

# Tarea Examen AED

Alejandro Sanz Sánchez

## Exploracion inicial de los datos

Se ha realizado una exploración de las tablas de excel de forma visual, para ver la clase de ficheros de datos con los que se deberá trabajar, de esta forma para el fichero de funcionamiento, observamos que algunos de los campos estarán vacíos o todos sus valores son nulos, por tanto podremos eliminar algunas de las columnas de datos, ya que no aportaran información útil para el uso de los datos proporcionados

## Importación de los ficheros de datos

```
## Warning in read_fun(path = path, sheet_i = sheet, limits = limits, shim =
## shim, : Expecting date in F16 / R16C6: got '-'
## Warning in read_fun(path = path, sheet_i = sheet, limits = limits, shim =
## shim, : Expecting date in F75 / R75C6: got '-'
```

## Análisis de los ficheros

### Fichero ParametrosFuncionamiento

```
## Observations: 635,023
## Variables: 17
## $ Reg_variable      <dbl> 2766475, 2766476, 2766477, 2766478, 27...
## $ Fecha_registro    <dttm> 2018-07-01 00:00:46, 2018-07-01 00:00...
## $ ID_Variable        <dbl> 267, 223, 224, 221, 220, 219, 218, 222...
## $ ID Grupo           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ ID_Control          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ ID_Agrupacion       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ Responsable_Registro <chr> "nd", "nd", "nd", "nd", "nd", "nd", ...
## $ Valor                <chr> "0", "0", "0", "0", "0", "0", "0"...
## $ Valor_Nominal         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ Valor_Maximo_Adminisble <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ Valor_Minimo_Adminisble <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ Alerta_Generada       <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, ...
## $ AR_Identificador     <chr> "106 TOTAL CAUDAL A DOSIFICAR", "78 DO...
## $ TRAZ_Id_Maquina       <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ...
## $ TRAZ_Id_Reg_Maquina   <chr> "636659999988854", "636659999988854", ...
## $ Observaciones          <chr> "nd", "nd", "nd", "nd", "nd", "n...
## $ Firma                 <lgl> NA, NA, NA, NA, NA, NA, NA, NA...
```

Vemos en el glimpse el tipo de variables contenidas en el dataframe importado y que en algunas de las variables de nuestro dataset ds contienen campos con el mismo numero, contiene todo NA o el mismo carácter. No obstante profundizaremos el estudio, antes de proceder a la eliminación de variables del mismo.

Sospechamos que las variables ID\_Grupo, ID\_Control, ID\_Agrupacion, Responsable\_Registro, Valor, Valor\_Nominal, Valor\_Maximo\_Admissible, Valor\_Minimo\_Admissible, Alerta\_generada, TRAZ\_ID\_Maquina, Observaciones, Firma contienen unicamente un valor o son directamente NA.

Para comprobar exactamente que es lo que tenemos, realizaremos la suma de todos los valores contenidos en las diferentes columnas que contienen valores numéricos, así como para las que contienen NA, cuantas de las filas no contiene NA.

```
## Tenemos un total 635023 de valores diferentes en la variable Reg_variable
## Tenemos un total 123178 fechas distintas
## Tenemos un total 90 de valores diferentes en la variable Reg_variable
## La suma de los numeros contenidos en la variable ID Grupo es: 0
## La suma de los numeros contenidos en la variable numID_Control es: 0
## En la variable ID_Agrupacion hay un total de 0 valores distintos a NA
## En la variable Responsable_Registro contiene los valores nd y NA
## La suma de los numeros contenidos en la variable Valor es: 1585231249
## La suma de los numeros contenidos en la variable Valor_Nominal es: 0
## La suma de los numeros contenidos en la variable Valor_Maximo_Adminisble es: 0
## La suma de los numeros contenidos en la variable Valor_Minimo_Adminisble es: 0
## En la variable Alerta_Generada hay un total de 0 valores distintos de el valor FALSE.
## Tenemos un total 90 codigos diferentes en la variable AR_Identificador
## En la variable TRAZ_ID_Maquina hay un total de 0 valores distintos de el numero 4.
## Tenemos un total 680 valores diferentes en la variable TRAZ_Id_Reg_Maquina
## En la variable Observaciones hay un total de 0 campos distintos de nd.
## En la variable Firma hay un total de 0 valores distintos a NA.
```

De lo anterior podemos ver que variables contienen datos distintos de cero o NA y que ademas cuantos valores distintos tiene cada una de las variables. Con lo especificado en el enunciado, todas las variables que contengan el mismo valor,cero o NA serán descartadas para el análisis posterior de los datos proporcionados.Cabe destacar la importancia de convertir la variable de Valor de carácter a numérico, ya que no eliminaremos la variable de nuestro conjunto de datos, considerando primero que dicha variable tiene la coma decimal como “,” en lugar de “.” , esto será importante ya que sino la transformación directa de carácter a variable nos modificará los valores de la misma , ya sea convirtiendo directamente a numérico o a factor y luego numérico.De igual forma, la variable Fecha\_Registro ya esta como variable de tipo fecha y hora, por lo tanto no realizares ninguna transformación(de momento), a la misma . Por otro lado la variable Responsable\_registro contiene los valores “ns”, así como NA y Firma contenga el valor “nd”, debe significar que no se conoce la persona encargada del registro y a raíz de esto, tampoco se ha realizado la firma. De la misma forma no se ha establecido ningún criterio para el valor nominal, el valor máximo/mínimo admisible, ya que estas 3 variables solo contienen ceros.

