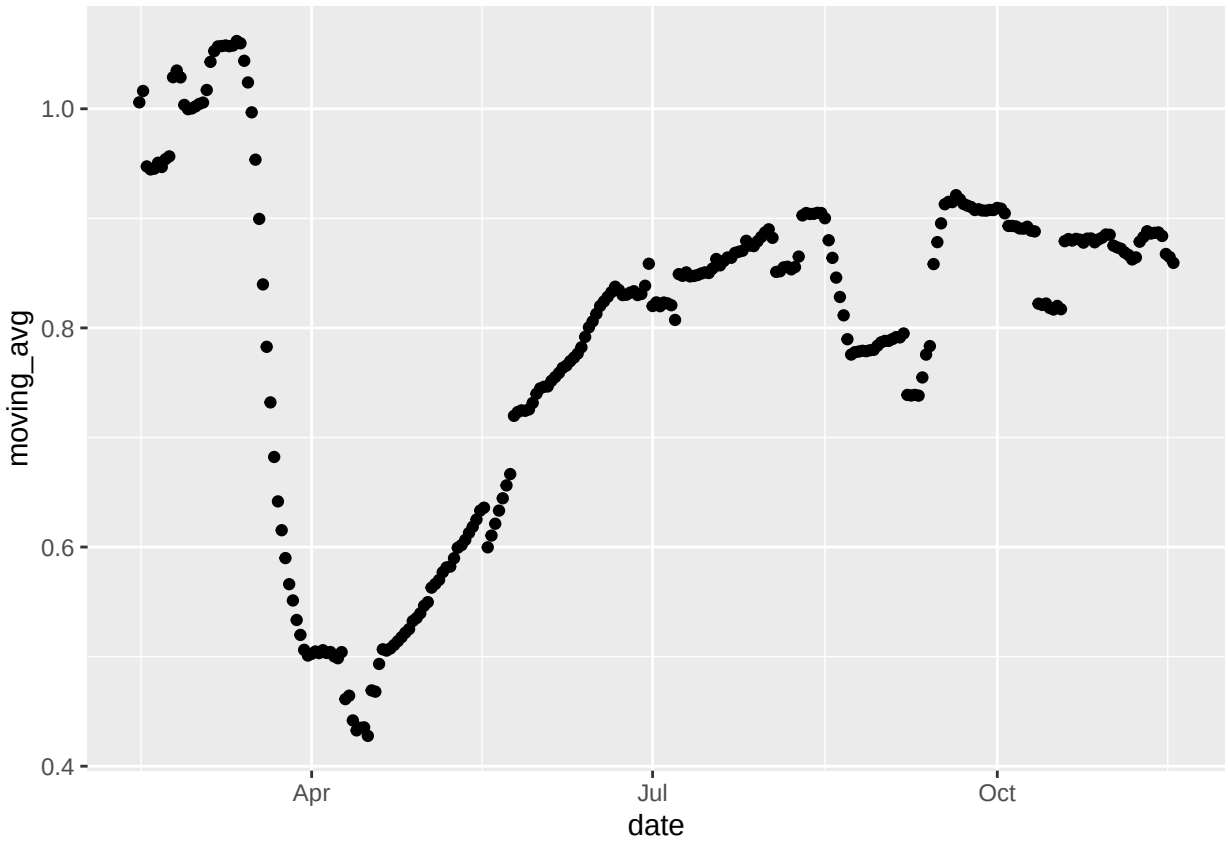
```
clean_google <- (google
    %>% filter(country_region == "Canada", sub_region_1 == "Ontario")
    %>% select(date, contains("baseline"))
    %>% pivot_longer(names_to="type", values_to="value", -c(date))
    %>% mutate_at("date", as.Date)
    %>% mutate_at("type", str_remove, "\\_percent.*")
    %>% mutate_at("value", ~./100+1)
)

clean_mobdat <- (clean_google
    %>% mutate(tvec=as.numeric(date-min(date,na.rm=TRUE)))
    %>% filter(type %in% c("retail_and_recreation","workplaces","driving"))
    %>% dplyr::select(date,value)
    %>% group_by(date)
    %>% summarise_at("value",mean,na.rm=TRUE)
    %>% na.omit()
    %>% rename(rel_activity = value)
#   %>% mutate_at("rel_activity", ~pmin(., 1))  ## cap at 100% (? should we ?)
    %>% mutate(moving_avg = zoo::rollmean(c(rep(1,6),rel_activity),k=7))
     %>% ungroup()
)
```

