In [3]:

```
library(dplyr)
library(reshape2)
```

In [4]:

```
# I analyzed the same datasets used in CA4.
# Data are related to the Covid pandemic and I tried to understand if the trend of c
ases/deaths has been impacted by the
# different categories of measures implemented by each country government.
# I used four different dataset: 1- Government measures dataset provided by ACAPS;
 2/3- Time series of confirmed
# cases/deaths provided by CSSE (John Hopkins); 4-Mapping file of country/region pro
vided by https://ourworldindata.org/
```

In [5]:

```
# Load the measures file and create a dataframe
measures <- read.csv("C:\\Users\\Romina\\Desktop\\DBSCourse\\ProgrammingBigData\\CA5
\\acaps_covid19_government_measures_dataset.csv", sep = ";", header = TRUE)

df_measures <- data.frame(measures)
```

In [6]:

```
# Let's have a look at the dataframe structure
str(df_measures)
```

```
'data.frame':	11200 obs. of  17 variables:
 $ ID                : int  1 2 3 4 5 6 8 10 12 17 ...
 $ COUNTRY           : Factor w/ 196 levels "Afghanistan",..: 1 1 1 1 1
1 1 1 1 1 ...
 $ ISO               : Factor w/ 193 levels "AFG","AGO","ALB",..: 1 1 1
1 1 1 1 1 1 1 ...
 $ ADMIN_LEVEL_NAME  : Factor w/ 595 levels ""," "," Al-Qatif ",..: 1 24
6 1 1 1 1 221 221 1 1 ...
 $ PCODE             : logi  NA NA NA NA NA NA ...
 $ REGION            : Factor w/ 6 levels "Africa","Americas",..: 3 3 3
3 3 3 3 3 3 3 ...
 $ LOG_TYPE          : Factor w/ 2 levels "Introduction / extension of m
easures",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ CATEGORY          : Factor w/ 6 levels "Governance and socio-economic
measures",..: 5 5 5 1 6 4 6 6 5 1 ...
 $ MEASURE           : Factor w/ 41 levels "Additional health/documents
requirements upon arrival",..: 18 21 3 14 24 5 34 24 37 22 ...
 $ TARGETED_POP_GROUP: Factor w/ 3 levels "No","No ","Yes": 1 1 1 1 1 1
1 1 1 1 ...
 $ COMMENTS          : Factor w/ 10877 levels "","-Hairdressing salons/b
arber shops/ beauty salons (as of May 21)",..: 1 1 1 1 6536 2549 1 1317
1 4204 ...
 $ NON_COMPLIANCE    : Factor w/ 13 levels "","Arrest/Detention",..: 1 1
1 1 1 1 1 1 1 1 ...
 $ DATE_IMPLEMENTED  : Factor w/ 152 levels "","01/01/2020",..: 61 61 61
61 62 117 4 4 4 4 ...
 $ SOURCE            : Factor w/ 1230 levels ""," Foreign Affairs, Forei
gn Trade and Development Cooperation",..: 671 671 671 671 8 457 837 837
837 1138 ...
 $ SOURCE_TYPE       : Factor w/ 8 levels "","Government",..: 2 2 2 2 3
3 8 8 8 2 ...
 $ LINK              : Factor w/ 6633 levels "","Canada Emergency Respon
se Benefit for individuals and families ",..: 2142 2142 2142 2142 2925 6
185 2429 2429 2430 1132 ...
 $ ENTRY_DATE        : Factor w/ 63 levels "01/04/2020","01/05/2020",..:
24 24 24 24 24 24 24 24 24 24 ...
```

In [7]:

```
# There are 11200 observations with 17 variables
# Let's have a look at some data
head(df_measures, 5)
```

ID	COUNTRY	ISO	ADMIN_LEVEL_NAME	PCODE	REGION	LOG_TYPE	CATEGORY	MEAS
1	Afghanistan	AFG		NA	Asia	Introduction / extension of measures	Public health measures	H screening airports bo cross
2	Afghanistan	AFG	Kabul	NA	Asia	Introduction / extension of measures	Public health measures	Isolation quarai pol
3	Afghanistan	AFG		NA	Asia	Introduction / extension of measures	Public health measures	Awarer campa
4	Afghanistan	AFG		NA	Asia	Introduction / extension of measures	Governance and socio-economic measures	Emerg administr struct activate establis
5	Afghanistan	AFG		NA	Asia	Introduction / extension of measures	Social distancing	Limit p gather

In [8]:

```
# I made the same decision as CA4. I decided to analyze the measures of type 'Introd
uction/extension of measures'
# As I want to match the data with the time series I decided to filter out the measu
res without the date implemented
df_measures <- subset(df_measures, LOG_TYPE == 'Introduction / extension of measure
s')

df_measures <- subset(df_measures, DATE_IMPLEMENTED != "")
```

In [9]:

```
# I create the dataframe with the desired columns
measures_by_country <- select(df_measures, COUNTRY ,REGION, CATEGORY, DATE_IMPLEMENT
ED)
```