&nbsp

## Fichero Averias

```
glimpse(daver)

## Observations: 94
## Variables: 9
## $ Maq          <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ...
## $ OrdenProduccion <dbl> 201832724, 201832762, 201832762, 201832771, ...
```

```

## $ Referencia      <chr> "A2X25 MI H-0055-02", "B1S25 G15 H-0007-02",...
## $ FechaInicioProd <dttm> 2018-06-26 17:36:38, 2018-06-27 02:16:15, 2...
## $ FechaFinProd    <dttm> 2018-06-27 02:16:09, 2018-06-27 09:57:48, 2...
## $ FechaInicioAveria <dttm> 2018-06-26 17:36:38, 2018-06-27 02:06:21, 2...
## $ TipoAveria       <chr> "PARADAS", "CAMBIO", "AVERIAS", "PARADAS", ...
## $ Averia            <chr> "LIMPIEZA", "CAMBIO", "BOMBA VACIO", "LIMPIE...
## $ Duración          <chr> "14", "53", "8", "5", "9", "24", "38", "0", ...

Vamos ahora a profundizar en los datos contenidos en este segundo conjunto de datos, para ver el numero de valores distintos por variables y realizar una selección de las variables que contengan información útil.

#Variable Maq
nummaq <- sum(which(daver$Maq!=4))
cat('En la variable Maq hay un total de ', nummaq, ' valores distintos de el numero 4.',," \n")

## En la variable Maq hay un total de 0 valores distintos de el numero 4.

#Variable OrdenProduccion
num_Orden <- length(unique(daver$OrdenProduccion))
cat('Tenemos un total ', num_Orden, 'valores distintos en la variable ordenProduccion ',," \n")

## Tenemos un total 49 valores distintos en la variable ordenProduccion

#Variable Referencia
num_ref <- length(unique(daver$Referencia))
cat('Tenemos un total ', num_ref, 'valores distintos en la variable Referencia ',," \n")

## Tenemos un total 31 valores distintos en la variable Referencia

#Variable FechaInicioProd
num_FechaInicioProd <- length(unique(daver$FechaInicioProd))
cat('Tenemos un total ', num_FechaInicioProd, 'fechas distintas en la variable FechaInicioProd ',," \n")

## Tenemos un total 49 fechas distintas en la variable FechaInicioProd

#Variable FechaFinProd
num_FechaFinProd <- length(unique(daver$FechaFinProd))
cat('Tenemos un total ', num_FechaFinProd, 'fechas distintas en la variable FechaFinProd',," \n")

## Tenemos un total 49 fechas distintas en la variable FechaFinProd

#Variable FechaInicioAveria

num_FechaInicioAveria <- length(unique(daver$FechaInicioAveria))
cat('Tenemos un total ', num_FechaInicioAveria, 'fechas distintas en la variable FechaInicioAveria',," \n")

## Tenemos un total 93 fechas distintas en la variable FechaInicioAveria

#Variable TipoAveria
num_TipoAveria <- length(unique(daver$TipoAveria))
cat('Tenemos un total ', num_TipoAveria, 'tipos de averias distintas en la variable TipoAveria ',," \n")

## Tenemos un total 6 tipos de averias distintas en la variable TipoAveria

#Variable Averia
num_Averia <- length(unique(daver$Averia))
cat('Tenemos un total ', num_Averia, ' averias distintas en la variable Averias ',," \n")

## Tenemos un total 15 averias distintas en la variable Averias

#Variable Duración
daver$Duración <- as.numeric(daver$Duración)

```

```

## Warning: NAs introducidos por coerción
num_duracion <- length(unique(daver$Duración))
cat('Tenemos un total ', num_duracion, ' valores distintos en la variable Duración ', "\n")

## Tenemos un total 47 valores distintos en la variable Duración
#Eliminamos las variables intermedias que hemos creado para realizar el conteo de valores

rm(nummaq,num_Orden,num_ref,num_FechaInicioProd,num_FechaFinProd,num_FechaInicioAveria,num_TipoAveria,n

```

Eliminaremos la variable “maq”, ya que no aporta nada todos los valore son los mismo, ya que como se nos ha dicho previamente todos los valores pertenecen a la misma máquina. Y hemos convertido la variable de duración a numérico, ya que parece adecuado el real izarlo para el uso futuro.

```
daver2 <- select(daver,-Maq)
```

Una vez llegados a este punto hemos conseguido limpiar los datos iniciales que se nos había presentado y se puede proceder a las siguientes secciones de la tarea.

&nbsp

## Analisis del fichero de averias.

```

library(knitr)
library(kableExtra)
Tablacon <- table(daver2$Averia,daver2$TipoAveria)

kable(Tablacon,align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "r

```

	-	AVERIAS	CAMBIO	LABORATORIO	MICROPARADA	PARADAS
-	2	0	0	0	0	0
BOMBA VACIO	0	9	0	0	0	0
CALEFACCION	0	1	0	0	0	0
CAMBIO	0	0	33	0	0	0
CAMBIO CABEZAL	0	0	0	0	0	2
CAMBIO MALLA	0	0	0	0	0	4
CAMBIO TALLARINA	0	0	0	0	0	4
CARGADORES	0	2	0	0	0	0
COLOR (AJUSTE)	0	0	0	2	0	0
DOSIFICADORES	0	2	0	0	0	0
ELECTRICA	0	1	0	0	0	0
LIMPIEZA	0	0	0	0	0	15
MICROPARADA	0	0	0	0	15	0
ROTURA DE HILOS	0	0	0	0	0	1
TRANSPORTE NEUMATICO	0	1	0	0	0	0

```
rm(Tablacon)
```

