Media & Data Science

Lecturer: Changjun Lee
changjunlee@hanyang.ac.kr

College of Communication
Media Informatics

HYU 한양대학교 HANYANG UNIVERSITY

Create a new project

*.Rproj
*.R

getwd()

Create a variable

<-

c()

Object? Variable?

rm()

1:10

sample(1:10, 10)

X[2:5]

{Base} functions

mean()

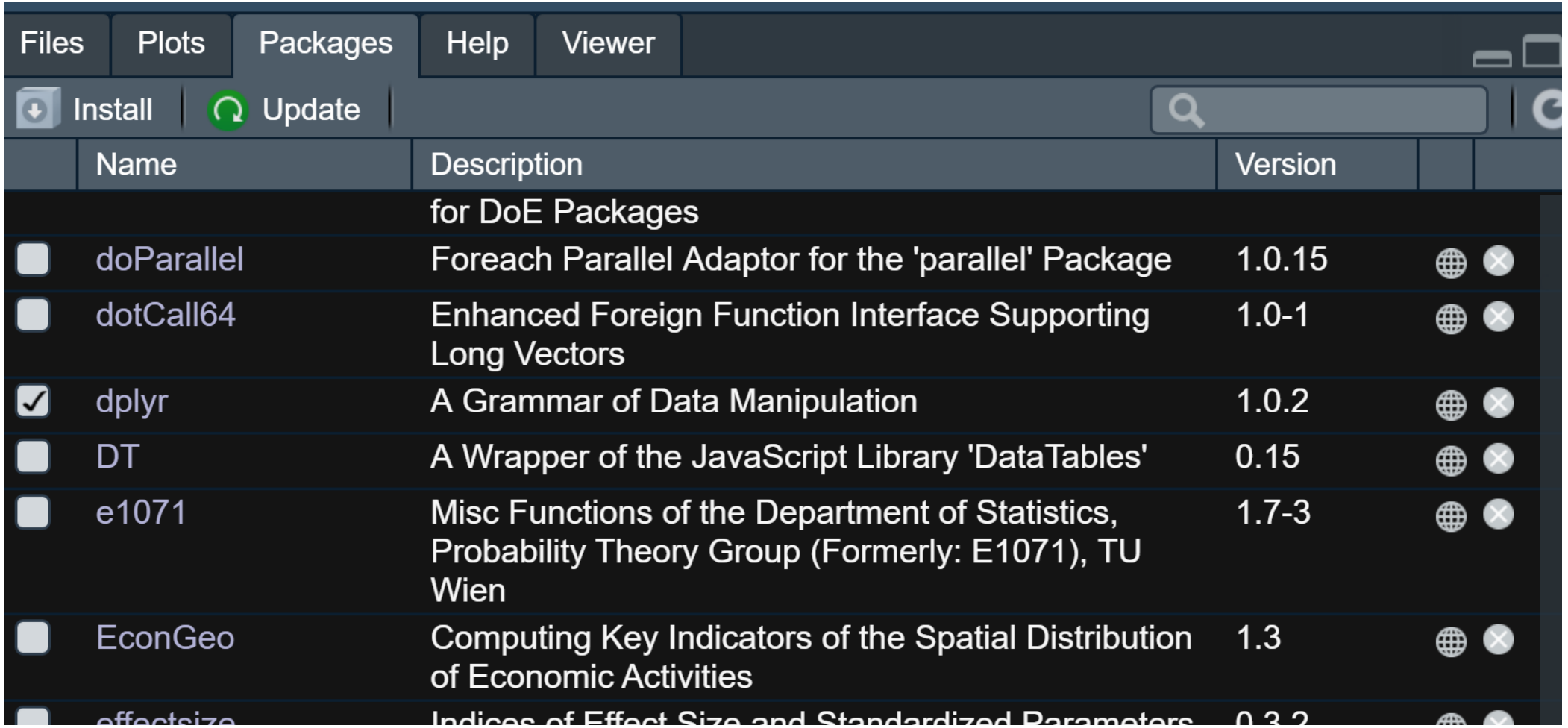
sd()

[

:

c()

Check your libraries



Function conflict

```
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

dplyr::filter
stats::filter

지난 시간 복습

(Basic syntax)

Choose two answers if its type cannot be 'factor' variable in R *

- [ ] GPA (Grade Point Average)
- [ ] Blood type
- [ ] Grade (A, B, C, D, F)
- [ ] Height
- [ ] Gender

Choose a line if it cannot be run in R *

- X <- 1
- X = 1
- Y <- X +1
- X+Y <- Z
- Z = X + T

Let X is a vector of {1,2,3,4,5,6,7,8,9,10}. If we run the sentence X ( ) 2 == 1. The result is {TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE}. Which operator can be suitable in ( ) ? *

- ○ +
- ○ &
- ○ %%
- ○ $
- ○ /

Choose the right codes below to print out (1) 1 2 3 1 2 3 1 2 3 (2) 1 1 1 2 2 2 3 3 3 *

- (1) rep(c(1,2,3), times=3) / (2) rep(c(1,2,3), each=3)
- (1) rep(c(1,2,3), each=3) / (2) rep(c(1,2,3), times=3)
- (1) rep(c(1,2,3), 3) / (2) rep(c(1,2,3), times=3)
- (1) rep(c(1,2,3), each=3) / (2) rep(c(1,2,3), 3)

To create a vector

```
> x<-vector(length=3)
> x
[1] FALSE FALSE FALSE
> x[2]=1
> x
[1] 0 1 0
```

- X<- c(1:5)
- c(1:5) * 2
- c(1:5) * 2 – 1
- Using seq() rep()

Vector Indexing

- Vector1[vector2]

```
> X<-1:10 * 2
> X
 [1]  2  4  6  8 10 12 14 16 18 20
> X[3:6]
[1]  6  8 10 12
>
```

- X[-1]
- X[c(2,2,5,7)]

Vector Operation

```
> x<-c(1:10)
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> all(x < 5)
[1] FALSE
> any(x < 5)
[1] TRUE
```

```
> x<5
 [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

Vector Filtering

Vector[condition]

```
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> x[x < 5]
[1] 1 2 3 4
```

Quiz

(Vector)

What will the following code return?

```
MyVector <- c(12, 456, 34.5, 23, 55, "34hello")
typeof(MyVector)
```

1. integer
2. double
3. character
4. FALSE

Which of these functions is NOT used to create vectors?

1. c()
2. typeof()
3. seq()
4. rep()

What is the name of the function used to install packages from the internet?

1. rnorm()
2. paste()
3. install.packages()
4. library

Create the vector below by using ‘seq’ function

2.0 2.5 3.0 3.5 4.0 4.5 5.0

Ans: seq(2, 5, 0.5) # or seq(from=2, to=5, by=0.5)

Create the vector below by using ‘rep’ function

3 3 3 3 3 3 3 3 3

Ans: rep(3, 9)

Create the vector below by using ‘rep’ function

80 20 80 20 80 20 80 20

Ans: x <- c(80,20)
rep(x,4)

Are these vectors possible forms in R?

“1”	“2”	“a”	“b”	“c”	“4”	“5”	“6”	“9”	“z”

“1”	“2”	“a”	“b”	“c”	“4”	5	6	“9”	“z”

1	100	1	20	20	29	128	23	78	2387

You have a vector:
mountain <- c("tree", "rock", "dirt", "dolphin", "waterfall")
How would you access the word "dolphin" in this vector?

1. c[4]
2. mountain[-2]
3. mountain[4]
4. mountain(4)

How to extract 3rd and 5th values from the vector below?

X <-

“1”	**“2”**	**“a”**	**“b”**	**“c”**	**“4”**	**“5”**	**“6”**	**“9”**	**“z”**

X[c(3,5)]

From x vector

X <-

“1”	**“2”**	**“a”**	**“b”**	**“c”**	**“4”**	**“5”**	**“6”**	**“9”**	**“z”**

How to extract vectors like below?

