Corona Analysis

Samuel Knapp - samuel.k@gmx.de

Download data

Data were downloaded from the github repository of the Johns Hopkins University. These are the same data, from which the famous GIS world map is created. See: https://github.com/CSSEGISandData/COVID-19

```
cases <- fread("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse
# turn into long format
datecols <- names(cases)[-c(1:4)]
idcols <- names(cases)[c(1:4)]
cases <- melt(cases,id.vars=idcols,measure.vars=datecols,variable.name="date")
cases$action<-"confirmed"

#add death
death <- fread("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse
datecols <- names(death)[-c(1:4)]
idcols <- names(death)[c(1:4)]
death <- melt(death,id.vars=idcols,measure.vars=datecols,variable.name="date")
death$action<-"death"
# bind
cases <- rbind(cases,death)

################################
# some renaming
setnames(cases,"Country/Region","country")
setnames(cases,"Province/State","province")
setnames(cases,"value","number")

# format date
cases[,date:=as.Date(date,tryFormats = c("%m/%d/%Y"))]
# number of days after first date in table
cases[,days:=as.numeric(date-min(date))]



# as Hong Kong is listed as country China, take out Hong Kong and set as country
cases[province=="Hong Kong",country:="Hong Kong"]


# sum over provinces for China
chinadat <- cases[country=="China"]
chinadat <- chinadat[,.(number=sum(number)),.(date,days,action,country)]
cut <- cases[!country=="China"]
cases <- rbind(cut,chinadat,fill=T)
```

```
# remove cruise ships
cases <- cases[!(country%in%c("Diamond Princess","MS Zaandam"))]

# remove * in Taiwan*
cases[country=="Taiwan*",country:="Taiwan"]

# some countries have outside provinces, mainland is identified by empty province

counts <- c("France","United Kingdom","Denmark","Netherlands","Canada")
for (counti in counts)
{
  changecases <- cases[country==counti& province==""]
  cases <- cases[country!=counti]
  cases <- rbind(cases,changecases)
}


# add population from https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population
pop <- fread("pop.csv")
#unique(cases$country)[!(unique(cases$country) %in% pop$country)]
cases <- merge(cases,pop,by="country",all.x=T)

# check how many countries and population
# contsum <- cases[,.(pop=unique(population)),country]
# nrow(contsum)
# sum(contsum$pop,na.rm=T)
```

Newest date

```
max(cases$date)
```

```
## [1] "20-04-19"
```

Selected countries

```
# number of countries to be plotted
nplot <- 20

# countries that shall definetly be selected
countadd <- c("Germany","Switzerland","Hong Kong","Singapore","Sweden")
# set actioni to confirmed or death
actioni <- "confirmed"

# table with most cases for given action
countover <- cases[action==actioni,.(maxnumber=max(number)),country]
countmost <- countover[order(-maxnumber),country]
# remove chosen countries
countmost <- countmost[!(countmost %in% countadd)]
countsel <- c(countadd,countmost[1:(nplot-length(countadd))])
#countsel
countsel <- sort(countsel)
```

