Visualització de Dades (Enginyeria de Dades – EE - UAB)
Examen Segon Parcial – 16 Juny 2025
RESPOSTES MODEL B

Nom i Cognom:_____

NIU:_____

Només es permet l'ús d'internet per l'accés al campus virtual en el moment de descarregar el full d'enunciats i d'entregar l'examen.

PART 1 (6 pts.)

Dataset: 25_noms_padro_any_sexe_1996_2019.csv

Agafarem aquest dataset que conté els noms de nens i nenes més freqüents dels nadons de Barcelona entre els anys 1996 i 2019. Podeu fer servir les llibreries R (plotly, gganimate, shiny, etc.) que creieu convenients i dibuixeu les gràfiques que us facin falta. Cal incloure les comandes R i una captura de pantalla de la gràfica que es demana.

Cada registre d'aquest data set conté informació d'un nom i any registrat. Conté les variables:

- **Ordre** → Número de ranking de nens o nenes d'un any, segons el nom
- **Nom** → Nom del nadó
- **Sexe** → Gènere. Té dos valors: "Dona", "Home"
- **Any** → Any de la dada
- **Nombre** → Nombre de nadons amb aquest nom i any.

Si necessiteu fer Data Massaging abans de dibuixar la gràfica, expliqueu quines operacions feu. Adjunteu el codi R.

Càrrega de les llibreries i dataset:

```
> library(tidyverse)
> library(dplyr)
> library(plotly)
> library(shiny)
> setwd("C:/Users/enric/Documents/R")
> NadonsBCN <- read.csv('./25_noms_padro_any_sexe_1996_2019.csv')
```

1.1 (1 pt.) Feu una gràfica interactiva de línies sobre l'evolució del nom de CARLA al llarg dels anys i contesta les preguntes sobre la gràfica.

RESPOSTA:

DATA MASSAGING: *Filtrar els registres amb valor de la variable Nom igual a CARLA:*

```
> NomCarla <- NadonsBCN %>% filter(Nom == 'CARLA')
```

PAS 2a. GRÀFICA de línies amb ggplotly:

```
> plotCarla <- ggplot(NomCarla, aes(x=Any,y=Nombre,color=Nom)) +
geom_line()
> ggplotly(plotCarla)
```

PAS 2b. GRÀFICA de línies amb plot_ly:

```
> plot_ly(NomCarla,x=~Any, y=~Nombre, color=~Nom, type='scatter',
mode='line')
>
```

Sobre aquesta gràfica contesta les preguntes:

a) **En quins anys hi ha el màxim nombre de nadons amb aquest nom i en quin any hi ha el mínim?. Dona l'any i el nombre.**

   RESPOSTA: Màxim l'any 2005 (171), mínim l'any 2019 (55).

b) **En quin any es puja per primer cop el nombre de nadons amb aquest nom a més de 150 i en quin any es baixa per primer cop el nombre de nadons amb aquest nom a menys de 100 nadons?. Dona l'any i el nombre**

   RESPOSTA:

- Es puja a més de 150 l'any 2001 amb 153 nadons.
- Es baixa a menys de 100 l'any 2013 amb 81 nadons.

c) **En quin parell d'anys s'assoleix el màxim descens de nadons amb aquest nom?.**

RESPOSTA: Entre 2012 (122) i 2013 (81) amb un descens de 41 nadons.

1.2. (1 pt.) Visualitza en gràfica de línies l'evolució temporal dels noms masculins (plotNomsMasculins).

RESPOSTA:

En primer lloc, hem de fer data massaging sobre el data set original i després dibuixem la gràfica.

PAS 1. DATA MASSAGING: *Filtrar els registres amb nom de dona:*

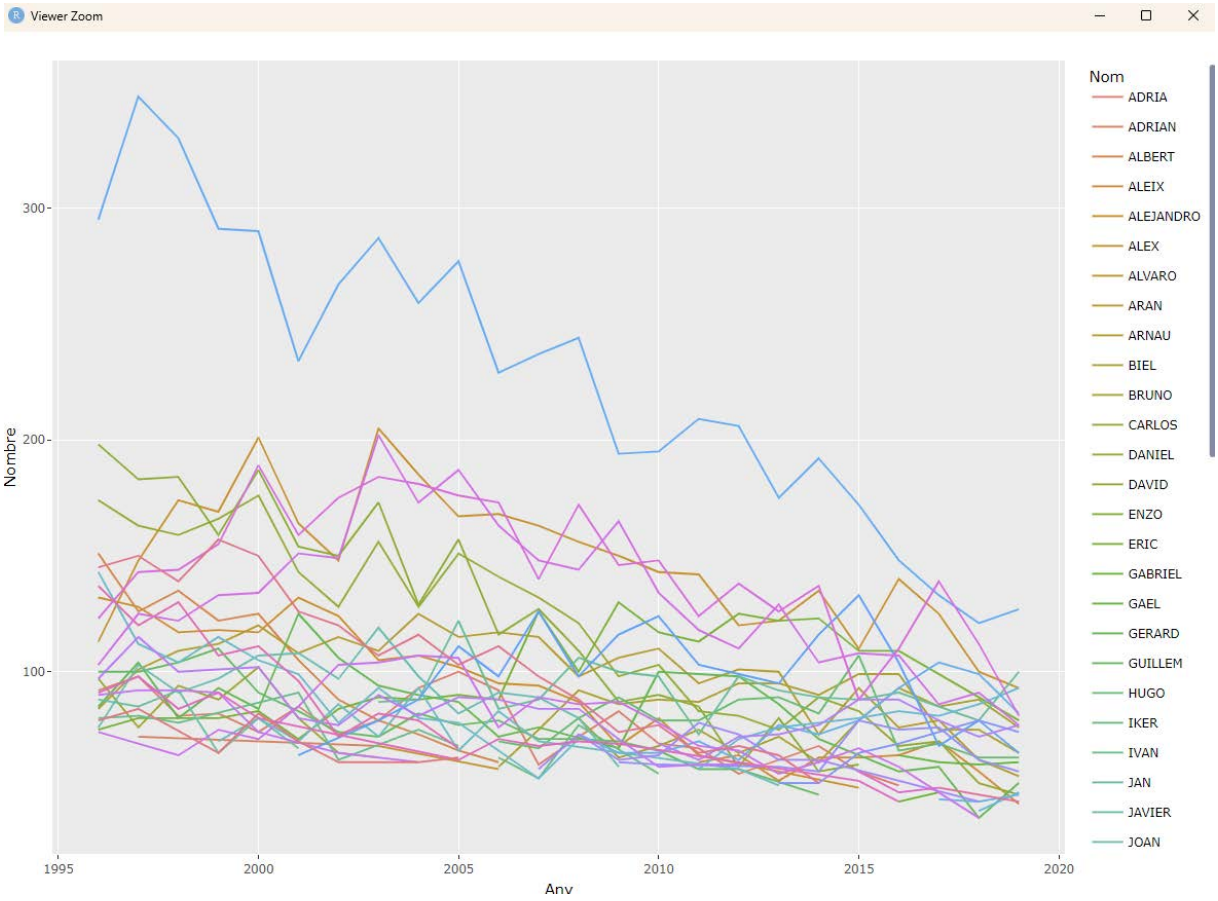
```
> NomsMasculins <- NadonsBCN %>% filter(Sexe=="Home")
```

