Pràctica 2 - Neteja i anàlisi de les dades

Albert Gil Devesa (agildeve) i Aleix Borrella Colomé (aborrellac)

Deadline: 05/01/2021

Contribucions	**Firma**
Investigació prèvia	ABC, AGD
Redacció de les respostes	ABC, AGD
Desenvolupament de codi	ABC, AGD

Table of Contents

0. Pràctica 2: Tipologia i cicle de vida de les dades.

En aquesta pràctica analitzarem el dataset "Novel Corona Virus 2019 Dataset - Day level information on ovid-19 affected cases" que podem trobar a Kaggle en el següent enllaç: "https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset"

La intenció d'aquesta pràctica és analitzar l'evolució de la Covid-19 a l'estat espanyol, a nivell global entre diferents regions, així com valorar l'eficàcia de les mesures implantades per lluitar contra aquesta nova malaltia.

1. Introducció

El 31 de Desembre de 2019 la Organització Mundial de la Salut (OMS, WHO en anglès) va ser alertada de diversos casos de pneumònia a la ciutat de Wuhan, de la província de Hubei, a la Xina. El virus no coincidia amb cap altre virus conegut, la qual cosa va generar preocupació perquè quan un virus és nou, no sabem com afecta a les persones.

El Novel Coronavirus de 2019 (2019-nCoV) és un virus (concretament un coronavirus) identificat com la causa d'un brot de malalties respiratòries detectat per primea vegada a la ciutat xinesa de Wuhan, de la província de Hubei. Al principi, molts dels pacients del brot de Wuhan van informar que tenien algun vincle amb un gran mercat de productes del mar i animals, cosa que suggeria la posible propagació d'animal a persona. No obstant, un nombre creixent de pacients van començar a indicar que no van tenir cap tipus d'exposició a mercats d'animals, cosa que indicava que estava succeint un contagi persona a persona.

1.1. Descripció del Dataset

El dataset conté informació diaria sobre el nombre de casos afectats, defuncions i recuperacions d'aquest nou coronavirus 2019-nCoV.

El principal arxiu amb el que treballarem és el 'covid_19_data.csv', i aquesta és la seva descripció detallada:

- **SNo**: Serial number.
- **ObservationDate**: Data de l'observació en format MM/DD/YYYY.
- **Provice/State**: Província o estat de l'observació.
- **Country/Region**: País d'observació.
- **Last Update**: Hora en UTC en que s'ha actualitzat la fila per a la província o país determinat.
- **Confirmed**: Nombre acumulat de casos confirmats fins a aquesta data.
- **Deaths**: Nombre acumulat de defuncions fins a aquesta data.
- **Recovered**: Nombre acumulat de casos recuperats fins a aquesta data.

1.2. Objectius de l'estudi

En primer lloc es realitzarà una primera exploració visual de les dades de Covid-19 a nivell europeu i, acte seguit, ens centrarem a analitzar el seu impacte en els dos països més castigats per aquest virus. Per fer-ho, el nostre objectiu serà el de realitzar tests estadístics de contrast d'hipòtesis per determinar si la incidència de la primera onada de la Covid-19 en els dos països més afectats va ser estadísticament diferent o no.

A continuació, ens centrarem amb les dades corresponents a les Comunitats Autonòmiques, i buscarem possibles correlacions entre les caracterítiques d'aquestes (tamany, població) i els efectes de la malaltia (casos confirmats, morts, recuperats).

Tot seguit, intentarem aplicar una classificació de clustering, per veure si podem agrupar les CCAA en funció de les seves característique si l'impacte del virus en cada cas, i estimarem el nombre òptim de agrupacions.

Per últim, intentarem crear un model de regressió lineal multivariant per intentar explicar el nombre de morts en cada Comunitat Autònoma espanyola en funció de les característiques demogràfiques d'aquestes així com de la incidència del virus en nombre de casos confirmats.

Per tant, algunes de les preguntes concretes a les que volem buscar resposta són:

- Podem considerar que les dades de la primera onada de la Covid-19 en els dos països europeus més afectats són estadísticament diferets?
- La Covid-19 és un coronavirus que majoritariament es contagia persona a persona. Per tant, existeix algun tipus de relació entre les característiques de cada CCAA (població, superfície) i l'impacte del virus? A priori esperarem que les comunitats amb més densitat de població haurien de veure's més afectades.
- Per fer front a la situació actual, és possible classificar les comunitats en diversos grups, per tal que cada grup pugui aplicar mesures adequades i més apropiades?
- Podem veure en aquestes dades un canvi degut a les polítiques adoptades per les diverses administracions competents?
- Podem construïr un model de regressió que ens permeti predir l'evolució de la Covid-19?

2. Integració i selecció de les dades d'interés a analitzar

En primer lloc, llegirem el fitxer obtingut de Kaggle *"covid_19_data.csv"* i guardarem les dades en un dataframe. Per fer-ho, començarem carregant i obrint el fitxer de dades que hem de treballar i analitzarem els tipus de dades amb els que R ha interpretat cada variable. Abans, però, matisarem alguns aspectes del fitxer *.csv* a utilitzar.

Com hem vist en els apunts, podem tenir dos tipus de fitxers *.csv*, el **format anglès**, en el qual les dades estan realment separades per comes (,); i el **format espanyol**, en el qual les dades estan separades per punts i coma (;). Si obrim el fitxer *.csv* amb qualsevol editor de text podem comprovar que es tracta del format anglès, així que podem carregar les dades directament amb la comanda *"read.csv()"* (si es tractessin del format espanyol hauríem d'utilitzar la comanda *"read.csv2()"*):

```
# Carreguem la llibreria necessària:
library(readr)

# Carreguem les dades del fitxer obtingut en un dataframe:
covid_19_data <- read.csv("covid_19_data.csv", header = TRUE, stringsAsFa
ctors = FALSE)
```

```
# Visualitzem les dades per pantalla:
head(covid_19_data)
```

```
##   SNo ObservationDate Province.State Country.Region     Last.Update Co
nfirmed
## 1   1       01/22/2020           Anhui Mainland China 1/22/2020 17:00
1
## 2   2       01/22/2020         Beijing Mainland China 1/22/2020 17:00
14
## 3   3       01/22/2020       Chongqing Mainland China 1/22/2020 17:00
6
## 4   4       01/22/2020          Fujian Mainland China 1/22/2020 17:00
1
## 5   5       01/22/2020           Gansu Mainland China 1/22/2020 17:00
0
## 6   6       01/22/2020       Guangdong Mainland China 1/22/2020 17:00
26
##                                          Deaths              Recovered
##    1                          0                                     0
##    2                          0                                     0
##    3                          0                                     0
##    4                          0                                     0
##    5                          0                                     0
## 6      0         0
```

Comentar que a l'hora de carregar les dades amb la funció *"read.csv()"* hem indicat que conté una primera fila amb els noms de cada atribut (*header = TRUE*), i també li hem ordenat que a l'hora de llegir les dades no converteixi els valors textuals com a factors (*stringsAsFactors = FALSE*).

A continuació, també és important verificar el tipus de cada variable, és a dir, determinar quines variables són de tipus numèric i quines variables són de tipus categòric. Per fer-ho, examinarem els tipus de dades amb els quals R ha interpretat cada variable:

```
# Explorem els tipus de dades de cada atribut:
str(covid_19_data)
```

```
##   'data.frame':              172480   obs.   of       8   variables:
##   $ SNo                     : int   1 2 3 4 5 6 7 8 9 10 ...
##  $ ObservationDate: chr  "01/22/2020" "01/22/2020" "01/22/2020" "01/22
/2020"                                                                ...
##   $ Province.State : chr  "Anhui" "Beijing" "Chongqing" "Fujian" ...
##  $ Country.Region : chr  "Mainland China" "Mainland China" "Mainland C
hina"                "Mainland                China"               ...
##  $ Last.Update    : chr  "1/22/2020 17:00" "1/22/2020 17:00" "1/22/202
0           17:00"           "1/22/2020           17:00"           ...
##   $ Confirmed           : num   1 14 6 1 0 26 2 1 4 1 ...
```

```
##    $ Deaths                : num    0 0 0 0 0 0 0 0 0 0 ...
##  $ Recovered      : num  0 0 0 0 0 0 0 0 0 0 ...
```

Podem observar que les nostres dades estan formades per 172.480 observacions referents a 8 atributs diferents, la qual cosa coincideix amb el fitxer *.csv* proporcionat, així que és una primera verificació de la transmissió de la informació.

Pel que fa a les **variables qualitatives (chr)**, podem veure que R n'ha interpretat aquells atributs que presenten valors textuals com poden ser *'Province.State'* o *'Country.Region'*, així com els atributs que fan referència a dates pel fet d'incorporar una barra / en la seva separació, com són *'ObservationDate'* i *'Last.Update'*.

Pel que fa a les **variables quantitatives (num o int)**, podem veure que R ha identificat la resta d'atributs com a aquest tipus de variable en presentar únicament valors numèrics. Només comentar que totes les dades numèriques que presenta aquest dataset haurien de ser interpretades com a **int** ja que són valors enters i en cap cas valors amb decimals (no tindria sentit tenir 2.5 casos nous), així que s'hauran de convertir els atributs *'Confirmed'*, *'Deaths'* i *'Recovered'* que R els ha interpretat com a **num**, al format **int**:

```
# Convertim les variables quantitatives que R ha intepretat com a 'num' a
tipus                                                                'int':
covid_19_data$Confirmed      <-      as.integer(covid_19_data$Confirmed)
covid_19_data$Deaths         <-         as.integer(covid_19_data$Deaths)
covid_19_data$Recovered      <-      as.integer(covid_19_data$Recovered)

#  Verifiquem  que  els  canvis  s'han  realitzar  satisfactoriament:
str(covid_19_data)
```

```
##    'data.frame':              172480   obs.    of       8    variables:
##    $  SNo                     :  int    1  2  3  4  5  6  7  8  9  10  ...
##  $ ObservationDate: chr  "01/22/2020" "01/22/2020" "01/22/2020" "01/22
/2020"                                                                  ...
##   $ Province.State : chr   "Anhui" "Beijing" "Chongqing" "Fujian" ...
##  $ Country.Region : chr  "Mainland China" "Mainland China" "Mainland C
hina"                 "Mainland                 China"                ...
##  $ Last.Update    : chr  "1/22/2020 17:00" "1/22/2020 17:00" "1/22/202
0           17:00"           "1/22/2020           17:00"           ...
##    $  Confirmed             :  int    1  14  6  1  0  26  2  1  4  1  ...
##    $  Deaths                :  int    0  0  0  0  0  0  0  0  0  0  ...
##  $ Recovered      : int  0 0 0 0 0 0 0 0 0 0 ...
```

Ara si que podem observar que R interpreta cada atribut correctament en funció de les dades que incorpora, és a dir:

- **'int'**: Aquells atributs **quantitatius** que incoporen dades numèriques enteres.
- **'chr'**: Aquells atributs **qualitatius** que incorporen dades textuals.

A més, a l'hora de realitzar els posteriors anàlisis sobre aquest dataset no tindrem en compte l'atribut *'Last.Update'* ja que no ens interesa el dia en què es van actualitzar les dades, sinó al dia al qual fan referència *'Observation.Date'*, així que eliminarem la variable en qüestió per no generar confusions:

```
# Eliminem la varialbe 'Last.Update' del dataset:
covid_19_data <- subset(covid_19_data, select = -Last.Update)

# Verifiquem que la hem eliminat correctament:
head(covid_19_data)
```

```
##   SNo ObservationDate Province.State Country.Region Confirmed Deaths R
ecovered
## 1   1      01/22/2020          Anhui Mainland China         1      0
0
## 2   2      01/22/2020        Beijing Mainland China        14      0
0
## 3   3      01/22/2020      Chongqing Mainland China         6      0
0
## 4   4      01/22/2020         Fujian Mainland China         1      0
0
## 5   5      01/22/2020          Gansu Mainland China         0      0
0
## 6   6      01/22/2020      Guangdong Mainland China        26      0
0
```

3. Neteja de les dades

Una vegada carregades i verificades les dades, procedirem a la seva neteja fent un primer cop d'ull mitjançant les funcions head i tail:

```
# Realitzem una exploració visual de les dades per detectar valors a nete
jar amb la funció head:
head(covid_19_data)
```

```
##   SNo ObservationDate Province.State Country.Region Confirmed Deaths R
ecovered
## 1   1      01/22/2020          Anhui Mainland China         1      0
0
## 2   2      01/22/2020        Beijing Mainland China        14      0
0
## 3   3      01/22/2020      Chongqing Mainland China         6      0
0
## 4   4      01/22/2020         Fujian Mainland China         1      0
0
## 5   5      01/22/2020          Gansu Mainland China         0      0
0
## 6   6      01/22/2020      Guangdong Mainland China        26      0
0
```

```
# Realitzem una exploració visual de les dades per detectar valors a nete
jar              amb              la              funció              head:
tail(covid_19_data)

##           SNo ObservationDate     Province.State Country.Region Confir
med
## 172475 172475      12/06/2020 Zakarpattia Oblast        Ukraine     24
541
## 172476 172476      12/06/2020  Zaporizhia Oblast        Ukraine     36
539
## 172477 172477      12/06/2020            Zeeland    Netherlands      6
710
## 172478 172478      12/06/2020           Zhejiang Mainland China      1
295
## 172479 172479      12/06/2020    Zhytomyr Oblast        Ukraine     31
967
## 172480 172480      12/06/2020       Zuid-Holland    Netherlands    154
813
##                                                    Deaths      Recovered
##      172475                        554                            15299
##      172476                      337                               6556
##     172477                    104                                   0
##     172478                           1                             1288
##      172479                        531                            22263
## 172480   2414         0
```

També podem extreure algunes estadístiques més detallades de les **variables quantitatives (int)** amb el següent codi:

```
# Obtenim algunes estadístiques més detallades de les dades numèriques:
summary(covid_19_data)

##       SNo         ObservationDate    Province.State     Country.Region
##  Min.   :     1   Length:172480      Length:172480      Length:172480
##  1st Qu.: 43121   Class :character   Class :character   Class :charact
er
##  Median : 86240   Mode  :character   Mode  :character   Mode  :charact
er
##                    Mean                                :          86240
##                                      3rd                    Qu.:129360
##                         Max.                                  :172480
##        Confirmed                        Deaths                Recovered
##    Min.       :-302844      Min.       :  -178      Min.       :-854405
##    1st  Qu.:       457      1st  Qu.:       6      1st  Qu.:         10
##    Median  :      4016      Median  :      76      Median  :        858
##    Mean      :    33232      Mean      :  1050      Mean      :    21028
##    3rd  Qu.:    18843      3rd  Qu.:    554      3rd  Qu.:       7229
##  Max.   :2290891   Max.   :54804   Max.   :5624444
```

Com podem observar, tenim valors negatius en els atributs *'Confirmed'*, *'Deaths'* i *'Recovered'*, els quals són valors clarament erronis, ja que no té cap sentit tenir un número negatiu de contagis, morts o recuperacions.

Si busquem en les dades aquests valors negatius, podem observar que només fan referència a 4 registres del total de 172.480 que en disposem, així que segurament seran deguts a algun tipus d'error en el procés de comunicació de les dades. D'aquesta manera, procedirem a eliminar les tuples amb valors negatius:

```
# Mostrem quants registres presenten valors negatius per a l'atribut 'Con
firmed':
covid_19_data[covid_19_data$Confirmed < 0, ]
```

```
##            SNo ObservationDate Province.State Country.Region Confirmed
Deaths
## 146717 146717      11/02/2020        Unknown       Colombia   -302844
0
##                                                                Recovered
## 146717         0
```

```
# Mostrem quants registres presenten valors negatius per a l'atribut 'Dea
ths':
covid_19_data[covid_19_data$Deaths < 0, ]
```

```
##            SNo ObservationDate Province.State Country.Region Confirmed
Deaths
## 117673 117673      09/24/2020        Unknown       Colombia         0
-178
## 140751 140751      10/25/2020        Unknown       Colombia         0
-154
##                                                                Recovered
##            117673                                                 -12684
## 140751     -8072
```

```
# Mostrem quants registres presenten valors negatius per a l'atribut 'Rec
overed':
covid_19_data[covid_19_data$Recovered < 0, ]
```

```
##            SNo ObservationDate Province.State Country.Region Confirmed
Deaths
## 117673 117673      09/24/2020        Unknown       Colombia         0
-178
## 140751 140751      10/25/2020        Unknown       Colombia         0
-154
## 144479 144479      10/30/2020        Unknown       Colombia         0
505
##                                                                Recovered
##            117673                                                 -12684
##          140751                                                    -8072
## 144479   -854405
```

```
# Eliminem els registres que presenten valors negatius:
covid_19_data <- covid_19_data[!(covid_19_data$Confirmed < 0 | covid_19_data$Deaths < 0 | covid_19_data$Recovered < 0),]

# Confirmem que les dades ja estan netes de valors negatius:
summary(covid_19_data)
```

```
##       SNo         ObservationDate    Province.State     Country.Region
##  Min.   :     1   Length:172476      Length:172476      Length:172476
##  1st Qu.: 43120   Class :character   Class :character   Class :character
##  Median : 86238   Mode  :character   Mode  :character   Mode  :character
##                   Mean                                :          86239
##                                       3rd                   Qu.:129358
##                       Max.                                    :172480
##        Confirmed                       Deaths                Recovered
##   Min.     :          0     Min.     :       0     Min.     :          0
##    1st  Qu.:        457      1st  Qu.:        6      1st  Qu.:         10
##    Median  :      4017      Median  :      76      Median  :        858
##    Mean      :    33234      Mean      :  1050      Mean      :    21034
##    3rd  Qu.:    18844      3rd  Qu.:     554      3rd  Qu.:       7229
##  Max.   :2290891   Max.   :54804   Max.   :5624444
```

Pel que fa a les **variables qualitatives (chr)** també en podem extreure algunes estadístiques més detallades, com per exemple els diferents valors possibles de cada atribut textual:

```
# Analitzem quins valors possibles presenta la variable 'ObservationDate':
print( paste("L'atribut 'ObservationDate' presenta valors de", length(unique(covid_19_data$ObservationDate)), "dates diferents."))
```

```
## [1] "L'atribut 'ObservationDate' presenta valors de 320 dates diferents."
```

```
# Analitzem quins valors possibles presenta la variable 'Province.State':
print( paste("L'atribut 'Province.State' presenta valors de", length(unique(covid_19_data$Province.State)), "províncies o estats diferents."))
```

```
## [1] "L'atribut 'Province.State' presenta valors de 736 províncies o estats diferents."
```

```
# Analitzem quins valors possibles presenta la variable 'Country.Region':
print( paste("L'atribut 'Country.Region' presenta valors de", length(unique(covid_19_data$Country.Region)), "països o regions diferents."))
```

```
## [1] "L'atribut 'Country.Region' presenta valors de 226 països o regions diferents."
```

3.1. Les dades contenen zeros o elements buits? Com gestionaries aquests casos?

També com a tasca de neteja de les dades, és important detectar si aquestes contenen valors zero o elements buits. Començarem per comptar els valors '0' presents en el nostre dataset:

```
# Detectem quants valors '0' presenten les nostres dades:
colSums(covid_19_data == 0)
```

```
##             SNo ObservationDate  Province.State  Country.Region       C
onfirmed
##                0                0               0                0
1983
##                                 Deaths                      Recovered
##           22076           36928
```

Podem observar que tenim 1983 valors '0' per a l'atribut *'Confirmed'*, 22076 per a l'atribut *'Deaths'* i 36928 per a l'atribut *'Recovered'*. Els zeros de Deaths i Recovered tenen sentit, ja que són persones malaltes que han mort o han superat la malaltia. En el cas de 'Confirmed', ho entenem com que no tenim casos de CoVid-19. Per tant, *els zeros en si no representen un problema*, però ho tractarem més detalladament en la identificació i tractament de valors extrems.