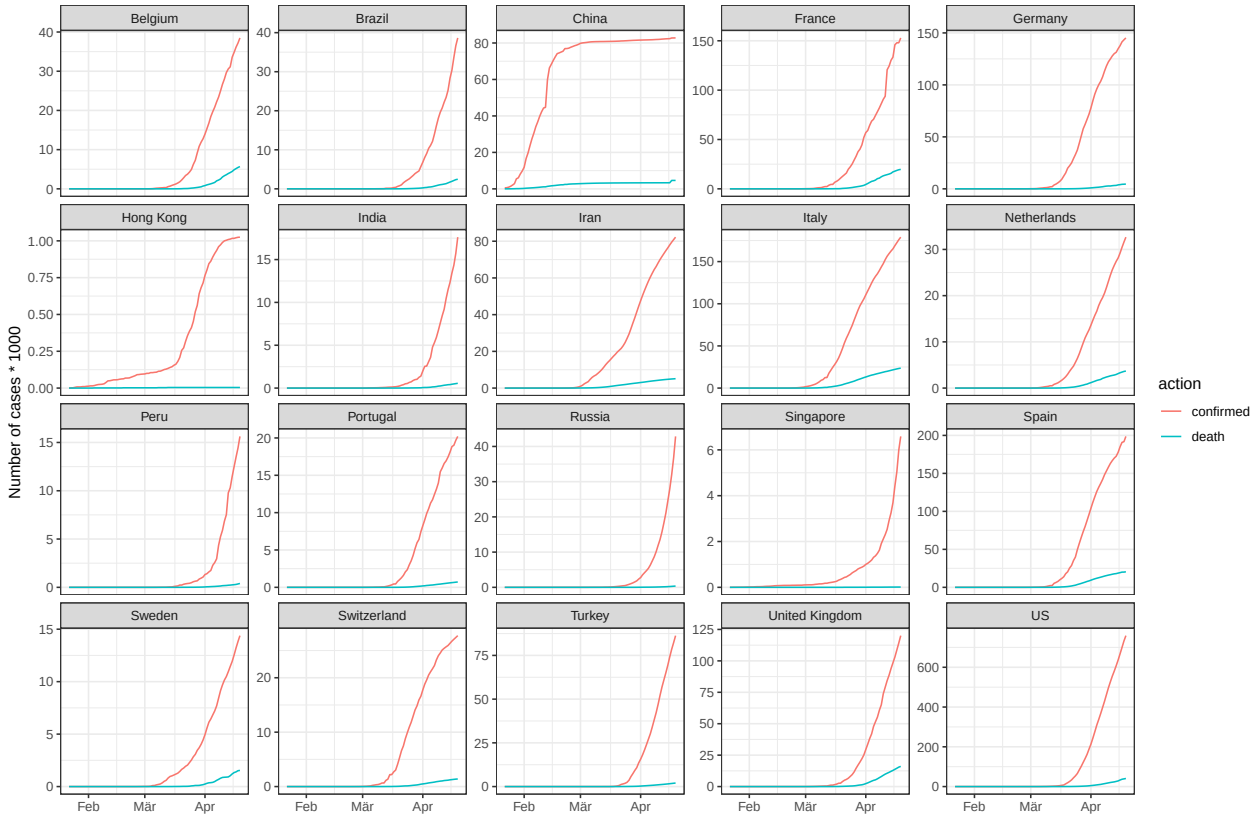
The following countries were set to be included: Germany, Switzerland, Hong Kong, Singapore, Sweden. Additionally, 15 countries with the highest number of confirmed cases were added.

Actual numbers

The number of confirmed and death cases for each day.

Hong Kong and Singapore both show two phases of linear growth. South Korea first had an exponential growth and then turned into linear growth.

```
# # cases per country, both confirmed and death
ggplot(cases[country%in%countsel],aes(date,number/1000,colour=action))+
  facet_wrap(vars(country),scales="free_y")+
  labs(x="",y="Number of cases * 1000")+
  geom_line()+
  theme_bw()
```