Make a plot of relative activity and explain how this might have an effect for disease transmission/dynamics.

```
ggmob <- (ggplot(clean_mobdat,aes(x=date))
)

print(ggmob + geom_point(aes(y=rel_activity)))
```




```
print(ggmob + geom_point(aes(y=moving_avg)))
```



Calibrating mobility model

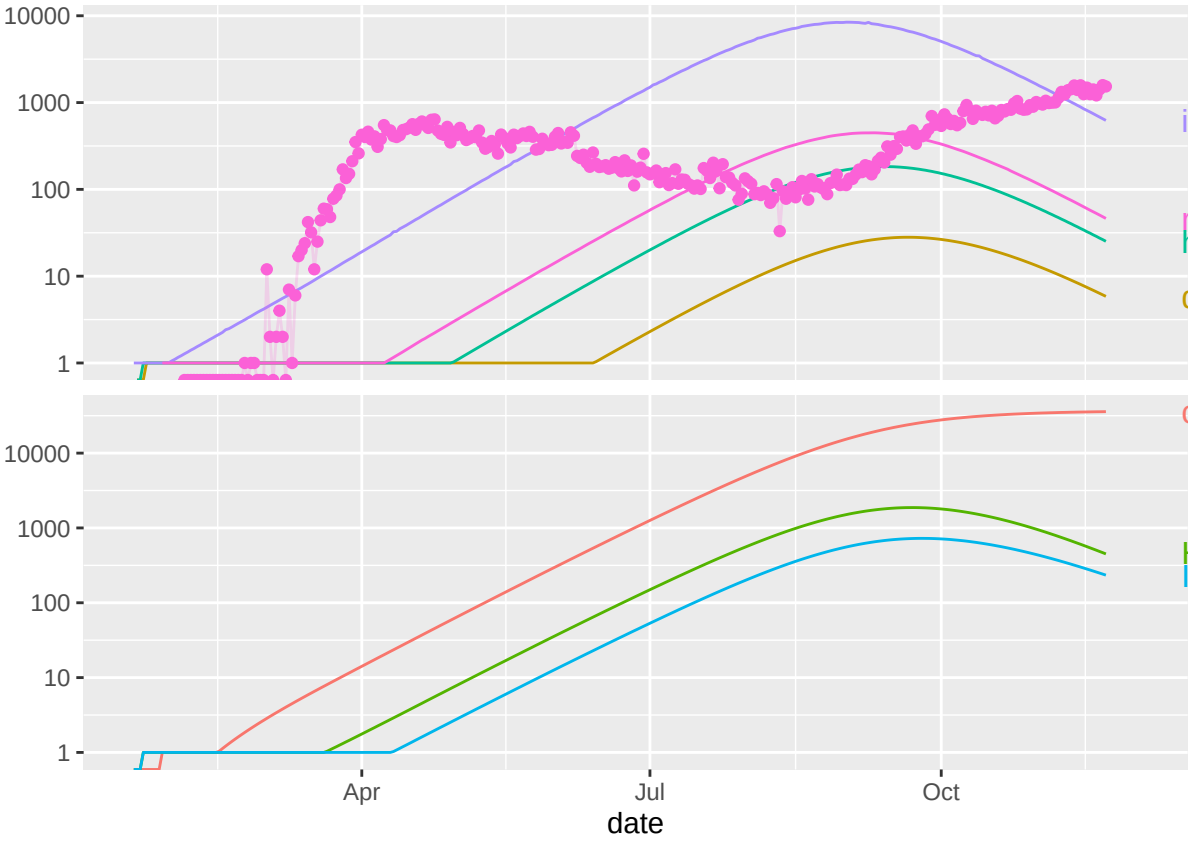
```
ontmod_mob <- calibrate_comb(data = calibrate_dat
    , params = newparams2
    , opt_pars = opt_pars
    , use_DEoptim = FALSE ## We don't want to wait that long
    , debug_plot = FALSE ## TRUE to watch fitting process, don't do it in rmd
    , mob_data = clean_mobdat
    , use_mobility = TRUE
)
```

```
# print(summary(ontmod_mob))
print(plot(ontmod_mob,data=calibrate_dat))
```

```
## Warning in (function (params, state = make_state(params[["N"]],
## params[["E0"]]), : dropped switch times on final day
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 11 row(s) containing missing values (geom_path).
```

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

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



MLi: It is trying to fit a bit better.

