UCI limitations

Brew, Chaccour, García-Basteiro, others?

2020-03-20

# ## Read first: Implicit non-exclusivity agreement  
#   
# - I (Joe) am working right now only under conditions of non-exclusivity.  
# - This means that if a collaborator (you) is working with me, you understand that I may publish, share, or disseminate any COVID-19 info (other than that which is related to private, personal data) to whomever I deem suitable  
# - The condition goes both ways: the collaborator (you) also can share publicly anything I do with you, or with whomever you deem suitable  
# - The reason for this non-exclusivity pact is principle: the COVID-19 crisis requires extremely fast action, and any delays in publication or dissemination for the purposes of getting an "exclusive" publication, authorship, ownership, recognition, etc. is unacceptable  
# - This non-exclusivity principile applies to everything we work on together, unless explicitly stated otherwise.  
# - If you don't like non-exclusivity (that is, you want me to "keep secret" any ideas we generate or work on together), that's fine. But find somebody else to work with.  
# - Just to be clear: If you are working with me, this means all of our code / data is public and can be shared by any of the collaborators (you and me) with anybody else, at any time.

## This document

This document seeks to answer the question: how soon until Spain runs out of ICU beds?

Click “Code” to see code snippets.

## The question?

How soon until Spain runs out of UCI beds?

## Methods

We define some parameters at the state level, generate a simple log-linear predictive model for daily UCI admissions weighted by days ago, and compare the capacity “ceiling” with prediction.

Here are the parameters:

beds\_public = 3508,  
 beds\_private = 896,  
 normal\_occupancy = 50,  
 need\_hospitalization = 15, # percent which requires hospitalization  
 need\_uci = 5, # percent which requires uci  
 beds = 4404,  
 average\_days\_in\_uci = 10 # total rough estimate

Here is pseudo-code of model definition:

model = log(daily UCI admission) ~ days, weights = 1 / (1 + today - date)

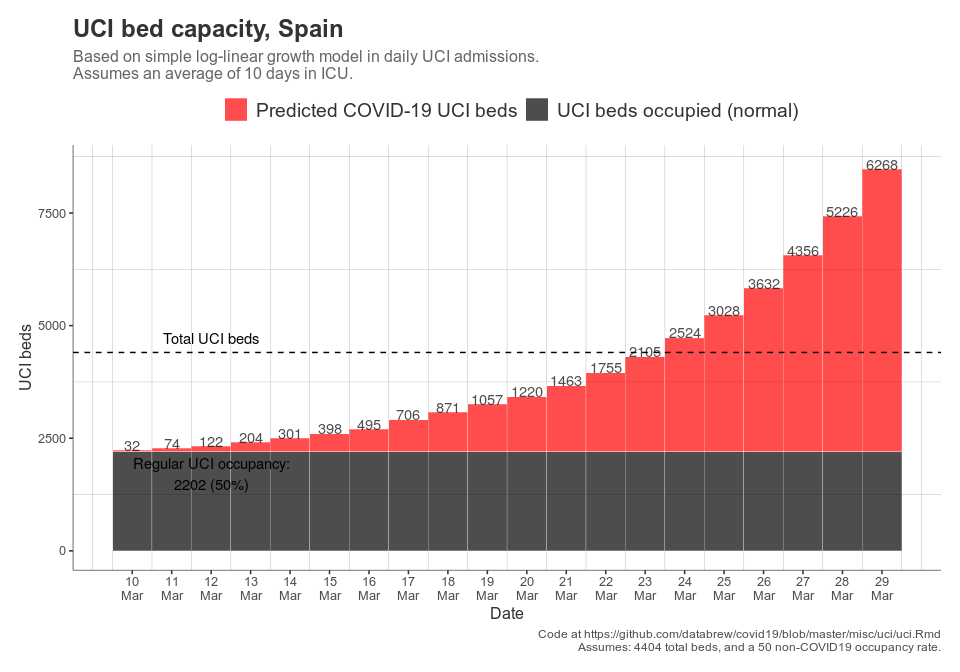
## Code

## Load libraries  
library(covid19) #devtools::install\_github('databrew/covid19')  
library(ggplot2)  
library(lubridate)  
library(dplyr)  
library(ggplot2)  
library(sp)  
library(raster)  
library(viridis)  
library(ggthemes)  
library(sf)  
library(rnaturalearth)  
library(rnaturalearthdata)