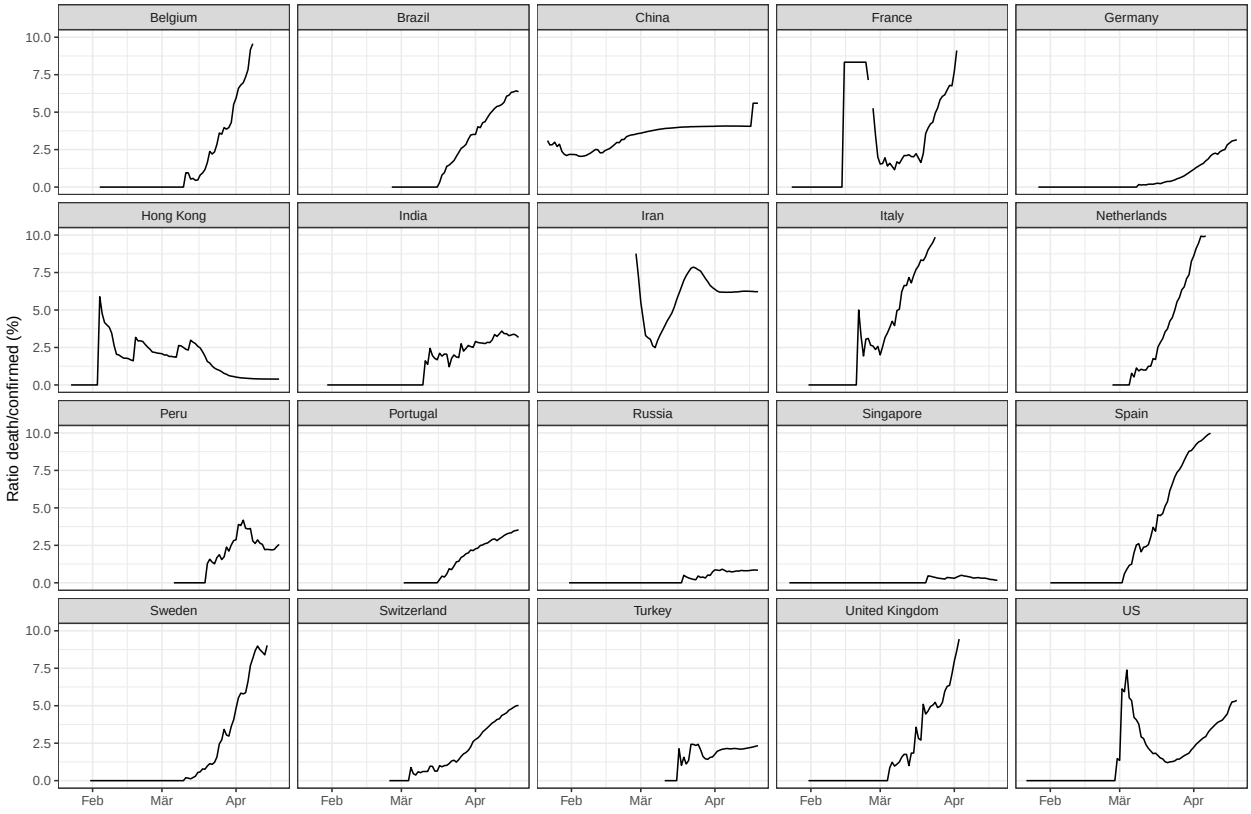
Ratio of death to infected

Simply the ratio of reported deaths divided by number of confirmed cases for each day. Interesting to see that this ratio increases i most countries. A particularly sharp increase can be observed for countries that start to struggle: Italy, Spain, and Belgium. However, this calculation is probably too simple, as it does not take account of recovered cases.

Note, that in Italy last points are not in plot anymore.

```
#
# #################
# # use wide form
# wide form with columns for confirmed and death
casw <- dcast(cases,country+province+Lat+Long+date~action,value.var="number")
```

```
casw[,ratio:=death/confirmed]
ggplot(casw[country%in%countsel],aes(date,ratio*100))+
  facet_wrap(vars(country),#scales="free_y"
             )+
  geom_line()+
  labs(x="",y="Ratio death/confirmed (%)")+
  lims(y=c(0,10))+
  theme_bw()
```




```
#
#
```

New cases

Simply the daily increase of confirmed cases.

```
cases[,newcases:=number-shift(number),country]
cases[,relnewcases:=number/shift(number),country]
# remove ones
cases <- cases[relnewcases!=1]
```