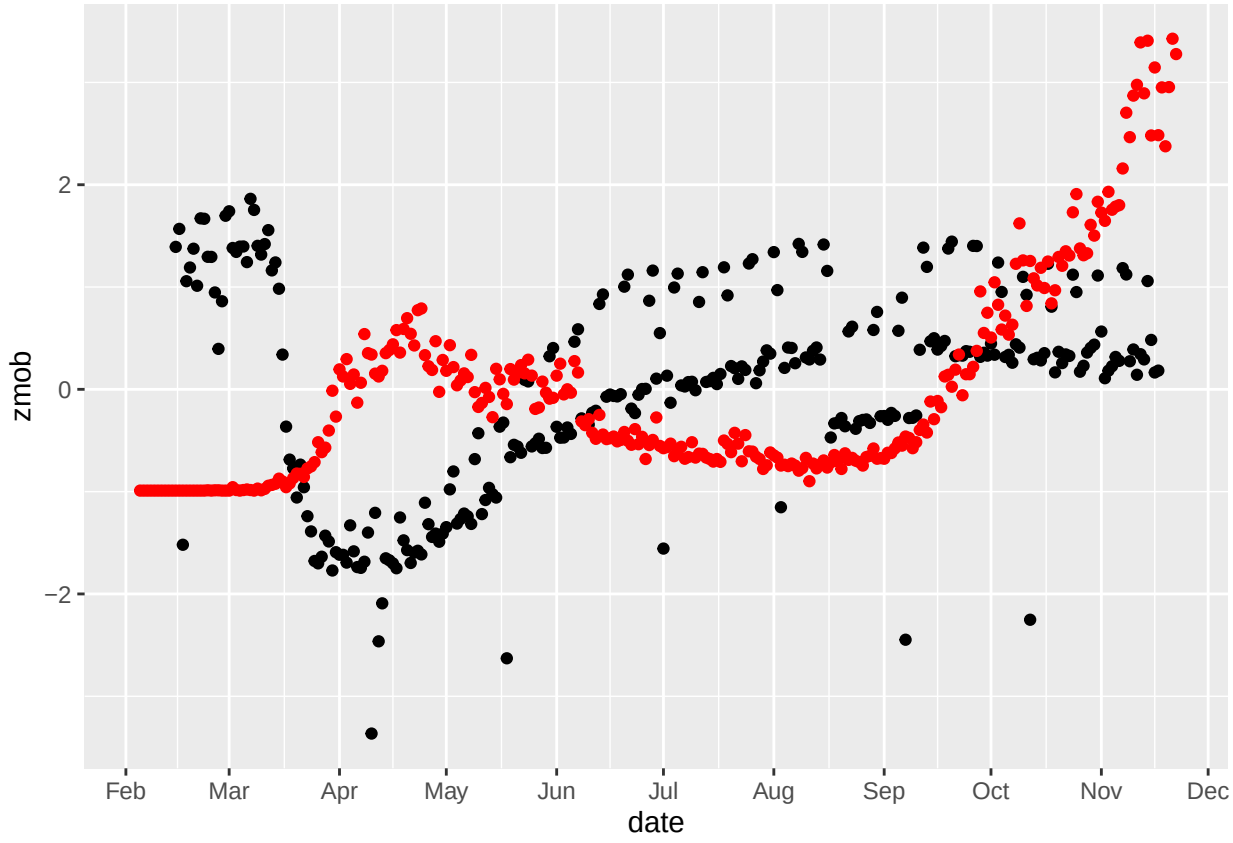
Question: What assumptions are we making? Do we think mobility have the same effect throughout the pandemic?