PAS 2a. GRÀFICA de línies amb ggplotly:

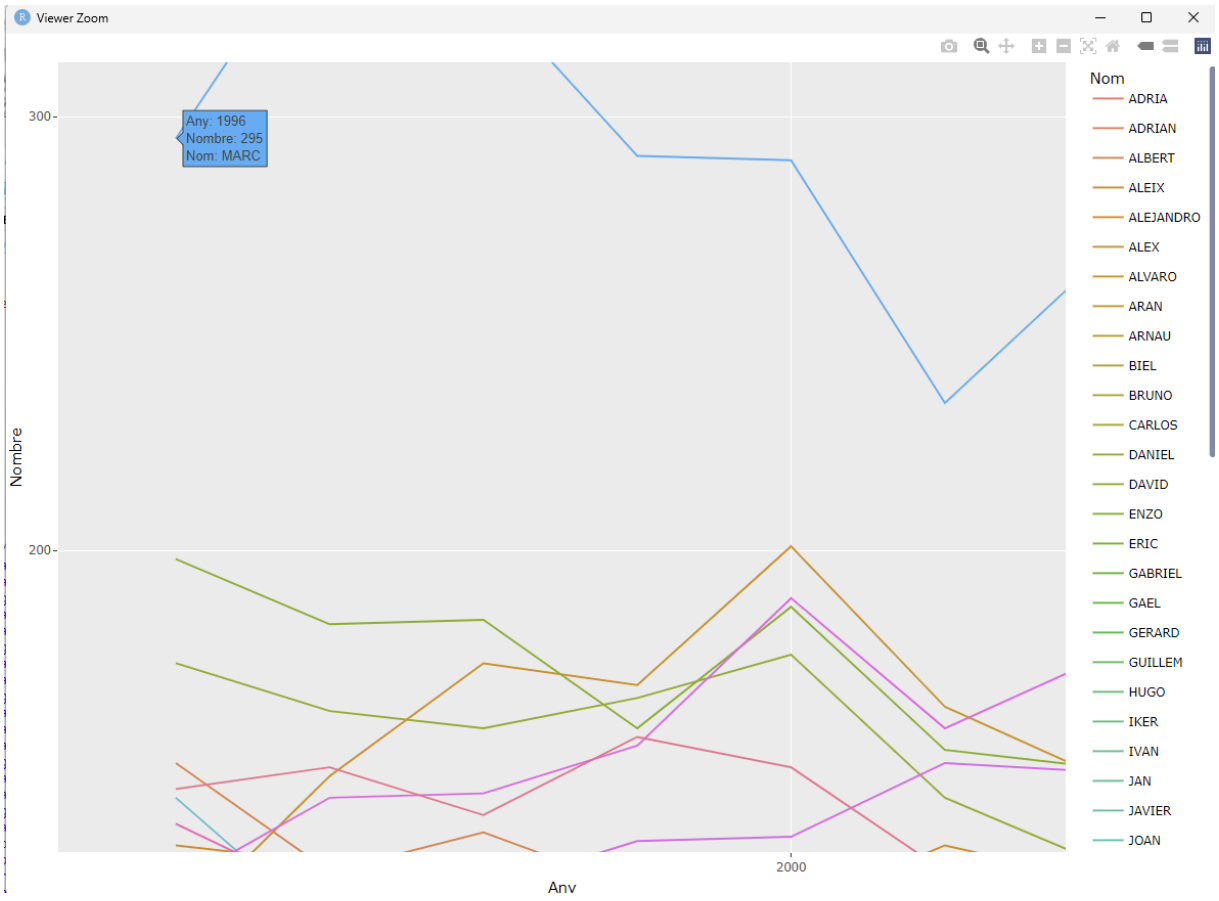
```
>    plotNomsMasculins<-    ggplot(NomsMasculins,aes(x=Any,    y=Nombre,
colour=Nom)) + geom_line()
> ggplotly(plotNomsMasculins)
>
```

PAS 2b. GRÀFICA de línies amb ggplotly:

```
> plot_ly(NomsMasculins,x=~Any, y=~Nombre, color=~Nom, type='scatter',
mode='line')
```


Viewer Zoom
Nombre
300
200
100
1995
2000
2005
2010
2015
2020
Any
Nom
ADRIA
ADRIAN
ALBERT
ALEIX
ALEJANDRO
ALEX
ALVARO
ARAN
ARNAU
BIEL
BRUNO
CARLOS
DANIEL
DAVID
ENZO
ERIC
GABRIEL
GAEL
GERARD
GUILLEM
HUGO
IKER
IVAN
JAN
JAVIER
JOAN