Relative increase per day

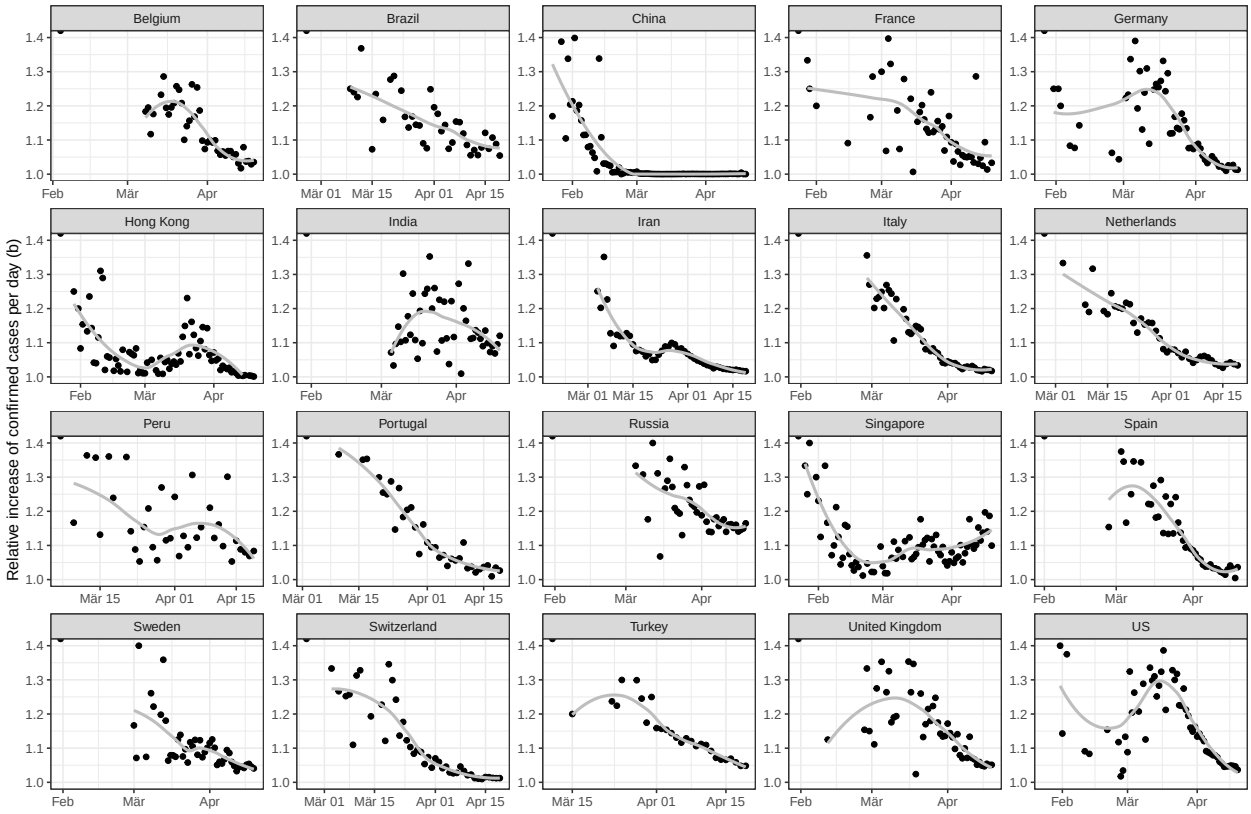
A relative increase of e.g. b=1.2 indicates that the number of confirmed cases increases by 20% in one day, e.g. from 1000 to 1200. This number (b) can be related to the number of days needed for doubling the number of confirmed cases by $b^x = 2$, with x as the number of days. The following shows the relation of b to x. The sometimes mentioned aim of a doubling time of ten days thus corresponds to $b \approx 1.07$.

```r
b=seq(1.05,1.4,0.05)
tab <- data.frame(b=b,
                  NumberOfDays=log(2)/log(b))
kable(tab,digits=2)
```

b	NumberOfDays
1.05	14.21
1.10	7.27
1.15	4.96
1.20	3.80
1.25	3.11
1.30	2.64
1.35	2.31
1.40	2.06

While the relative increase was at around $b \approx 1.3$ to $b \approx 1.4$ (meaning a doubling of confirmed cases every 2 to 2.6 days), this rate has dropped to around $b \approx 1.1$ in most countries. This might be most probably due to the imposed measures.

```r
ggplot(cases[country%in%countsel&action=="confirmed"],aes(date,relnewcases))+
  facet_wrap(vars(country),scales = "free")+
  geom_point()+
  geom_smooth(col="grey",se=F)+
  #geom_smooth(col="grey",se=F,method="lm",formula = y ~ x + I(x^2))+
  #scale_y_continuous(expand = expand_scale(mult = c(0.0001, .2))) +
  labs(x="",y="Relative increase of confirmed cases per day (b)")+
  lims(y=c(1,1.4))+
  theme_bw()
```