Adding more mobility flexibilities

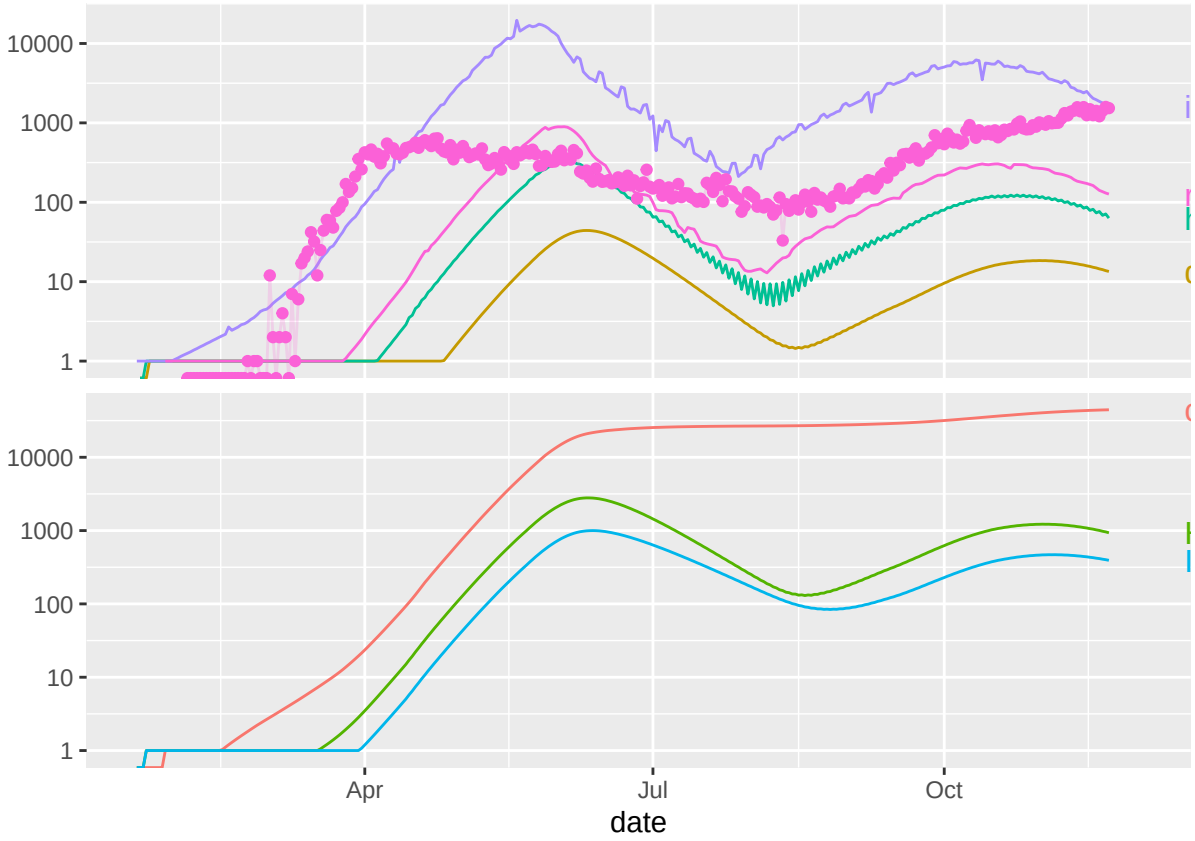
- mob break
- new mob intercept and slope
- smoother on the breakpoint

```
## normalize mobility and reports

clean_mobdat_z <- (clean_mobdat
    %>% mutate(zmob = (rel_activity - mean(rel_activity))/sd(rel_activity))
)

calibrate_dat_z <- (calibrate_dat
    %>% mutate(zreport = (value - mean(value,na.rm=TRUE))/sd(value,na.rm=TRUE))
)

print(ggplot(clean_mobdat_z, aes(x=date,y=zmob))
    + geom_point(color="black")
    + geom_point(data=calibrate_dat_z, aes(x=date,y=zreport), color="red")
```

```
    + scale_x_date(date_breaks = "1 month",date_labels = "%b")
)
```




```
ontmod_mob2 <- calibrate_comb(data = calibrate_dat
    , params = newparams2
    , opt_pars = opt_pars
    , use_DEoptim = FALSE ## We don't want to wait that long
    , debug_plot = FALSE ## TRUE to watch fitting process, don't do it in rmd
    , mob_data = clean_mobdat
    , use_mobility = TRUE
    , mob_breaks = c("2020-03-01", "2020-06-01","2020-08-01")
    , mob_breaks_int = TRUE
    , mob_logist_scale = 3
)


# print(summary(ontmod_mob2))

print(plot(ontmod_mob2,data=calibrate_dat))
```



MLi: The last curve down is probably due to new lockdown/stage restrictions. Maybe add another breakpoint, idk.