

Backcasting with Deconvolution

Eric Marty, Tim Wildauer

Current Outbreak Data

```
library(tidyverse)
library(tibbletime)

source('R/package.R'); source('R/deconcolve.R'); source('R/simple_tdar.R'); source('R/new_package.R');

## also possible to link directly to google sheet. see mosaic::fetchGoogle(), requires publishing sheet
china_province_data_cases <- read.csv("data/china_province_data_cases.csv",stringsAsFactors = FALSE)

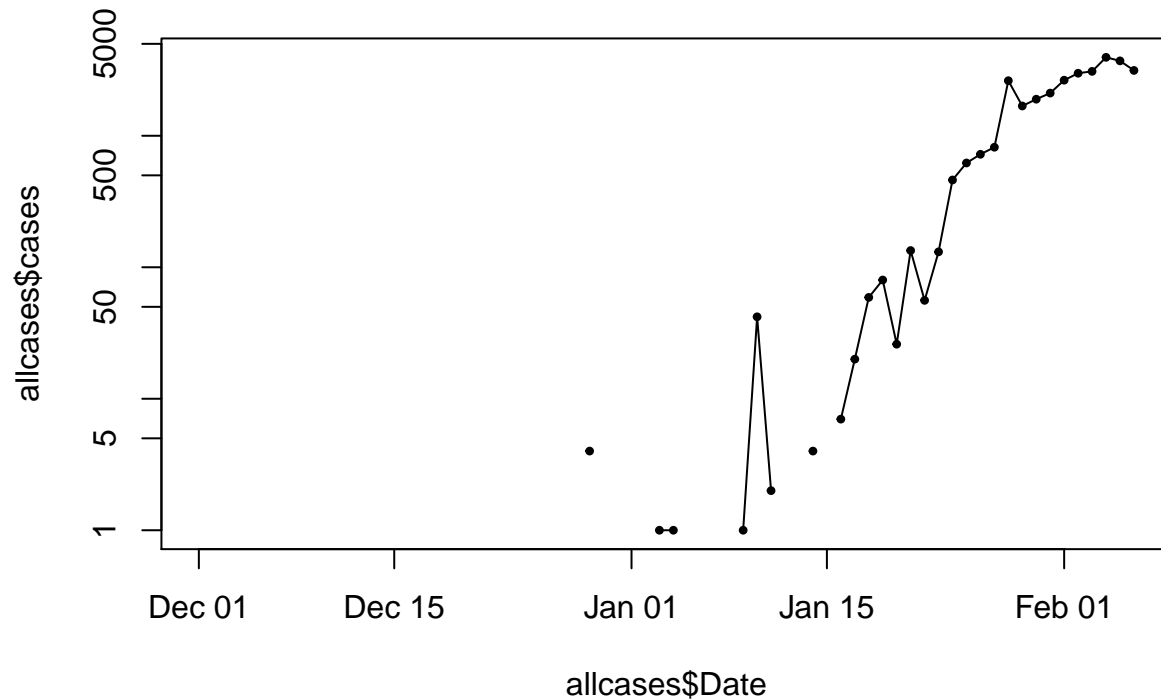
cases <- china_province_data_cases %>%
  select(-Type) %>%
  mutate(Date = parse_date(Date,format="%m-%d-%Y")) %>%
  drop_na(Date) %>%
  tibbletime::as_tbl_time(index=Date) %>%
  mutate(China = rowSums(.,-1],na.rm = TRUE))

china <- cases %>% select(Date, cases = China)

prefix <- data.frame(Date=seq(as.Date("2019-12-01"), as.Date("2019-12-28"), by="days"), cases=rep(0, times=length(Date)))

allcases <- rbind(prefix, china)
parms <- data.frame(mean=rep(0, times=length(allcases[,1])), sd=rep(0, times=length(allcases[,1])))
allcases <- cbind(allcases, parms)

plot(x=allcases$Date,y=allcases$cases,type="o",pch=20,cex=.7,log='y')
```



Deconvolution overview

During the early stages of the outbreak, we were able to estimate the incubation period. Deconvolving the curve with just the incubation period is easy. However, if we use officially confirmed cases, we need to take into account the time between symptom onset and case confirmation. The distribution for this time changes throughout the early stages of the outbreak. Because there are two different time delays, we need two separate deconvolutions. First we will deconvolve the time variable portion, and then the incubation period.