Absolute increase

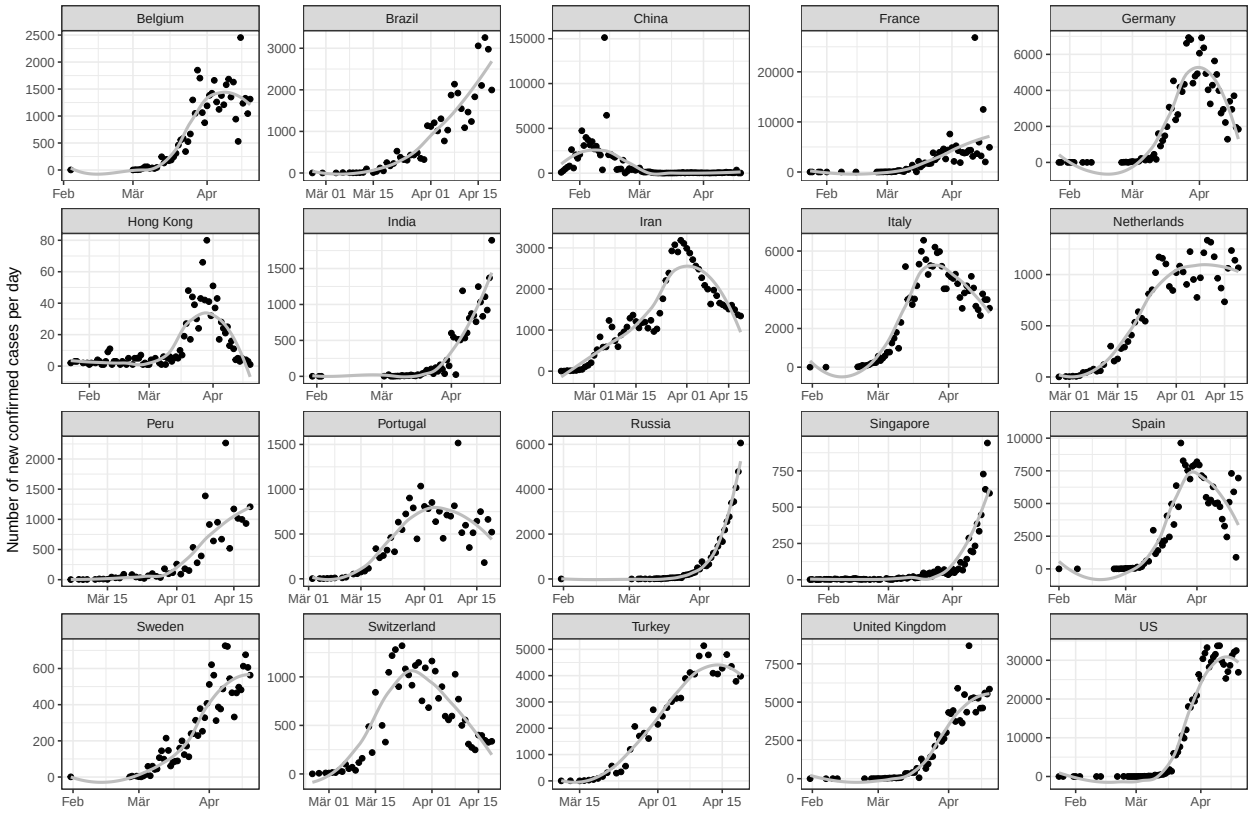
Absolute numbers

For the capacity of the health systems, it is more important to look at the absolute numbers of new confirmed cases. The aim should be to get a constant number of new cases at a niveau which can be handled by the health system.

Austria and Switzerland have managed to drop the increase to a constant level. In many other countries (also Germany) the daily increases are still increasing.

In South Korea it can be nicely see how the exponential growth was lowered to a linear growth. This could/should be the aim...