A continuació, realitzarem la identificació de valors buits en les dades:

```
# Detectem quants valor 'NA' presenten les nostres dades:
summary(is.na(covid_19_data))
```

```
##       SNo             ObservationDate  Province.State   Country.Region
##   Mode :logical     Mode :logical     Mode :logical     Mode :logical
##   FALSE:172476      FALSE:172476      FALSE:172476      FALSE:172476
##     Confirmed                        Deaths                    Recovered
##       Mode   :logical          Mode   :logical          Mode   :logical
##  FALSE:172476    FALSE:172476    FALSE:172476
```

Podem veure que les dades estan netes de valors 'NA', però també cal comprovar si ho estan de valors en blanc '', així que ho podem comprovar amb el següent codi:

```
# Detectem quants valor en blanc '' presenten les nostres dades:
colSums(covid_19_data == '')
```

```
##             SNo ObservationDate  Province.State  Country.Region       C
onfirmed
##                0                0           47883                0
0
##                                 Deaths                      Recovered
##               0              0
```

Com podem veure, tenim 47883 valors en blanc per a l'atribut *'Province.State'*.

Només hem trobat valors buits en el camp *'Province.State'*, així que els interpretem com que s'ha reportat la informació general de casos per país, i per tant no fa falta especificar la sub-regió de les mesures. De totes maneres, igual que en el cas dels zeros, ho estudiarem en detall en el següent punt.

3.2. Identificació i tractament de valors extrems

Com hem comentat, els zeros presents en els atributs *'Confirmed'*, *'Deaths'* i *'Recovered'* no comporten cap contradicció, però per sentit comú, sempre s'ha de complir que els valors confirmats siguin més grans o iguals al número de morts més els casos recuperats, ja que els valors son acumulatius, com hem vist en la descripció del dataset.

Per tant, els valors que no compleixin aquesta condició els tractarem com a valors anomals (podent considerar-los valors extrems) i els excourem del nostre anàlisi. En primer lloc detectarem quants casos tenim:

```
# Carreguem la llibreria necessària:
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

# Detectem en quants registres el número de casos confirmats acumulats és
inferior a la suma de casos que han mort més els que s'han recuperat:
count(covid_19_data[covid_19_data$Confirmed < (covid_19_data$Deaths + cov
id_19_data$Recovered),])

##                                                                      n
## 1 1624
```

Una vegada detectats, com representen una part molt poc representativa de la mostra (1624 registres dels 172476 totals, és a dir, un 0.9 %), hem decidit que la millor manera de tractar-los és eliminant-los:

```
# Eliminem els registres detectats anteriorment:
covid_19_data <- covid_19_data[!(covid_19_data$Confirmed < (covid_19_data
$Deaths + covid_19_data$Recovered)),]

# Confirmem que ara en cap registre el número de casos confirmats acumula
ts és inferior a la suma de casos morts més recuperats:
count(covid_19_data[covid_19_data$Confirmed < (covid_19_data$Deaths + cov
id_19_data$Recovered),])
```

```
##                                                                   n
## 1 0
```

En l'apartat anterior també havíem detectat valors en blanc '' així com valors 'Unknown' per a la variable *'Province.State'*, els quals els tractarem tots de la mateixa manera. Per tal de tenir-los controlats i identificats conjuntament (ja que tots dos casos ens informen de que no es disposa informació sobre la provincia o estat al qual fan referència les dades), els susbtituirem pel valor '-' i, si en algun posterior anàlisi és necessari eliminar-los, ja ho realitzarem:

```
# Carreguem la llibreria necessària:
library(stringr)

# Donem el valor de '-' per als registres que presenten 'Unknown' a l'atr
ibut 'Province.State':
covid_19_data$Province.State <- gsub("Unknown","-", covid_19_data$Provinc
e.State)

# Donem el valor de '-' per als registres que presenten valor en blanc ''
a l'atribut 'Province.State':
covid_19_data$Province.State <- gsub("^$","-", covid_19_data$Province.Sta
te)

# També hem detectat que alguns valors comencen amb un espai en blanc, el
qual eliminem:
covid_19_data$Province.State <- str_trim(covid_19_data$Province.State, "l
eft")

# Comprovem que hem realitzat el canvi correctament:
colSums(covid_19_data == '')

##             SNo ObservationDate  Province.State  Country.Region       C
onfirmed
##               0                0               0                0
0
##                                 Deaths                      Recovered
##              0              0

colSums(covid_19_data == 'Unknown')

##             SNo ObservationDate  Province.State  Country.Region       C
onfirmed
##               0                0               0                0
0
##                                 Deaths                      Recovered
##              0              0
```

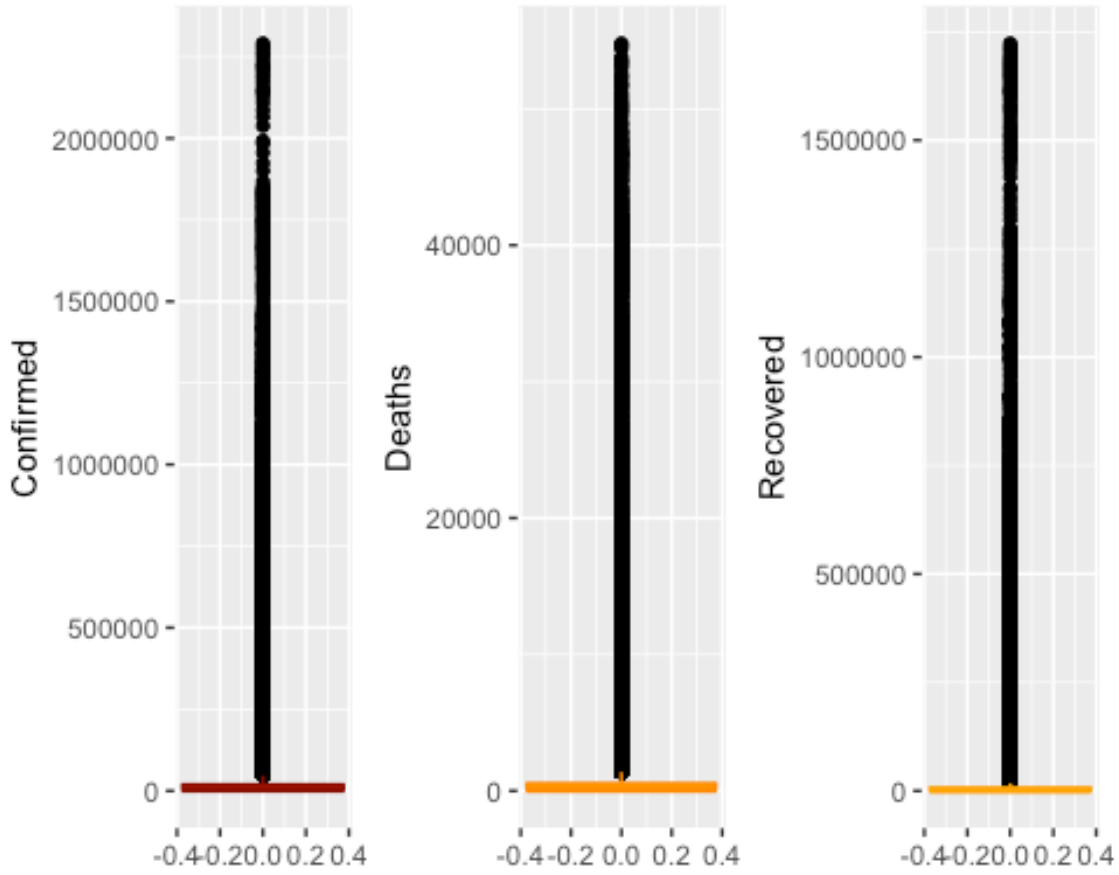
Una vegada tractades les dades perdudes, en blanc, nul·les, etc, ja ens podrem centrar en analitzar si els diferents atributs incorporen valors extrems.

Tal i com vam veure en els apunts, un **valor atípic o outlier** es pot definir com aquella observació (o grups d'observacions) que semblen ser inconsistents amb el conjunt de les dades. No obstant això, és important no confondre'ls amb valors sentinelles. En el nostre cas, no tenim informació que les dades incorporin cap tipus de valor sentinella, així que si observem algun valor inconsistent, segurament es tracti d'un valor atípic.

Per consultar-ho, en la teoria hem vist que una manera senzilla de veure-ho és a través dels diagrames de caixes o box plots, els quals crearem per a les diferents **variables quantitatives (int)**.

Per tal de no saturar l'informe amb gràfiques, les agruparem totes en un panell:

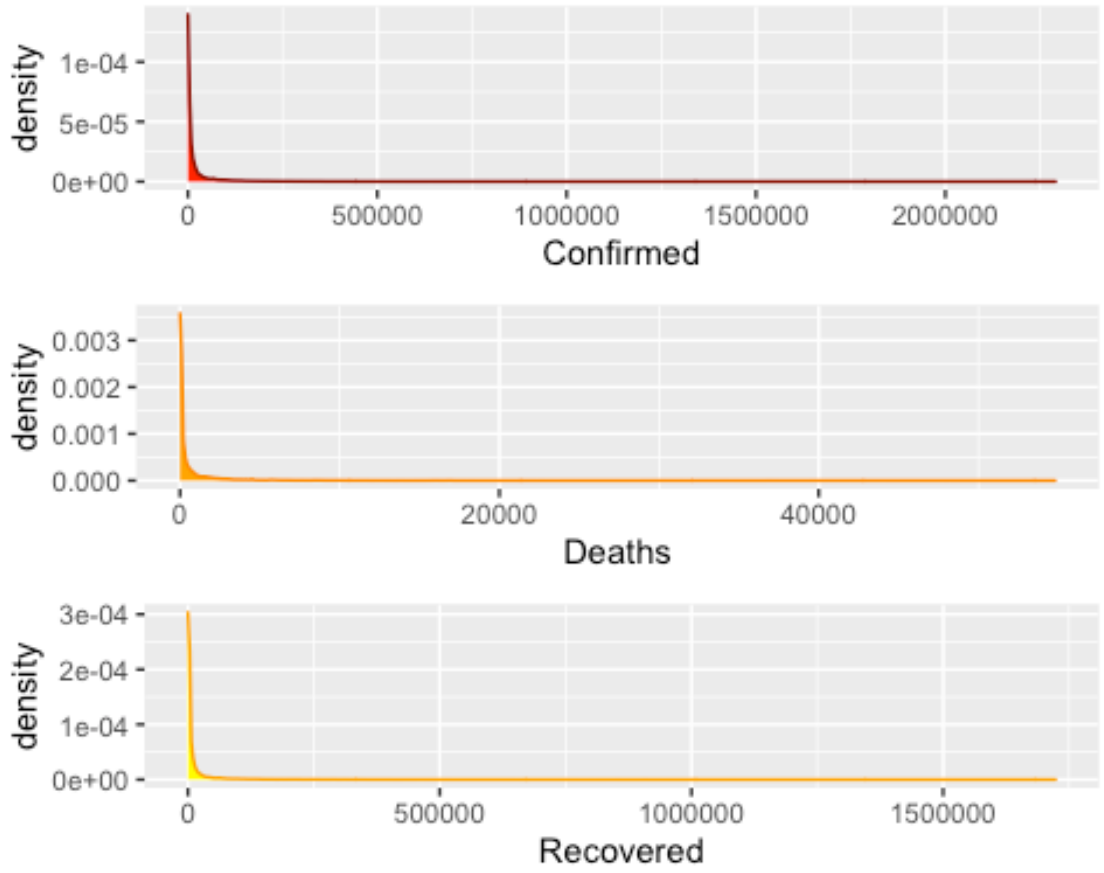
```
#         Carreguem         les         llibreries         necessàries:
library(ggplot2)
library(gridExtra)

##
## Attaching package: 'gridExtra'

##    The    following    object    is    masked    from    'package:dplyr':
##
##     combine

#   Creem   un   gràfic   boxplot   per   a   la   variable   'Confirmed':
a1 <- ggplot() + geom_boxplot(aes(y = covid_19_data$Confirmed), color="da
rk  red",  fill  =  "red",  outlier.colour="black")  +  ylab("Confirmed")

#   Creem   un   gràfic   boxplot   per   a   la   variable   'Deaths':
a2 <- ggplot() + geom_boxplot(aes(y = covid_19_data$Deaths), color="dark
orange",  fill  =  "orange",  outlier.colour="black")  +  ylab("Deaths")

#   Creem   un   gràfic   boxplot   per   a   la   variable   'Recovered':
a3 <- ggplot() + geom_boxplot(aes(y = covid_19_data$Recovered), color="or
ange",  fill  =  "yellow",  outlier.colour="black")  +  ylab("Recovered")

#    Agrupem    els    gràfics    anteriors    en    un    mateix    panell:
grid.arrange(a1, a2, a3, nrow = 1)
```



Com podem veure, el fet que la major part de les dades del nostre dataset estiguin centrades en valors pròxims a 0 implica que qualsevol registre amb valor superior quedi fora del boxplot i ens faci pensar que es tracta d'un valor atípic o outlier. No obstant això, si realment fossin valors extrems estarien tots o bé molts pròxims a les caixes boxplot, o bé molt lluny, és a dir, agrupats en alguna zona, mentre que aquests es troben homogèniament distribuits, la qual cosa també es pot veure si grafiquem el seu diagrama de densitat:

```
# Carreguem les llibreries necessàries:
library(ggplot2)
library(gridExtra)

# Creem un gràfic de densitat per a la variable 'Confirmed':
b1 <- ggplot(mapping = aes(covid_19_data$Confirmed)) + geom_density(color
= "dark red", fill = "red") + xlab("Confirmed")

# Creem un gràfic de densitat per a la variable 'Deaths':
b2 <- ggplot(mapping = aes(covid_19_data$Deaths)) + geom_density(color= "
dark orange", fill = "orange") + xlab("Deaths")

# Creem un gràfic de densitat per a la variable 'Recovered':
b3 <- ggplot(mapping = aes(covid_19_data$Recovered)) + geom_density(color
= "orange", fill = "yellow") + xlab("Recovered")
```

```r
# Agrupem els gràfics anteriors en un mateix panell:
grid.arrange(b1, b2, b3, nrow = 3)
```



En els diagrames de densitat, un clar valor outlier donaria lloc a un pic allunyat del pic principal de densitat, mentre que podem veure que en cap variable ens apareix. Per tant, **podem afirmar que les dades no presenten valors outliers o atípics.**

Per últim, abans de seguir amb els següents punts, comentar que l'atribut *'ObservationDate'* està expressat en el format mm/dd/yyyy i, per tal de poder ordenar les dates de forma més precisa, utilitzarem el format de yyyy-mm-dd, conversió que realitzem a continuació:

```r
# Canviem el format de la columna 'ObservationDate' a un que sigui més fàcil per ordenar els resultats:
covid_19_data$Date <- as.Date(covid_19_data$ObservationDate, format = "%m/%d/%y")

# Reordenem les columnes:
covid_19_data <- covid_19_data[c("SNo","Date", "Province.State", "Country.Region","Confirmed", "Deaths", "Recovered")]
```

4. Anàlisi de les dades

Una vegada realitzada la càrrega, integració i neteja de les dades, és hora de procedir amb el seu anàlisi.

Tal i com s'ha indicat en els objectius, s'intentaran abordar els diferents anàlisis següents:

1. **Tests estadístics de contrast d'hipòtesis**
2. **Correlació en les dades**
3. **Clustering**
4. **Regressió lineal multivariant**

Per tant, desenvoluparem aquests 4 anàlisis en els corresponents apartats 4.1, 4.2, 4.3 i 4.4.

4.1. Tests estadístics

A continuació, procedirem amb un conjunt de tests estadístics tot estudiant la normalitat i homogeneïtat de la variància.

4.1.1. Seleció dels grups de dades que es volen analitzar/comparar (Planificació dels anàlisis a aplicar).

L'objectiu d'aquest primer anàlisi, és determinar si la incidència de la primera onada de covid-19 en els dos països europeus més afectats va ser estadísticament diferent o no, la qual cosa resoldrem a partir de contrastos d'hipòtesis entre els casos confirmats així com el nombre de morts.

Abans però, analitzarem visualment les dades. El nostre dataset conté informació referent als casos confirmats, morts i recuperats de covid-19 a escala mundial, així que començarem limitant una mica l'àmbit d'aquest primer anàlisi, centrant-nos únicament en els països europeus.