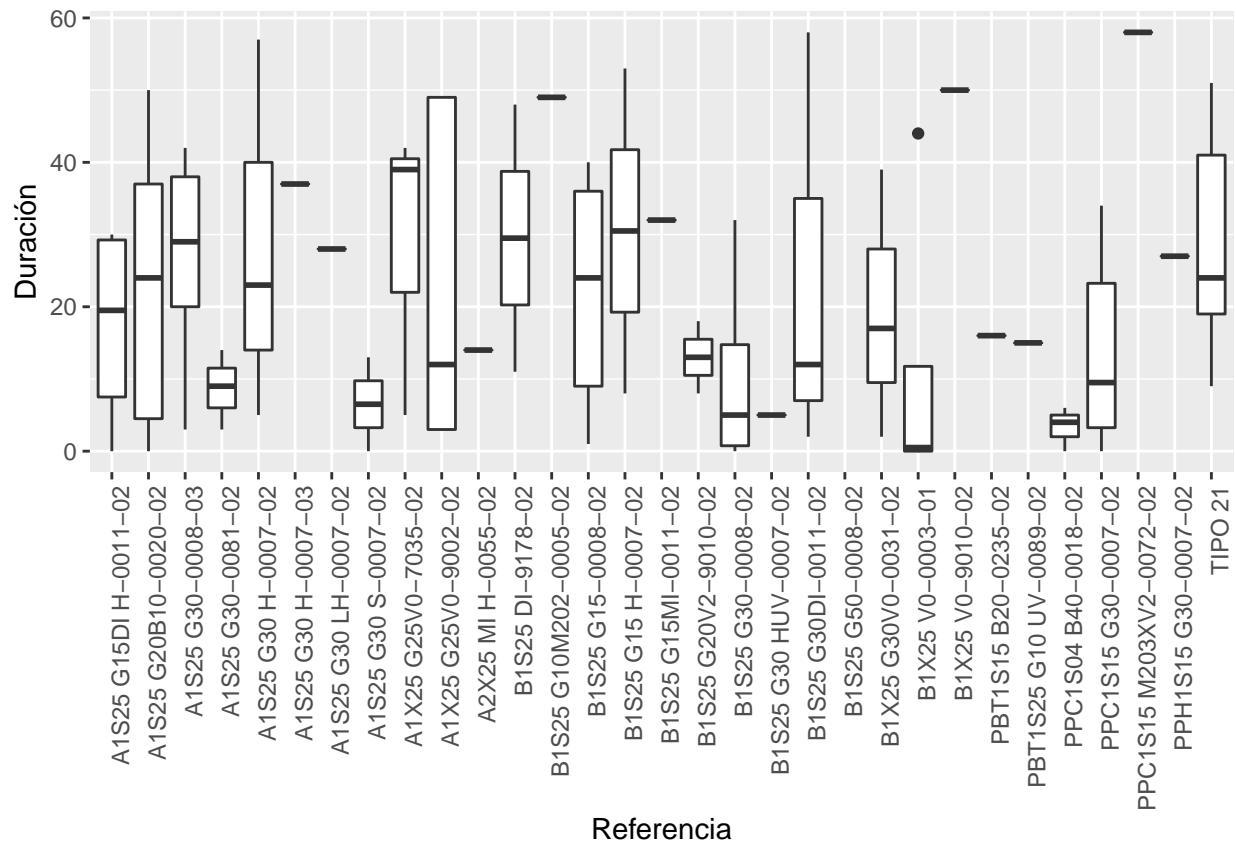
Las averías de tipo “PARADA” se deben 2 casos de Cambio de cabezal, 4 casos a cambio de malla, 4 a cambio de tallarina, 15 operaciones de limpieza y 1 caso de rotura de hilos.

Las averías de tipo “AVERIA” se deben a 9 casos de fallo de la bomba de vacío, 1 caso de fallo de calefacción, 2 fallos de los cargadores, 2 fallos de los dosificadores, 1 caso de avería electiva y 1 fallo de transporte

neumático.