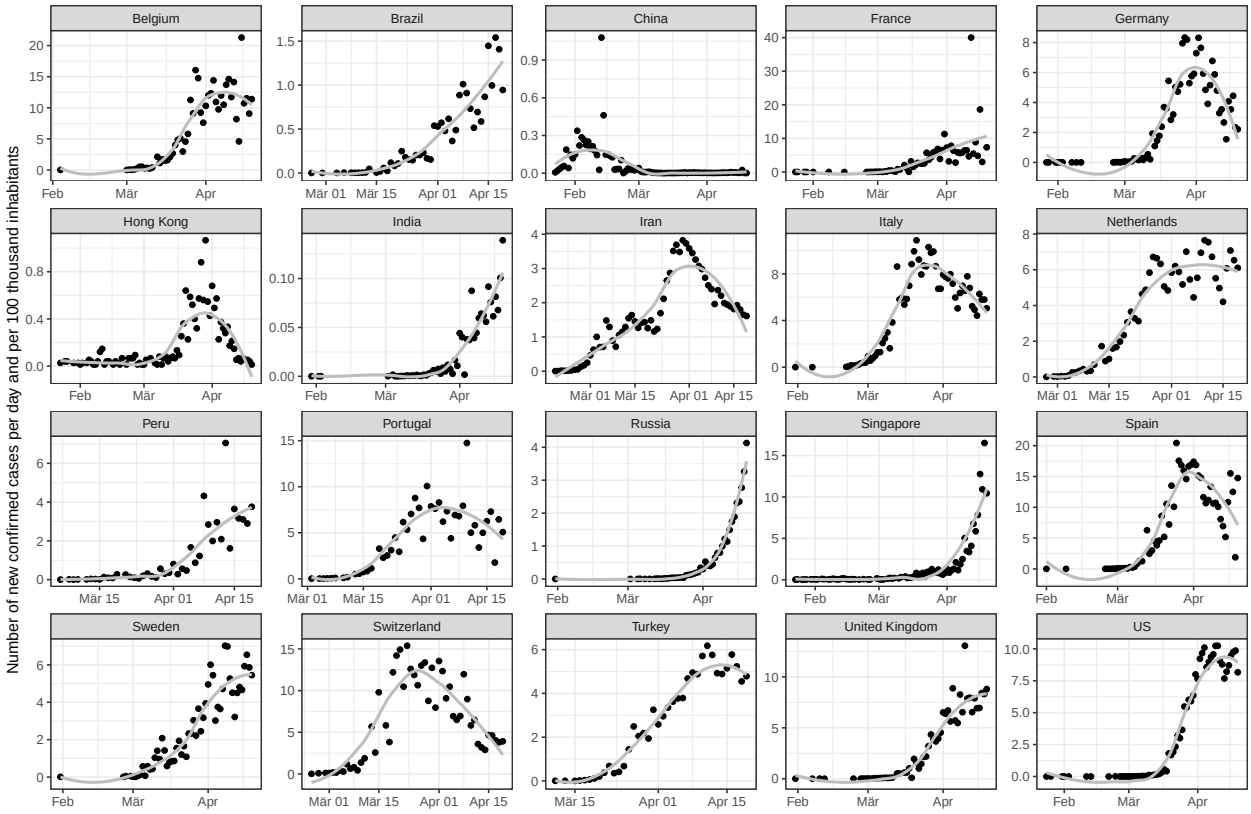
```
ggplot(cases[country%in%countsel&action=="confirmed"],aes(date,newcases))+
  facet_wrap(vars(country),scales = "free")+
  #geom_hline(aes(yintercept=newvent))+
  geom_point()+
  geom_smooth(col="grey",se=F)+
  labs(x="",y="Number of new confirmed cases per day")  +
  #lims(y=c(1,1.5))+
  theme_bw()
```



As number per 100 thousand inhabitants

Relating the absolute number of new cases to the total population per country. All in similar range, but still different. Not sure about the interpretation.

```
cases[,newcases_pop:=newcases/population]
ggplot(cases[country%in%countsel&action=="confirmed"],aes(date,newcases_pop*100000))+
  facet_wrap(vars(country),scales = "free")+
  #geom_hline(aes(yintercept=newvent))+
  geom_point()+
  geom_smooth(col="grey",se=F)+
  #scale_y_continuous(expand = expand_scale(mult = c(0.0001, .2))) +
  labs(x="",y="Number of new confirmed cases per day and per 100 thousand inhabitants")  +
  #lims(y=c(1,1.5))+
  theme_bw()
```