Per exemple, podem començar obtenint un subdataframe que inclogui alguns dels països europeus on el covid-19 ha tingut més incidència, com són:

```
# Creem un nou dataframe que només inclogui dades d'alguns països europeus:
Europe <- covid_19_data[covid_19_data$Country.Region == 'Spain' |
                        covid_19_data$Country.Region == 'Italy' |
                        covid_19_data$Country.Region == 'Germany' |
                        covid_19_data$Country.Region == 'Belgium'|
                        covid_19_data$Country.Region == 'UK' |
                        covid_19_data$Country.Region == 'France',]
```

```
#            El            mostrem            per            pantalla:
head(Europe)

##     SNo       Date Province.State Country.Region Confirmed Deaths Reco
vered
## 125 125 2020-01-24              -          France         2      0
0
## 166 166 2020-01-25              -          France         3      0
0
## 212 212 2020-01-26              -          France         3      0
0
## 259 259 2020-01-27              -          France         3      0
0
## 310 310 2020-01-28              -          France         4      0
0
## 319 319 2020-01-28        Bavaria         Germany         4      0
0
```

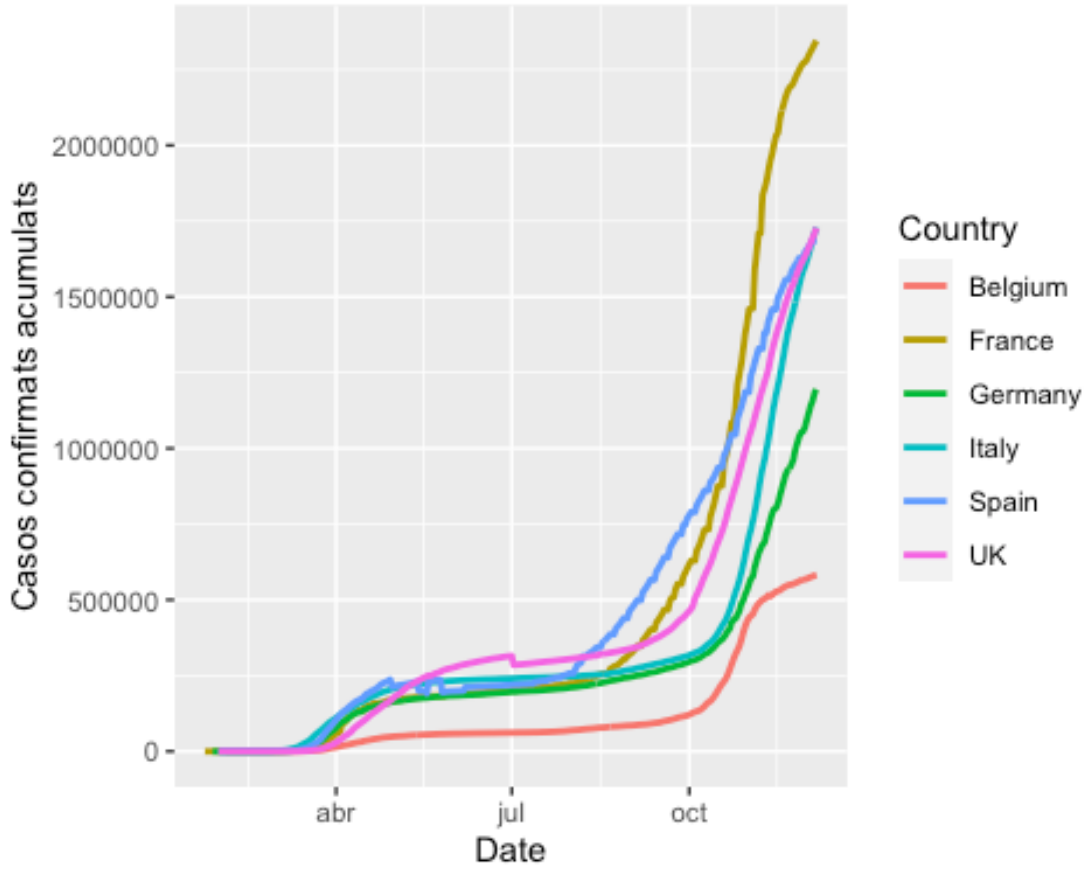
No obstant això, les dades estan classificades per cada divisió territorial de província o estat referent al país en qüestió, mentre que nosaltres volem representar-les per a tot el país de forma global, així que les agruparem totes en funció de la data per obtenir el total de cada dia pel país en general:

```
# Agrupem les dades en funció de la data per obtenir els casos globals a
tot                              el                              país:
Europe <- aggregate(list(Europe$Confirmed, Europe$Deaths, Europe$Recovere
d),    by    =    list(Europe$Date,    Europe$Country.Region),    sum)

#      Recodifiquem      el      nom      de      les      columnes:
colnames(Europe) <- c("Date", "Country", "Confirmed", "Deaths", "Recovere
d")

#Mostrem          els          canvis          per          pantalla:
head(Europe)

##                        Date   Country   Confirmed   Deaths   Recovered
##  1  2020-02-04  Belgium                1           0                0
##  2  2020-02-05  Belgium                1           0                0
##  3  2020-02-06  Belgium                1           0                0
##  4  2020-02-07  Belgium                1           0                0
##  5  2020-02-08  Belgium                1           0                0
## 6 2020-02-09 Belgium         1      0         0
```

Una vegada les dades es troben correctament indexades, ja podem realitzar una primera exploració visual, per exemple, obtenint la corba de casos confirmats acumulats cada dia per país de la unió europea:

```
#          Carreguem          la          llibreria          necessària:
library(ggplot2)

# Creem un gràfic que representi els casos confirmats acumulats cada dia
```

```r
per                                                                    país:
Europe %>% ggplot(aes(x=Date, y=Confirmed, group=Country, color=Country))
+ geom_line(size=1) + ylab("Casos confirmats acumulats")
```



Com podem observar, la majoria de països europeus van patir la **primera onada** entre el març i l'abril del 2020, on es pot observar un clar increment dels casos confirmats acumulats, mentre que la **segona onada** va arribar entre setembre i novembre, on el nombre de casos confirmats acumulats van patir un increment molt més significatiu que no pas en la primera onada, principalment degut a l'increment de testos de diagnòsi realitzats.

Com les dades no estan actualitzades fins al 2021, no es podria analitzar amb detall la segona onada, per aquest motiu en aquest anàlisi ens centrarem en la primera.

Concretament, compararem la incidència del covid-19 en la primera onada per a dos dels països més castigats segons el següent article: "https://elpais.com/sociedad/2020-12-07/la-segunda-ola-de-covid-19-ya-ha-provocado-mas-muertes-en-la-union-europea-que-la-primera.html", com són **Itàlia** i **França**.

D'aquesta manera, començarem seleccionant les dades referents a cada país necessàries per a poder-los comparar:

```r
# Seleccionem les dades referents a França en un nou dataframe:
France <- Europe[Europe$Country == 'France',]

# El mostrem per pantalla:
head(France)
```

```
##                         Date  Country  Confirmed  Deaths  Recovered
##  308  2020-01-24    France               2          0               0
##  309  2020-01-25    France               3          0               0
##  310  2020-01-26    France               3          0               0
##  311  2020-01-27    France               3          0               0
##  312  2020-01-28    France               4          0               0
## 313 2020-01-29  France         5      0         0
```

I realitzem el mateix per Itàlia:

```r
# Seleccionem les dades referents a França en un nou dataframe:
Italy <- Europe[Europe$Country == 'Italy',]

# El mostrem per pantalla:
head(Italy)
```

```
##                         Date  Country  Confirmed  Deaths  Recovered
##  940  2020-01-31     Italy               2          0               0
##  941  2020-02-01     Italy               2          0               0
##  942  2020-02-02     Italy               2          0               0
##  943  2020-02-03     Italy               2          0               0
##  944  2020-02-04     Italy               2          0               0
## 945 2020-02-05   Italy         2      0         0
```

Ara bé, per a poder delimitar correctament on i quan acaba la primera onada de covid-19, és millor treballar amb els nous casos diaris i no amb els acumulats, així que crearem dues noves columnes que calculin els nous casos i morts diàries a partir de les acumulades:

```r
# Carreguem la llibreria necessària:
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
##   The   following   objects   are   masked   from   'package:dplyr':
##
##     between, first, last
```

```r
# Creem una nova columna que calculi els nous casos diaris a partir dels acumulats:
setDT(France)[, Confirmed_Daily := Reduce(`-`, shift(Confirmed, 0:1))]

# Creem una nova columna que calculi les noves morts diaries a partir de les acumulades:
```

```
setDT(France)[,  Deaths_Daily  :=  Reduce(`-`,  shift(Deaths,  0:1))]
France[France            <            0]            <-            0

#        Mostrem        els        canvis        per        pantalla:
head(France)

##          Date Country Confirmed Deaths Recovered Confirmed_Daily Death
s_Daily
## 1: 2020-01-24  France         2      0         0               NA
NA
## 2: 2020-01-25  France         3      0         0                1
0
## 3: 2020-01-26  France         3      0         0                0
0
## 4: 2020-01-27  France         3      0         0                0
0
## 5: 2020-01-28  France         4      0         0                1
0
## 6: 2020-01-29  France         5      0         0                1
0
```

I ho repetim per a Itàlia:

```
#          Carreguem          la          llibreria          necessària:
library(data.table)

# Creem una nova columna que calculi els nous casos diaris a partir dels
acumulats:
setDT(Italy)[,  Confirmed_Daily  :=  Reduce(`-`,  shift(Confirmed,  0:1))]

# Creem una nova columna que calculi les noves morts diaries a partir de
les                                                          acumulades:
setDT(Italy)[,  Deaths_Daily  :=  Reduce(`-`,  shift(Deaths,  0:1))]
Italy[Italy            <            0]            <-            0

#        Mostrem        els        canvis        per        pantalla:
head(Italy)

##          Date Country Confirmed Deaths Recovered Confirmed_Daily Death
s_Daily
## 1: 2020-01-31   Italy         2      0         0               NA
NA
## 2: 2020-02-01   Italy         2      0         0                0
0
## 3: 2020-02-02   Italy         2      0         0                0
0
## 4: 2020-02-03   Italy         2      0         0                0
0
## 5: 2020-02-04   Italy         2      0         0                0
0
```

```
## 6: 2020-02-05    Italy          2       0          0                 0
0
```

Ja per últim, agruparem les dades de les dues noves columnes que hem creat per setmana i no per dia per tal que la seva representació gràfica sigui més visual i delimitada. Això ho podem realitzar amb el següent codi:

```
# Carreguem la llibreria necessària:
library(zoo)

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

# Agrupem les dades dels nous casos diaris per setmana:
Confirmed_by_week_1 <- rollapply(France$Confirmed_Daily, 7, sum, by = 7)

# Agrupem les dades de les noves morts diaries per setmana:
Deaths_by_week_1 <- rollapply(France$Deaths_Daily, 7, sum, by = 7)

# Creem un dataframe per aquestes noves dades:
France_by_week <- data.frame(Confirmed_by_week_1, Deaths_by_week_1)

# Agefim una columna per indicar quina setmana és:
France_by_week$Setmana <- 1:nrow(France_by_week)

# Mostrem els canvis per pantalla:
tail(France_by_week)

##                   Confirmed_by_week_1     Deaths_by_week_1     Setmana
##  40                        285861                    1821          40
##  41                        321137                    3030          41
##  42                        299145                    3936          42
##  43                        188962                    4177          43
##  44                         98441                    3840          44
## 45               74734             3190      45
```

I repetim el procediment per Itàlia:

```
# Carreguem la llibreria necessària:
library(zoo)

# Agrupem les dades dels nous casos diaris per setmana:
Confirmed_by_week_2 <- rollapply(Italy$Confirmed_Daily, 7, sum, by = 7)

# Agrupem les dades de les noves morts diaries per setmana:
Deaths_by_week_2 <- rollapply(Italy$Deaths_Daily, 7, sum, by = 7)
```

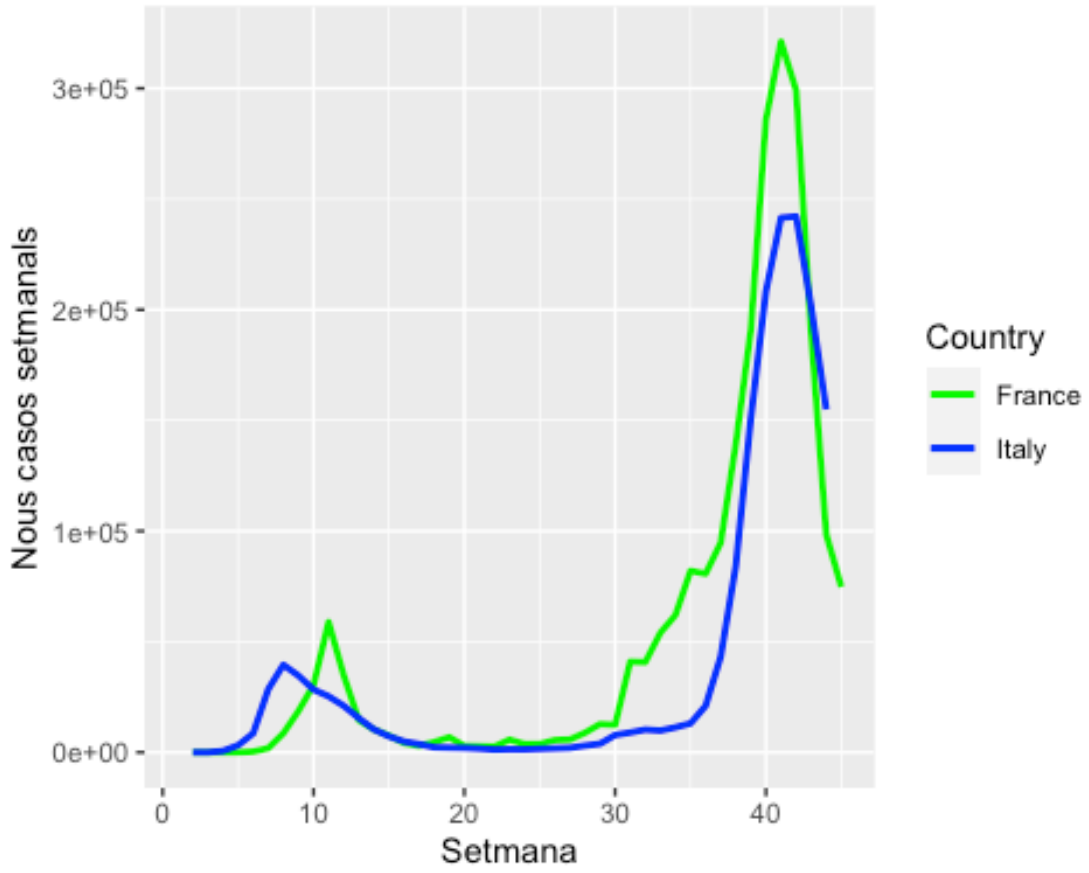
```
# Creem un dataframe per aquestes noves dades:
Italy_by_week <- data.frame(Confirmed_by_week_2, Deaths_by_week_2)

# Agefim una columna per indicar quina setmana és:
Italy_by_week$Setmana <- 1:nrow(Italy_by_week)

# Mostrem els canvis per pantalla:
tail(Italy_by_week)
```

```
##    Confirmed_by_week_2 Deaths_by_week_2 Setmana
## 39              150869              1154      39
## 40              208284              2070      40
## 41              241522              3397      41
## 42              242127              4281      42
## 43              201347              4980      43
## 44              154954              5188      44
```

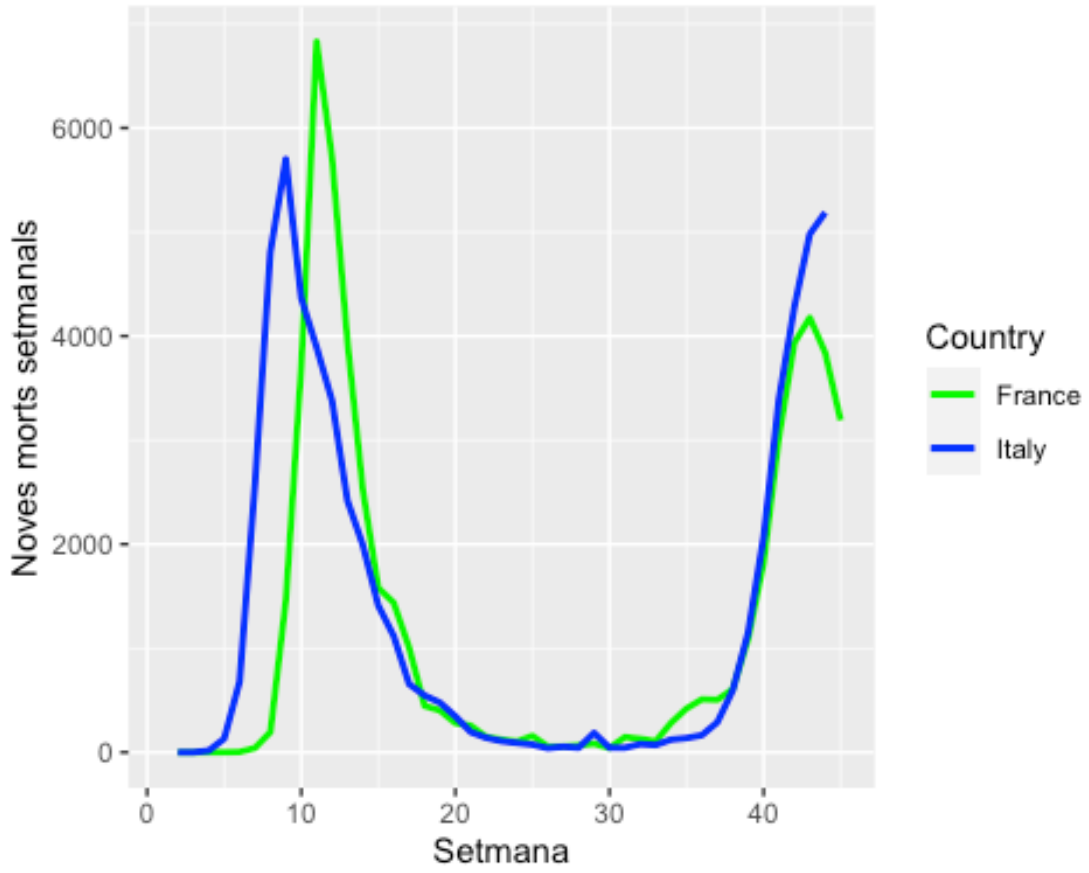
Una vegada obtinguts els nous casos i morts diàries per a Itàlia i França, i agrupats per setmana, ja podem representar-los gràficament per a poder delimitar correctament la primera onada i procedir amb els test d'hipòtesis:

```
# Carreguem la llibreria necessària:
library(ggplot2)

# Creem un gràfic que representi els nous casos diàris agrupats per setmana:
ggplot() +
geom_line(data=France_by_week, aes(x=Setmana, y=Confirmed_by_week_1, colour = 'France'), size=1) + ylab('Nous casos setmanals') +
geom_line(data=Italy_by_week, aes(x=Setmana, y=Confirmed_by_week_2, colour = 'Italy'), size=1) + scale_colour_manual(name= 'Country', values=c("green", "blue"))
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```
# Creem un gràfic que representi les noves morts diàries agrupades per se
tmana:
ggplot()                                                                +
geom_line(data=France_by_week, aes(x=Setmana, y=Deaths_by_week_1, colour
=    'France'),    size=1)    +    ylab('Noves    morts    setmanals')    +
geom_line(data=Italy_by_week, aes(x=Setmana, y=Deaths_by_week_2, colour =
'Italy'), size=1) + scale_colour_manual(name ='Country', values=c("green"
, "blue"))

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```



A partir d'aquests gràfics, podem extreure tres tipus d'informació:

- **Primera:** Podem observar que tant pel nombre de casos nous setmanals com per les morts, la primera onada estaria delimitada entre les 20 primeres setmanes de pandèmia, és a dir, entre els primers 5 mesos.

- **Segona:** Durant la primera onada, tot i presentar 3 vegades menys número de contagis nous setmanals que no pas la segona onada, es van produir moltes més morts, la qual cosa reforça la idea anterior que durant la segona onada s'han realitzat molts més tests de diagnòsi que han permès detectar superiors casos confirmats.

- **Tercera:** Observant únicament les dades dels gràfics, sembla ser que la incidència pel que fa al nombre de casos confirmats així com de morts durant la primera onada va ser superior en França que no pas Itàlia (gairabé 20.000 contagis i 1.000 morts més). No obstant això, cal tractar bé les dades i realitzar els càlculs per poder determinar si estadísticament es pot afirmar que França va ser més castigada que Itàlia.

Així doncs, per a realitzar el contrast d'hipòtesis utilitzarem els primers 5 mesos de pandèmia dels quals tenim dades, és a dir, des del 24.01.2020 fins al 24.06.2020. Ara

bé, per poder disposar de major quantitat de dades, utilitzarem les dades diàries i no les agrupades per setmana:

```
# Limitem les dades diàries entre les dates comentades:
France_test                        <-                      France[1:153,]

# Mostrem per pantalla les dades:
head(France_test)
```

```
##          Date Country Confirmed Deaths Recovered Confirmed_Daily Death
s_Daily
## 1: 2020-01-24  France         2      0         0               NA
NA
## 2: 2020-01-25  France         3      0         0                1
0
## 3: 2020-01-26  France         3      0         0                0
0
## 4: 2020-01-27  France         3      0         0                0
0
## 5: 2020-01-28  France         4      0         0                1
0
## 6: 2020-01-29  France         5      0         0                1
0
```

Repetim el mateix per Itàlia:

```
# Limitem les dades diàries entre les dates comentades:
Italy_test                        <-                      Italy[1:153,]

# Mostrem per pantalla les dades:
head(Italy_test)
```

```
##          Date Country Confirmed Deaths Recovered Confirmed_Daily Death
s_Daily
## 1: 2020-01-31   Italy         2      0         0               NA
NA
## 2: 2020-02-01   Italy         2      0         0                0
0
## 3: 2020-02-02   Italy         2      0         0                0
0
## 4: 2020-02-03   Italy         2      0         0                0
0
## 5: 2020-02-04   Italy         2      0         0                0
0
## 6: 2020-02-05   Italy         2      0         0                0
0
```

4.1.2. Comprovació de la normalitat i homogeneïtat de la variància

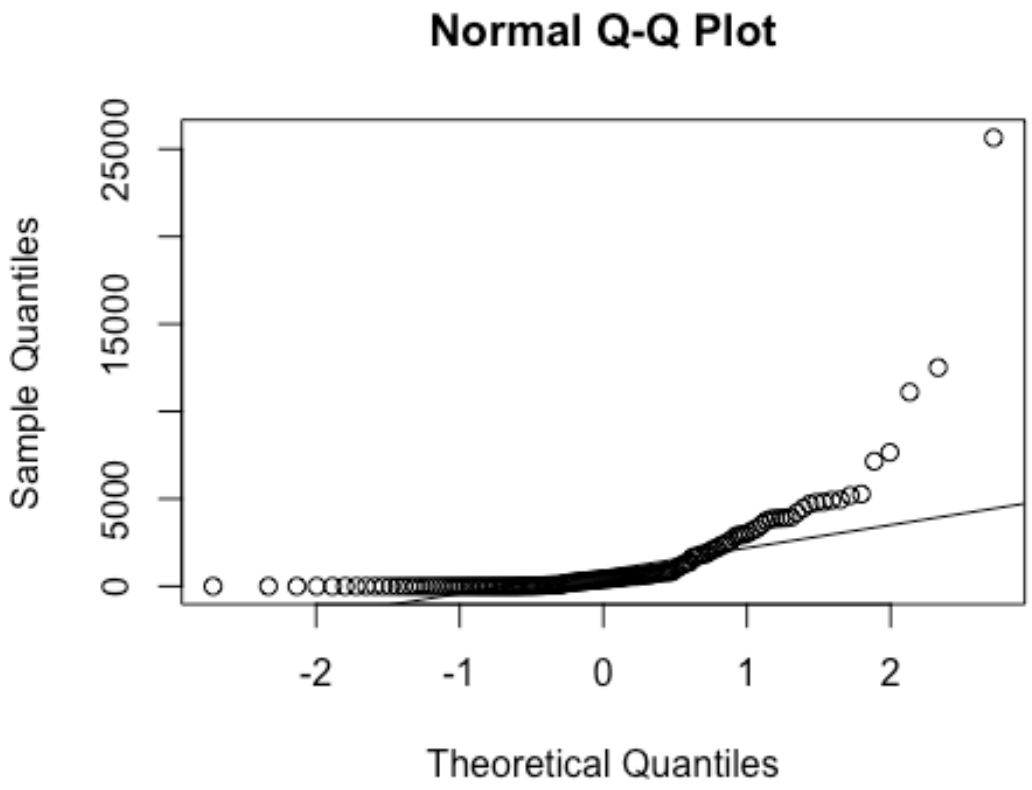
Una vegada argumentat com s'han seleccionat les dades que es desitgen analitzar i comparar, ja estem en disposició de comprovar-ne la seva normalitat i homogeneïtat de la variància.

Com sabem, a l'hora de revisar la normalitat de les dades cal tenir en compte la mida de la mostra. D'aquesta manera, **per a mostres suficientment grans (on per grans se sol considerar n > 30 elements) es pot considerar l'aplicació del teorema del límit central (TLC)**. Per altra banda, **si la mida de la mostra és petita (normalment n < 30 elements), es poden utilitzar altres tests de normalitat de les dades, com per exemple el test de normalitat Shapiro-Wilk**.

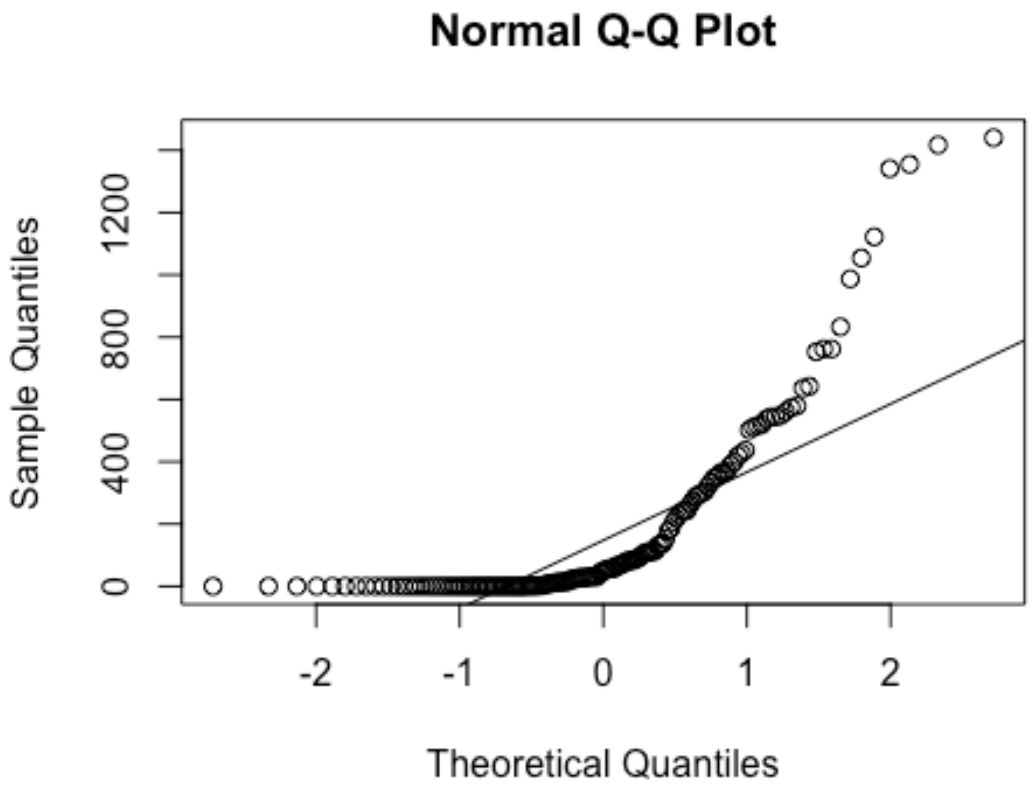
En el nostre exemple, tant pel cas de França com Itàlia, les variables referents als casos nous diaris confirmats, així com les morts, estan formats per més de 150 elements (n > 30), així que per revisar l'assumpció de normalitat utilitzarem el teorema del límit central (TLC).

Per tant, **ja tindríem una primera assumpció de normalitat de les nostres dades**. A més, en els apunts també hem vist que de manera complementària es poden emprar visualitzacions de dades per comprovar la normalitat de les dades. Concretament, hem vist la utilització del gràfic Q-Q, on la Q denota quantil, i es tracta d'un tipus de visualització que s'utilitza per a diagnosticar la desviació de les dades de la mostra en relació amb una població normal. Aquest es pot aplicar amb el següent codi:

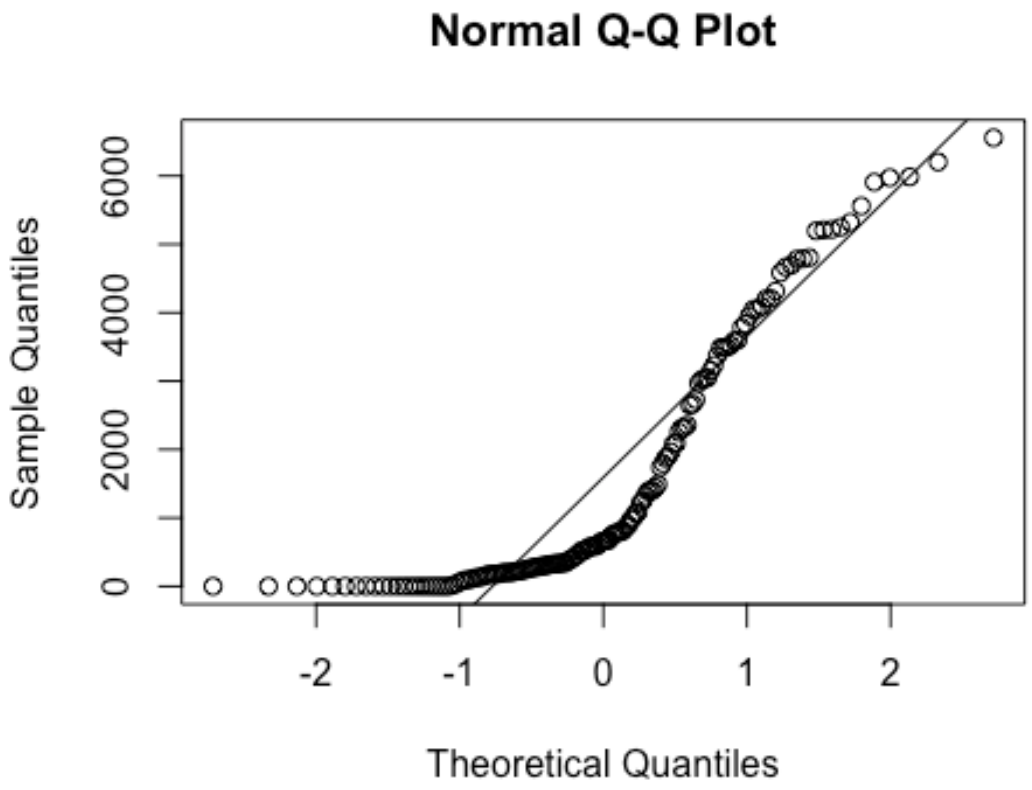
```
# Creem el gràfic Q-Q per a la variable de casos nous diàris confirmats e
n              el              cas              de              França:
qqnorm(France_test$Confirmed_Daily)
qqline(France_test$Confirmed_Daily)
```



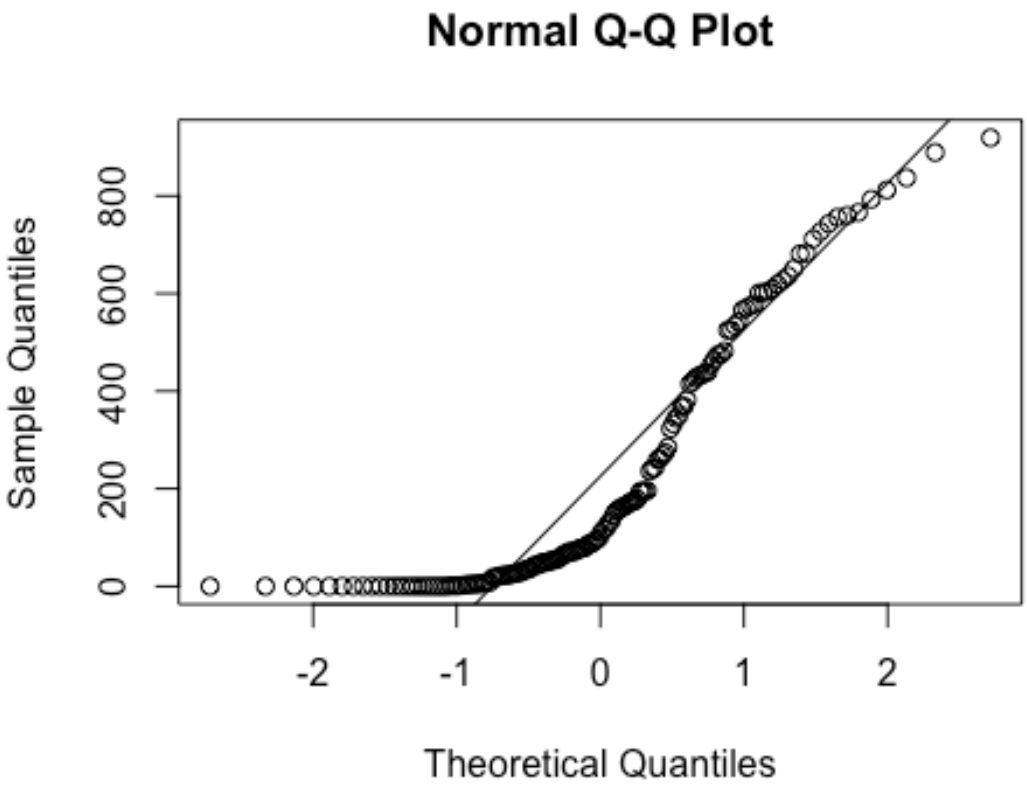

```r
# Creem el gràfic Q-Q per a la variable de noves morts diàries confirmades en el cas de França:
qqnorm(France_test$Deaths_Daily)
qqline(France_test$Deaths_Daily)
```




```r
# Creem el gràfic Q-Q per a la variable de casos nous diàris confirmats e
n                    el                    cas                    d'Itàlia:
qqnorm(Italy_test$Confirmed_Daily)
qqline(Italy_test$Confirmed_Daily)
```




```r
# Creem el gràfic Q-Q per a la variable de noves morts diàries confirmades en el cas d'Itàlia:
qqnorm(Italy_test$Deaths_Daily)
qqline(Italy_test$Deaths_Daily)
```



Com podem observar, aquests gràfics mostren una línia diagonal que correspondria a la distribució normal teòrica (generada per la funció qqline) i cada un dels elements de la mostra són els punts del conjunt de dades distribuïts segons els quantils teòrics. Així doncs, quan la mostra segueix una distribució normal, els punts queden representats sobre de la línia diagonal.

A partir del Teorema del Límit Central (TLC) hem fet una primera justificació de que les dades complien la normalitat, i **a partir dels gràfics Q-Q podem observar que les dades referents a Itàlia presenten una major normalitat que no pas les de França** en adaptar-se millor a la línia diagonal.

Pel que fa a les variàncies d'aquestes variables, podem obtenir-les amb el següent codi:

```
# Calculem la variància de la variable referent als nous casos diàris confirmats a França:
print(paste("Els nous casos diàris confirmats a França presenten un valor de variància de", var(France_test$Confirmed_Daily, na.rm=TRUE)))
```

```
## [1] "Els nous casos diàris confirmats a França presenten un valor de variància de 7974417.40811258"
```

```
# Calculem la variància de la variable referent a les noves morts diàries de França:
```

```r
print(paste("Les noves morts diàries a França presenten un valor de varià
ncia de", var(France_test$Deaths_Daily, na.rm=TRUE)))
```

```
## [1] "Les noves morts diàries a França presenten un valor de variància
de 95757.8323021959"
```

```r
# Calculem la variància de la variable referent als nous casos diàris con
firmats a Itàlia:
print(paste("Els nous casos diàris confirmats a França presenten un valor
de variància de", var(Italy_test$Confirmed_Daily, na.rm=TRUE)))
```

```
## [1] "Els nous casos diàris confirmats a França presenten un valor de v
ariància de 3314558.69344719"
```

```r
# Calculem la variància de la variable referent a les noves morts diàries
a Itàlia:
print(paste("Les noves morts diàries a França presenten un valor de varià
ncia de", var(Italy_test$Deaths_Daily, na.rm=TRUE)))
```

```
## [1] "Les noves morts diàries a França presenten un valor de variància
de 65426.9285029627"
```

A primera vista els valors de variància poden semblar desorbitats, però cal tenir en compte que, com sabem, la variància és una mesura de dispersió que representa la variabilitat d'una sèrie de dades respecte la seva mitjana i en aquest cas les dades poden prendre valors des de 0 fins a més de 25.000 casos diàris confirmats.

4.1.3. Càlculs

Una vegada analitzada la normalitat i homogeneïtat de la variança, podem procedir a aplicar proves estadístiques, com pot ser un **test d'hipòtesis de dues mostres independents sobre la seva mitjana**. Concretament, volem calcular si la mitjana de casos diaris confirmats a França durant la primera onada de covid-19 és significativament superior del cas d'Itàlia, i el mateix per les morts diàries (tal i com suggerien els gràfics a primera vista).

Començarem tractant primer els nous casos diaris, així que, com a **hipotesi nul·la** considerarem que les mitjanes poblacionals dels casos diàris registrats de covid-19 a França (μ1) i a Itàlia (μ2) són iguals, és a dir:

$$H_0: \mu_1 = \mu_2$$

Mentre que, com a **hipòtesi alternativa** considerarem que la mitjana poblacional dels casos diàris registrats de covid-19 a França (μ1) és superior a la mitjana poblacional dels casos diàris d'Itàlia (μ2) tal i com semblava apuntar el gràfic, és a dir:

$$H_1: \mu_1 > \mu_2$$

Per tant, ens trobem en la següent situació: **assumim que les dues mostres provenen de dues poblacions normals independents de les quals desconeixem la mitjana i la variància**. La normalitat la hem garantit amb el raonament anterior, tot i que per a

aplicar el test estadístic adequat, cal comprovar si les variàncies de les dues poblacions són iguals. Per això, **apliquem primer el test d'igualtat de variàncies**:

1) La hipòtesi nul·la i l'alternativa per al test de variàncies són:

$$H_0: \sigma_1^2 = \sigma_2^2$$

$$H_1: \sigma_1^2 \neq \sigma_2^2$$

2) Tal i com sabem, cal aplicar l'estadístic d'igualtat de variàncies següent:

$$f_{obs} = \frac{s_1^2}{s_2^2} \approx F_{n_1-1, n_2-1}$$

3) Calculem el valor observat a partir de les variàncies mostrals:

```
# Càlcul del valor observat a partir de les variàncies de les dues mostres:
f_obs <- (sd(France_test$Confirmed_Daily, na.rm=TRUE)^2)/(sd(Italy_test$Confirmed_Daily, na.rm=TRUE)^2)

# Mostrem el valor observat per pantalla:
f_obs
```

```
## [1] 2.405876
```

4) Es calculen els límits de la zona d'acceptació de la hipòtesi nul·la per a una distribució F amb n1 − 1 = 151 i n2 − 1 = 151 graus de llibertat. Recordem que θ = 0.05 i estem treballant amb un test bilateral. Per tant, cal calcular els llindars tals que la probabilitat a l'esquerra i a la dreta són iguals a θ/2 = 0.025:

```
# Calculem el llindar inferior:
L <- qf(0.025, df1=151, df2=151)

# Calculem el llindar superior:
U <- qf(1-0.025, df1=151, df2=151)

# Mostrem els llindars per pantalla, així com el valor observat:
c(L, U, f_obs)
```

```
## [1] 0.7259989 1.3774125 2.4058761
```

5) Es comprova si el valor observat cau en la zona d'acceptació o de rebuig. Com podem veure, **aquest valor cau fora de la zona d'acceptació de la hipòtesi nul·la de les variàncies**.

6) Anàlogament, podem calcular el valor p. Donat que la distribució F és asimètrica, una manera aproximada de calcular el valor p és la següent:

```
# Calculem el valor p de forma aproximada:
p_value_1 <- min(pf(f_obs, df1=151, df2=151, lower.tail=FALSE), pf(f_obs, df1=151, df2=151))*2
```

```
# Mostrem per pantalla els llindars, el valor obsrvat i el valor p:
c(L, U, f_obs, p_value_1)
```

```
## [1] 7.259989e-01 1.377412e+00 2.405876e+00 1.143522e-07
```

7) Podem observar que el valor p és molt petit (0.0000001), inferior al nostre nivell de significança (θ = 0.05), així que **podem rebutjar la hipòtesi nul·la i concloure que les dues mostres presenten variàncies diferents amb un nivell de confiança del 95 %**.

Per últim, amb la intenció de comprovar si hem realitzat els càlculs correctament, podem realitzar el test estadístic de variàncies amb la funció següent:

```
# Comprovem si hem realitzat correctament el test estadístic de variàncies:
var.test(France_test$Confirmed_Daily, Italy_test$Confirmed_Daily)
```

```
##
##             F       test       to       compare       two       variances
##
##  data:    France_test$Confirmed_Daily  and  Italy_test$Confirmed_Daily
##  F  =  2.4059,  num  df  =  151,  denom  df  =  151,  p-value  =  1.144e-07
##  alternative  hypothesis:  true  ratio  of  variances  is  not  equal  to  1
##           95           percent           confidence           interval:
##                                     1.746663                  3.313884
##                            sample                              estimates:
##                   ratio                   of                   variances
##           2.405876
```

En vista dels resultats, observem que hem d'utilitzar el **test d'hipòtesis de dues mostres independents sobre la mitjana amb variàncies desconegudes diferents**.

Com sabem, el test d'hipòtesis de dues mostres de poblacions independents amb distribucions normals i variàncies desconegudes diferents es realitza amb l'estadístic de contrast següent:

$$t = \frac{\hat{x}_1 - \hat{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \approx t_\upsilon$$

El qual segueix una distribució t d'Student amb υ graus de llibertat, on υ es calcula com:

$$\upsilon = \frac{(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})^2}{\frac{(s_1^2/n_1)^2}{(n_1 - 1)} + \frac{(s_2^2/n_2)^2}{(n_2 - 1)}}$$

Així doncs, comencem a calcular els diferents paràmetres que necessitarem per a realitzar el càlcul:

```r
# Calculem les mitjanes de la variable de casos nous diàris per a la mostra de França i Itàlia:
x1 <- mean(France_test$Confirmed_Daily, na.rm = TRUE)
x2 <- mean(Italy_test$Confirmed_Daily, na.rm = TRUE)

# Calculem les desviacions estàndard de la variable de casos nous diàris per a la mostra de França i Itàlia:
sd1 <- sd(France_test$Confirmed_Daily, na.rm = TRUE)
sd2 <- sd(Italy_test$Confirmed_Daily, na.rm = TRUE)

# Calculem les variàncies de la variable de casos nous diàris per a la mostra de França i Itàlia
var1 <- var(France_test$Confirmed_Daily, na.rm = TRUE)
var2 <- var(Italy_test$Confirmed_Daily, na.rm = TRUE)

# Calculem la mida de les mostres dels casos nous diàris a França i a Itàlia:
n1 <- length(France_test$Confirmed_Daily)
n2 <- length(Italy_test$Confirmed_Daily)

# Calculem els cocients enter les variàncies i la mida de les mostres per simplificar càlculs:
SN1 <- var1/n1
SN2 <- var2/n2

# Mostrem els valors per pantalla:
c(x1, x2); c(sd1, sd2); c(var1, var2); c(n1, n2)
```

```
## [1] 1402.125 1584.908
```

```
## [1] 2823.901 1820.593
```

```
## [1] 7974417 3314559
```

```
## [1] 153 153
```

Amb els paràmetres bàsics calculats, ja ens podem centrar a calcular el valor t observat i els graus de llibertat υ:

```r
# Calculem el valor t observat:
t_obs <- (x1-x2)/(sqrt((SN1+SN2)))

# Calculem els graus de llibertat υ:
v <- ((SN1+SN2)^2)/(((SN1^2)/(n1-1))+((SN2^2)/(n2-1)))

# Mostrem els valors per pantalla:
c(t_obs, v)
```

```
## [1]  -0.6729054 259.7431505
```

A continuació, per a la distribució t d'Student amb υ graus de llibertat, es calculen els marges de la zona d'acceptació.

Recordem que θ = 0.05 i estem treballant amb un test unilateral per la dreta, per tant, només cal calcular el llindar superior dret per tal de determinar la zona d'acceptació de la hipòtesi nul·la:

```
# Calculem el marge superior dret:
MS <- qt(0.05, df=v, lower.tail = FALSE)

# Mostrem el marge calculat per pantalla, així com el valor t observat:
c(MS, t_obs)

## [1]  1.6507412 -0.6729054
```

Com podem observar, **la zona d'acceptació de la hipòtesi nul·la es trobarà entre [-∞ , 1.65], així que el nostre valor t observat (-0,67) queda dins de la zona d'acceptació de la hipotesi nul·la. Per tant, podem concloure que en primera instància els casos nous diàris de covid-19 durant la primera onada a França i a Itàlia són significativament similars**.

Anàlogament, podem calcular el valor p amb el següent codi:

```
# Calculem el valor p:
p_value_2 <- pt(t_obs, df=v, lower.tail=FALSE)

# El mostrem per pantalla:
p_value_2

## [1] 0.7491971
```

Donat que el valor p és superior a θ = 0.05, **concloem que no podem rebutjar la hipòtesi nul·la** i, per tant, **podem afirmar que els casos nous diàris de covid-19 a França i Itàlia són iguals amb un nivell de confiança del 95 %**.

Ja per últim, podem comprovar els càlculs realitzats amb la funció que ho calcula directament:

```
# Comprovem els càlculs realitzats amb una funció que ho calcula directament:
t.test(France_test$Confirmed_Daily, Italy_test$Confirmed_Daily, alternative="greater", var.equal=FALSE)

##
##  Welch Two Sample t-test
##
##  data:  France_test$Confirmed_Daily and Italy_test$Confirmed_Daily
##  t = -0.6707, df = 258.03, p-value = 0.7485
##  alternative hypothesis: true difference in means is greater than 0
##  95 percent confidence interval:
##  -632.6609 Inf
```

```
##                             sample                             estimates:
##          mean          of          x          mean          of          y
##  1402.125  1584.908
```

A continuació, procedirem a repetir el procés per a **les noves morts diàries**, és a dir, calcularem si les morts diàries per covid-19 durant la primera onada a França són superiors a les d'Itàlia, o no.

Per tant, com a **hipotesi nul·la** considerarem que les mitjanes poblacionals de les morts diàries registrades de covid-19 a França (μ1) i a Itàlia (μ2) són iguals, és a dir:

$$H_0: \mu_1 = \mu_2$$

Mentre que, com a **hipòtesi alternativa** considerarem que la mitjana poblacional de les morts diàries registrades de covid-19 a França (μ1) és superior a la mitjana poblacional de les morts diàries d'Itàlia (μ2) tal i com semblava apuntar el gràfic, és a dir:

$$H_1: \mu_1 > \mu_2$$

No obstant això, **en aquest cas no realitzarem tot el desenvolupament teòric per tal de no saturar l'informe d'aquesta pràctica, sinó que realitzarem els càlculs directament amb les funcions var.test i t.test**.

De nou, **assumirem que les dues mostres provenen de dues poblacions normals independents de les quals desconeixem la mitjana i la variància**. La normalitat la hem garantit amb el raonament prèvi a les proves estadístiques, tot i que per a aplicar el test estadístic adequat, cal comprovar si les variàncies de les dues poblacions són iguals. Per això, **apliquem primer el test d'igualtat de variàncies**:

```r
# Realitzem el test d'igualtat de variàncies per a les dues noves mostres
:
var.test(France_test$Deaths_Daily, Italy_test$Deaths_Daily)
```

```
##
##            F       test       to       compare       two       variances
##
##   data:       France_test$Deaths_Daily   and   Italy_test$Deaths_Daily
##  F  =  1.4636,  num  df  =  151,  denom  df  =  151,  p-value  =  0.01984
##  alternative  hypothesis:  true  ratio  of  variances  is  not  equal  to  1
##           95           percent           confidence           interval:
##                                      1.062561                  2.015959
##                              sample                              estimates:
##                   ratio                  of                  variances
##           1.463584
```

La zona d'acceptació de la hipòtesi nul·la serà la mateixa que en el cas anterior, ja que les mostres tenen exactament la mateixa mida, és a dir, entre [0.73 1.38], mentre que el valor f observat és de 1.46.

Per tant, observem que cau fora de la zona d'acceptació i podem afirmar que les mostres presenten variàncies diferents, la qual cosa també es confirma amb el valor p (0.02), el qual inferior al nostre nivell de significança (θ = 0.05), així que **podem rebutjar la hipòtesi nul·la i concloure que les dues mostres presenten variàncies diferents amb un nivell de confiança del 95 %**.

En vista dels resultats, observem que de nou haurem d'utilitzar el **test d'hipòtesis de dues mostres independents sobre la mitjana amb variàncies desconegudes diferents**, el qual podem calcular directament amb el següent codi:

```
# Realitzem el test d'hipòtesis de dues mostres independents sobre la mit
jana        amb        variàncies        desconegudes        diferents:
t.test(France_test$Deaths_Daily, Italy_test$Deaths_Daily, alternative="gr
eater", var.equal=FALSE)

##
##                     Welch          Two          Sample          t-test
##
##   data:       France_test$Deaths_Daily   and   Italy_test$Deaths_Daily
##    t    =    -0.98348,    df    =    291.67,    p-value    =    0.8369
## alternative hypothesis: true difference in means is greater than 0
##           95           percent           confidence           interval:
##                -85.7603                                          Inf
##                              sample                             estimates:
##          mean          of          x          mean          of          y
##  197.0461  229.0724
```

En aquest cas, si es realitzen els càlculs, es pot observar que els graus de llibertat υ són molt similars en el cas anterior (293 respecte els 259 anteriors), la qual cosa implica que quan es calcula el marge superior de la zona d'acceptació de la hipòtesi nul·la, ens torna a donar el valor de 1.65, així que acceptarem la hipòtesi nul·la sempre que el valor observat es trobi entre [-∞ , 1.65].

Podem observar que el nostre valor t observat (-0,98) queda dins de la zona d'acceptació de la hipotesi nul·la. Per tant, **podem concloure que en primera instància les morts diàries de covid-19 durant la primera onada a França i a Itàlia són significativament similars**. A més, el valor p associat (0.84) és superior a θ = 0.05, així que **concloem que no podem rebutjar la hipòtesi nul·la** i, per tant, **podem afirmar que les morts diàries de covid-19 a França i Itàlia són iguals amb un nivell de confiança del 95 %**.

D'aquesta manera podem concloure que cal interpretar els gràfics amb molta cura, ja que, en un primer moment semblava que França presentava més casos confirmats i morts diàries que no pas Itàlia durant la primera onada de covid-19, mentre que, en realitzar els diferents tests estadístics, **hem pogut observar que ni els casos confirmats diaris ni les morts són estadísticament diferents entre tots dos països**.

4.2. Correlació

Una vegada realitzat un anàlisi sobre les dades a nivell europeu, passarem a limitar una mica més l'àmbit d'aplicació del següent cas d'estudi al territori espanyol. Concretament, ens proposem determinar si existeix alguna correlació entre la incidència del virus (casos confirmats, morts i casos recuperats) i característiques demogràfiques de les comunitats autònomes de les quals es disposen dades, com podrien ser la seva superfície, població, densitat de població, etc.

En primer lloc, però, cal obtenir les dades referents al territori Espanyol:

```
# Creem un dataframe per les dades referents a Espanya ordenades per comu
nitat                                                          autònoma:
Spain_by_ccaa <- covid_19_data[covid_19_data$Country.Region == 'Spain',]

#             El             mostrem             per             pantalla:
head(Spain_by_ccaa)

##     SNo       Date Province.State Country.Region Confirmed Deaths Reco
vered
## 552 552 2020-02-01              -          Spain         1      0
0
## 619 619 2020-02-02              -          Spain         1      0
0
## 688 688 2020-02-03              -          Spain         1      0
0
## 758 758 2020-02-04              -          Spain         1      0
0
## 828 828 2020-02-05              -          Spain         1      0
0
## 899 899 2020-02-06              -          Spain         1      0
0
```

Si donem un cop d'ull a les dades, podem observar que entre el 01.02.2020 i el 13.05.2020 les dades referents a Espanya no eren reportades en funció de la comunitat autònoma, sinó de forma general, per aquest motiu dins d'aquest periode de temps l'atribut *'Province.State'* presenta un valor de '-' en comptes del nom de cada comunitat.

Com en aquest anàlisi es desitja cercar correlacions entre les dades i certes característiques de les comunitats autònomes, únicament utilitzarem les dades que estiguin classificades per comunitat, eliminant les del periode de temps anteriorment comentat:

```
# Eliminem les dades d'Espanya que no estan reportades per comunitat autò
noma:
Spain_by_ccaa   <-   Spain_by_ccaa[!(Spain_by_ccaa$Province.State=="-"),]

#        Mostrem        els        canvis        per        pantalla:
head(Spain_by_ccaa)
```

```
##         SNo       Date       Province.State Country.Region Confirmed D
eaths
## 24687 24687 2020-05-14             Andalusia          Spain     12359
1336
## 24690 24690 2020-05-14                Aragon          Spain      5389
836
## 24694 24694 2020-05-14              Asturias          Spain      2356
308
## 24697 24697 2020-05-14              Baleares          Spain      1958
216
## 24712 24712 2020-05-14              Canarias          Spain      2275
151
## 24714 24714 2020-05-14 Castilla - La Mancha          Spain     16470
2852
##                                                                Recovered
##         24687                                                       9918
##         24690                                                       3471
##         24694                                                       1046
##         24697                                                       1492
##         24712                                                       1496
## 24714      6244
```

Una vegada disposem de les dades referents a cada comunitat autònoma, crearem dos nous dataframes, un amb la informació més actualitzada (és a dir, els últims valors acumulats) i un segon amb la més antiga possible (és a dir, els primers valors dels quals disposem), i analitzarem la correlació de tots dos dataframes.

A priori esperem que hi hagi una certa correlació entre la població, o la densitat d'aquesta, i la expansió de la Covid-19 degut a que és una malaltia que es contagia majoritariament per contacte directe (estant a menys de 1.8 metres d'una persona infectada, un temps igual o superior a 15 minuts).

Per crear aquests dos nous dataframes, en primer lloc obtindrem la primera i última data per a les quals es disposen dades de Covid-19. Si fem una primera ullada podem observar que la primera data és el 14.05.2020, tot i que no es disposen de dades per a totes les comunitats autònomes fins al 19.05.2020, així que prendrem com a referència aquesta última data:

```
# Anotem la primera data per a la qual es disposen dades de totes les com
unitats                                                        autònomes:
first_date                          <-                        "2020-05-19"

#  Busquem  la  última  data  per  a  la  qual  disposem  de  dades:
last_date            <-            tail(Spain_by_ccaa$Date,            1)

# Capturem els valors corresponents a la primera data disponible:
first_confirmed <- Spain_by_ccaa$Confirmed[Spain_by_ccaa$Date == first_da
te]
first_deaths  <-  Spain_by_ccaa$Deaths[Spain_by_ccaa$Date  ==  first_date]
first_recovered <- Spain_by_ccaa$Recovered[Spain_by_ccaa$Date == first_da
```

```
te]

# Capturem els valors corresponents a l'última data disponible:
last_confirmed <- Spain_by_ccaa$Confirmed[Spain_by_ccaa$Date == last_date
]
last_deaths <- Spain_by_ccaa$Deaths[Spain_by_ccaa$Date == last_date]
last_recovered <- Spain_by_ccaa$Recovered[Spain_by_ccaa$Date == last_date
]
```

Per completar aquesta informació amb dades oficials, emprarem una informació trobada en el document oficial "España en cifras 2020 - Instituto Nacional de Estadísitca", obtingut de la pàgina del INE (www.ine.es). D'aquí obtenim la superficie en KM^2 i la població en nº d'habitants de cada Comunitat Autònoma, i afegirem totes aquestes dades en els nous dataframes:

```
# Creem un DF amb les dades referents a la última data (2020-12-06):
end_covid_spain <- data.frame(
              poblacio = c(8476718, 1330445, 1018775, 1210750, 2237309,
582357, 2401230, 2045384, 7652069, 5028650, 1061768, 2702244, 6747425, 15
04607,      656487,      218310,      315926,      84032,      84496),
              superficie = c(87600, 47700, 10600, 5000, 7450, 5300, 9420
0, 79500, 32100, 23300, 41600, 29500, 8000, 11300, 10400, 7250, 5050, 19,
13),
              confirmed = last_confirmed,
              deaths = last_deaths,
              recovered = last_recovered)

# Creem un DF amb les dades referents a la primera data (2020-05-19):
start_covid_spain <- data.frame(
              poblacio = c(8476718, 1330445, 1018775, 1210750, 2237309,
582357, 2401230, 2045384, 7652069, 5028650, 1061768, 2702244, 6747425, 15
04607,      656487,      218310,      315926,      84032,      84496),
              superficie = c(87600, 47700, 10600, 5000, 7450, 5300, 9420
0, 79500, 32100, 23300, 41600, 29500, 8000, 11300, 10400, 7250, 5050, 19,
13),
              confirmed = first_confirmed,
              deaths = first_deaths)

# Agefim els noms de les comunitats autònomes al primer DF:
row.names(end_covid_spain) <- c('Andalucia','Aragon','Asturias','Balears'
,'Canarias','Cantabria','Castilla y Leon','Castilla la Mancha','Catalunya
','Comunitat Valenciana','Extremadura','Galicia','Madrid','Murcia','Navar
ra','Pais Vasco','La Rioja','Ceuta','Melilla')

# Agefim els noms de les comunitats autònomes al segon DF:
row.names(start_covid_spain) <- c('Andalucia','Aragon','Asturias','Balear
s','Canarias','Cantabria','Castilla y Leon','Castilla la Mancha','Catalun
ya','Comunitat Valenciana','Extremadura','Galicia','Madrid','Murcia','Nav
arra','Pais Vasco','La Rioja','Ceuta','Melilla')
```