In [10]:

```
# I realized that countries have a different definition from the files with time ser
ies
# In order to avoid thta the joins don't work I remapped some countries that I know
 to be different in the next dataframes
measures_by_country$COUNTRY <- as.character(measures_by_country$COUNTRY)
measures_by_country$COUNTRY[tolower(measures_by_country$COUNTRY) == "czech republic"
] <- "Czechia"
measures_by_country$COUNTRY[tolower(measures_by_country$COUNTRY) == "moldova republi
c of"] <- "Moldova"
measures_by_country$COUNTRY[tolower(measures_by_country$COUNTRY) == "North Macedonia
Republic Of"] <- "North Macedonia"
measures_by_country$COUNTRY[measures_by_country$COUNTRY == "Russian Federation"] <-
"Russia"
measures_by_country$COUNTRY <- as.factor(measures_by_country$COUNTRY)
```

In [11]:

```
# Load the file with time series on covid confirmed cases and create the dataframe
cases <- read.csv("C:\\Users\\Romina\\Desktop\\DBSCourse\\ProgrammingBigData\\CA5\\t
ime_series_covid19_confirmed_global.csv", sep = ",", header = TRUE)

cases <- data.frame(cases)
```

In [12]:

```
# Let's have a look at the dataframe structure
str(cases)
```

```
'data.frame':	266 obs. of  129 variables:
 $ Province.State: Factor w/ 82 levels "","Alberta","Anguilla",..: 1 1 1
1 1 1 1 1 6 50 ...
 $ Country.Region: Factor w/ 188 levels "Afghanistan",..: 1 2 3 4 5 6 7
8 9 9 ...
 $ Lat           : num  33 41.2 28 42.5 -11.2 ...
 $ Long          : num  65 20.17 1.66 1.52 17.87 ...
 $ X1.22.20      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ X1.23.20      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ X1.24.20      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ X1.25.20      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ X1.26.20      : int  0 0 0 0 0 0 0 0 0 3 ...
 $ X1.27.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X1.28.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X1.29.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X1.30.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X1.31.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.1.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.2.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.3.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.4.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.5.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.6.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.7.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.8.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.9.20       : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.10.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.11.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.12.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.13.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.14.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.15.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.16.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.17.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.18.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.19.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.20.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.21.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.22.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.23.20      : int  0 0 0 0 0 0 0 0 0 4 ...
 $ X2.24.20      : int  1 0 0 0 0 0 0 0 0 4 ...
 $ X2.25.20      : int  1 0 1 0 0 0 0 0 0 4 ...
 $ X2.26.20      : int  1 0 1 0 0 0 0 0 0 4 ...
 $ X2.27.20      : int  1 0 1 0 0 0 0 0 0 4 ...
 $ X2.28.20      : int  1 0 1 0 0 0 0 0 0 4 ...
 $ X2.29.20      : int  1 0 1 0 0 0 0 0 0 4 ...
 $ X3.1.20       : int  1 0 1 0 0 0 0 1 0 6 ...
 $ X3.2.20       : int  1 0 3 1 0 0 0 1 0 6 ...
 $ X3.3.20       : int  1 0 5 1 0 0 1 1 0 13 ...
 $ X3.4.20       : int  1 0 12 1 0 0 1 1 0 22 ...
 $ X3.5.20       : int  1 0 12 1 0 0 1 1 0 22 ...
 $ X3.6.20       : int  1 0 17 1 0 0 2 1 0 26 ...
 $ X3.7.20       : int  1 0 17 1 0 0 8 1 0 28 ...
 $ X3.8.20       : int  4 0 19 1 0 0 12 1 0 38 ...
 $ X3.9.20       : int  4 2 20 1 0 0 12 1 0 48 ...
 $ X3.10.20      : int  5 10 20 1 0 0 17 1 0 55 ...
 $ X3.11.20      : int  7 12 20 1 0 0 19 1 0 65 ...
 $ X3.12.20      : int  7 23 24 1 0 0 19 4 0 65 ...
 $ X3.13.20      : int  7 33 26 1 0 1 31 8 1 92 ...
 $ X3.14.20      : int  11 38 37 1 0 1 34 18 1 112 ...
 $ X3.15.20      : int  16 42 48 1 0 1 45 26 1 134 ...
```

```
 $ X3.16.20       : int  21 51 54 2 0 1 56 52 2 171 ...
 $ X3.17.20       : int  22 55 60 39 0 1 68 78 2 210 ...
 $ X3.18.20       : int  22 59 74 39 0 1 79 84 3 267 ...
 $ X3.19.20       : int  22 64 87 53 0 1 97 115 4 307 ...
 $ X3.20.20       : int  24 70 90 75 1 1 128 136 6 353 ...
 $ X3.21.20       : int  24 76 139 88 2 1 158 160 9 436 ...
 $ X3.22.20       : int  40 89 201 113 2 1 266 194 19 669 ...
 $ X3.23.20       : int  40 104 230 133 3 3 301 235 32 669 ...
 $ X3.24.20       : int  74 123 264 164 3 3 387 249 39 818 ...
 $ X3.25.20       : int  84 146 302 188 3 3 387 265 39 1029 ...
 $ X3.26.20       : int  94 174 367 224 4 7 502 290 53 1219 ...
 $ X3.27.20       : int  110 186 409 267 4 7 589 329 62 1405 ...
 $ X3.28.20       : int  110 197 454 308 5 7 690 407 71 1617 ...
 $ X3.29.20       : int  120 212 511 334 7 7 745 424 77 1791 ...
 $ X3.30.20       : int  170 223 584 370 7 7 820 482 78 2032 ...
 $ X3.31.20       : int  174 243 716 376 7 7 1054 532 80 2032 ...
 $ X4.1.20        : int  237 259 847 390 8 7 1054 571 84 2182 ...
 $ X4.2.20        : int  273 277 986 428 8 9 1133 663 87 2298 ...
 $ X4.3.20        : int  281 304 1171 439 8 15 1265 736 91 2389 ...
 $ X4.4.20        : int  299 333 1251 466 10 15 1451 770 93 2493 ...
 $ X4.5.20        : int  349 361 1320 501 14 15 1451 822 96 2580 ...
 $ X4.6.20        : int  367 377 1423 525 16 15 1554 833 96 2637 ...
 $ X4.7.20        : int  423 383 1468 545 17 19 1628 853 96 2686 ...
 $ X4.8.20        : int  444 400 1572 564 19 19 1715 881 99 2734 ...
 $ X4.9.20        : int  484 409 1666 583 19 19 1795 921 100 2773 ...
 $ X4.10.20       : int  521 416 1761 601 19 19 1975 937 103 2822 ...
 $ X4.11.20       : int  555 433 1825 601 19 21 1975 967 103 2857 ...
 $ X4.12.20       : int  607 446 1914 638 19 21 2142 1013 103 2857 ...
 $ X4.13.20       : int  665 467 1983 646 19 23 2208 1039 102 2863 ...
 $ X4.14.20       : int  714 475 2070 659 19 23 2277 1067 103 2870 ...
 $ X4.15.20       : int  784 494 2160 673 19 23 2443 1111 103 2886 ...
 $ X4.16.20       : int  840 518 2268 673 19 23 2571 1159 103 2897 ...
 $ X4.17.20       : int  906 539 2418 696 19 23 2669 1201 103 2926 ...
 $ X4.18.20       : int  933 548 2534 704 24 23 2758 1248 103 2936 ...
 $ X4.19.20       : int  996 562 2629 713 24 23 2839 1291 103 2957 ...
 $ X4.20.20       : int  1026 584 2718 717 24 23 2941 1339 104 2963 ...
 $ X4.21.20       : int  1092 609 2811 717 24 23 3031 1401 104 2969 ...
 $ X4.22.20       : int  1176 634 2910 723 25 24 3144 1473 104 2971 ...
 $ X4.23.20       : int  1279 663 3007 723 25 24 3435 1523 104 2976 ...
 $ X4.24.20       : int  1351 678 3127 731 25 24 3607 1596 105 2982 ...
 $ X4.25.20       : int  1463 712 3256 738 25 24 3780 1677 106 2994 ...
  [list output truncated]
```