Fit exponential function

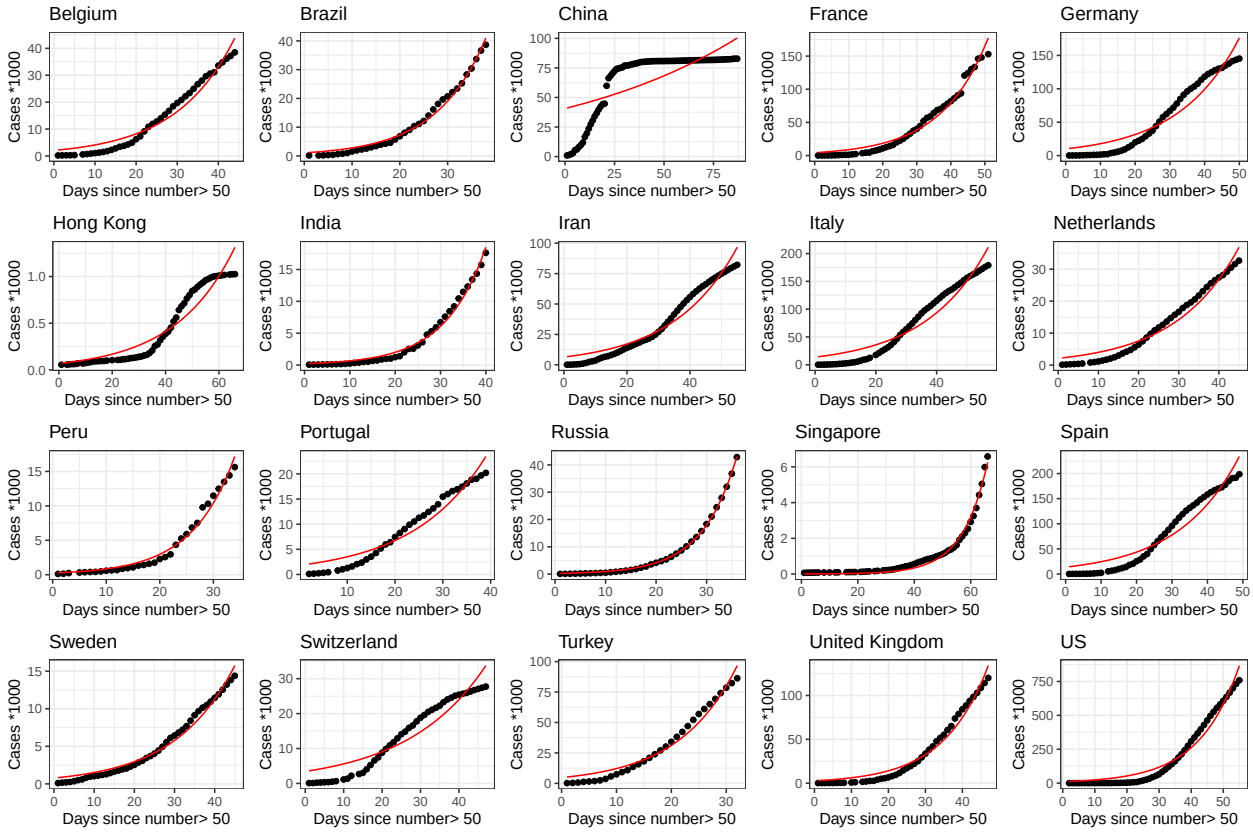
An exponential function ($y = a * b^x$) is fit using only the data from when there were more than 50 confirmed cases per country. While the exponential used to fit very well up to around 1 or 2 weeks ago, they don't fit that well anymore (fortunately!). But in some countries (US), the exponential function still fits very well.