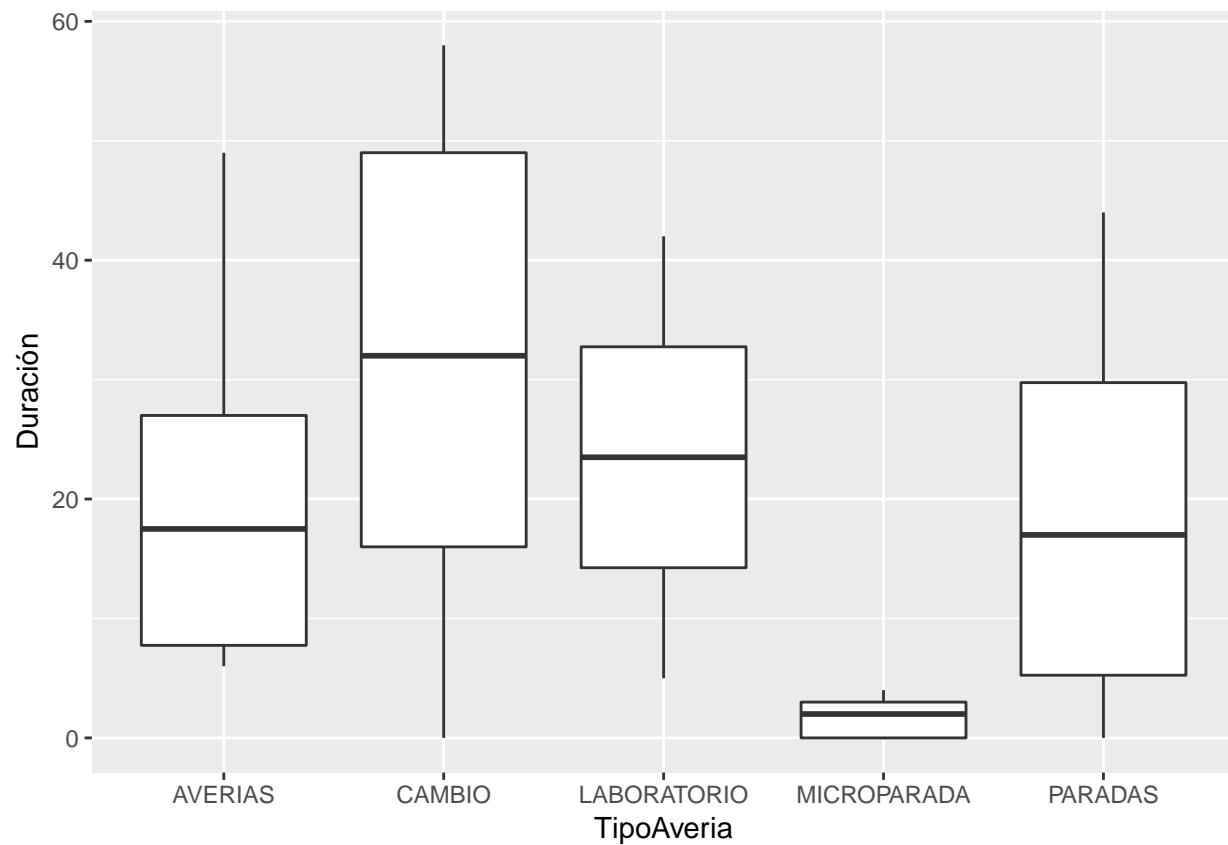
Representamos las duraciones de las averías por tipos de plásticos fabricados.

```
library(ggplot2)
ggplot(daver2)+geom_boxplot(aes(x=Referencia,y=`Duración`))+ theme(axis.text.x = element_text(angle = 90))
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```

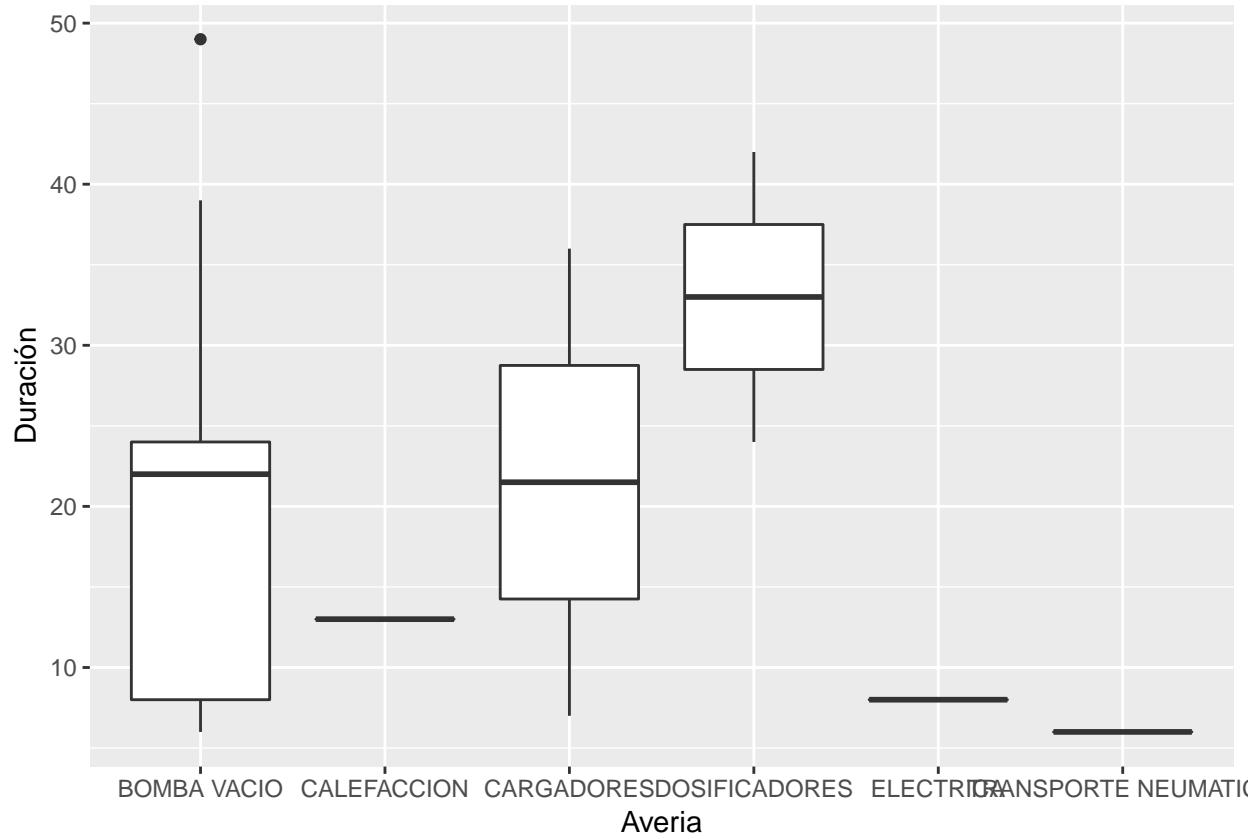


Podemos tambien realizar un boxplot para ver las duraciones medias de las averías y los tipos de averia:

```
ggplot(daver2%>%filter(TipoAveria!="-"))+geom_boxplot(aes(x=TipoAveria,y=`Duración`))
```



```
ggplot(daver2%>%filter(TipoAveria=="AVERIAS"))+  
  geom_boxplot(aes(x=Averia,y=`Duración`))
```



## Periodicidad del fichero de Averías

Vamos a estudiar ahora la periodicidad del fichero Averías.

```
daver2 <- mutate(daver2, IncFechaProd=FechaFinProd-FechaInicioProd, IncFechaInicioAveria=FechaInicioAveria-FechaInicioProd)

daver2$IncFechaInicioAveriaInicioProd <- as.difftime(as.numeric(daver2$IncFechaInicioAveriaInicioProd), units="hours")