```
# Creem una nova columna que calculi la densitat de població (habitants/km^2):
end_covid_spain$population_dens <- end_covid_spain$poblacio / end_covid_spain$superficie
start_covid_spain$population_dens <- start_covid_spain$poblacio / start_covid_spain$superficie
```

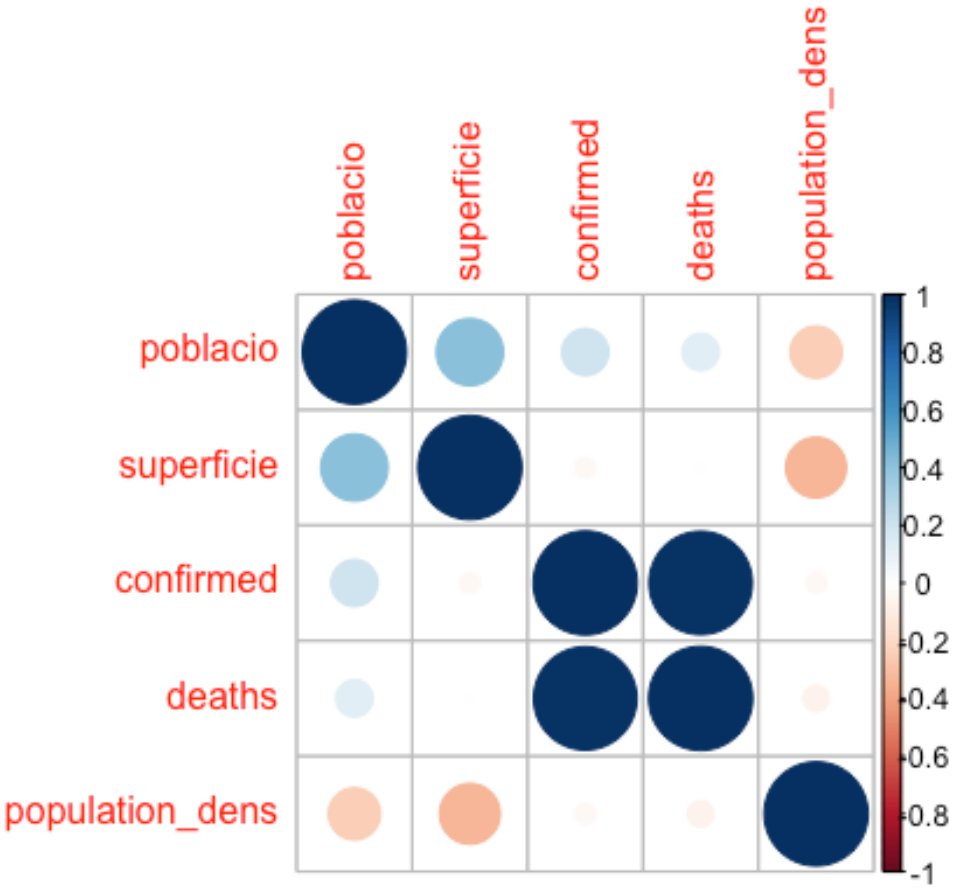
Per al dataframe referent a la primera data a partir de la qual disposem dades no tenim informació dels casos recuperats, per tant hem exclos aquest factor. Preferim excloure'l abans de cercar una data posterior en la que tinguem la informació perquè com veurem posteriorment hi ha una forta correlació entre Infectats-Morts-Recuperats, i si trobem alguna correlaciò amb Infectats i Morts, inevitablement també ho serà amb Recuperats.

Una vegada tenim les dades per a cada comunitat autònoma en dos dataframes diferents, un per les primeres dades disponibles i un altre per les últimes, ja podem analitzar-ne la seva correlació. Per fer-ho, en primer lloc, si és necessàri, cal instal·lar el paquet corresponent:

```
# Instal·lem el paquet a utilitzar en cas que sigui necessari:
# install.packages("corrplot")
```

A continuació, ja podem calcular la correlació de les primeres dades disponibles (19 de Maig, 2020):

```
# Carreguem la llibreria necessaria:
library(corrplot)

## corrplot 0.84 loaded

# Calculem les correlacions:
start_covid_spain.cor <- cor(start_covid_spain)
corrplot(start_covid_spain.cor)
```

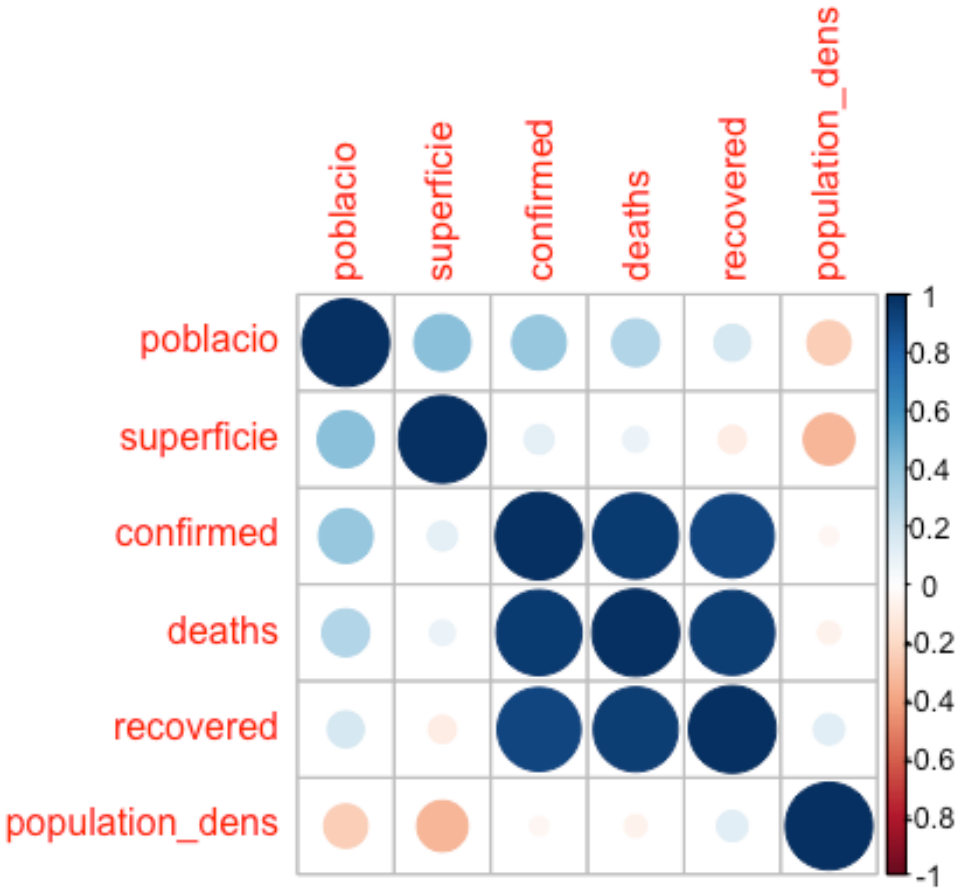



```
start_covid_spain.cor
```

```
##                   poblacio  superficie   confirmed      deaths populat
ion_dens
## poblacio         1.0000000  0.41447974  0.20112218  0.12581172     -0.
24881409
## superficie       0.4144797  1.00000000 -0.03477931  0.00358474     -0.
33729380
## confirmed        0.2011222 -0.03477931  1.00000000  0.98671412     -0.
03542453
## deaths           0.1258117  0.00358474  0.98671412  1.00000000     -0.
06145275
## population_dens -0.2488141 -0.33729380 -0.03542453 -0.06145275      1.
00000000
```

I repetir el procés amb les últimes dades disponibles (6 de Desembre, 2020):

```
#          Carreguem          la          llibreria          necessaria:
library(corrplot)

#                Calculem                les                correlacions:
end_covid_spain.cor                <-                cor(end_covid_spain)
corrplot(end_covid_spain.cor)
```

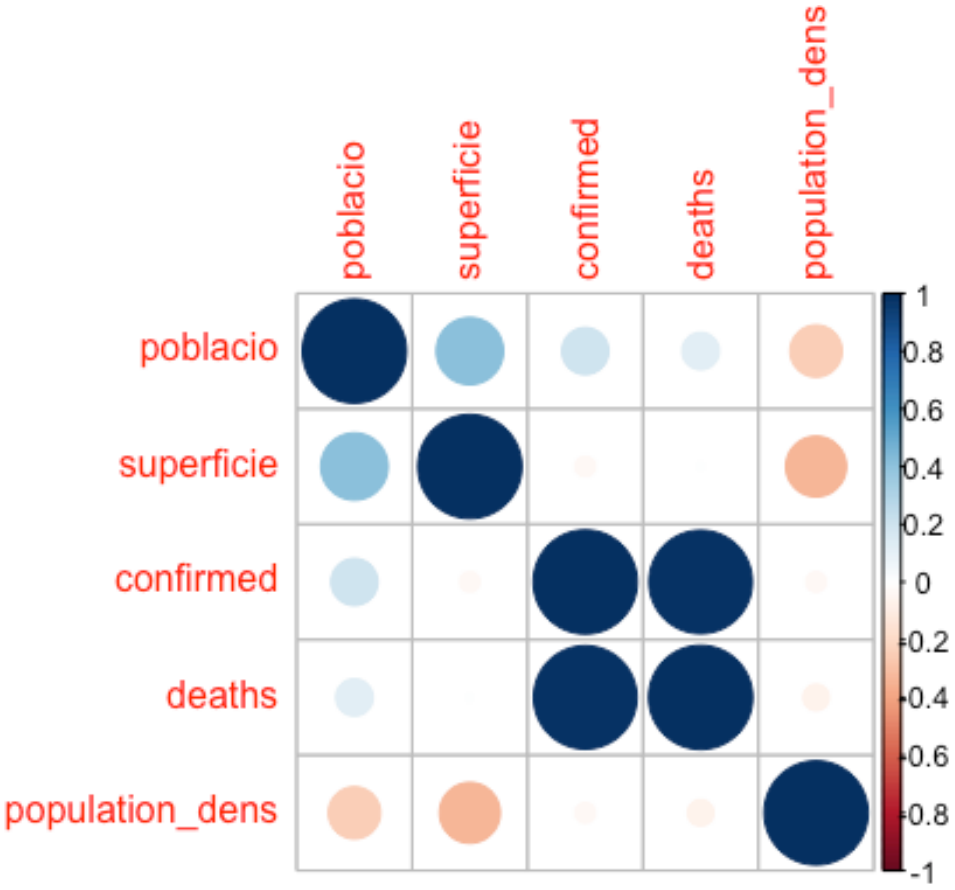



```
end_covid_spain.cor

##                    poblacio  superficie   confirmed      deaths  recove
red
## poblacio          1.0000000  0.41447974  0.38204287  0.29181195  0.1712
443
## superficie        0.4144797  1.00000000  0.11072731  0.08179018 -0.0975
157
## confirmed         0.3820429  0.11072731  1.00000000  0.95819611  0.9173
899
## deaths            0.2918119  0.08179018  0.95819611  1.00000000  0.9405
930
## recovered         0.1712443 -0.09751570  0.91738991  0.94059300  1.0000
000
## population_dens -0.2488141 -0.33729380 -0.04220743 -0.06392848  0.1226
615
##                                                        population_dens
##    poblacio                                                -0.24881409
##     superficie                                             -0.33729380
##     confirmed                                              -0.04220743
##    deaths                                                  -0.06392848
##     recovered                                               0.12266153
## population_dens      1.00000000
```

Tal i com podem observar, no es detecta cap diferència significativa entre les correlacions del 19/05/2020 i el 06/12/2020.

Per últim, repetirem l'anàlisi per a les dades inicials però ara normalitzant-les prèviament per veure si el fet de normalitzar-les té algun tipus d'influència:

```
# Normalitzem les dades referents a la primera data:
covid_spain_normal <- scale(start_covid_spain)

# Calculem de nou les correlacions:
covid_spain_normal.cor <- cor(covid_spain_normal)
corrplot(covid_spain_normal.cor)
```




```
covid_spain_normal.cor

##                  poblacio  superficie   confirmed      deaths populat
ion_dens
## poblacio         1.0000000  0.41447974  0.20112218  0.12581172     -0.
24881409
## superficie       0.4144797  1.00000000 -0.03477931  0.00358474     -0.
33729380
## confirmed        0.2011222 -0.03477931  1.00000000  0.98671412     -0.
03542453
## deaths           0.1258117  0.00358474  0.98671412  1.00000000     -0.
```

```
06145275
## population_dens -0.2488141 -0.33729380 -0.03542453 -0.06145275      1.
00000000
```

Observem com normalitzar les dades no té cap impacte en la corelació ja que obtenim exactament les mateixes correlacions, però ens servirà per aplicar a continuació tècniques de clustering.

4.3. Clustering

Amb les dades més recents aplicarem el mètode de k-means, per buscar

Una vegada estudiada la correlació entre les diferents variables per a les dades referents a Espanya classificades per Comunitat Autònoma, en aquest anàlisi **ens proposem buscar una possible classificació de les Comunitats Autònomes en funció de les seves característiques i les dades de la Covid-19 de que disposem**.

Per fer-ho, **utilitzarem el mètode de k-means** sobre les dades anteriors normalitzades i partirem de la hipòtesis que si som capaços trobar una classificació adequada, es poden implementar mesures específiques per a cada grup, podent així definir mesures més específiques i efectives.

Partirem per instal·lar els paquets necessaris que utilitzarem en cas que fos necessari:

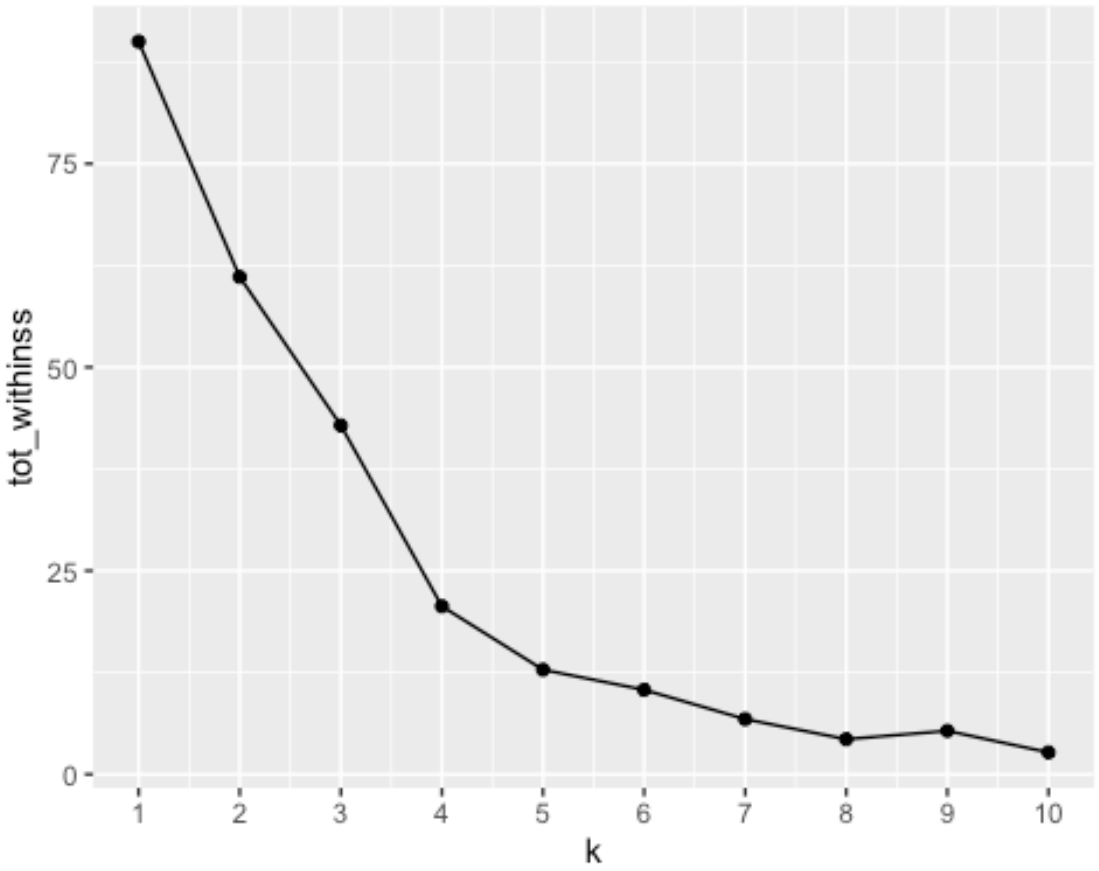
```
# En cas necessari, instal·lem les llibreries a utilitzar:
#                                   install.packages("factoextra")
# install.packages("tidyverse")
```

Per tal d'aplicar el mètode k-means, primer haurem d'estimar el nombre de Clusters. Farem això mitjançant dos mètodes diferents: **"l'Elbow Method"** i el **"Silhouette"**.

Començarem amb el mètode anomenat "Elbow Method", que consisteix en executar un conjunt de tests per a diferent nombre de clusters i representar la distància mitja entre les dades i el centre del seu cluster. El valor de 'k' que escollim serà el punt d'inflexió en el que la corva s'aplani. Això ho podem realitzar amb el següent codi:

```
# Carreguem les llibreries necessàries:
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https
://goo.gl/ve3WBa