“2”	**“a”**	**“b”**	**“c”**	**“5”**	**“6”**	**“9”**	**“z”**

You have two vectors:
number.of.unit <- c(10, 12, 10, 9)
price.per.unit <- c(200, 250, 230, 190)
How would you calculate the *total.price.unit* vector in which each element is the product of the corresponding elements of the two given vectors?

Y is a vector of a sequence from 1 to 5000

Y <- 1:5000

Write an R code for counting the number of values in Y that are divisible by 26 (0 remains)

#hint: [], %%, length

Modify Vector by using filtering

```
x
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
x[x >= 10] <- 10
x
```

```
##  [1]  2  4  6  8 10 10 10 10 10 10
```

X<-c(1, 2, 3, 4, NA, NA, 7, 8, NA, 10)

Let’s convert all NA values to 0

Modify Vector by using filtering

There is a vector X below

X <- sample(1:30, 30)

Modify all the values > 24 to 24

Matrix

```
> x<-cbind(1:4, 12:15)
> x
     [,1] [,2]
[1,]    1   12
[2,]    2   13
[3,]    3   14
[4,]    4   15
> dim(x)
[1] 4 2
```

```
> x<-matrix(1:10, ncol=2)
> x
     [,1] [,2]
[1,]    1    6
[2,]    2    7
[3,]    3    8
[4,]    4    9
[5,]    5   10
> x<-matrix(1:10, ncol=2, byrow=T)
> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
[5,]    9   10
```

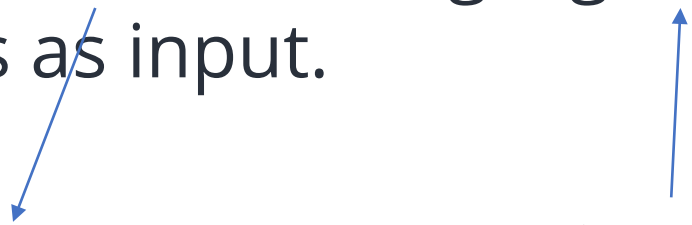
Practice

(Array)

Create **a blank** matrix with **10** rows and **5** columns

```
m = matrix(, nrow = 10, ncol = 5)
```

Write an R program to create a matrix taking a given vector of numbers as input.



```
c(1:16)
```

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

```
M = matrix(c(1:16), nrow = 4, byrow = TRUE)
```

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Write an R program to access the element from the matrix left

1. Access the element at 3rd column and 2nd row
2. Access only the 3rd row
3. Access only the 4th column

Run the code below and create a matrix M

```
row_names = c("row1", "row2", "row3", "row4")
col_names = c("col1", "col2", "col3", "col4")
M = matrix(c(1:16), nrow = 4, byrow = TRUE, dimnames = list(row_names, col_names))
```

Write an R program to extract a TRUE FALSE matrix
if the matrix M's value > 7 and show the result.

Run the code below and create a matrix M

```
row_names = c("row1", "row2", "row3", "row4")
col_names = c("col1", "col2", "col3", "col4")
M = matrix(c(1:16), nrow = 4, byrow = TRUE, dimnames = list(row_names, col_names))
```

Write an R program to convert a matrix to a 1 dimensional array.

1 dimensional array (column wise):

```
[1] 1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16
```

```
as.vector(M)
```

1 dimensional array (column wise):

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

```
as.vector(t(M))
```

Vectorized codes

```
c(1, 2, 4) + c(2, 3, 5)
```

```
X<-c(1,2,4,5)
X * 2
```

Recycling rule

```
1:4 + c(1, 2)

X<-c(1,2,4,5)
X * 2

1:4 + 1:3
```

Practice with the real-world data

Tidying the new Johns Hopkins Covid-19 time-series datasets

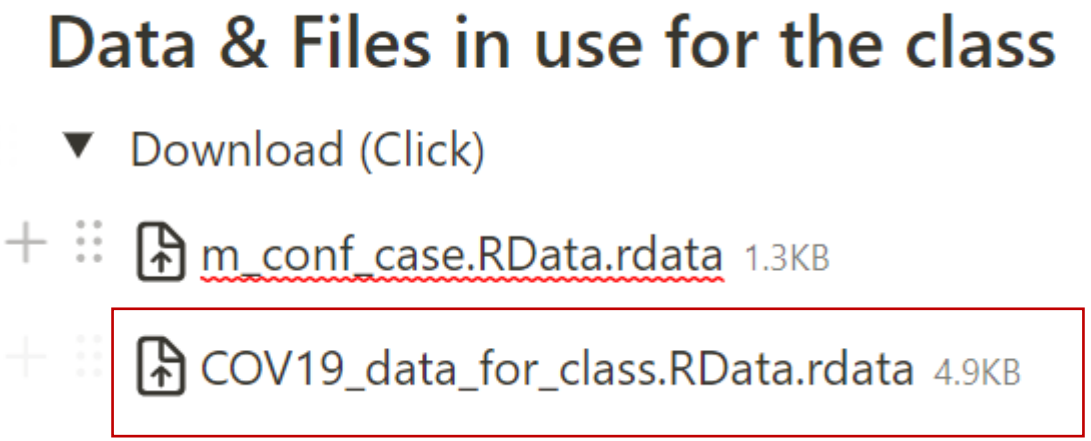
March 23, 2020

By An Accounting and Data Science Nerd's Corner

[This article was first published on **An Accounting and Data Science Nerd's Corner**, and kindly contributed to R-bloggers]. (You can report issue about the content on this page here)

my old blog post about tidying Johns Hopkins CSSE

Download the data for class



Matrix 1: Confirmed cases

	2020-02-01	…	2020-10-01	…	2022-09-01
China	11891	…	90567	…	2510703
Italy					
Japan					
Korea					
Spain					
UK					
US	8	…	7279272	…	94665567

7 rows and 32 columns

Matrix 2: Death cases

	2020-02-01	…	2020-10-01	…	2022-09-01
China	18	…		…	
Italy					
Japan					
Korea					
Spain					
UK					
US	0	…		…	

Access to the matrix

1. UK’s total confirmed cases on 2022-09-01
2. South Korea’s total confirmed cases on 2022-09-01
3. China’s total confirmed cases on 2022-09-01
4. South Korea’s increasing confirmed cases on 2022-09-01 compared to the previous month
5. Japan’s increasing confirmed cases on 2022-09-01 compared to the previous month

Vector 1: GDP

```
country.name<-c("China","Italy","Japan","Korea","Spain","UK","US")
GDP<-c(12237700479375,
1943835376342,
4872415104315,
1530750923149,
1314314164402,
2637866340434,
19485394000000)
names(GDP)<-country.name
```