In [14]:

```
# There are 266 observations with 129 variables.
# The variables (columns) are the dates from 22/01/2020 to 25/05/2020 in which the d
ata have been registered
# These columns are integer and have a fomrat not esaily readable (see below for con
version)
# Let's have a look at some data
head(cases, 5)
```

Province.State	Country.Region	Lat	Long	X1.22.20	X1.23.20	X1.24.20	X1.25.20	X1.2
	Afghanistan	33.0000	65.0000	0	0	0	0	
	Albania	41.1533	20.1683	0	0	0	0	
	Algeria	28.0339	1.6596	0	0	0	0	
	Andorra	42.5063	1.5218	0	0	0	0	
	Angola	-11.2027	17.8739	0	0	0	0	

In [15]:

```
# I select the rows with Province.State empty in order to avoid getting data for the
overseas territorial collectivity
# I exclude the column not useful
# I reshape the dataframe in order to transpose the columns in rows
cases <- subset(cases, Province.State == "" )
cases <- select(cases, -Province.State ,-Lat, -Long)
cases <- melt(cases, id = "Country.Region")
```

In [16]:

```
# Let's have a look at the reshaped dataframe
head(cases,5)
```

Country.Region	variable	value
Afghanistan	X1.22.20	0
Albania	X1.22.20	0
Algeria	X1.22.20	0
Andorra	X1.22.20	0
Angola	X1.22.20	0

In [17]:

```
# I convert the column 'variable' in a date with the desired format
# I rename the columns with a more meaningful description
cases$variable <- as.Date(as.Date(cases$variable , format = "X%m.%d.%y"), format ="%
d/%m/%y")

colnames(cases)[2] <- "date"
colnames(cases)[3] <- "cases"
```

In [21]:

```
# Load the file with covid deaths and create the dataframe
# The structure of this dataset is the same of the previous one

deaths <- read.csv("C:\\Users\\Romina\\Desktop\\DBSCourse\\ProgrammingBigData\\CA4
\\time_series_covid19_deaths_global.csv", sep = ",", header = TRUE)

deaths <- data.frame(deaths)
```

In [22]:

```
# I approach with the same logic used for the cases dataframe
deaths <- subset(deaths, Province.State == "" )
deaths <- select(deaths, -Province.State ,-Lat, -Long)
deaths <- melt(deaths, id = "Country.Region")
```