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyvers
e 1.3.0 ──

##   ✓   tibble       3.0.3                ✓   purrr          0.3.4
## ✓ tidyr   1.1.2     ✓ forcats 0.5.0
```

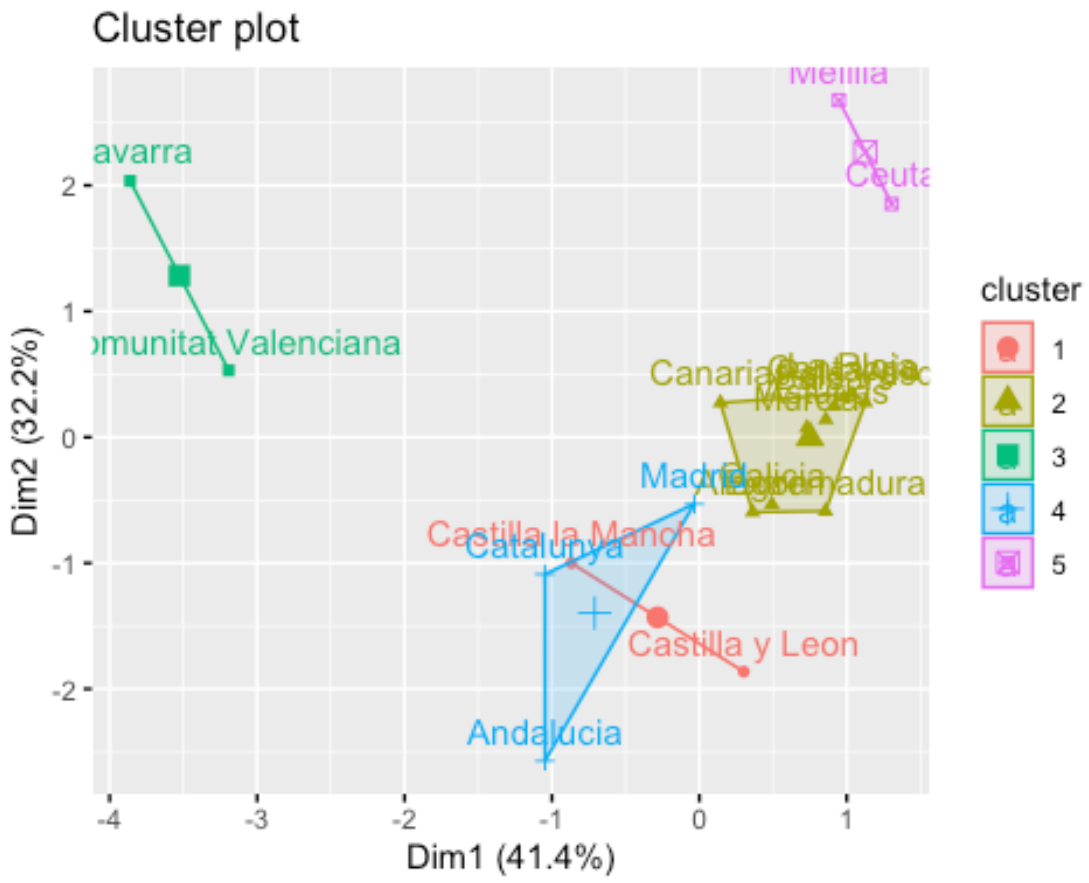
```
## ── Conflicts ─────────────────────────────────────── tidyverse_conf
licts() ──
##       x       data.table::between()       masks       dplyr::between()
##      x      gridExtra::combine()           masks      dplyr::combine()
##    x    dplyr::filter()                       masks    stats::filter()
##      x      data.table::first()               masks      dplyr::first()
##   x   dplyr::lag()                               masks   stats::lag()
##     x     data.table::last()                    masks     dplyr::last()
## x purrr::transpose()    masks data.table::transpose()

# Fem servir map_dbl per executar varis models canviant el valor de k:
tot_withinss        <-        map_dbl(1:10,                function(k){
  model    <-    kmeans(x    =    covid_spain_normal,    centers    =    k)
  model$tot.withinss})

# Generem un data frame que contingui els diferents valors possibles de '
k'                            i                            'tot_withinss':
elbow_df                          <-                          data.frame(
  k                                =                                1:10,
  tot_withinss                       =                      tot_withinss)

#           Representem            el            "elbow            plot":
ggplot(elbow_df,     aes(x     =     k,     y     =     tot_withinss))     +
  geom_line()                       +                      geom_point()+
  scale_x_continuous(breaks = 1:10)
```



Segons aquest mètode, podem agrupar les dades en **cinc grups** de comunitats diferens.

A continuació procedim a aplicar el k-means:

```
# Càlcul per a 5 clústers:
test_5k <- kmeans(covid_spain_normal, centers = 5)
fviz_cluster(test_5k, geom = c("point","text"), data = covid_spain_norma
l)
```



Així doncs, les comunitats queden classificades en els següents grups:

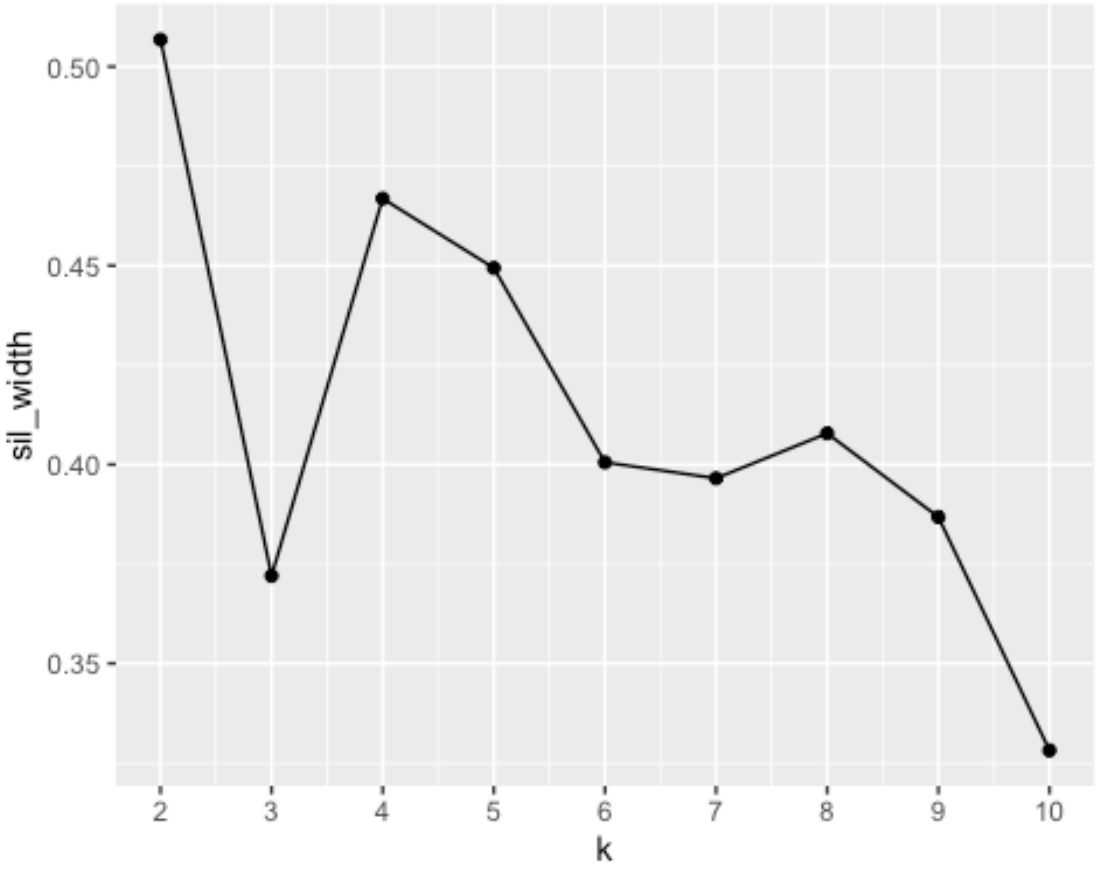
```
# Mostrem per pantalla la classificació de les diferents comunitats autòn
omes:
print(test_5k$cluster)
```

```
##               Andalucia                  Aragon                 Asturias
##                       4                       2                       2
##                 Balears                Canarias                Cantabria
##                       2                       2                       2
##        Castilla y Leon    Castilla la Mancha                  Catalunya
##                       1                       1                       4
## Comunitat Valenciana             Extremadura                    Galicia
##                       3                       2                       2
##                  Madrid                  Murcia                  Navarra
##                       4                       2                       3
##             Pais Vasco                La Rioja                   Ceuta
##                       2                       2                       5
##                                                                   Melilla
##                      5
```

Amb el mètode de Silhouette, també executem un test per a diferents valors de 'k', però amb aquest mètode calculem com de similar és un objecte al cúmul d'objectes al qual

pertany. Aquest valor anirà de +1 a -1, +1 indicant un bon aparellament al seu grup i mal aparellament als altres grups:

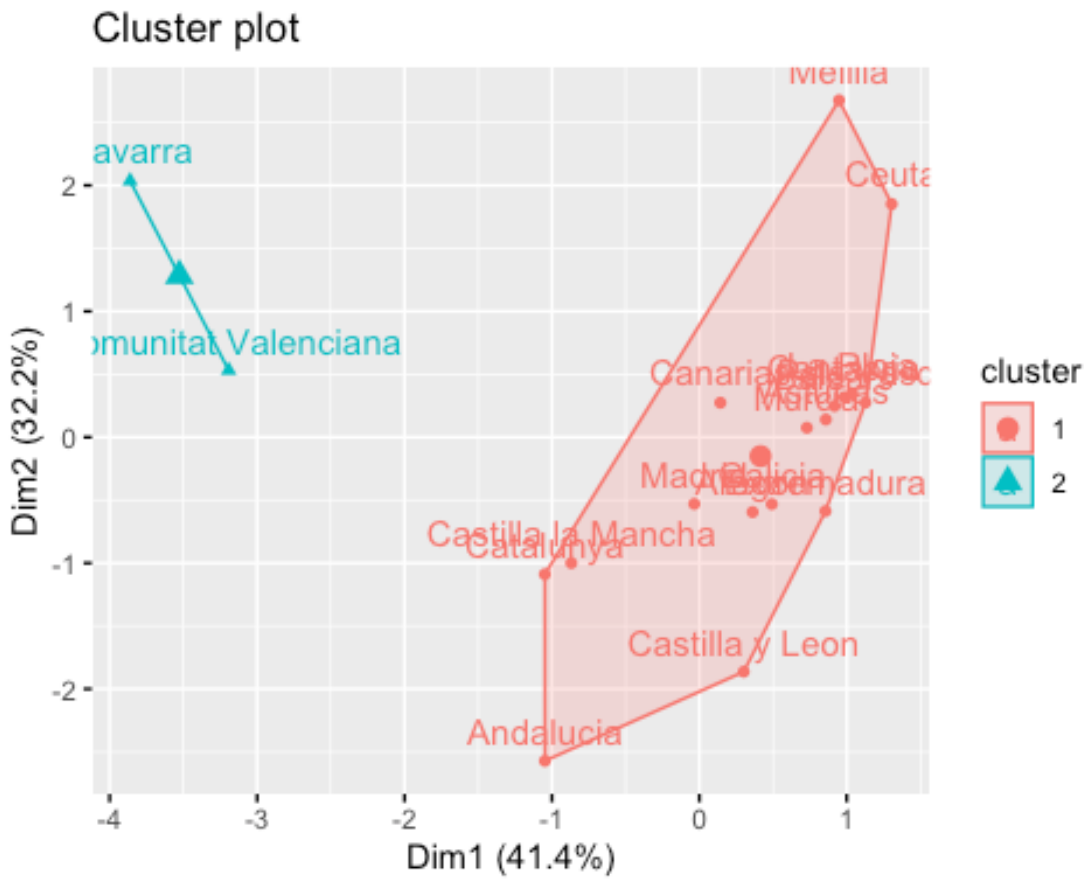
```
# Carreguem les llibreries necessàries:
library(factoextra)
library(tidyverse)
library(cluster)

# Fem servir map_dbl per executar diversos models amb múltiples valors de k:
sil_width <- map_dbl(2:10, function(k){
  model <- pam(x = covid_spain_normal, k = k)
  model$silinfo$avg.width})

# Generem un data frame que contingui 'k' i 'sil_width':
sil_df <- data.frame(
  k = 2:10,
  sil_width = sil_width)

# Representem la relació entre 'k' i 'sil_width':
ggplot(sil_df, aes(x = k, y = sil_width)) +
  geom_line() + geom_point() +
  scale_x_continuous(breaks = 2:10)
```



El test de Silhouette ens diu que el nombre òptim de clusters és dos.

A continació repetirem el k-means per a **tres clusters**:

```
# Càlcul per a 2 clústers:
test_2k <- kmeans(covid_spain_normal, centers = 2)
fviz_cluster(test_2k, geom =  c("point","text"), data = covid_spain_normal)
```



Com podem observar, en les grafiques no tenim un conjunt de grups clarament separats. Tenim algunes agrupacions molt clares (Navarra - Comunitat Valenciana o l'aglomerat de La Rioja, Pais Vasc, Murcia, Asturias, Cantabria, Balears) però la resta estant més disperses i poden ser classificades tant en dos com en cinc grups. Per a aquest problema en questió trobem més adequat el test de Silhouette, ja que té en compte la pertanyensa al grup i la no-pertanyensa als altres grups i per tant triariem la classificació en dos clusters.

Les agrupacions que hem trobat per aquest últim cas són les següents:

- Un primer grup compost per:

```
# Mostrem per pantalla la primera agrupació trobada:
print(test_2k$cluster[test_2k$cluster==1])
```

```
##          Andalucia             Aragon           Asturias             Ba
lears
##                  1                  1                  1                  1
##           Canarias          Cantabria    Castilla y Leon Castilla la M
ancha
##                  1                  1                  1                  1
##          Catalunya        Extremadura            Galicia             M
adrid
##                  1                  1                  1                  1
##             Murcia         Pais Vasco           La Rioja             C
euta
##                  1                  1                  1                  1
##                                                                   Melilla
##                  1
```

- Un segon grup compost per:

```
#      Mostrem      per      pantalla      la      segona      agrupació:
print(test_2k$cluster[test_2k$cluster==2])

##   Comunitat   Valenciana                                        Navarra
##                     2                    2
```

Finalment, en cas que volguem podem exportar els dataframes que hem emprat en l'anàlisi en un arxiu '.csv' (normalitzat i sense normalitzar).

4.4. Regressió

Ja per últim, després de realitzar tests estadístics per a contrastar hipòtesis, analitzar correlacions i generar agrupacions de clustering, en aquest últim anàlisi ens proposem calcular un model de regressió lineal múltiple que permeti explicar les morts de cada comunitat autònoma en funció de les característiques d'aquesta, així com dels casos de Covid-19 confirmats.

Per fer-ho, partirem de les dades anteriors classificades per comunitat autònoma:

```
# Seleccionem únicament les variables amb les quals treballarem:
Spain_by_ccaa <- Spain_by_ccaa[c("Date","Province.State","Confirmed", "De
aths")]

#       Canviem       de       nom       a       les       columnes:
colnames(Spain_by_ccaa) <- c("Date", "Comunitat", "Confirmed", "Deaths")

#    Mostrem    el    dataset    que    utilitzarem    per    pantalla:
head(Spain_by_ccaa)

##                    Date                 Comunitat  Confirmed  Deaths
##  24687  2020-05-14                    Andalusia         12359     1336
##  24690  2020-05-14                       Aragon          5389      836
##  24694  2020-05-14                     Asturias          2356      308
##  24697  2020-05-14                     Baleares          1958      216
```

```
##  24712  2020-05-14                      Canarias          2275       151
## 24714 2020-05-14 Castilla - La Mancha     16470   2852
```

A partir d'aquest dataset, li afegirem les dades referents a cada comunitat autònoma que hem obtingut anteriorment a través de la web del INE, però en aquest cas no ens limitarem a treballar amb la primera i última data, sinó que realitzarem la regressió a partir de totes les dades disponibles:

```
# Creem un dataframe per a les dades referents a cada comunitat autònoma
obtingudes                              del                              INE:
dades_INE                                <-                            data.frame(
               Poblacio = c(8476718, 1330445, 1018775, 1210750, 2237309,
582357, 2401230, 2045384, 7652069, 5028650, 1061768, 2702244, 6747425, 15
04607,       656487,       218310,       315926,       84032,       84496),
               Superficie = c(87600, 47700, 10600, 5000, 7450, 5300, 9420
0, 79500, 32100, 23300, 41600, 29500, 8000, 11300, 10400, 7250, 5050, 19,
13))

# Calculem una nova variable per a la densitat de població (habitants/km^
2):
dades_INE$Densitat    <-    dades_INE$Poblacio    /    dades_INE$Superficie

#   Afegim   els   noms   de   cada   comunitat   autònoma   al   dataframe:
dades_INE$Comunitat <- c('Andalusia','Aragon','Asturias','Baleares','Cana
rias','Cantabria','Castilla y Leon','Castilla - La Mancha','Catalonia','C
.Valenciana','Extremadura','Galicia','Madrid','Murcia','Navarra','Pais Va
sco','La                                          Rioja','Ceuta','Melilla')

#                 Reordenem                  les                 columnes:
dades_INE <- dades_INE[c("Comunitat","Poblacio","Superficie", "Densitat")
]

# Agrupem els dos dataframes, el que conté informació sobre Covid-19 i el
de              les              dades              de              l'INE:
Spain_test    <-    merge(Spain_by_ccaa,    dades_INE,    by="Comunitat")

#                 Reordenem                  les                 columnes:
Spain_test <- Spain_test[c("Date", "Comunitat","Poblacio","Superficie", "
Densitat",                    "Confirmed",                    "Deaths")]

# Mostrem el dataframe obtingut (i el qual utilitzarem) per pantalla:
head(Spain_test)
```

```
##          Date Comunitat Poblacio Superficie Densitat Confirmed Deaths
## 1 2020-07-03 Andalusia  8476718      87600 96.76619     13205   1433
## 2 2020-09-29 Andalusia  8476718      87600 96.76619     59873   1780
## 3 2020-10-20 Andalusia  8476718      87600 96.76619     96224   2176
## 4 2020-11-12 Andalusia  8476718      87600 96.76619    184649   3105
## 5 2020-09-15 Andalusia  8476718      87600 96.76619     43678   1614
## 6 2020-07-11 Andalusia  8476718      87600 96.76619     13386   1435
```

Ja per últim, abans de generar els models de regressió lineal, factoritzarem la variable *'Comunitat'* per tal de treballar amb valors numèrics i no textuals:

```
# Factoritzem la variable 'Comunitat':
Spain_test$Comunitat[Spain_test$Comunitat == "Andalusia"] <- "1"
Spain_test$Comunitat[Spain_test$Comunitat == "Aragon"] <- "2"
Spain_test$Comunitat[Spain_test$Comunitat == "Asturias"] <- "3"
Spain_test$Comunitat[Spain_test$Comunitat == "Baleares"] <- "4"
Spain_test$Comunitat[Spain_test$Comunitat == "Canarias"] <- "5"
Spain_test$Comunitat[Spain_test$Comunitat == "Cantabria"] <- "6"
Spain_test$Comunitat[Spain_test$Comunitat == "Castilla - La Mancha"] <- "
7"
Spain_test$Comunitat[Spain_test$Comunitat == "Castilla y Leon"] <- "8"
Spain_test$Comunitat[Spain_test$Comunitat == "Catalonia"] <- "9"
Spain_test$Comunitat[Spain_test$Comunitat == "Ceuta"] <- "10"
Spain_test$Comunitat[Spain_test$Comunitat == "Extremadura"] <- "11"
Spain_test$Comunitat[Spain_test$Comunitat == "Galicia"] <- "12"
Spain_test$Comunitat[Spain_test$Comunitat == "La Rioja"] <- "13"
Spain_test$Comunitat[Spain_test$Comunitat == "Madrid"] <- "14"
Spain_test$Comunitat[Spain_test$Comunitat == "Melilla"] <- "15"
Spain_test$Comunitat[Spain_test$Comunitat == "Murcia"] <- "16"
Spain_test$Comunitat[Spain_test$Comunitat == "Navarra"] <- "17"
Spain_test$Comunitat[Spain_test$Comunitat == "Pais Vasco"] <- "18"