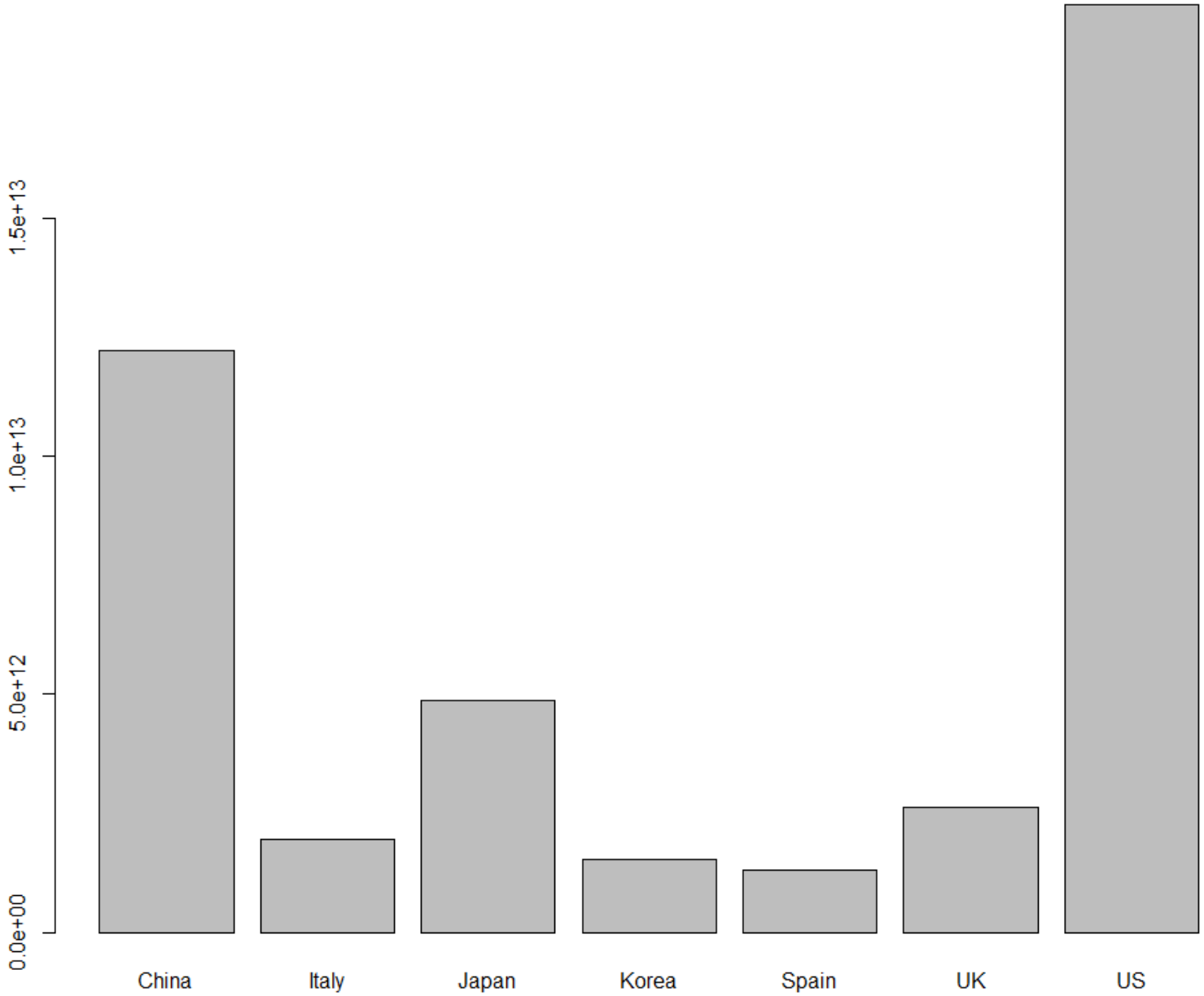
Vector 2: Population

```
pop<-c(1439323776, 60461826, 126476461, 51269185, 46754778,
67886011, 331002651)
names(pop)<-country.name
```

Vector 3: Population Density

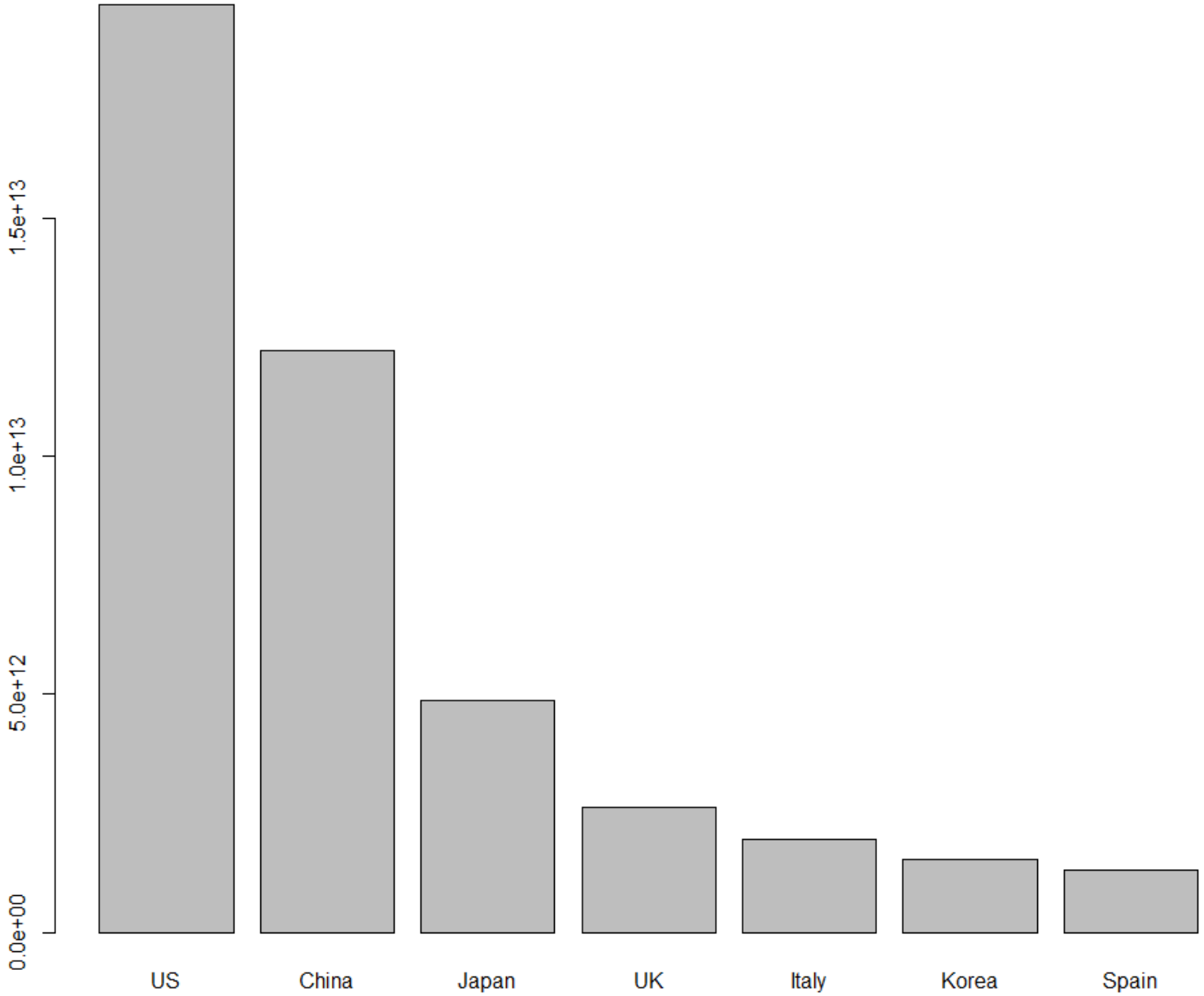
```
country.name<-c("China","Italy","Japan","Korea","Spain","UK","US")
pop.density<-c(148, 205, 347, 530, 94, 275, 36)
names(pop.density)<-country.name
```

barplot(GDP)



barplot(sort(GDP))

barplot(sort(GDP, decreasing = T))