In [23]:

```
deaths$variable <- as.Date(as.Date(deaths$variable , format = "X%m.%d.%y"), format =
"%d/%m/%y")

colnames(deaths)[2] <- "date"
colnames(deaths)[3] <- "deaths"
```

In [24]:

```
# Let's have a look at the final file
head(deaths,5)
```

Country.Region	date	deaths
Afghanistan	2020-01-22	0
Albania	2020-01-22	0
Algeria	2020-01-22	0
Andorra	2020-01-22	0
Angola	2020-01-22	0

In [25]:

```
# I merge the two dataset with covid data.
# The inner join is used because the two datframe have the same countries and dates.
No risk of loosing information from
# one of the two datasets
# I filter out the rows with deaths and cases = 0
covid_data <- merge(cases, deaths, by = c("Country.Region", "date"))
covid_data <- subset(covid_data, cases > 0 & deaths > 0)
```

In [26]:

```
# Before joining the covid_data dataframe with the measures one I need to associate
 the region to the covid_data
# I load a file containing the mapping of country/region in the world
mapping <- read.csv("C:\\Users\\Romina\\Desktop\\DBSCourse\\ProgrammingBigData\\CA5
\\continents-according-to-our-world-in-data.csv", sep = ",", header = TRUE)
```

In [27]:

```
# Let's have a look at the mapping dataframe
head(mapping,5)
```

Entity	Code	Year	X
Abkhazia	OWID_ABK	2015	Asia
Afghanistan	AFG	2015	Asia
Akrotiri and Dhekelia	OWID_AKD	2015	Asia
Albania	ALB	2015	Europe
Algeria	DZA	2015	Africa

In [31]:

```
# I have to merge the covid_data region column with the Entity column of mapping df.
# In order to avoid loosing data from the covid df I choose a left join wiht the cov
id_data df on the left side
covid_data_region <- merge(covid_data, mapping, by.x = "Country.Region", by.y = "Ent
ity", all.x = TRUE )
```

In [32]:

```
# I exclude the columns not useful and rename the X one
covid_data_region <- select(covid_data_region, -Code ,-Year)
colnames(covid_data_region)[5] <- "Region"
```

In [33]:

```
#Let's have a look at the covid_data dataframe
head(covid_data_region,5)
```

Country.Region	date	cases	deaths	Region
Afghanistan	2020-03-25	84	2	Asia
Afghanistan	2020-03-24	74	1	Asia
Afghanistan	2020-04-15	784	25	Asia
Afghanistan	2020-03-26	94	4	Asia
Afghanistan	2020-04-14	714	23	Asia

In [34]:

```
# I remap a specific country name in order to avoid error during the join
covid_data_region$Country.Region <- as.character(covid_data_region$Country.Region)
covid_data_region$Country.Region[tolower(covid_data_region$Country.Region) == "czech
republic"] <- "Czechia"
covid_data_region$Country.Region <- as.factor(covid_data_region$Country.Region)
```

In [35]:

```
# Before joining the covid data with the measures dataframe I convert the 'DATE_IMPL
EMENTED' to a date
measures_by_country$DATE_IMPLEMENTED <- as.Date(measures_by_country$DATE_IMPLEMENTED
, format = "%d/%m/%y")
```

In [36]:

```
# I join the covid data with the measures dataframe.
# I use a full outer join because: there are some measures implemented before the fi
rst date available in the time series,
# measures have not been implemented every day so we have covid data in some days wi
thout measures assigned
dataset <- merge(measures_by_country, covid_data_region, by.x = c("COUNTRY","DATE_IM
PLEMENTED","REGION"), by.y = c("Country.Region","date","Region"), all = TRUE)
```

In [37]:

```
# I decide analysing the European situation. I create a subset from the previous dat
aframe
Europe_df <- subset(dataset, REGION == "Europe" )
```

In [38]:

```
# I decided to analyze the trend of cases/deaths for each country
# In order to avoid a messy plot I decided to plot the top 10 countries for # of dea
ths and # of cases
# To aim this plot I proceed in this way:
# 1- Creating two dataframes with the max # of deaths and max # of cases for each co
untry (I decided for the max because
# the data from time series datasets are cumulative):

Europe_cases_max <- Europe_df %>% group_by(COUNTRY) %>% summarise_at(vars(cases), ~
max(., na.rm = TRUE))
Europe_deaths_max <- Europe_df %>% group_by(COUNTRY) %>% summarise_at(vars(deaths),
~ max(., na.rm = TRUE))
```

In [39]:

```
#2 - Creating two dataframes with decreasing order of deaths/cases from previous dat
aframes
top10_Europe_cases <- Europe_cases_max$COUNTRY [order (Europe_cases_max$cases,  decr
easing = TRUE)]
top10_Europe_deaths <- Europe_deaths_max$COUNTRY [order (Europe_deaths_max$deaths,
decreasing = TRUE)]
```