# La convertim a tipus 'integer' ja que ara està formada únicament per va
lors enters:
Spain_test$Comunitat <- as.integer(Spain_test$Comunitat)
```

Una vegada les dades estan preparades, partirem per realitzar a través de mínims quadrats ordinaris un model lineal que expliqui la variable *'Deaths'* en funció de les variables *Poblacio*, *Superficie* i *Densitat* i *'Confirmed'*.

Tal i com hem vist en els apunts, el **model de regressió lineal múltiple** no és més que una generalització del model de regressió lineal simple, en el qual relacionem la variable que volem explicar Y amb les k variables explicatives X1, X2, ..., Xk. Per tant, l'expressió general d'aquest model és la següent:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$$

En el nostre cas k = 4, ja que pretenem explicar la variable Y a partir de 4 variables X. Per tant, l'expressió anterior es podria reformular com:

$$ Deaths = \beta _0 + \beta _1·Població + \beta _2·Superfície + \beta _3·Densitat + \beta _4·Casos \space confirmats $$

Per a construir el model, cal buscar la suma dels residus al quadrat i després determinar els paràmetres del model que fan que aquesta suma tingui un valor mínim, la qual cosa es pot implementar amb el següent codi

```
# Creem el model de regressió lineal múltiple:
regressio_lineal_multiple_1 <- lm(Deaths ~ Poblacio + Superficie + Densit
```

```
at + Confirmed, data = Spain_test)

# El mostrem per pantalla:
regressio_lineal_multiple_1

##
##                                                                Call:
## lm(formula = Deaths ~ Poblacio + Superficie + Densitat + Confirmed,
##                                     data = Spain_test)
##
##                                                          Coefficients:
##  (Intercept)       Poblacio     Superficie       Densitat      Confirmed
##    1.449e+02     3.172e-04    -1.049e-03     5.568e-03     2.531e-02
```

A més, també podem consultar algunes estadístiques més detallades sobre el model amb el següent codi:

```
# Obtenim algunes estadístiques més detallades del model:
summary(regressio_lineal_multiple_1)

##
##                                                                Call:
## lm(formula = Deaths ~ Poblacio + Superficie + Densitat + Confirmed,
##                                     data = Spain_test)
##
##                                                           Residuals:
##            Min              1Q     Median              3Q             Max
##    -4645.5         -539.9         -191.4             302.8         4977.9
##
##                                                          Coefficients:
##                               Estimate  Std.  Error  t  value  Pr(>|t|)
##   (Intercept)      1.449e+02      3.850e+01         3.763   0.000171   ***
##  Poblacio           3.172e-04     1.197e-05     26.505     <  2e-16  ***
##    Superficie       -1.049e-03       8.327e-04       -1.260    0.207672
##   Densitat              5.568e-03      1.546e-02         0.360   0.718745
##  Confirmed          2.531e-02     4.739e-04     53.410     <  2e-16  ***
##                                                                      ---
##  Signif.  codes:   0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1
##
##   Residual   standard   error:   1294   on   3236   degrees   of   freedom
##   Multiple   R-squared:      0.7371,   Adjusted   R-squared:      0.7367
## F-statistic:  2268 on 4 and 3236 DF,  p-value: < 2.2e-16
```

Una vegada obtingut el model de regressió lineal múltiple, és molt important fer una bona interpretació dels resultats obtinguts. Per veure quines variables predictores són més significatives, partirem per analitzar la taula de coeficients, la qual mostra l'estimació dels coeficients beta de regressió i els valors t-estatístics i p associats:

```
# Mostrem per pantalla la taula de coeficients del model:
summary(regressio_lineal_multiple_1)$coefficient
```

```
##                      Estimate    Std. Error     t value       Pr(>|t|)
##  (Intercept)    1.448740e+02  3.849678e+01    3.7632763    1.706769e-04
##  Poblacio         3.171583e-04  1.196604e-05  26.5048672  2.935357e-140
##  Superficie    -1.049426e-03  8.327165e-04  -1.2602442    2.076721e-01
##  Densitat         5.567929e-03  1.545923e-02   0.3601686    7.187445e-01
## Confirmed    2.531199e-02 4.739145e-04 53.4104670  0.000000e+00
```

A partir d'aquesta taula, per a cada variable predictora, el valor t-estadístic evalua si existeix una relació significativa entre l'atribut en qüestió i la variable a predir (*'Deaths`*), la qual cosa es tradueix en veure si el coeficient beta obtingut és significativament diferent de 0 o no.

Per tant, cal interpretar correctament els diferents coeficients beta obtinguts. Anirem pas per pas:

- **Interpretació del coeficient B_0**

Com sabem, aquest paràmetre representa l'estimació del valor de Y quan totes les Xj prenen valor zero tot i que no sempre té una interpretació lligada al context (geomètrica, física, econòmica…). Perquè sigui possible interpretar-lo, s'ha de complir que:

1. Sigui realment possible que les Xj = 0.
2. S'han de tenir suficients observacions a prop dels valors Xj = 0.

Com es pot intuir, en aquest cas els diferents valors de Xj no poden ser realment 0, ja que no tindria sentit tenir una comunitat autònoma amb 0 habitants o 0 km2 de superfície, així que no es poden extreure conclusions d'aquest paràmetre.

- **Interpretació dels coeficients B_k**

Aquests coeficients, com hem vist en els apunts, representen l'estimació de l'increment que experimentaria la variable Y quan Xk augmentés el seu valor en una unitat i la resta de les variables es mantinguessin constants.

Com podem observar, per una banda els paràmetres *'Població'*, *'Densitat'* i *'Confirmed'* tenen un coeficient associat amb valor positiu, la qual cosa ens indica que, a major número de persones, densitat de població i casos confirmats per comunitat autònoma, major nombre de morts s'assoliran.

Per altra banda, el paràmetre *'Superfície'* té associat un coeficient amb valor negatiu, la qual cosa indica que a major superfície menys nombre de morts es produiran, la qual cosa està deguda a que, una major superfície dona lloc a una inferior densitat de població.

- **Interpretació dels valors p associats als coeficients**

En treure les estadístiques del model, també hem pogut observar els diferents valors p associats als coeficients de cada variable regressora. Aquests valors ens permeten

determinar si la relació que existeix entre l'atribut regressor i la variable a predir és estadísticament significativa o no.

Si partim d'un nivell de confiança del 95 %, ens podem fixar que tots els valors p obtinguts són inferiors a 0.05 i, per tant, podem afirmar que la relació de les 4 variables regressores quantitatives utilitzades en aquest model (*'Població'*, *'Densitat'*, *'Superfície'* i *'Casos confirmats'*) i la variable a predir (*'Deaths'*) és estadísticament significativa.

- **Interpretació de la bondat del model a partir del coeficient de determinació R2**

Com sabem, per comprovar la idoneïtat d'un model, s'utilitza el coeficient de determinació per a la regressió múltiple com a indicador de la qualitat de l'ajust.

De la mateixa manera que en la regressió lineal simple, **el coeficient de determinació R2 es pot definir com la proporció de variabilitat explicada pel model respecte a la variabilitat total**, és a dir:

$$ R^2 = \frac{Variabilitat \space explicada \space pel \space model}{Variabilitat \space total \space de \space la \space mostra} $$

El valor de R2 que hem obtingut per aquest model és de 0.7371, així que podríem afirmar que el nostre model és capaç d'explicar prop d'un 75 % de la variabilitat total de la mostra.

Ja per acabar, amb la intenció de millorar encara més el model, l'ampliarem de tal manera que la variable *'Deaths'*, a més de ser explicada pels atributs predictors quantitatius anteriors, també ho faci ara per les variable *'Comunitat Autònoma'*. Per tant, en aquest cas k = 5, ja que pretenem explicar la variable Y a partir de 5 variables X i l'expressió anterior es podria reformular com:

$$ Deaths = \beta _0 + \beta _1· Comunitat \space Autònoma + \beta _2 ·Població + \beta _3·Superfície + \beta _4·Densitat + \beta _5·Casos \space confirmats $$

De nou, cal buscar la suma dels residus al quadrat i després determinar els paràmetres del model que fan que aquesta suma tingui un valor mínim, la qual cosa es pot implementar amb el següent codi

```
# Creem el model de regressió lineal múltiple amb regressors quantitatius
i                                                                qualitatius:
regressio_lineal_multiple_2 <- lm(Deaths ~ Comunitat + Poblacio + Superfi
cie     +     Densitat     +     Confirmed,     data     =     Spain_test)

#            El            mostrem            per            pantalla:
regressio_lineal_multiple_2

##
##                                                                  Call:
## lm(formula = Deaths ~ Comunitat + Poblacio + Superficie + Densitat +
##                             Confirmed,     data     =     Spain_test)
##
```

```
##                                                          Coefficients:
## (Intercept)    Comunitat     Poblacio   Superficie     Densitat    Con
firmed
##  -8.977e+02    9.885e+01    3.729e-04    3.920e-03   -4.035e-02    2.2
61e-02
```

També podem treure les estadístiques més detallades per obtenir el coeficient de determinació R2:

```
# Obtenim algunes estadístiques més detallades del model:
summary(regressio_lineal_multiple_2)

##
##                                                                  Call:
## lm(formula = Deaths ~ Comunitat + Poblacio + Superficie + Densitat +
##                             Confirmed,      data      =      Spain_test)
##
##                                                               Residuals:
##           Min              1Q     Median              3Q           Max
##    -4004.3        -555.5            -27.4            549.8        4442.8
##
##                                                             Coefficients:
##                               Estimate  Std.  Error  t  value  Pr(>|t|)
##   (Intercept)   -8.977e+02      6.209e+01   -14.457      <   2e-16   ***
##   Comunitat         9.885e+01     4.784e+00     20.663     <   2e-16   ***
##   Poblacio           3.729e-04     1.157e-05     32.235     <   2e-16   ***
##   Superficie        3.920e-03     8.189e-04        4.787   1.77e-06   ***
##   Densitat          -4.035e-02      1.470e-02     -2.745     0.00609   **
##   Confirmed         2.261e-02     4.643e-04     48.708     <   2e-16   ***
##                                                                    ---
##  Signif.  codes:   0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1
##
##   Residual   standard   error:   1216   on   3235   degrees   of   freedom
##   Multiple   R-squared:      0.7677,   Adjusted   R-squared:      0.7674
## F-statistic:  2139 on 5 and 3235 DF,  p-value: < 2.2e-16
```

En vista d'aquests resultats podem observar que el fet d'ampliar el primer model de regressió lineal per tal de que l'atribut *'Deaths'* també sigui explicat per la variable *'Comunitat Autònoma'* permet incrementar el coeficient de determinació de 0.7371 a 0.7677, és a dir, explicar gairabé el 77 % de la variabilitat total de la mostra.

5. Representació dels resultats a partir de taules i gràfiques.

Hem generat aquest document mitjançant el llenguatge R i Rmarkdown, així que tots els resultats els hem anat representant a través de taules i gràfiques dins dels diferents anàlisi que hem realitzat en el punt 4.

6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclussions? Els resultats permeten respondre al problema?

Tal i com s'ha indicat en el punt 4, en aquestà pràctica hem realizat 4 anàlisis diferents de les dades, les conclusions dels quals detallem a continuació:

- **Tests estadístics per realitzar contrasts d'hipòtesis**

En primer lloc hem realitzat una exploració gràfica de les dades de Covid-19 a nivell europeu i ens hem centrat a estudiar la incidència del virus en els dos països europeus més afectats durant la primera onada: França i Itàlia.

Per aquests dos països hem representat la evolució dels casos i morts diàries agrupats setmanalment i a partir del gràfic semblava que França hagués sigut més castigada que no pas Itàlia. No obstant això, per comprovar-ho estadísticament, hem realitzat un **test d'hipòtesis de dues mostres independents sobre la seva mitjana amb variàncies desconegudes diferents** tant pel nombre de casos confirmats com per les morts i, una vegada realtizats els càlculs, hem pogut arribar a la conclusió que no es pot afirmar que la incidència de la Covid-19 durant la primera onada a França hagi sigut superior que no pas a Itàlia, reforçant la idea de que cal saber interpretar cautelarment bé els gràfics.

- **Determinació de la correlació entre variables**

Una vegada realitzats els tests estadístics per a les dades de Covid-19 a nivell europeu, hem delimitat l'àmbit d'estudi al territori espanyol, concretament hem utiltizat les dades classificades per comunitat autònoma i, a més d'utilitzar les dades referents al Covid-19 (és a dir, els casos confirmats, recuperats i les morts), també hem afegit informació demogràfica de cada comunitat autònoma obtingut a través del INE.

Pel que fa l'estudi de correlació en si, hem intentat determinar si hi havia alguna diferència significativa entre la correlació de les dades inicials del Covid-19 a espanya (19.05.20) i les últimes dades recullides en aquest dataset (06.12.20). No obstant això, no hem trobat cap relació entre la superfície o població de cada comunitat autònoma i les dades de la Covid-19 a data de 6 de desembre de 2020, ni tampoc hem trobat cap diferència significativa entre el comportament de la Covid-19 a 19 de Maig i el que té a finals d'any (6 de desembre).

Intepretem aquestes dades com que des de després de la primera onada, la malaltia ja està present en tot el territori i paràmetres com la població o la densitat de població ja no tenen cap correlació amb el nombre d'infectats, morts o recuperats.

De la comparativa de correlacions entre el 19 de Maig i el 6 de Desembre, que són iguals, interpretem que la malaltia té el mateix comportemanet en ambdues dates. Per tant, podem concloure que les politiques aplicades des de les diverses administracions han servit per evitar que el sistema sanitari colapsi, però no han modificat el comportament de la malaltia.

- **Agrupació de les dades en clusters**

Després d'analitzar la correlació entre variables, ens hem proposat buscar una possible classificació de les Comunitats Autònomes en funció de les seves característiques i les dades de la Covid-19 de que disposem (de nou únicament per al territori espanyol).

Per fer-ho, s'han utilitzat 2 mètodes diferents per a calcular el nombre de clústers 'k' òptims a utilitzar: "L'Elbow Method" i el "Silhouette". A partir del primer s'ha obtingut que el k òptim passava per utilitzar 5 clústers, mentre que en el segon mètode només en calien 3 (sent millor aquest últim resultat).

Així que podem concloure que una classificació en tres grups és prou bona, amb una pertanyensa clarament marcada. La classificació en cinc també és bona, però determinades comunitats estàn al llindar de varis grups i tan poden caure en un com en un altre, en funció de com s'executi l'algosisme. Per tant, considerem que la classificació en tres grups és més adequada.

- **Regressió lineal múltiple per explicar la variable 'Deaths'**

Ja per últim, després de realitzar tests estadístics per a contrastar hipòtesis, analitzar correlacions i generar agrupacions de clustering, hem realitzat un últim anàlisi per calcular un model de regressió lineal múltiple que permetés explicar les morts de cada comunitat autònoma en funció de les característiques d'aquesta, així com dels casos de Covid-19 confirmats.

Per fer-ho, en primer lloc hem creat un primer model on k = 4, ja que preteníem explicar la variable Y (*'Deaths'*) a partir de 4 variables X (*'Població'*, *'Superfície'*, *'Densitat'* i *'Casos confirmats'*) i hem obtingut un coeficient de correlació R2 = 0.7371, el qual permetia explicar prop d'un 75 % de la variabilitat total de la mostra.

Després, amb la intenció de millorar el model, l'hem ampliat per tal que l'atribut *'Deaths' també sigui explicat per la variable 'Comunitat Autònoma'*, així que hem passat d'un k = 4 a un k = 5, obtenint ara un valor R2 = 0.7677, és a dir, explicant gairabé el 77 % de la variabilitat total de la mostra.

- **Preguntes concretes plantejades inicialment**

Les preguntes concretes que voliem respondre en iniciar aquest document eren les següents:

1. Podem considerar que les dades de la primera onada de la Covid-19 en els dos països europeus són estadísticament diferets?

Hem pogut veure com estadísticament no hi ha diferencia entre les dades de França i Itàlia, els països europeus més afectats per la primera onada de la Covid-19.

2. La Covid-19 és un coronavirus que majoritariament es contagia persona a persona. Per tant, existeix algun tipus de relació entre les característiques de cada CCAA (població, superfície) i l'impacte del virus? A priori esperarem que les comunitats amb més densitat de població haurien de veure's més afectades.

La única corrlació que hem trobat ha estat entre el nombre de casos confirmats, el nombre de morts i el nombre de casos superats. La superfície o el nombre d'habitans de les comunitats autònomes no te cap correlació amb l'impacte de la malaltia en aquesta, ni a mitjans de Maig, ni a principis de Decembre.

3. Per fer front a la situació actual, és possible classificar les comunitats en diversos grups, per tal que cada grup pugui aplicar mesures adequades i més apropiades?

Hem trobat, que gràcies al mètode de Silhouette, podem classificar les comunitats en dos clusters. Tot i això, la classificació no és massa estable i algunes comunitats poden canviar de grup en funció de com s'executi l'algorisme, fet que no genera massa confiança.

4. Podem veure en aquestes dades un canvi degut a les polítiques adoptades per les diverses administracions competents?

Com que la matriu de correlacions es pràcticament idèntica a mitjans de Maig que a principis de Decembre, no sembla que cap política hagi tingut efectes positius com una reducció de la mortalitat, per exemple.

5. Podem construïr un model de regressió que ens permeti predir l'evolució de la Covid-19?

Com hem vist en el punt 4.4, hem pogut construir un model que explica gairebé el 77% de variabilitat total de la mostra.

7. Codi: Cal adjuntar el codi, preferiblement R, amb el que s'ha realitzat la neteja, anàlisi i rempresentació de les dades.

Aquest document ha estat generat mitjançant el llenguatge R i Rmarkdown. Tot el codi apareix de forma transparent en cada punt en que s'ha implementat.

Adicionalment, els fitxers "PRA_2-Entrega.Rmd", "covid_19_data.csv" i "scholar.bib" seran adjuntats en l'entrega d'aquesta pràctica.

8. Agraïments.

- Johns Hopkins University for making the data available for educational and academic research purposes
- https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset
- MoBS lab - https://www.mobs-lab.org/2019ncov.html
- World Health Organization (WHO): https://www.who.int/
- DXY.cn. Pneumonia. 2020. http://3g.dxy.cn/newh5/view/pneumonia.
- BNO News: https://bnonews.com/index.php/2020/02/the-latest-coronavirus-cases/

- National Health Commission of the People's Republic of China (NHC):
- http://www.nhc.gov.cn/xcs/yqtb/list_gzbd.shtml
- China CDC (CCDC): http://weekly.chinacdc.cn/news/TrackingtheEpidemic.htm
- Hong Kong Department of Health: https://www.chp.gov.hk/en/features/102465.html
- Macau Government: https://www.ssm.gov.mo/portal/
- Taiwan CDC: https://sites.google.com/cdc.gov.tw/2019ncov/taiwan?authuser=0
- US CDC: https://www.cdc.gov/coronavirus/2019-ncov/index.html
- Government of Canada: https://www.canada.ca/en/public-health/services/diseases/coronavirus.html
- Australia Government Department of Health: https://www.health.gov.au/news/coronavirus-update-at-a-glance
- European Centre for Disease Prevention and Control (ECDC): https://www.ecdc.europa.eu/en/geographical-distribution-2019-ncov-cases
- Ministry of Health Singapore (MOH): https://www.moh.gov.sg/covid-19
- Italy Ministry of Health: http://www.salute.gov.it/nuovocoronavirus