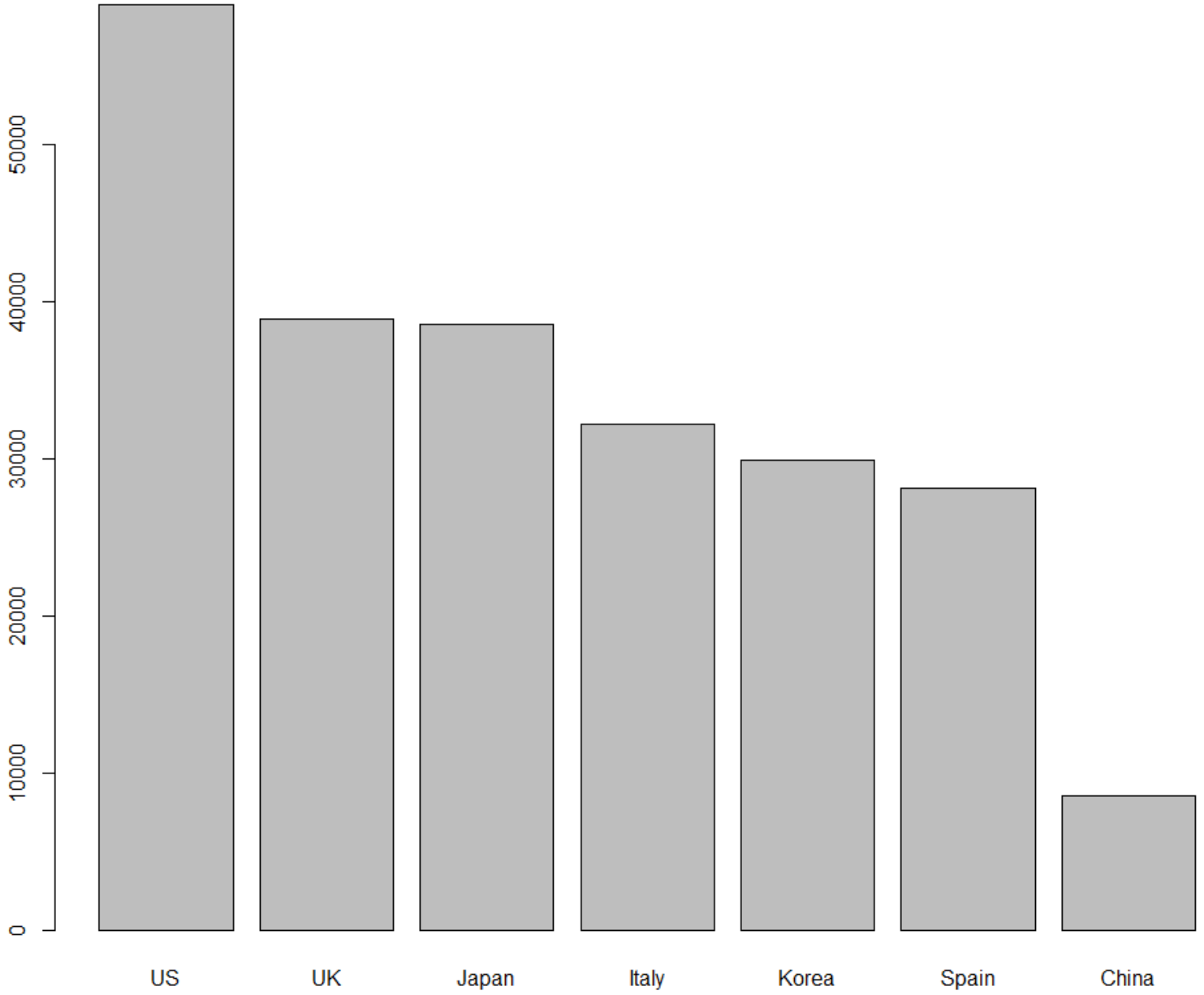


5 mins

Bar graph of GDP per capita

We have
1. GDP vector
2. Population vector

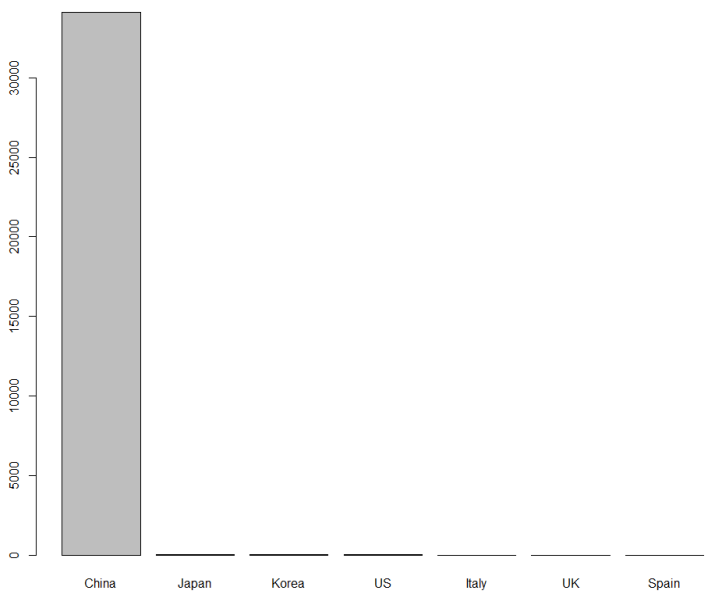
We know
1. barplot()
2. Vector calculation
3. Sort()
4. Decreasing=T option