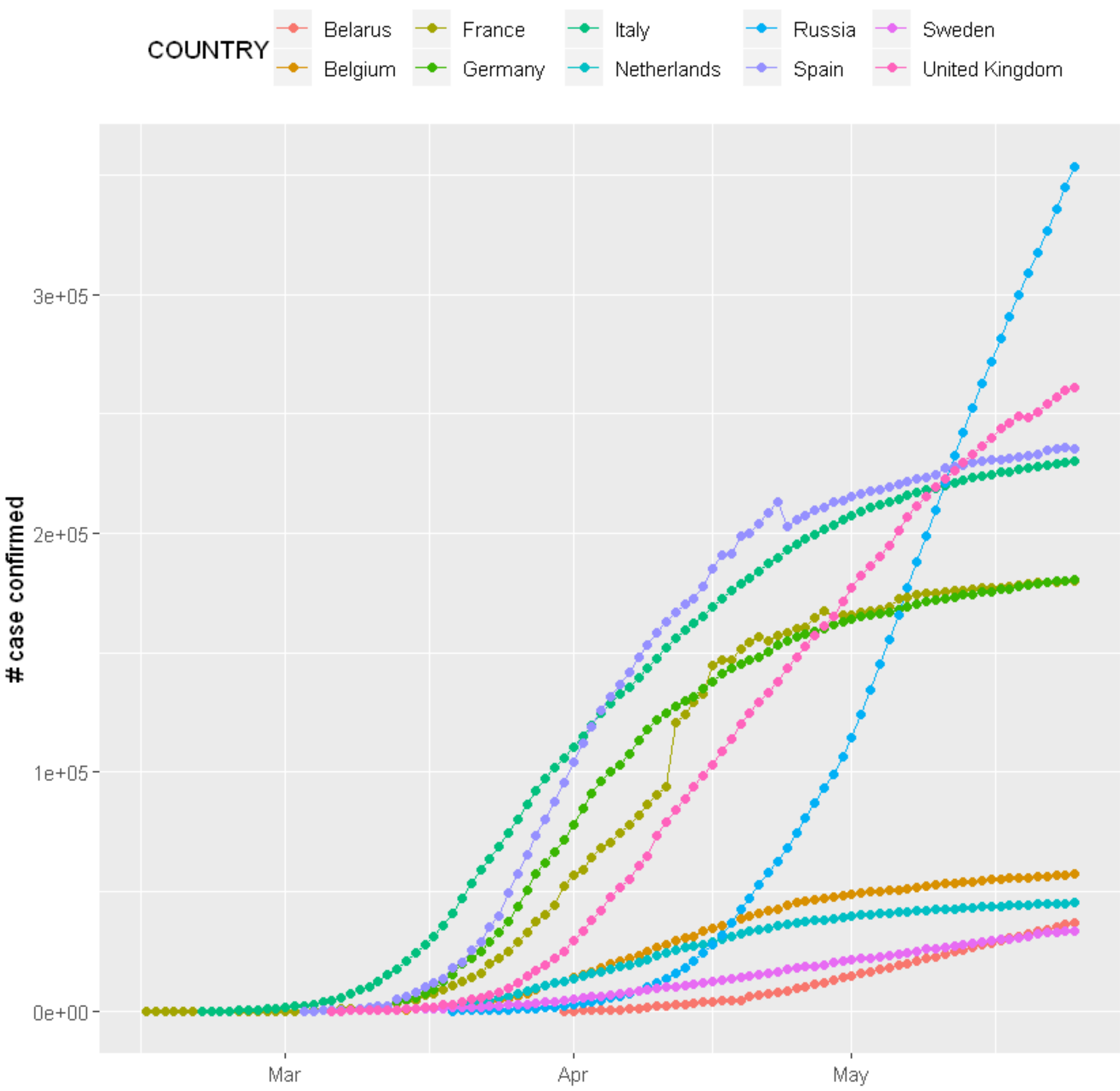
In [40]:

```
library(ggplot2)
```

```
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang
```