Confirmed Cases to Symptom Onset

There are four different stages of the outbreak. Pre-January 9, January 9 to January 18, January 19 to January 23, and post-January 23. These timeframes are defined by Paige in <https://docs.google.com/document/d/1GpTvsoTZBWphn2tLbKHE5Z1ZE8RuD3caTHar-DqtqHc/edit?pli=1>. The time between onset of symptoms and confirmation is distinct between these time frames.

```
for (row in 1:length(allcases[,1])) {
  if(allcases[row,"Date"] < as.Date("2020-01-09")){
    allcases[row,"mean"] = 6.57
    allcases[row,"sd"] = 4.78
  }
  else if(allcases[row,"Date"] < as.Date("2020-01-19")){
    allcases[row,"mean"] = 4.77
    allcases[row,"sd"] = 3.11
  }
  else if(allcases[row,"Date"] < as.Date("2020-01-24")){
    allcases[row,"mean"] = 0.812
    allcases[row,"sd"] = 1.12
  }
}
```

```

else {
  allcases[row,"mean"] = 0
  allcases[row,"sd"] = 0
}
}

```

We can use a combination of the ridge regression transition matrix and random deconvolution to deconvolve the time-variant convolution. This method has not been tested yet, but it seems to work.

```

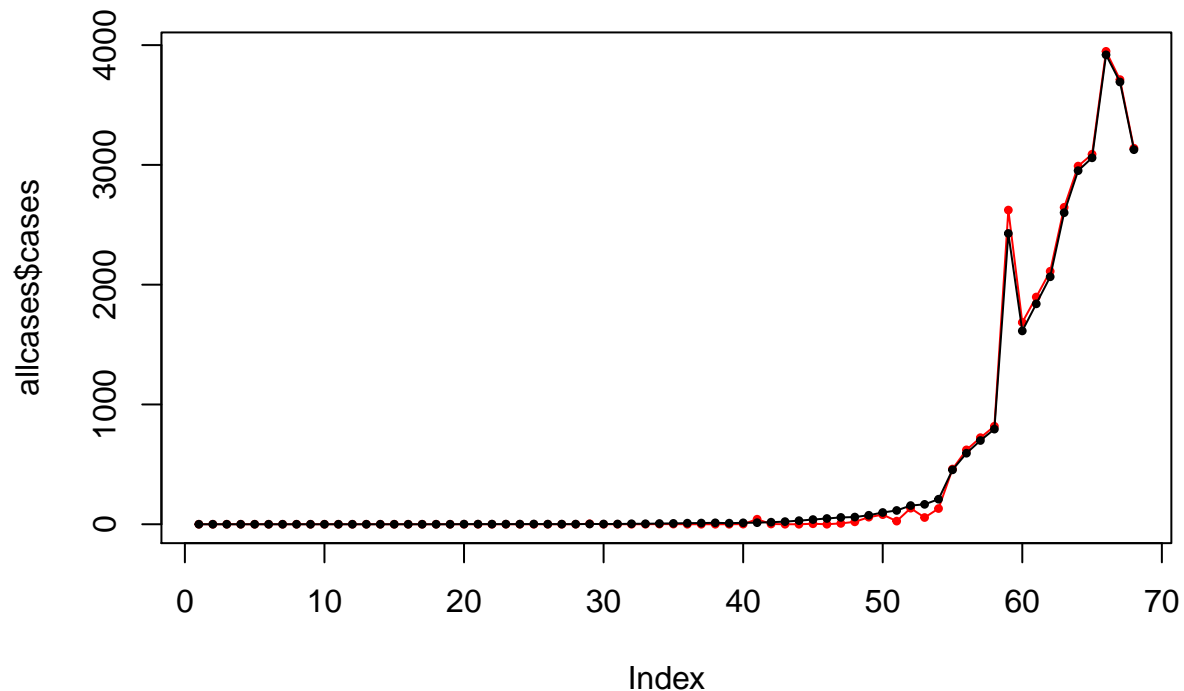
deconvolve_infection_curve_random_tv = function(casemix){
  matrix = incubation_period_distribution_matrix_tv(casemix)
  estimates = rep(0, times=length(matrix[1,]))

  incubation_period = generate_incubation_period
  distribution = "lognorm"
  for (distro in 1:length(matrix[1,])) {
    for (day in 1:length(matrix[,1])) {
      if(casemix$cases[day]!=0 & matrix[day, distro]!=0){
        for (patient in 1:casemix$cases[day]) {
          parms = c(casemix$mean[distro], casemix$sd[distro])
          if(sum(parms)!=0){
            estimated_infection_day = day - incubation_period(distribution, parms = c(casemix$mean[distro], casemix$sd[distro]))
          }
          else{
            estimated_infection_day = day
          }
          if(estimated_infection_day <= 0)
          {
            estimated_infection_day = 1
          }
          estimates[estimated_infection_day] = estimates[estimated_infection_day] + matrix[day, distro]
        }
      }
    }
  }
  return(estimates)
}

estimates = deconvolve_infection_curve_random_tv(allcases)
allcases$estimate = round_infection_curve(allcases$cases, estimates)

plot(allcases$cases,col='RED',type="o",pch=20,cex=.7)
lines(allcases$estimate,type="o",pch=20,cex=.7)