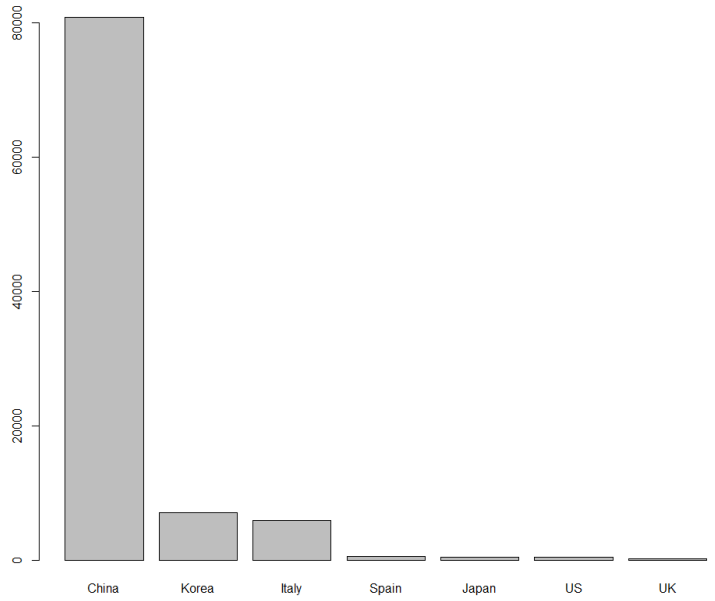
10 mins

Confirmed cases bar graph

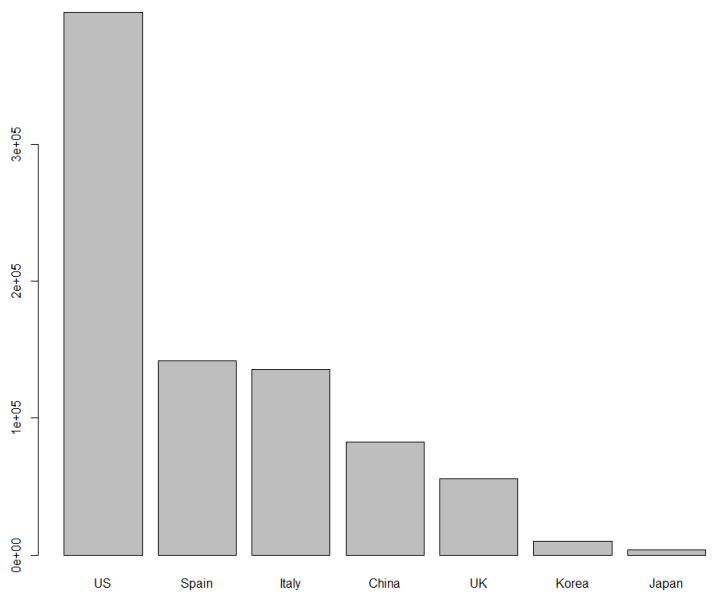
2020-03-01



2020-05-01



2021-10-01



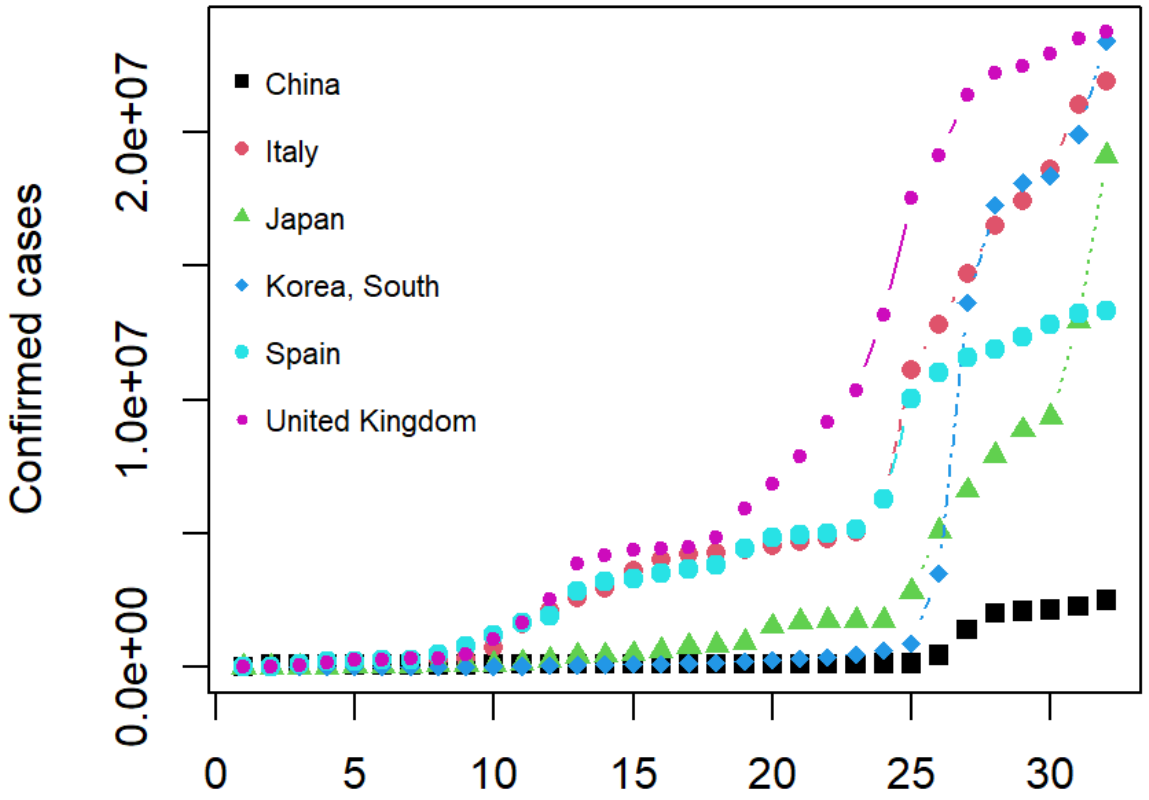
See which columns indicate the dates above (2020-03-01 / 2020-05-01 / 2021-10-01) in the matrix

- Mat.name[,]

Use barplot() , sort() , decreasing=T option

matplot

	2020-01-23	…	2020-02-15	…	2020-04-07
China	643	…	64813	…	82718
Italy					
Japan					
Korea					
Spain					
UK					
US	1	…	13	…	396223



matplot

First, try matplot(m.conf.case)

You must transpose the matrix

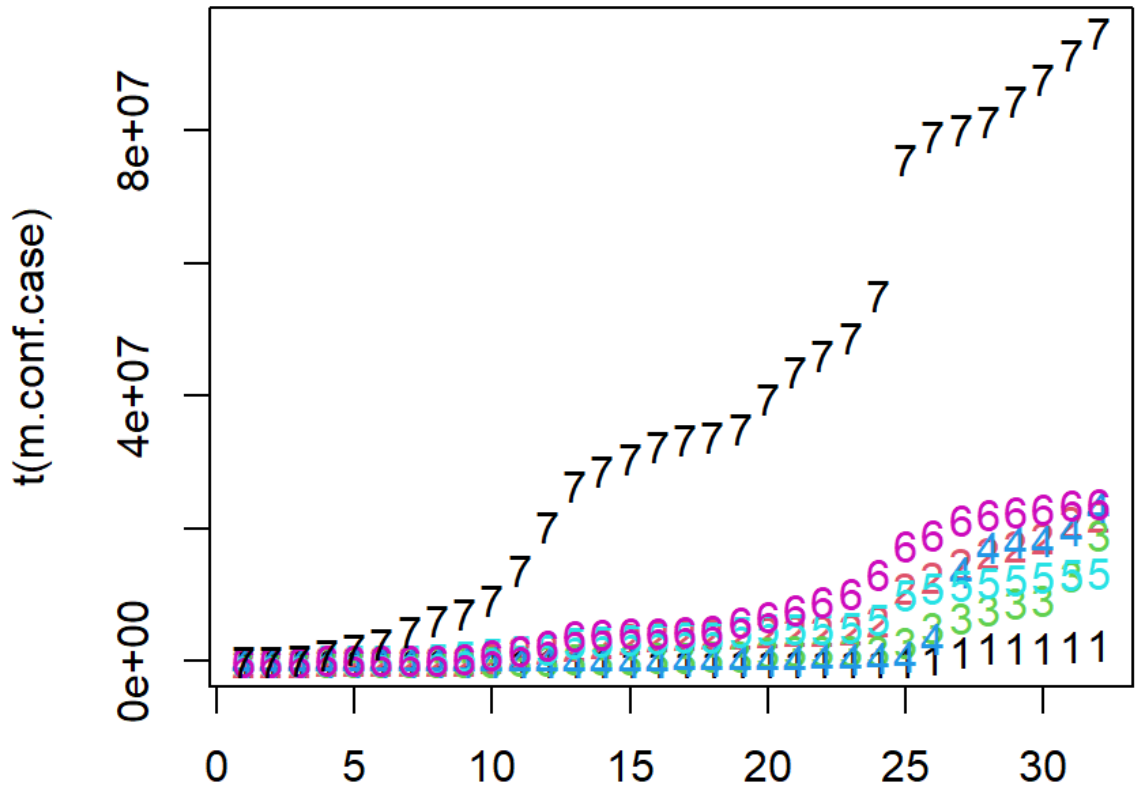
t(m.conf.case)

	2020-01-23	…	2020-02-15	…	2020-04-07
China	643	…	64813	…	82718
Italy					
Japan					
Korea					
Spain					
UK					
US	1	…	13	…	396223

	China	…	Korea	…	US
2020-01-23	643	…	64813	…	82718
…					
…					
…					
…					
…					
2020-04-07	1	…	13	…	396223

matplot

Let's try matplot(t(m.conf.case))



matplot

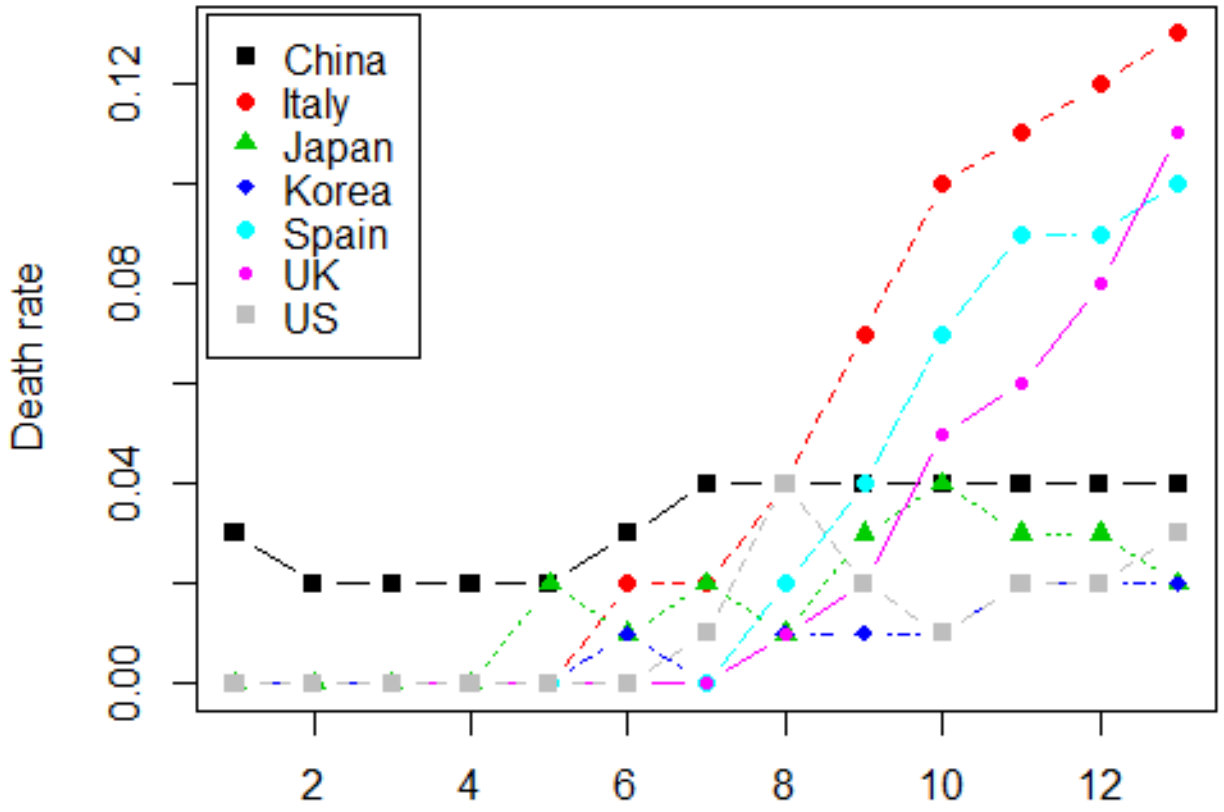
This is the cascade

```
matplot(t(m.conf.case))

matplot(t(m.conf.case), type='b')

matplot(t(m.conf.case), type='b', pch=15:20)

matplot(t(m.conf.case), type='b', pch=15:20, col=c(1:6, 8),
        ylab="Confirmed cases")

legend("topleft", inset=0.01, legend=country.name, pch=15:20,
col=c(1:6, 8), horiz=F)
```