daver2$IncFechaInicioAveriaFinProd <- as.difftime(as.numeric(daver2$IncFechaInicioAveriaFinProd), units="hours")
```

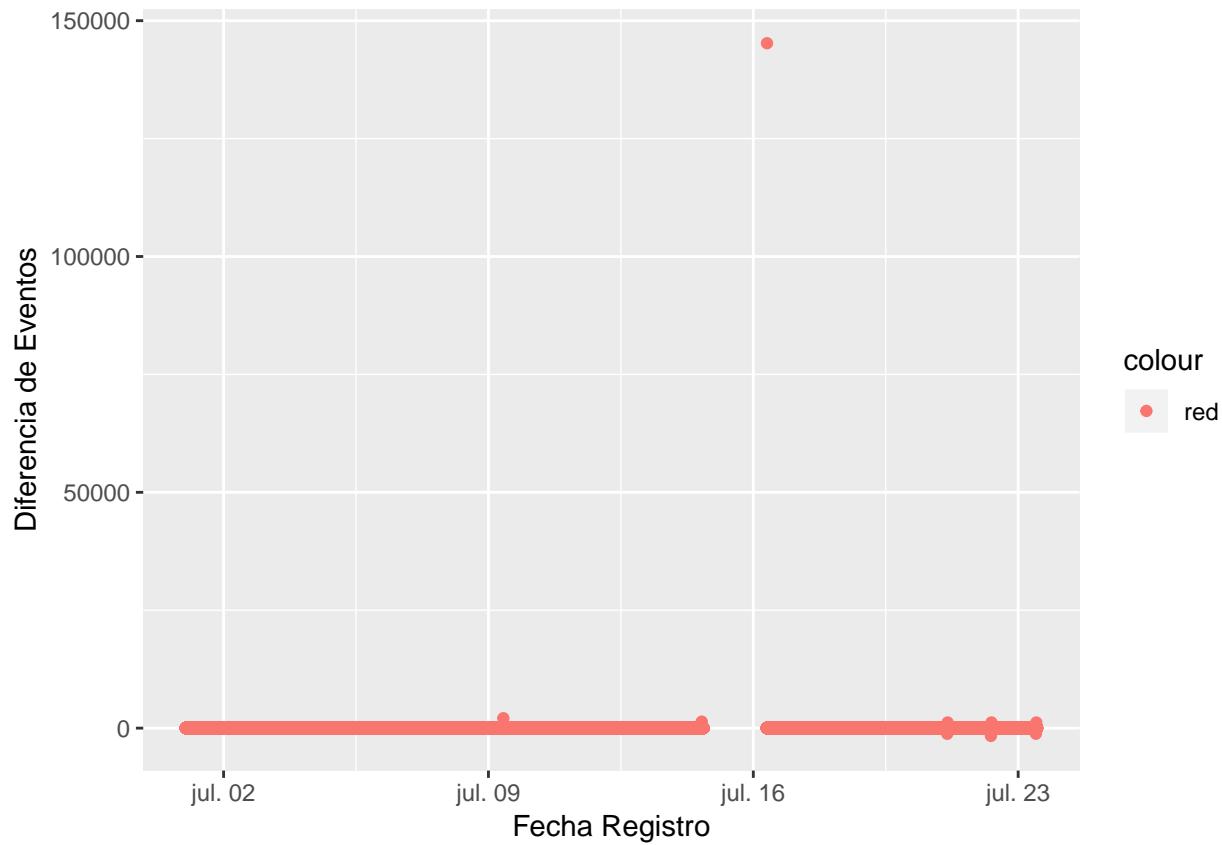
Hemos tenido que transformar algunas de las diferencias temporales a horas, ya que así se especificaba en el enunciado.

Ahora realizamos las diferencias temporales para el conjunto de datos ds

```
ds <- mutate(ds, difer.eventos=c(0,diff(Fecha_registro)))
```

Realizamos ahora la representación gráfica de la diferencia del la Fecha\_registro y la diferencia de tiempos, para ver si podemos encontrar algún tipo de patrón.

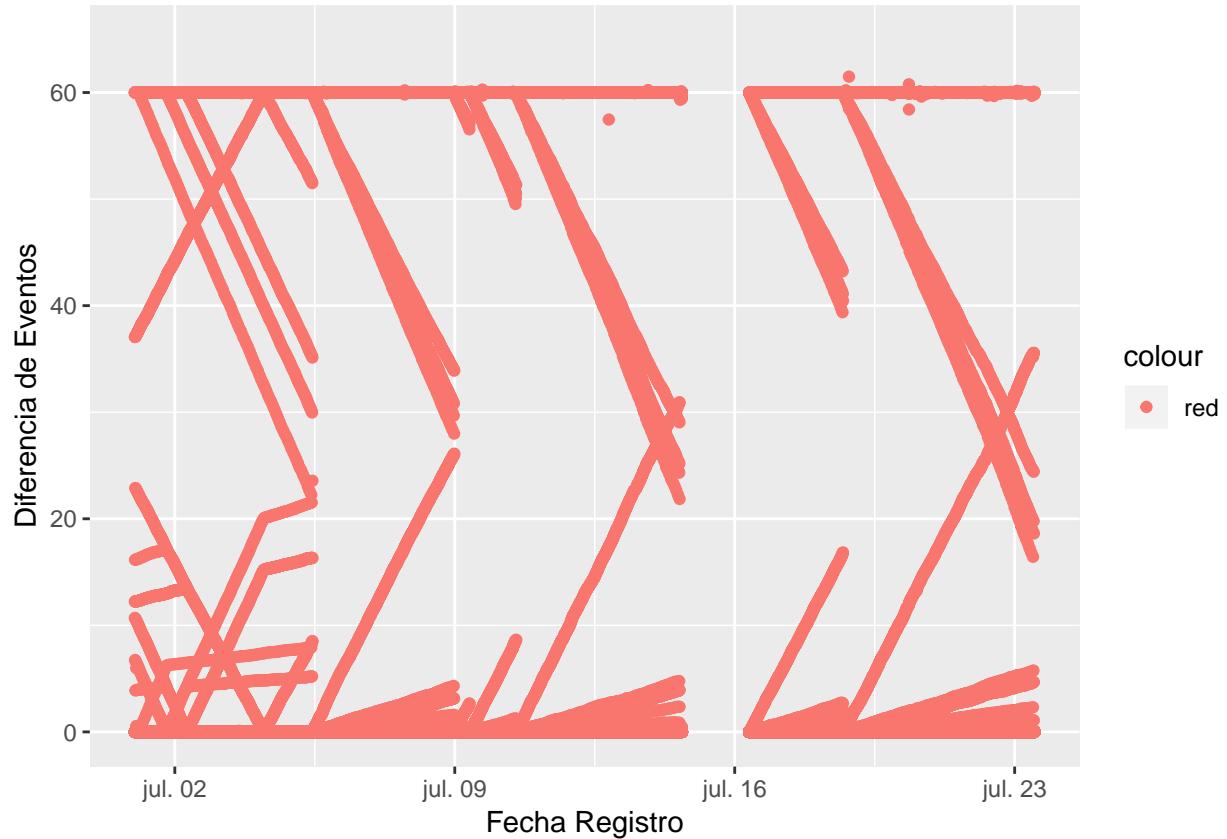
```
library(ggplot2)
ggplot(ds,aes(ds$Fecha_registro,ds$difer.eventos,color="red"))+geom_point()+labs(x = "Fecha Registro", y = "Diferencia de Tiempo")
```



Hemos encontrado que tenemos un posible outlier para los valores de las diferencias temporales, a consecuencia de esto la escala con la que se ha representado no permite discernir ningún tipo de patrón. Procederé a representar la misma gráfica, limitando los valores representados en el eje y ,para ver si así podemos observar algún tipo de patrón.