Viewer Zoom
Nombre
Any: 2016
Nombre: 48
Nom: ROC
Any
Nom
ADRIA
ADRIAN
ALBERT
ALEIX
ALEJANDRO
ALEX
ALVARO
ARAN
ARNAU
BIEL
BRUNO
CARLOS
DANIEL
DAVID
ENZO
ERIC
GABRIEL
GAEL
GERARD
GUILLEM
HUGO
IKER
IVAN
JAN
JAVIER
JOAN



Sobre aquesta gràfica interactiva, respon a les següents preguntes

a) **Quins són els 3 noms masculins menys i més utilitzats els anys 1996 i 2016, especificant el nombre de nadons per a cada nom?.**

RESPOSTA:

- 1996: Menys utilitzats: PABLO (74), ERIC (75), JOAN (76).
- 1996: Més utilitzats: MARC (295), DAVID (198), DANIEL (174).
- 2016: Menys utilitzats: ROGER i GABRIEL (44), ROC (48), ADRIÀ (51).
- 2016: Més utilitzats: MARC (148), ALEX (140), POL (110).

b) **Quin és el nom o noms masculins del que es té la darrera referència l'any 2005?. Digues nom i nombre de nadons aquell any.**

RESPOSTA:

- XAVIER (63), SERGI (67) i IVAN (68).

1.3. (1 pt.) Reproduïu l'aplicació shiny amb entrada per desplegable que dibuixi la gràfica de línies amb tots els noms femenins de nadons del dataset, de forma que es puguin seleccionar els noms en el desplegable. Posa com a nom per defecte el nom 'SARA'. Utilitza la funció `plot_ly()` o `ggplot()`, la que vulguis.

RESPOSTA:

PAS 1. DATA MASSAGING: *Filtrar els registres amb nom de dona:*

```
> NomsFemenins <- NadonsBCN %>% filter(Sexe=="Dona")
```

PAS 2a. GRÀFICA SHINY amb ggplot:

```
ui <- fluidPage(   selectizeInput( inputId = "Noms",
                        label = "Selecciona un Nom:",
                        choices = unique(NomsFemenins$Nom),
                        selected = "SARA",
                        multiple = TRUE
                       ),
        plotlyOutput(outputId = "plot")
       )

server <- function(input, output, ...)
   { output$plot <- renderPlotly(
      { plotNomS <- ggplot(filter(NomsFemenins,Nom %in% input$Noms),
                      aes(x=Any, y=Nombre,color=Nom)) + geom_path()
        ggplotly(plotNomS)
      })
   }
shinyApp(ui, server)
```

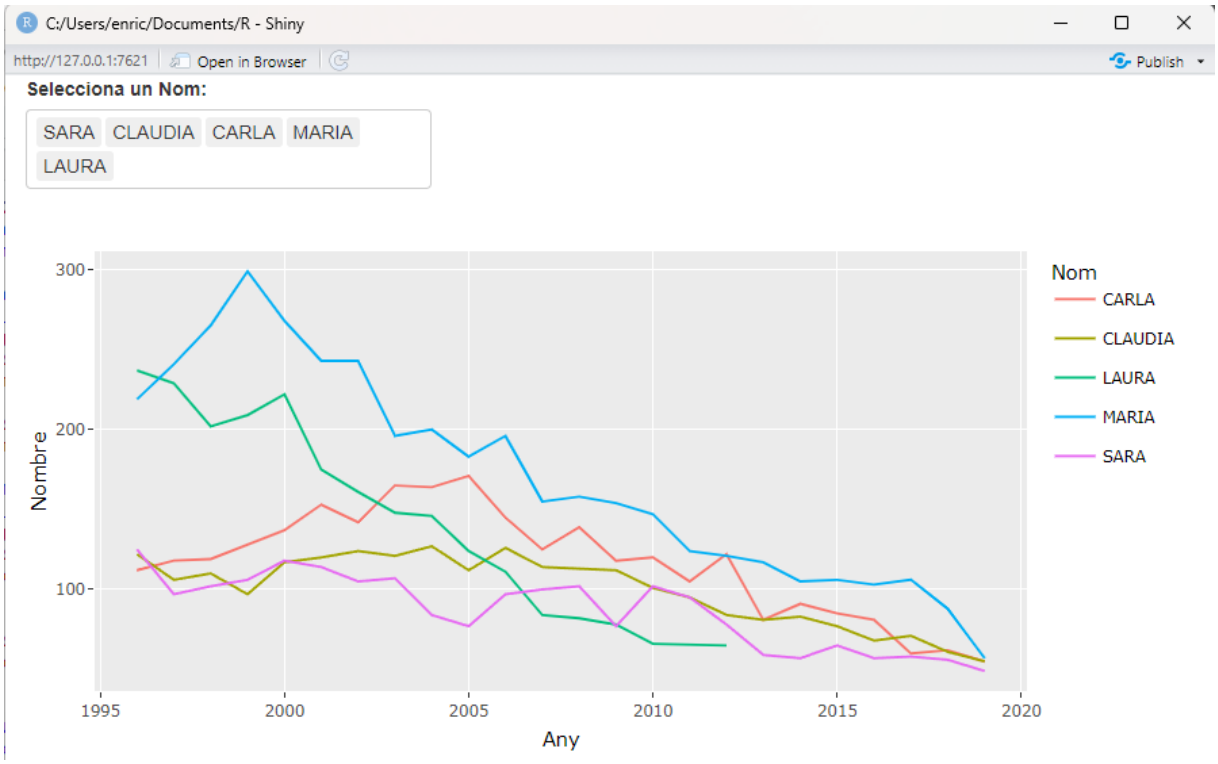
PAS 2a. GRÀFICA SHINY amb plot_ly:

```
ui <- fluidPage(   selectizeInput( inputId = "Noms",
                        label = "Selecciona un Nom:",
                        choices = unique(NomsFemenins$Nom),
                        selected = "ENRIC",
                        multiple = TRUE
                       ),
        plotlyOutput(outputId = "p")
       )

server <- function(input, output, ...)
  {     output$p <- renderPlotly (
        { plot_ly(NomsFemenins, x = ~Any, y = ~Nombre, color=~Nom) %>%
                    filter(Nom %in% input$Noms) %>%
                    group_by(Nom) %>%
                    add_lines()
        })
  }
shinyApp(ui, server)
```

Selecciona els noms de CLAUDIA, CARLA, MARIA, PAULA i SARA. Mostra la gràfica.