10 mins

matplot

Try the same graph but now use the death rate

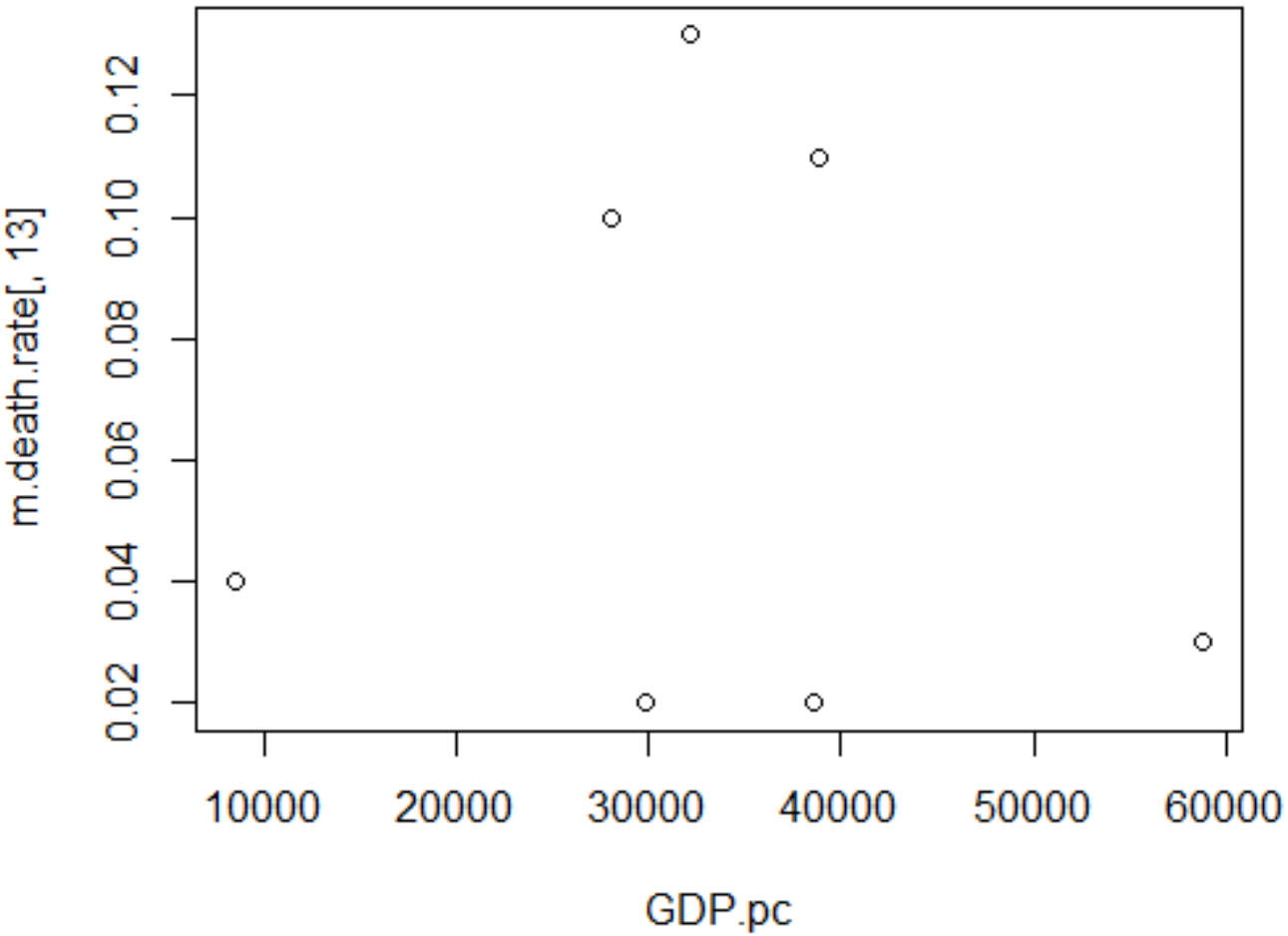


County’s wealth and COVID19

I’m now curious about the relationship between countries’ GDP per capita and the death rate at the latest time

plot

plot(GDP.pc, m.death.rate[,16])