```
ggplot(ds,aes(ds$Fecha_registro,ds$difer.eventos,color="red"))+geom_point() +ylim(0,65)+labs(x = "Fecha Registro",y = "Diferencia de Eventos")
```

```
## Warning: Removed 52 rows containing missing values (geom_point).
```

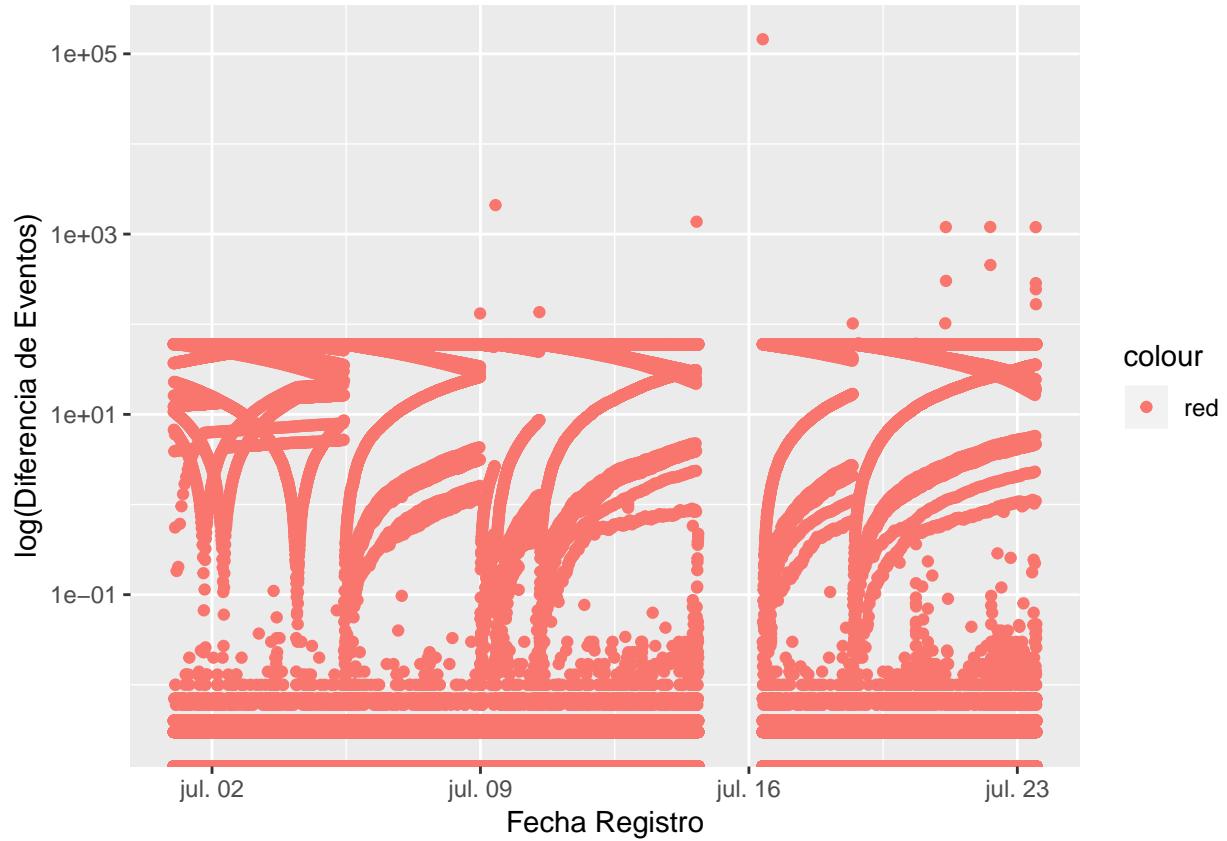


Observamos que tenemos algún tipo de patrón, ya que las gráficas se parecen para días diferentes, requerimos de mas estudio para ver si tenemos algún tipo de patrón. Se ha Representado entre 0 y 75 para poder observar mejor el patrón, ya que tenemos ciertos puntos por encima de 100, pero es una cantidad despreciable, en comparación con la cantidad de puntos a representar.

observamos así mismo que para diferentes días las diferencias tienen tanto regiones donde la diferencia temporal se mantiene constante en algunas regiones , no obstante a o largo del día también hay zonas donde la diferencias temporales decrecen y en crecen.

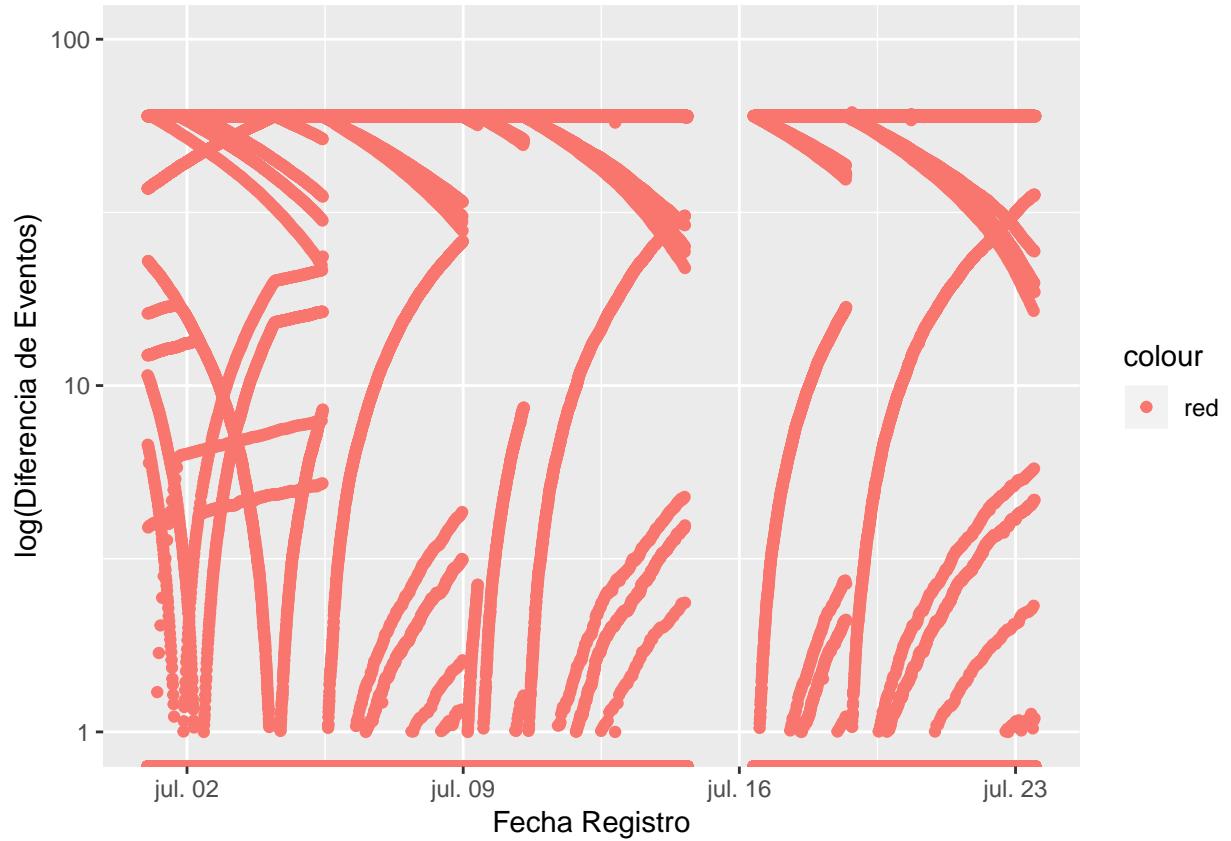
Vamos ahora a tomar el logaritmo en base 10 para transformar los valores de las diferencias de los eventos, con el fin de intentar observar algún patrón mas claro, que para las representaciones anteriores.