RESPOSTA:



Sobre aquesta gràfica contesta a les preguntes:

a) Troba tres característiques de la gràfica.

RESPOSTA:

- La majoria de noms ha tingut una davallada al llarg dels anys.
- El nom de MARIA és el que ha estat en primera posició en tots els anys excepte els dos últims.
- El nom de SARA és el menys utilitzat dels cinc noms majoritàriament.
- Els noms de MARIA i PAULA han destacat dels anys 1995 a 2005, I el de PAULA ha anat decaient més en els anys següents.
- El nom de CARLA s'ha mantingut en una posición Mitjana durant tot el període del dataset.

b) Quins estan els tres primers l'any 2004?. Dona noms i nombre de nadons

RESPOSTA:

1. MARIA (200)
2. CARLA (164)
3. LAURA (146)

c) Quins estan els tres últims l'any 2017?.

RESPOSTA:

1. SARA (58)
2. CARLA (60)
3. CLAUDIA (71)

1.4. (2 pts.) Mostra el codi per a generar el Ranking de Barres Animades (*Animated Bar Race Ranking*) sobre els 10 noms masculins o femenins menys utilitzats cada any. Fes un parell de captures de pantalla de l'animació. . Fes un parell de captures de pantalla de l'animació.

RESPOSTA:

Càrrega de les llibreries:

```
> library(gganimate)    # Generació de frames i la seva compilació per
a generar fitxer animació
```

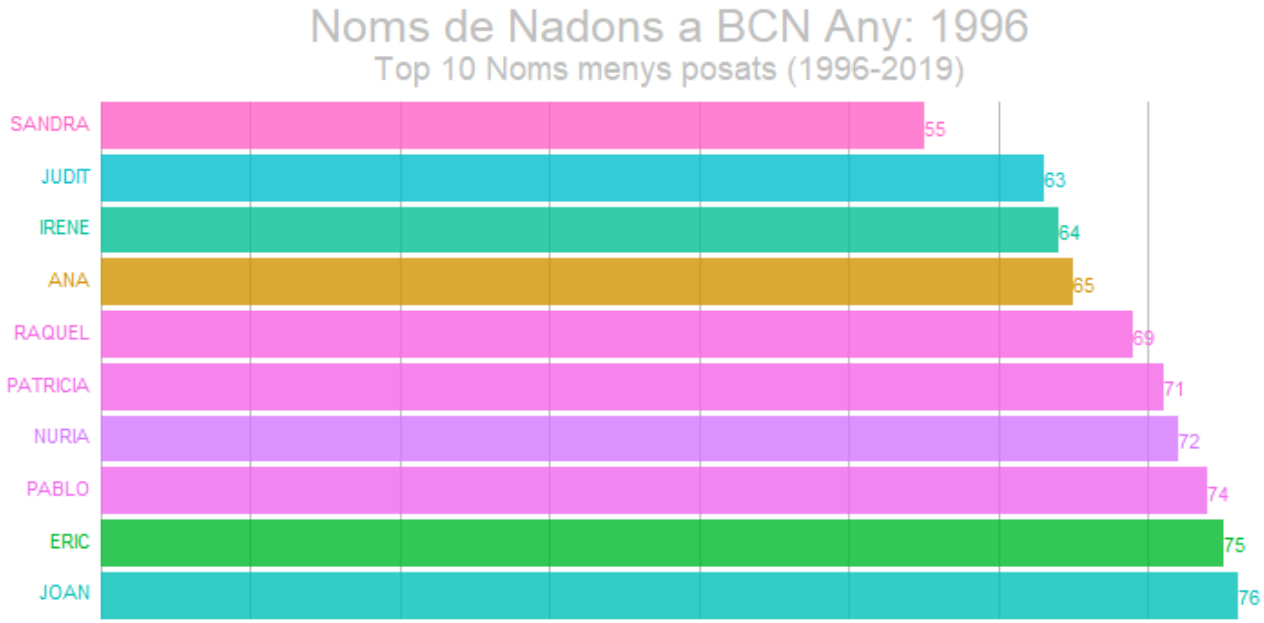
PAS 1. DATA MASSAGING: *Definir nova variable rank amb la posició creixent en nombre de nadons i filtrar els 10 primers noms per a cada any segons rank:*

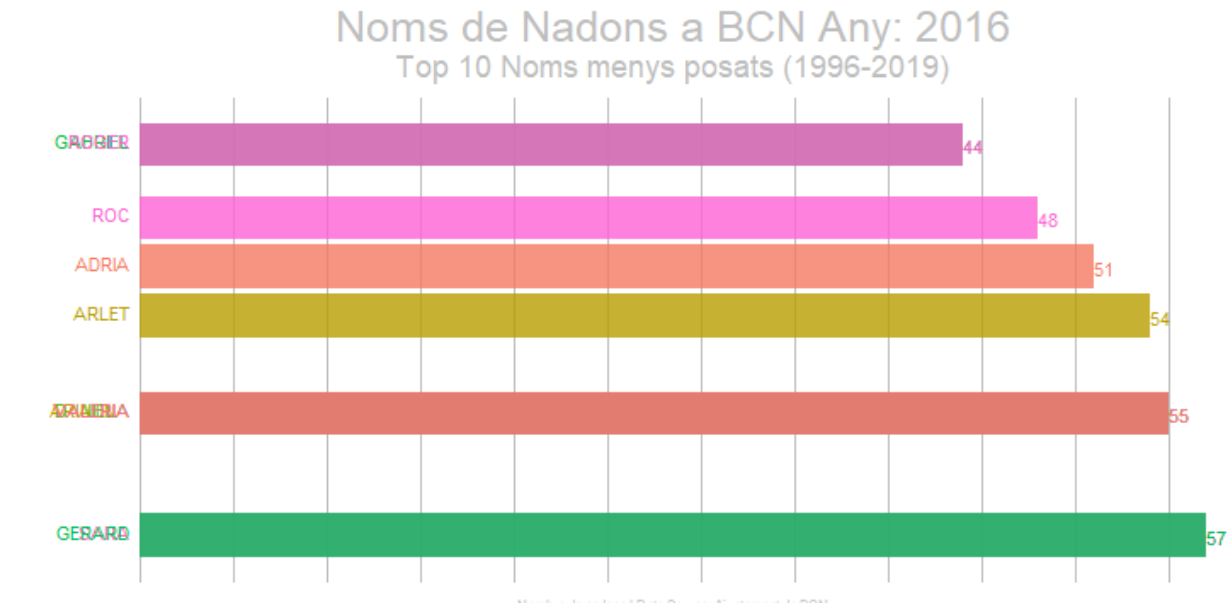
```
> NadonsBCN_formatted <- NadonsBCN %>%
  group_by(Any) %>%
  mutate(rank = rank(Nombre)) %>%
  filter(rank <=10)
```