# Define parameters  
# Basing off: https://www.niusdiario.es/multimedia/nius-te-explica/colpaso-sistema-sanitario-espana-uci-unidad-cuidados-intensivos-hospitales-coronavirus\_18\_2914020237.html  
p <- list(  
 beds\_public = 3508,  
 beds\_private = 896,  
 normal\_occupancy = 50,  
 need\_hospitalization = 15, # percent which requires hospitalization  
 need\_uci = 5, # percent which requires uci  
 beds = 4404,  
 average\_days\_in\_uci = 10 # total rough estimate  
)  
options(scipen = '999')  
  
  
make\_prediction <- function(data,  
 n\_start = 5,  
 cumulative = FALSE,  
 time\_ahead = 7,  
 var = 'uci'){  
   
 sub\_data <- data  
 # Get the var  
 the\_var <- paste0(var, ifelse(cumulative, '', '\_non\_cum'))  
 sub\_data$var <- as.numeric(unlist(sub\_data[,the\_var]))  
 # narrow down  
 sub\_data <- sub\_data %>%  
 dplyr::select(date, var)  
  
 pd <- sub\_data %>%  
 filter(!is.na(var)) %>%  
 mutate(start\_date = min(date[var >= n\_start])) %>%  
 mutate(days\_since = date - start\_date) %>%  
 filter(days\_since >= 0) %>%  
 mutate(days\_since = as.numeric(days\_since)) %>%  
 mutate(the\_weight = 1/(1 + (as.numeric(max(date) - date))))  
 fit <- lm(log(var) ~ days\_since,  
 weights = the\_weight,  
 data = pd)   
  
   
 # Predict days ahead  
 day0 <- pd$date[pd$days\_since == 0]  
 fake <- tibble(days\_since = seq(0, max(pd$days\_since) + time\_ahead, by = 1))  
 fake <- fake %>%mutate(date = seq(day0, day0+max(fake$days\_since), by = 1))  
 fake <- left\_join(fake, pd %>% dplyr::select(days\_since, var, date))  
 fake$predicted <- exp(predict(fit, newdata = fake))  
 # fake$predictedlo <- predict(fitlo, newdata = fake)  
 ci <- exp(predict(fit, newdata = fake, interval = 'prediction'))  
 # cilo <- predict(fitlo, newdata = fake, interval = 'prediction')  
  
 fake$lwr <- ci[,'lwr']  
 fake$upr <- ci[,'upr']  
 # fake$lwrlo <- ci[,'lwr']  
 # fake$uprlo <- ci[,'upr']  
 # Doubling time  
 dt <- log(2)/fit$coef[2]  
 fake %>% mutate(doubling\_time = dt)  
}  
  
plot\_prediction <- function(data, ylog = F,  
 ci = FALSE){  
 long <- data %>%  
 tidyr::gather(key, value, var:predicted) %>%  
 mutate(key = ifelse(key == 'var', 'Observed', key)) %>%  
 mutate(key = Hmisc::capitalize(key))  
 g <- ggplot()  
 if(ci){  
 g <- g +  
 geom\_ribbon(data = data %>% filter(date > max(long$date[!is.na(long$value) & long$key == 'Observed'])),  
 aes(x = date,  
 ymax = upr,  
 ymin = lwr),  
 alpha =0.6,  
 fill = 'darkorange')  
 }  
 g <- g +  
 geom\_line(data = long,  
 aes(x = date,  
 y = value,  
 group = key,  
 lty = key)) +  
 geom\_bar(data = long %>% filter(key == 'Observed'),  
 stat = 'identity',  
 alpha = 0.6,  
 aes(x = date,  
 y = value)) +  
 theme\_simple() +  
 theme(legend.position = 'right',  
 legend.title = element\_blank())   
 if(ylog){  
 g <- g + scale\_y\_log10()  
 }  
 return(g)  
}  
  
  
spain\_data <-   
 esp\_df %>% group\_by(date) %>%  
 summarise\_at(.vars = vars(uci, deaths, confirmed\_cases,  
 uci\_non\_cum,  
 deaths\_non\_cum,  
 confirmed\_cases\_non\_cum),  
 .fun = function(x){sum(x, na.rm = TRUE)})  
  
pd <- make\_prediction(data = spain\_data,  
 n\_start = 20,  
 cumulative = FALSE,  
 time\_ahead = 10,  
 var = 'uci')  
# Get the number of daily admissions to "spillover" for the number of days they need to be in UCI  
# this function not estimating confidence bounds at this point  
spill\_over <- function(data,  
 days = p$average\_days\_in\_uci){  
 out <- data  
 out$predicted\_spilled\_over <- NA  
 out\_list <- list()  
 for(i in 1:nrow(out)){  
 # Get the sub data for up to days days before  
 sub\_data <- out %>%  
 filter(date >= out$date[i] -( days-1),  
 date <= out$date[i])  
 # Get the sum of ingresado people during that window  
 sum\_ingresado <- sum(sub\_data$var, na.rm = T)  
 # Get the predicted sum too  
 sum\_predicted <- mean(sub\_data$predicted, na.rm = TRUE) \* nrow(sub\_data)  
 # Manually replace with observed  
 out\_predicted <- ifelse(!is.na(sub\_data$var[i]),  
 sum\_ingresado,  
 sum\_predicted)  
 message(i, ": ", round(out\_predicted))  
 # pop back into dataframe  
 # out\_list[[i]] <- out\_predicted  
 out$predicted\_spilled\_over[i] <- out\_predicted  
 }  
 # out <- unlist(out\_list)  
 return(out)  
}  
# preds <- spill\_over(pd)  
pd <- pd %>% spill\_over(days = p$average\_days\_in\_uci)