```
ggplot(ds,aes(ds$Fecha_registro,ds$difer.eventos,color="red"))+geom_point()+scale_y_log10()+labs(x = "F
## Warning in self$trans$transform(x): Se han producido NaNs
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 37 rows containing missing values (geom_point).
```



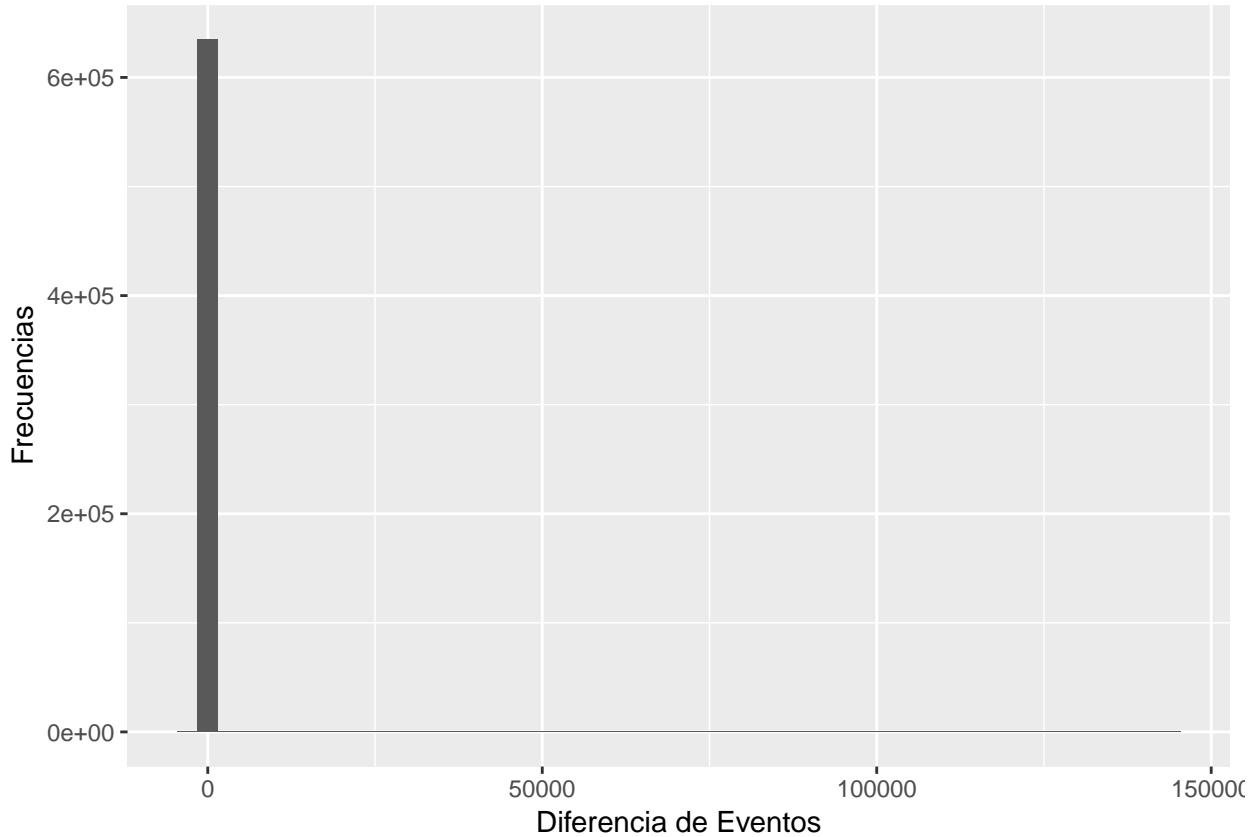
Observamos de nuevo un patrón, de nuevo los valores que podrían tratarse de outliers nos modifican de forma importante la escala de representación, vamos a limitar de nuevo la escala, para ver la representación de forma mas detallada.