PAS 2. ANIMATED BAR RACE RANKING:

```
> anim <- ggplot(NadonsBCN_formatted, aes(rank, group = Nom,
                fill = as.factor(Nom), color = as.factor(Nom))) +
  geom_tile(aes(y = Nombre/2,
                height = Nombre,
                width = 0.9), alpha = 0.8, color = NA) +
  geom_text(aes(y = 0, label = paste(Nom, " ")), vjust = 0.2, hjust =
1) +
  geom_text(aes(y=Nombre,label = Nombre, hjust=0)) +
  coord_flip(clip = "off", expand = FALSE) +
  scale_x_reverse() +
  guides(color = FALSE, fill = FALSE) +
  theme(axis.line=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks=element_blank(),
        axis.title.x=element_blank(),
         axis.title.y=element_blank(),
        legend.position="none",
        panel.background=element_blank(),
        panel.border=element_blank(),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank(),
        panel.grid.major.x = element_line( size=.1, color="grey" ),
        panel.grid.minor.x = element_line( size=.1, color="grey" ),
        plot.title=element_text(size=25, hjust=0.5, face="bold",
      colour="grey", vjust=-1),
        plot.subtitle=element_text(size=18, hjust=0.5, face="italic",
      color="grey"),
        plot.caption =element_text(size=8, hjust=0.5, face="italic",
color="grey"),
```

```
        plot.background=element_blank(),
        plot.margin = margin(2,2, 2, 4, "cm")) +
  transition_states(Any, transition_length = 4, state_length = 1,
 wrap = FALSE) +
  view_follow(fixed_x = TRUE) +
  labs(title = 'Noms de Nadons a BCN Any: {closest_state}',
       subtitle  =  "Top 10 Noms menys posats (1996-2019)",
       caption  = "Nombre de nadons | Data Source: Ajuntament de BCN")
> anim
```





ANIMATED BAR RACE RANKING. ALTERNATIVA: Llibreria ddplot *que defineix una gràfica de forma més senzilla i clara.*

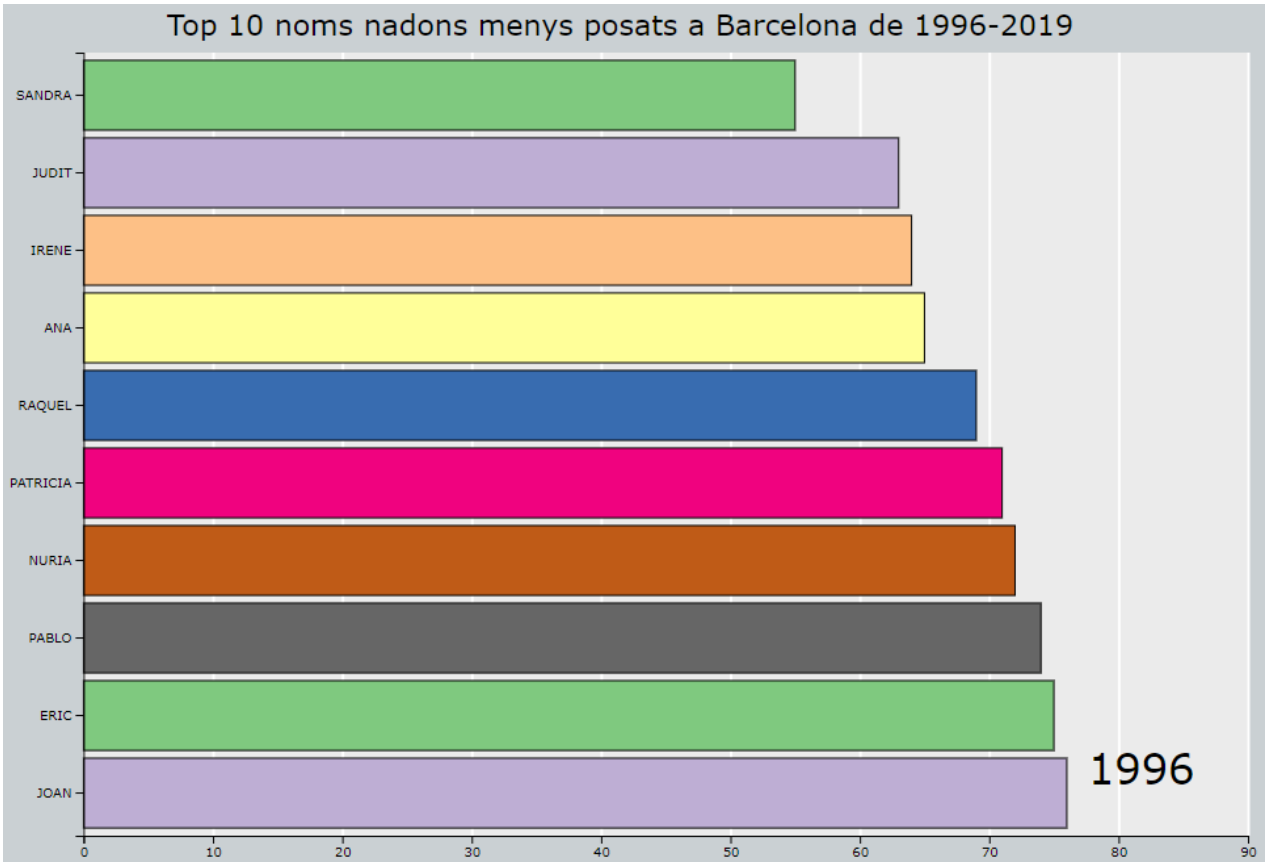
```
> library(ddplot)              # Generació de Racing Bar Charts
```