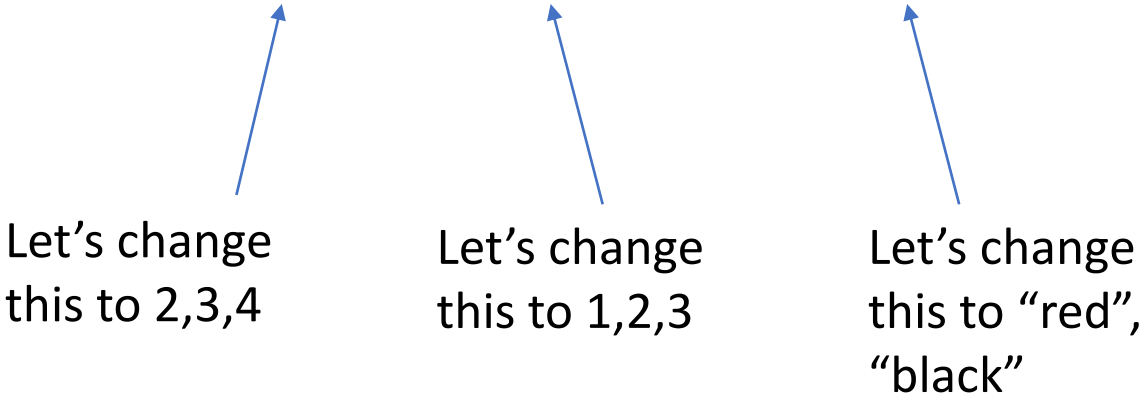


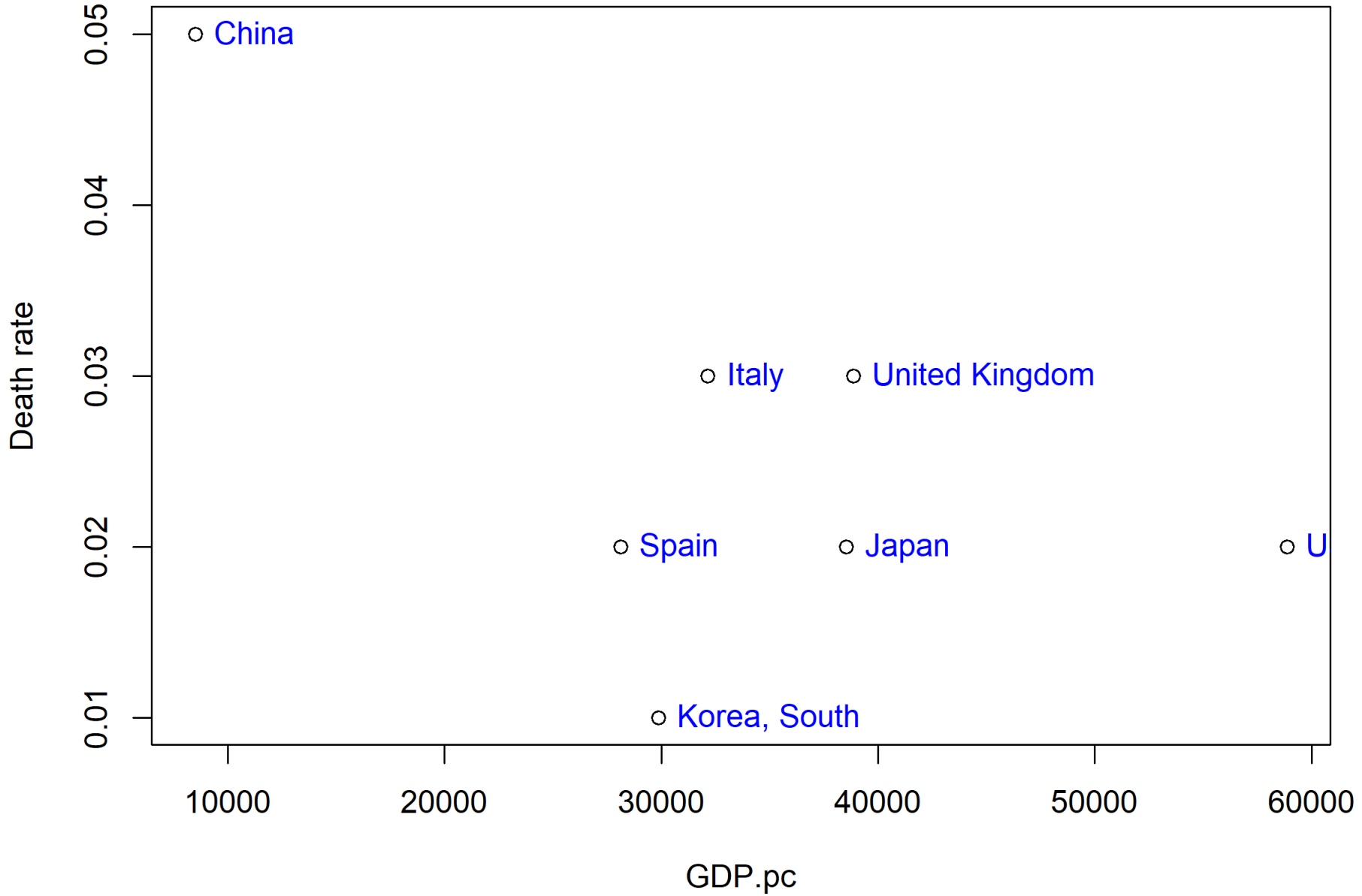
plot

```
plot(GDP.pc, m.death.rate[,32])
```

```
plot(GDP.pc, m.death.rate[,32], ylab="Death rate")
```

```
text(GDP.pc, m.death.rate[,32],  row.names(m.death.rate),
cex=1, pos=4, col="blue")
```




China
Italy
United Kingdom
Spain
Japan
U
Korea, South
Death rate
0.01
0.02
0.03
0.04
0.05
10000
20000
30000
40000
50000
60000
GDP.pc

10 mins

County’s pop density and COVID19

Let’s explore the relationship between countries’ population density and the death rate at the latest time

Increasing rate!

When you wonder if the number of confirmed cases is in increasing phase or flattened..

Increasing rate!

	2020-02-01	…	2020-10-01	…	2022-09-01
China					
Italy					
Japan					
Korea					
Spain					
UK					
US					

m1<-m.conf.case[,-1]

	2020-02-01	…	2020-10-01	…	2022-09-01
China					
Italy					
Japan					
Korea					
Spain					
UK					
US					

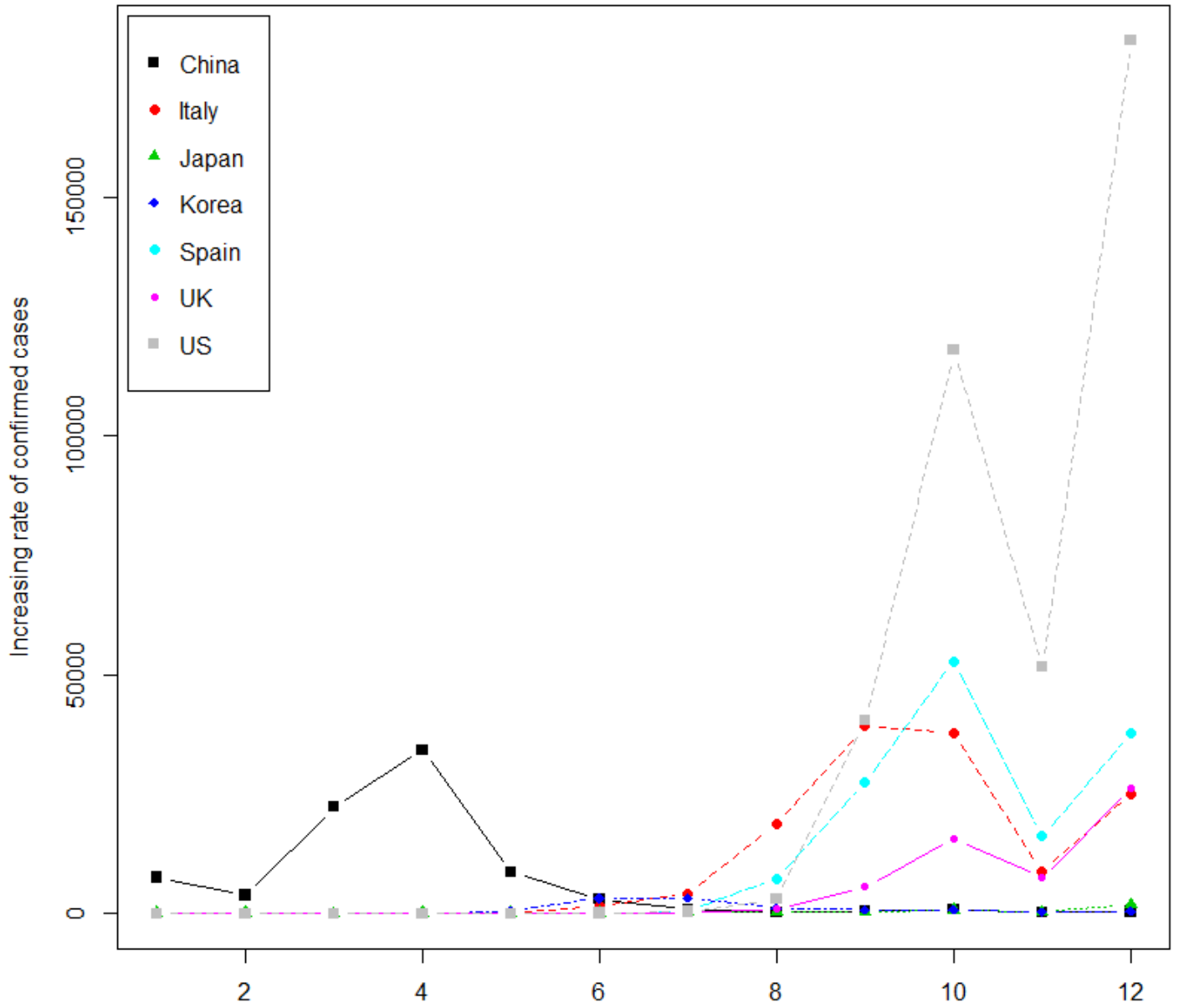
m2<-m.conf.case[,-32]