```



Symptom-onset to Infection

Now we have a rough estimate of the number of people beginning to show symptoms each day. Deconvolution to get the number of people infected each day is straightforward.

```
#standard model inputs
shape=6.2
mean=5.6
scale=mean/shape
sd=sqrt(shape*scale^2)

parms <- c(mean,sd)
distribution <- "gamma"

confidence_interval <- function(vector, interval) {
  vector=as.numeric(vector[1,])
  vec_sd <- sd(vector)
  n <- length(vector)
  vec_mean <- mean(vector)
  error <- qt((interval + 1)/2, df = n - 1) * vec_sd / sqrt(n)
  result <- c("lower" = vec_mean - error, "upper" = vec_mean + error)
  return(result)
}

results = deconvolve_single_curve(allcases$estimate, parms)
```

Forecast Prep

Since the deconvolution backs up the symptom-onset curve an average of 5.6 days, the last 8-10 days of the infection curve are not accurate due to the curve being truncated. Any sharp downward trend towards

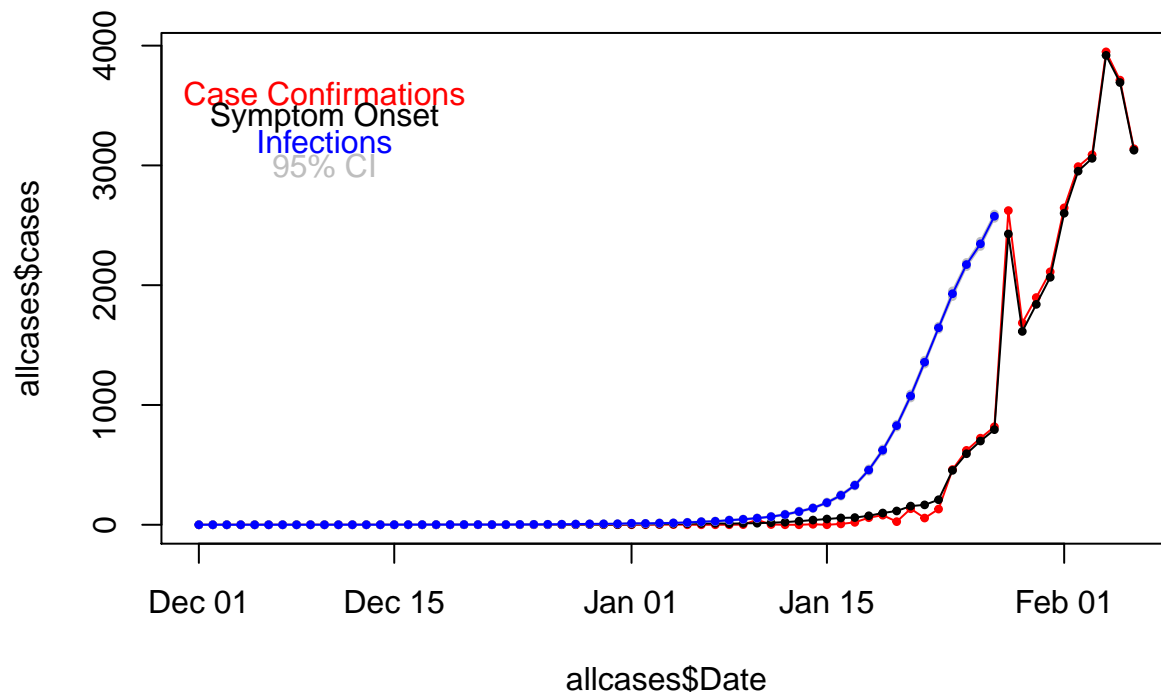
the end of the outbreak is likely due to errors from the truncated outbreak (run during the duration of the outbreak instead at the end looking back). Estimations should be good up until then, but not all the way to the last row of the dataframe.

```
estimates = results$random_estimates
forecast = data.frame(curve = results$random)
forecast$lower = NA
forecast$upper = NA
for (i in 1:length(estimates[,1])) {
  ci = confidence_interval(estimates[i,1:length(estimates[1,])],0.95)
  forecast[i,"lower"] = ci[1]