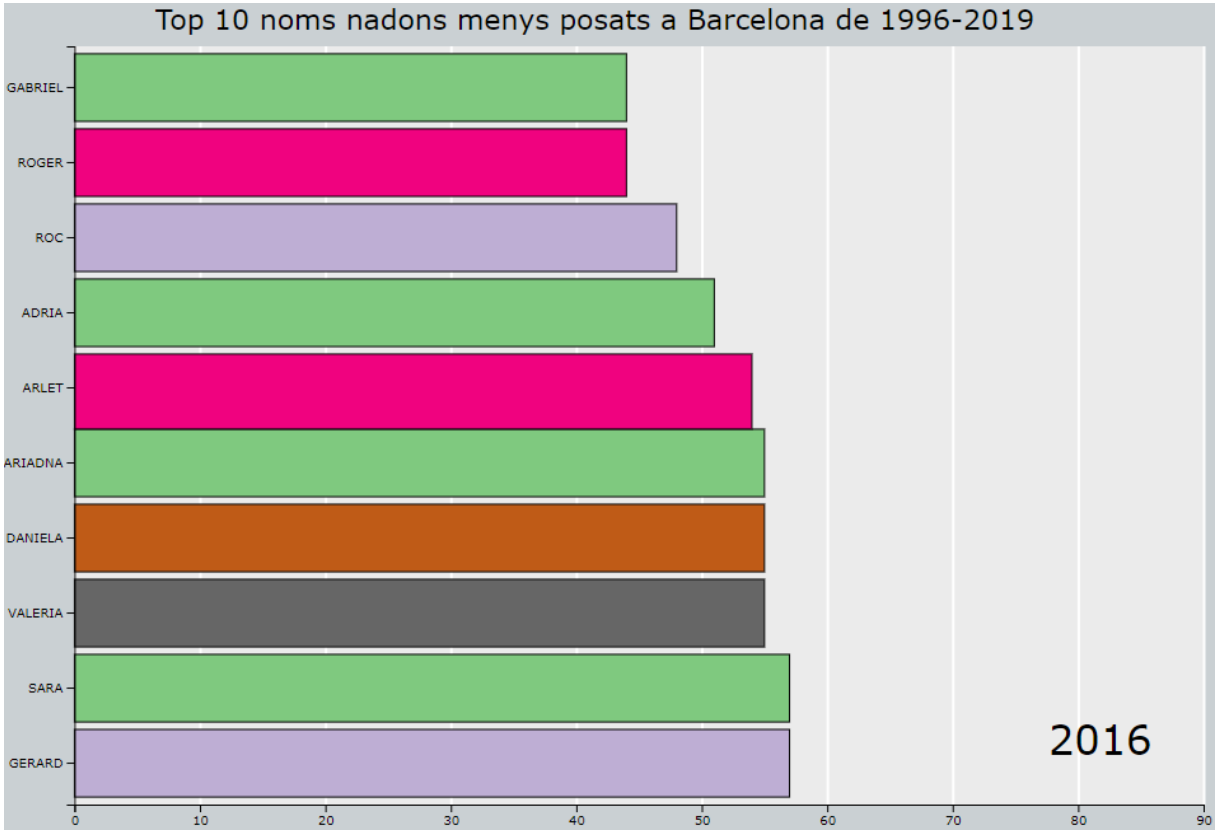
PAS 1. DATA MASSAGING: *Definir nova variable rank amb la posició creixent en nombre de nadons i filtrar els 10 primers noms per a cada any segons rank:*

```
> NadonsBCN_formatted <- NadonsBCN %>%
  group_by(Any) %>%
  mutate(rank = rank(Nombre)) %>%
  filter(rank <=10)
```

PAS 2: ANIMATED BAR RACE RANKING:

```
>  barChartRace(NadonsBCN_formatted, x="Nombre", y="Nom", time="Any", sort="ascending", title = "Top 10 noms nadons menys posats a Barcelona de 1996-2019")
```





1.5. (1 pt.) Defineix 4 dels següents conceptes en animació, interactivitat, usabilitat i Experiència d'Usuari en Visualització de Dades:

- ***SUS***
- ***Utility***
- ***UEQ***
- ***Data linking***
- **Event**
- ***Checkbox***

PART 2 (4 pts.)

Dataset: filmdeathcounts_with_main_genre.csv

- **Si utilitzeu Tableau, cal incloure una captura de pantalla de l'aplicació Tableu on es vegin les configuracions actives i la gràfica.**
- **Si utilitzeu R, cal incloure les comandes R i una captura de pantalla de la gràfica.**

2.1 (Total: 2,5 pts.) Fes un *treemap* que mostri quins directors són els més "letalment productius" dins de cada gènere.

2.1.1 (0,75 pts.) Data Massage necessari per la visualització

a. **Llegeix les dades.**

b. **Agrupa les dades pel director (Director) i el gènere principal (Main_Genre).**

c. **Calcula el total de morts (Body_Count) per a cada combinació de Director i Main_Genre.**

d. **Filtra directors amb almenys 300 morts acumulades.**

e. **Desa el resultat en un dataframe anomenat dataT.**

2.1.2.a (0,75 pts.) Mostra un mapa d'arbre (*treemap*) que et permeti saber quins directors (*Director*) fan pel·lícules amb més de 300 morts totals per cada gènere principal (Main_Genre).