# Results

already\_occupied <- (p$normal\_occupancy/100) \* p$beds  
total\_beds <- p$beds  
  
# Shape data for plotting  
plot\_data <- pd %>%  
 dplyr::select(date, var, predicted, predicted\_spilled\_over) %>%  
 mutate(already\_occupied = already\_occupied) %>%  
 tidyr::gather(key, value, var:already\_occupied) %>%  
 mutate(key = ifelse(key == 'already\_occupied',  
 'UCI beds occupied (normal)',  
 ifelse(key == 'predicted\_spilled\_over',  
 'Predicted COVID-19 UCI beds',  
 key)))  
  
basic\_data <- plot\_data %>%  
 filter(key %in% c('UCI beds occupied (normal)',  
 'Predicted COVID-19 UCI beds'))  
# basic\_data$key <- factor(basic\_data$key,  
# levels = rev(c('UCI beds occupied (normal)',  
# 'Predicted COVID-19 UCI beds')))  
  
ggplot() +  
 geom\_bar(data = basic\_data,  
 aes(x = date,  
 y = value,  
 fill = key),  
 stat = 'identity',  
 position = position\_stack(),  
 alpha = 0.7, width = 1,  
 color = 'white',  
 lwd = 0.1) +  
 geom\_text(data = basic\_data %>% filter(key == 'Predicted COVID-19 UCI beds'),  
 aes(x = date,  
 y = value + already\_occupied \* 1.05,  
 label = round(value, digits = 0)),  
 # stat = 'identity',  
 position = position\_stack(),  
 alpha = 0.7) +  
 # scale\_y\_log10() +  
 geom\_hline(yintercept = total\_beds,  
 lty = 2) +  
 geom\_text(data = tibble(x = min(plot\_data$date) +2,  
 y = total\_beds + 300,  
 label = 'Total UCI beds'),  
 aes(x = x,  
 y = y,  
 label = label)) +  
 geom\_text(data = tibble(x = min(plot\_data$date) +2,  
 y = already\_occupied - 500,  
 label = paste0('Regular UCI occupancy:\n', round(already\_occupied), ' (', p$normal\_occupancy, '%)')),  
 aes(x = x,  
 y = y,  
 label = label)) +  
 theme\_simple() +  
 theme(legend.position = 'top') +  
 scale\_fill\_manual(name = '',  
 values = c('red', 'black')) +  
 labs(x = 'Date',  
 y = 'UCI beds',  
 title = 'UCI bed capacity, Spain',  
 subtitle = paste0('Based on simple log-linear growth model in daily UCI admissions.\nAssumes an average of ', p$average\_days\_in\_uci, ' days in ICU.'),  
 caption = paste0('Code at https://github.com/databrew/covid19/blob/master/misc/uci/uci.Rmd\nAssumes: ', total\_beds, ' total beds, and a ', p$normal\_occupancy, ' non-COVID19 occupancy rate.')) +  
 scale\_x\_date(breaks = sort(unique(plot\_data$date)),  
 labels = format(sort(unique(plot\_data$date)), '%d\n%b'))



ggsave('~/Desktop/uci.png',  
 height = 7, width = 10)

Table of results:

Note: Predicted including stay-over time means that amount of people in the UCI taking into account the fact that they stay for p$average\_days\_in\_uci days.

pd %>% dplyr::select(date, `Observed daily admissions` = var,   
 `Predicted daily admissions based on model` = predicted,   
 `Predicted including stay-over time` = predicted\_spilled\_over) %>%   
 kable

|  |  |  |  |
| --- | --- | --- | --- |
| date | Observed daily admissions | Predicted daily admissions based on model | Predicted including stay-over time |
| 2020-03-10 | 32 | 39.26825 | 32.000 |
| 2020-03-11 | 42 | 47.10136 | 74.000 |
| 2020-03-12 | 48 | 56.49699 | 122.000 |
| 2020-03-13 | 82 | 67.76682 | 204.000 |
| 2020-03-14 | 97 | 81.28473 | 301.000 |
| 2020-03-15 | 97 | 97.49915 | 398.000 |
| 2020-03-16 | 97 | 116.94796 | 495.000 |
| 2020-03-17 | 211 | 140.27636 | 706.000 |
| 2020-03-18 | 165 | 168.25824 | 871.000 |
| 2020-03-19 | 186 | 201.82185 | 1057.000 |
| 2020-03-20 | NA | 242.08062 | 1219.534 |
| 2020-03-21 | NA | 290.37008 | 1462.803 |
| 2020-03-22 | NA | 348.29217 | 1754.598 |
| 2020-03-23 | NA | 417.76836 | 2104.600 |
| 2020-03-24 | NA | 501.10345 | 2524.418 |
| 2020-03-25 | NA | 601.06194 | 3027.981 |
| 2020-03-26 | NA | 720.95984 | 3631.993 |
| 2020-03-27 | NA | 864.77458 | 4356.491 |
| 2020-03-28 | NA | 1037.27702 | 5225.510 |
| 2020-03-29 | NA | 1244.18969 | 6267.878 |