  forecast[i,"upper"] = ci[2]
}
```

The data frame estimates has 20 estimates of the infection curve (down from 35 to make the CI wider). If you would like to use that, wonderful. If not, the data frame forecast has the curve estimate with a 95% confidence interval.

```
allcases$curve = NA; allcases$curve[1:58] = forecast$curve[1:58]
allcases$upper = NA; allcases$upper[1:58] = forecast$upper[1:58]
allcases$lower = NA; allcases$lower[1:58] = forecast$lower[1:58]

plot(x=allcases$Date,y=allcases$cases,col='RED',type="o",pch=20,cex=.7)
lines(x=allcases$Date,y=allcases$estimate,type="o",pch=20,cex=.7)
lines(x=allcases$Date,y=allcases$upper,type="o",pch=20,cex=.7,col='GRAY')
lines(x=allcases$Date,y=allcases$lower,type="o",pch=20,cex=.7,col='GRAY')
lines(x=allcases$Date,y=allcases$curve,type="o",pch=20,cex=.7,col='BLUE')
text(x=as.Date("2019-12-10"),y=3600,labels="Case Confirmations",col='RED')
text(x=as.Date("2019-12-10"),y=3400,labels="Symptom Onset",col='BLACK')
text(x=as.Date("2019-12-10"),y=3200,labels="Infections",col='BLUE')
text(x=as.Date("2019-12-10"),y=3000,labels="95% CI",col='GRAY')
```



```

plot(x=allcases$Date,y=allcases$cases,col='RED',type="o",pch=20,cex=.7,log = 'y')
lines(x=allcases$Date,y=allcases$estimate,type="o",pch=20,cex=.7)
lines(x=allcases$Date,y=allcases$upper,type="o",pch=20,cex=.7,col='GRAY')
lines(x=allcases$Date,y=allcases$lower,type="o",pch=20,cex=.7,col='GRAY')
lines(x=allcases$Date,y=allcases$curve,type="o",pch=20,cex=.7,col='BLUE')

```