2.1.2.b (0,75 pts.) Argumenta com és un *treemap* en general i detalla els passos que has de fer per construir la visualització d'aquest exercici (és a dir quina variable utilitzes per l'àrea de les graelles, variables d'agrupació, etc. i per què?).

2.1.2.c (0,25 pts.) Quin director és més "letalment productiu" quan el gènere principal és aventura? I quan el gènera principal és crim?

NOTA: En R, l'opció `reflow = TRUE` **dins de** `geom_treemap_text()` **controla com es disposa el text dins de cada casella del *treemap*. D'aquesta manera el text s'adapta dinàmicament a l'espai disponible. És més llegible. (Podeu usar-lo, opcionalment en aquest exercici, per veure més informació).**

RESPOSTES:

RESPOSTA 2.1.1:

```
# Llibreries necessàries
library(tidyverse)

# a) Llegir les dades
 > data <- read.csv("filmdeathcounts_with_main_genre.csv")

# b-d) Agrupar i sumar
>  dataT <- data %>%
group_by(Director, Main_Genre) %>%
summarise(Total_Deaths = sum(Body_Count, na.rm = TRUE), .groups =
"drop")

# e) Filtrar directors amb almenys 300 morts acumulades
dataT <- dataT %>%
group_by(Director) %>%
filter(sum(Total_Deaths) >= 300) %>%
ungroup()
```

NOTA: No es baixarà nota per no haver fet les parts en negreta tot i els warnings, ja que el resultat és el mateix.

RESPOSTA 2.1.2.a. i b. :

Com vam veure a classe, un mapa d'arbre és un dibuix rectangular dividit en caselles, i cada casella representa una sola observació. Vam veure que era una bona manera de mostrar dades jeràrquiques mitjançant rectangles imbricats. I l'àrea relativa de cada casella expressava una variable contínua. Per tant, una possibilitat amb aquest dataset seria:

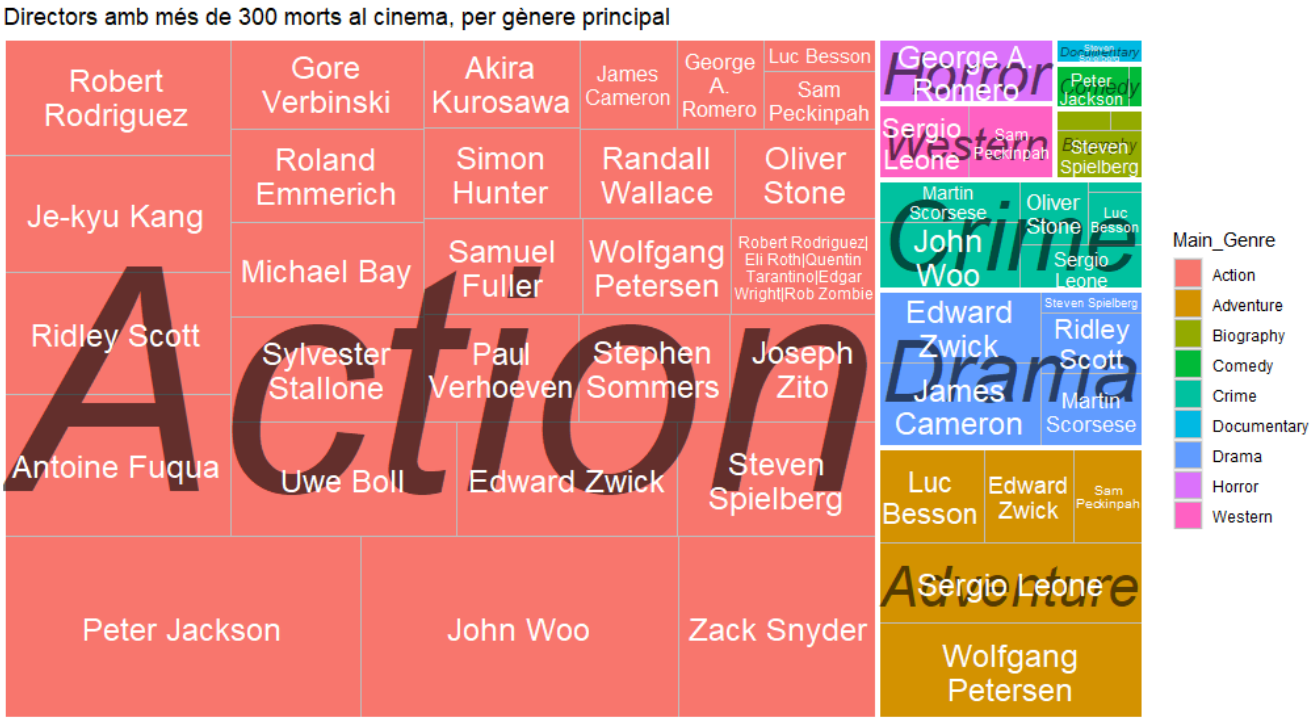
- Per definir el color (`fill`) i actuar doncs com un 'grup pare' utilitzem la variable Main_Genre Agafem aquesta com grup de color, ja que té menys nivells que l'altra variable categòrica que tenim (Director).
- Com a 'label' pel text utilitzaríem el Director.
- Com a variable que descrigui l'àrea de les caselles triarem la variable que hem creat durant el data massatge: Total_Deaths. Aquesta tria ve donada perquè de totes les variables que ens pregunten quelcom, és l'única numèrica.