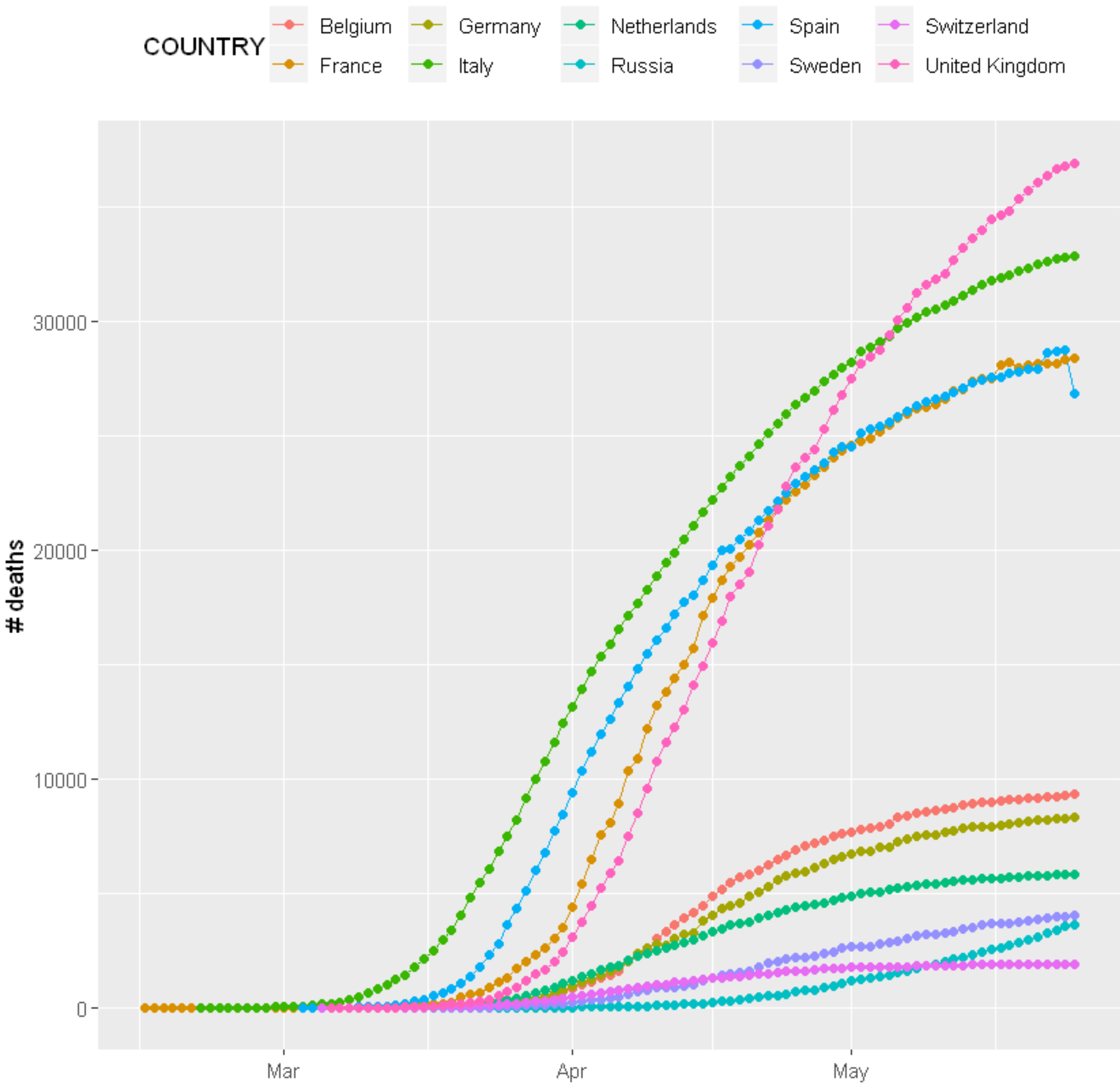
In [41]:

```
# Plot the trend of cases for the top 10 countries
ggplot(data = subset(Europe_df, COUNTRY %in% top10_Europe_cases [1 : 10] & !(is.na(E
urope_df$cases))), aes(x=DATE_IMPLEMENTED, y=cases, group=COUNTRY)) +
  geom_line(aes(color=COUNTRY))+
  geom_point(aes(color=COUNTRY))+
  xlab("")+
  ylab("# case confirmed")+
  theme(legend.position="top")
```



In [42]:

```
# Plot the trend of deaths for the top 10 countries
ggplot(data = subset(Europe_df, COUNTRY %in% top10_Europe_deaths [1 : 10] & !(is.na(
Europe_df$deaths))), aes(x=DATE_IMPLEMENTED, y=deaths, group=COUNTRY)) +
  geom_line(aes(color=COUNTRY))+
  geom_point(aes(color=COUNTRY))+
  xlab("")+
  ylab("# deaths")+
  theme(legend.position="top")
```



In [43]:

```
# Most part of the countries are falling in the top 10 for both # of deaths and # of cases (Italy, Germany, France, Spain,
# United Kingdom, Russia)
# The Russian trends have picked my attention because the # of cases seem increased in May more than in the other countries
# the # of deaths, on the otherhand, has been the lowest.
# The German situation is interesting as well because,despite the high # of cases (very similar to the French trend),
# it has registered a lower trend of deaths.
```

In [44]:

```
# I decided to analyze the relation (if existing) between the # of measures implemented by each government and the
# of deaths through a scatter plot

# To build the plot I proceed in this way:
# 1- Create a first dataframe with max # of detahs and max # of cases for each countries
first_df <- Europe_df %>% group_by(COUNTRY) %>% summarise(max(deaths, na.rm = TRUE), max(cases, na.rm = TRUE))
```

In [45]:

```
# 2- Create a second dataframe with # of measures for each country:
second_df <- subset(Europe_df, !(is.na(Europe_df$CATEGORY))) %>% group_by(COUNTRY) %>% count()
```

In [46]:

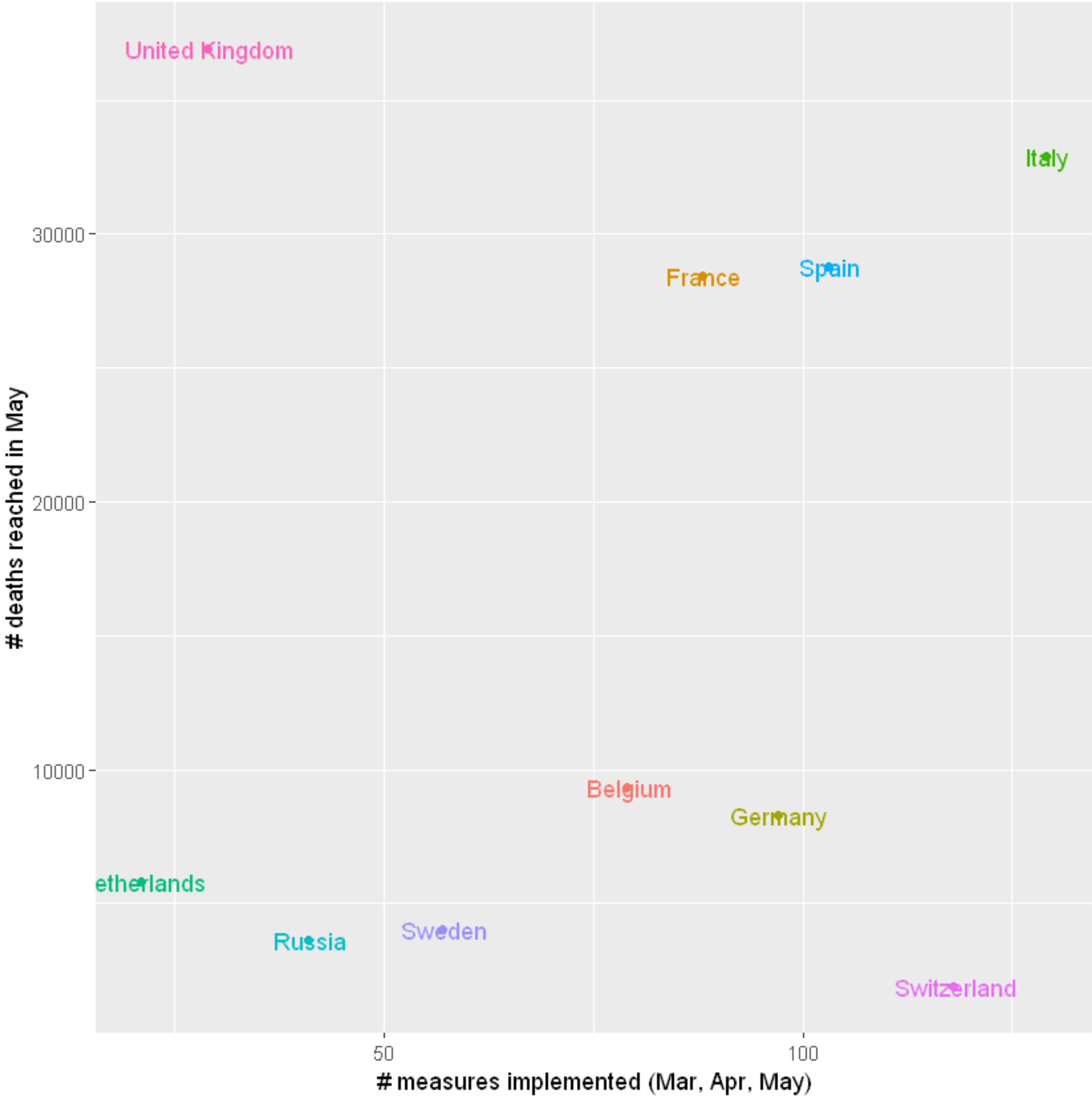
```
# 3- Create the dataframe for the scatter plot merging the two previous ones:
scatter_df <- merge(first_df, second_df)

colnames(scatter_df)[2] <- "deaths"
colnames(scatter_df)[3] <- "cases"
colnames(scatter_df)[4] <- "measures"
```

In [47]:

```
# I plot only the top 10 countries for # of deaths. I create a subset from the previous dataframe:
scatter_df <- subset(scatter_df, COUNTRY %in% top10_Europe_deaths [1 : 10])
```

In [48]:

```
# Scatterplot with # of measures and # of deaths for the top 10 countries:
ggplot(scatter_df, aes(x=measures, y = deaths, color = COUNTRY)) +
  geom_point() +
  geom_text(
    label = scatter_df$COUNTRY,
    nudge_x = 0.25, nudge_y = -0.25,
    check_overlap = T
  )+
  xlab("# measures implemented (Mar, Apr, May)")+
  ylab("# deaths reached in May")+
  theme(legend.position = "none")
```



In [49]:

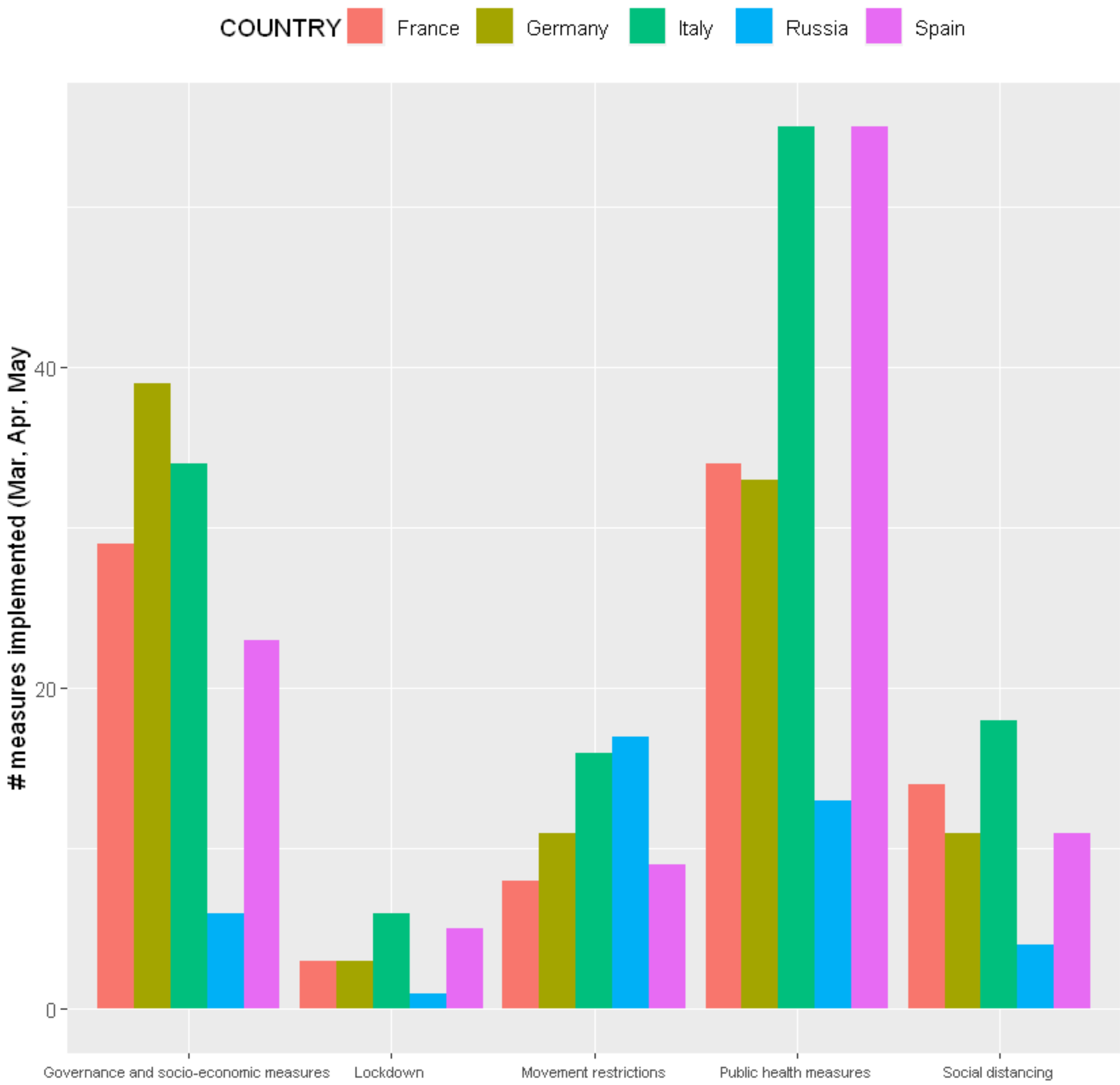
```
# The United Kingdom and Switzerland seem to be the only ones that can demonstrate a
direct correlation
# between the # of measures implemented and the effects on covid letality (United Ki
ngdom with the highest # of deaths
# has implemented the lowest # of measures; Switzerland with the lowest # of deaths
 has implemented one of the highest
# # of measures).
# What about Russia with low # of measures and low # of deaths?  What about Germany
 with the same # of measures as Spain
# but with a lower # of deaths? What about Italy with the highest # of measures and
 # of deaths close to england?
```

In [50]:

```
# I decided to analyze for the countries highlighted above the categories of measure
s implemented
# To plot them I created a dataframe containing the # of measures for each category
 in each country
measures_by_category <- subset(Europe_df, !(is.na(Europe_df$CATEGORY))) %>% group_by
(COUNTRY, CATEGORY) %>% count()
colnames(measures_by_category)[3] <- "measures"
```

In [56]:

```
# Plot the # of measures for each category for 5 countries with opposite behaviour
 (Russia, Germany, France, Italy, Spain)
ggplot(data = subset(measures_by_country, COUNTRY %in% list('Russia','France','Ital
y','Spain','Germany')), aes(x=CATEGORY, fill = COUNTRY )) +
  geom_bar(position = position_dodge(width = 0.9))+
  xlab("")+
  ylab("# measures implemented (Mar, Apr, May")+
  theme(legend.position = "top", axis.text.x = element_text(size = 6.5))
```


COUNTRY
France
Germany
Italy
Russia
Spain
measures implemented (Mar, Apr, May
40
20
0
Governance and socio-economic measures
Lockdown
Movement restrictions
Public health measures
Social distancing

In [57]:

```
# The German trend could be explained by the higher # of both 'Governance and socio-
economic measures' and 'Public health'
# Russia have implemented the higher # of 'Movement restrictions' measures. Neverthe
less the difference with the Italian
# number seems not enough to explain the big difference in the # of deaths.
# Analyzing the different categories of measures is not enough to explain the deaths
trend.
# Probably it could be interesting to match these data with different information su
ch as : population, density population,
# type of health system, type of lockdown implemented.
```

In [ ]:

```

```