It would be nice now to fit some kind of logistic growth function to determine if and when there was a turning point.

```
# set actioni to confirmed or death
actioni <- "confirmed"
# and start case number
startnumber <- 50



# for each country add days since first case
#cases<-cases[number>100]
cases[,firstday:=min(days[action==actioni&number>startnumber]),country]
cases[,dayfirst:=days-firstday]

par(mfrow=c(4,4))
coltab<-data.frame()
countri <- countsel[1]#"United Kingdom"

i=0
plotcollect<-list()
countsel <- sort(countsel)
```

```
for(countri in countsel){
  countsub <- cases[country==countri&action==actioni&dayfirst>0]

# exp-models
# e0: e^(bx), start b=1
# e1: e^(a+bx), start a=0 and b=1, b around 0.3, best fit
# e2: a*e^(bx), start a=1 and b=1, equi to e1
# e3: a+e^(bx), start a=0 and b=1, but doesn't fit so well

#   SS<-getInitial(number~SSexp(dayfirst,b,y0),
#                  data=countsub)
#   b <- SS["b"]
#   y0 <- SS["y0"]
# model <- nls(number ~ y0*10^(b*dayfirst),
#              data = countsub,
#              start = list(y0=y0,b=b))


# ^x models
# 0: b^x, start b=2
# 1: b^(a+x), start a=1,b=2, fits also goot, b around 1.4, b is exp() of b in e1
# 2: a*b^x, start a=1,b=2, same b estimated as in 1
# 3: a+b^x, start a=0, b=1, b around 1.5
model <- nls(number ~ a*b^(dayfirst),
             data = countsub,
             start = list(a=startnumber,b=1.2))
             #start = list(a=N0_start,b=exp(R_start)))
#
# ### collect coefficients and model stats
coltab[countri,"a"] <- coefficients(model)[1]
coltab[countri,"b"] <- coefficients(model)[2]
# residual standard error
coltab[countri,"RSE"] <- summary(model)$sigma
coltab[countri,"maxnumber"] <- max(countsub$number)
#coltab[countri,"number10"] <- countsub[dayfirst==10,number]
coltab[countri,"days800"] <- countsub[number>800,min(dayfirst)]
coltab[countri,"days1600"] <- countsub[number>1600,min(dayfirst)]


# add predicted to countsub
countsub[,predicted:=predict(model,data=list(dayfirst=countsub$dayfirst))]

#### plot
i=i+1
plotcollect[[i]]<-
  ggplot(countsub,aes(dayfirst,number/1000))+
  geom_point()+
  geom_line(aes(y=predicted/1000),col="red")+
  labs(title=countri,x=paste("Days since number>",startnumber),
       y="Cases *1000")+
  theme_bw()
```

```
# observed
# plot(number~dayfirst,countsub,main=countri,
#      xlab=paste("Days since number>",startnumber))#,xlim=c(0,15),ylim=c(0,2000))
# # predicted as line
# lines(countsub$dayfirst,predict(model,data=list(dayfirst=countsub$dayfirst)),col="red")

}


#arrangeGrob(grobs = plotcollect, ncol = 4)
ggarrange(plotlist=plotcollect,ncol=5,nrow=4)
```




```
#coltab$ratio <- coltab$RSE/coltab$maxnumber

# calculate double time from number of days to increase from 100 to 800, resp. 1600
coltab$doubtime800 <- coltab$days800^(1/3)
coltab$doubtime1600 <- coltab$days1600^(1/4)
coltab$doubtime_b <- log(2)/log(coltab$b)
coltab$days800to1600<-coltab$days1600-coltab$days800
#coltab
# coltab$country<-rownames(coltab)
# setDT(coltab)
# coltab[order(b)]

# par(mfrow=c(1,1))
# hist(coltab$b)
# plot(density(coltab$b))
```

Percentage of population

Number of confirmed cases (most recent day) divided by the total population.

```
ratab <- cases[action=="confirmed",.(RatioPercent=max(number)/max(population)*100),.(country)]
kable(head(ratab[order(-RatioPercent)],50),digits=3)
```

country	RatioPercent
San Marino	1.373
Holy See	1.001
Andorra	0.919
Luxembourg	0.578
Iceland	0.486
Spain	0.422
Belgium	0.334
Switzerland	0.323
Ireland	0.310
Italy	0.297
Monaco	0.245
US	0.230
France	0.228
Liechtenstein	0.209
Qatar	0.198
Portugal	0.197
Netherlands	0.187
United Kingdom	0.181
Germany	0.175
Austria	0.166
Israel	0.147
Sweden	0.139
Norway	0.132
Denmark	0.127
Bahrain	0.122
Singapore	0.116
Estonia	0.115
Turkey	0.104
Panama	0.101
Iran	0.099
Moldova	0.092
Serbia	0.091
Cyprus	0.088
Malta	0.087
Djibouti	0.078
United Arab Emirates	0.069
Finland	0.068
Slovenia	0.064
Czechia	0.063
North Macedonia	0.058
Ecuador	0.054
Chile	0.053
Belarus	0.051
Montenegro	0.049
Peru	0.049
Lithuania	0.046

country	RatioPercent
Croatia	0.046
Dominican Republic	0.045
Romania	0.045
Armenia	0.044