El codi:

```
# Llibreries necessàries
library(treemapify)

# Crear treemap jeràrquic: Director (nivell inferior) dins de
Main_Genre (nivell superior)

> ggplot(dataT, aes(area = Total_Deaths,
  fill = Main_Genre,
  label = Director,
  subgroup = Main_Genre
)) +
  geom_treemap() +
  geom_treemap_subgroup_border(color = "white") +
  geom_treemap_subgroup_text(place = "centre", grow = TRUE,
                             alpha = 0.6, colour = "black",
fontface = "italic") +
  geom_treemap_text(colour = "white", place = "centre", reflow =
TRUE) +
  labs(title = "Directors amb més de 300 morts al cinema, per gènere
principal")
```



RESPOSTA 2.1.2.c:

En aventura Wolfgang Petersen i en crim John Woo.

2.2. (Total: 1,5 pts.)

a) Ajudat d'un gràfic per mostrar si la puntuació mitjana de la pel·lícula IMDb (IMDB_Rating) està correlacionada amb alguna de les variables numèriques d'aquest dataframe. (0,75 pts.)

b) Mostra el gràfic, tot intentant que no hi hagi informació repetida, i argumenta l'elecció del gràfic. En pots extreure alguna conclusió? (0,5 pts.)

c) Descriu com faries un gràfic per veure varia la valoració d'IMDB en funció d'aquestes variables numèriques, separant la informació per la classificació per edats establerta per la Motion Picture Association of America (MPAA_Rating). Nota: En aquest apartat, no es demana fer el gràfic, solament dir quin gràfic podria mostrar-ho i com. (0,25 pts.)

RESPOSTES:

RESPOSTA 2.2.a):

Primer ens quedem amb les dades numèriques:

```
num_data <- data %>% select(Year, Body_Count, Length_Minutes,
IMDB_Rating)
```

Ens demanen la correlació entre la variable IMDB_Rating i la resta de dades numèriques del nostre dataframe. Podríem fer totes les combinacions amb gràfic de punts, però una

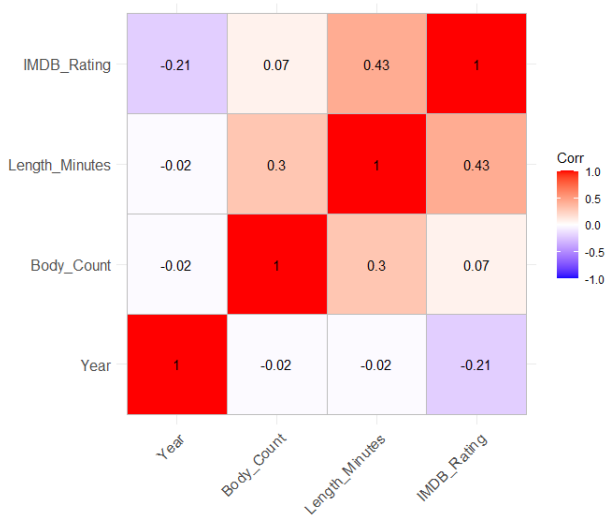
millor opció és mostrar una matriu de correlació. Creem la matriu de correlació i donat que tenim valors amb dos decimals com a molt, no ens cal arrodonir.

Carreguem la llibreria necessària:

```
> library(ggcorrplot)
```

```
> cormat <- cor(num_data)
```

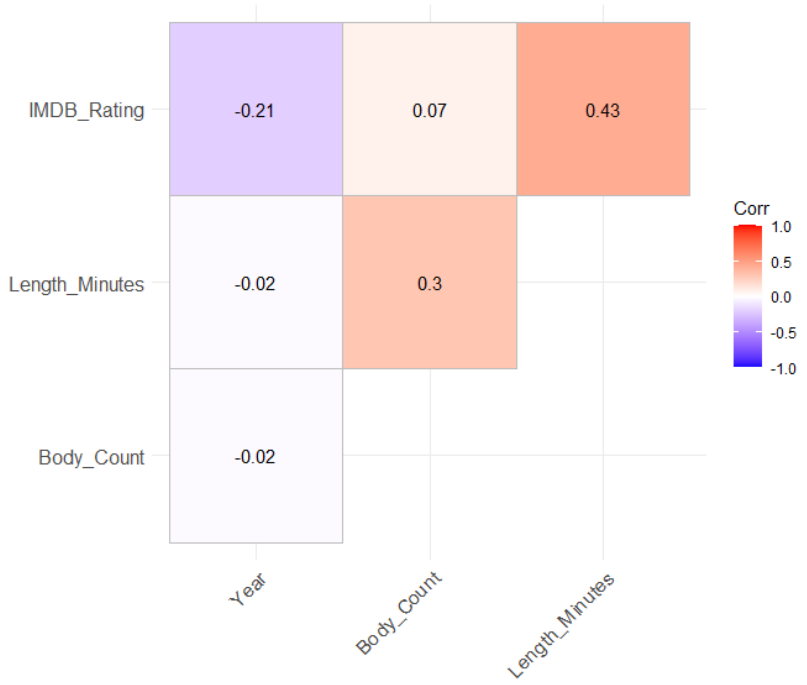
```
> ggcorrplot(cormat,lab=TRUE)
```



RESPOSTA 2.2.b):

Com ens demanen: “tot intentant que no hi hagi informació repetida”, mostrem només la part d’adalt (per exemple) d’aquest matriu simètrica:

```
>ggcorrplot(cormat, lab=TRUE, type = "upper")
```



Sembla que no hi ha correlació entre el IMDB_Rating i el Body_Count. A mesura que augmenta l'any (pel·lícules més recents), la puntuació IMDb tendeix a baixar lleugerament, però és una relació feble (només -0.21). Sí que hi ha una correlació positiva però de 0.43 (no forta). Recordem que la correlació positiva ens diu que quan una variable augmenta, l'altra també tendeix a augmentar.

RESPOSTA 2.2.c):

En lloc d'una matriu de correlació, una molt bona alternativa per explorar la relació entre IMDB_Rating i altres variables numèriques (com ara *Body_Count*, *Length_Minutes* o *Year*) és fer un **gràfic de dispersió (scatter plot) amb facets**. Això et permet veure com varia la valoració d'IMDB en funció d'aquestes variables, separant-ho per alguna variable categòrica com la **MPAA_Rating**.