m3<-m1-m2

Increasing rate of confirmed cases

Let’s visualize like an example →

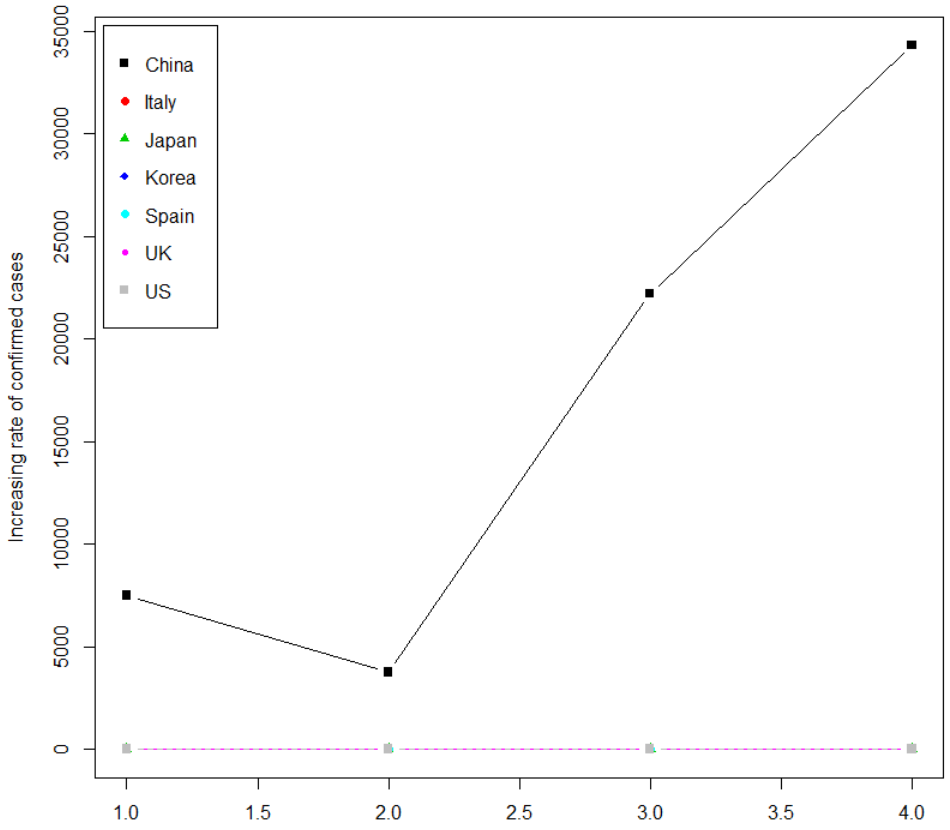
5 mins

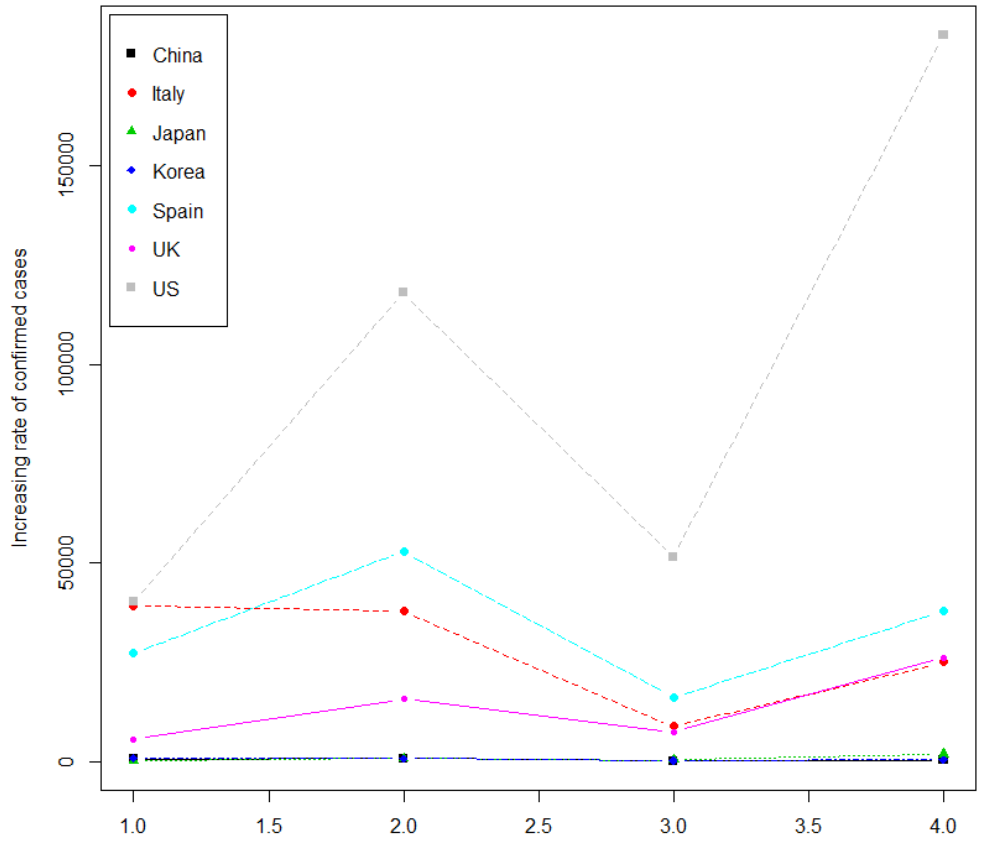


5 mins

Increasing rate of confirmed cases

Let’s also visualize the first four periods and the last (recent) four periods



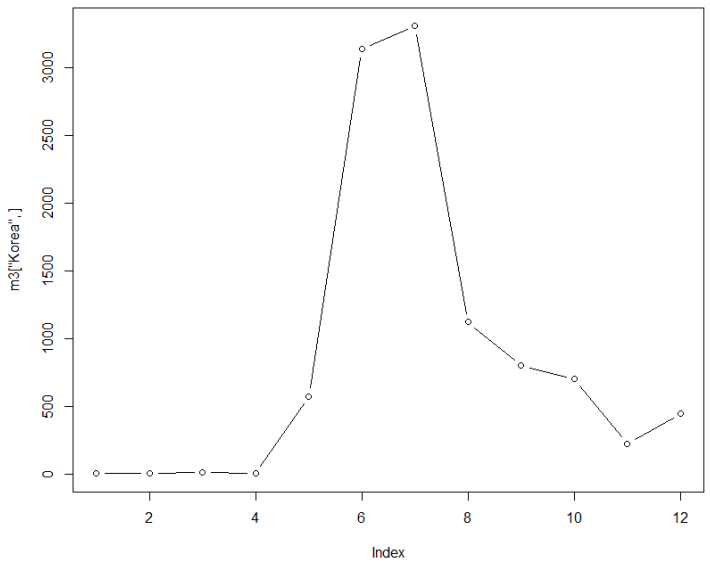


10 mins

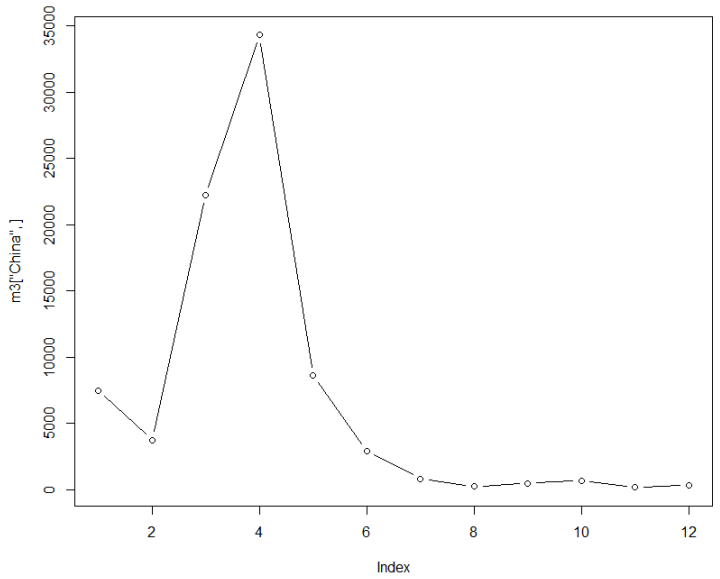
Increasing rate of confirmed cases

Can you also do for the specific countries like Korea, China, and Japan?

Korea



China



Japan