```
ggplot(ds,aes(ds$Fecha_registro,ds$difer.eventos,color="red"))+geom_point()+scale_y_log10(limits = c(1,100000))
## Warning in self$trans$transform(x): Se han producido NaNs
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 84347 rows containing missing values (geom_point).
```



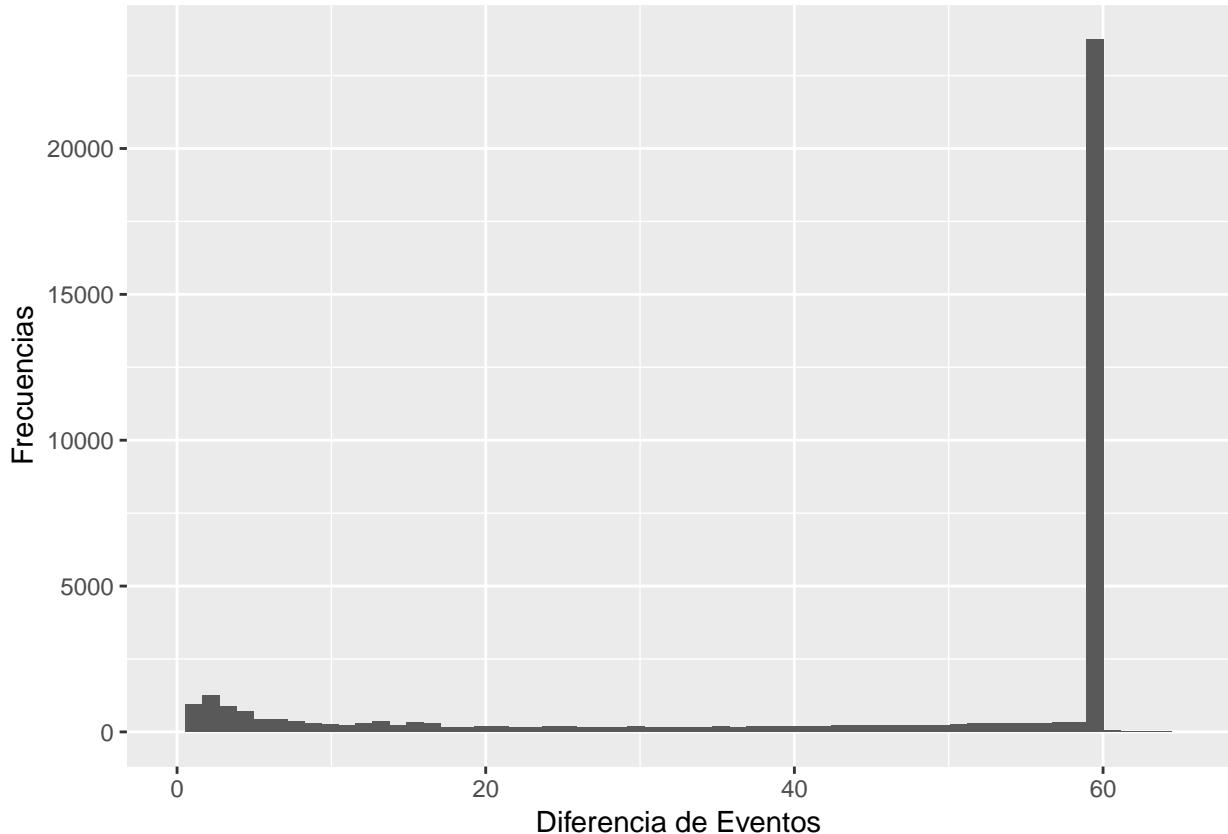
Ahora se puede discernir mejor la gráfica, no obstante vemos patrones de crecimiento y decrecimiento, así como líneas rectas para los valores de 100 y de 60 segundos, como en el caso de la representación sin logaritmo.

```
ggplot(ds)+geom_histogram(aes(ds$difer.eventos),bins = 50)+labs(x = "Diferencia de Eventos", y = "Frecuencia")
```



No se puede observar nada en dicho histograma debido a que la mayoría de los valores están agrupados entre los valores de 0 y 150. Ademas cabe destacar que la escala que obtenemos se debe a que tenemos un posible valor outlier que tiene como valor

```
ggplot(filter(ds,difer.eventos>1 & difer.eventos<120),aes(difer.eventos))+geom_histogram(bins = 60)+xlin  
## Warning: Removed 2 rows containing non-finite values (stat_bin).  
## Warning: Removed 2 rows containing missing values (geom_bar).
```



Tras representar el histograma, vemos que principalmente se agrupan entorno al valor 60 segundos, siendo la frecuencia de este mucho mayor que el resto, esto se debe a que se toman varias medidas para un mismo tiempo y al minuto se vuelve a realizar una medida. Por eso el periodo de muestreo mas frecuente es el de los 60 segundos. Cabe destacar que se ha representado para un valor máximo de 60 segundos, ya que tras el filtrado aplicado, no se obtenía para el rango especificado ningún valor por encima de 60.

Creamos la nueva variable que contiene los tiempos de registro redondeados al minuto.

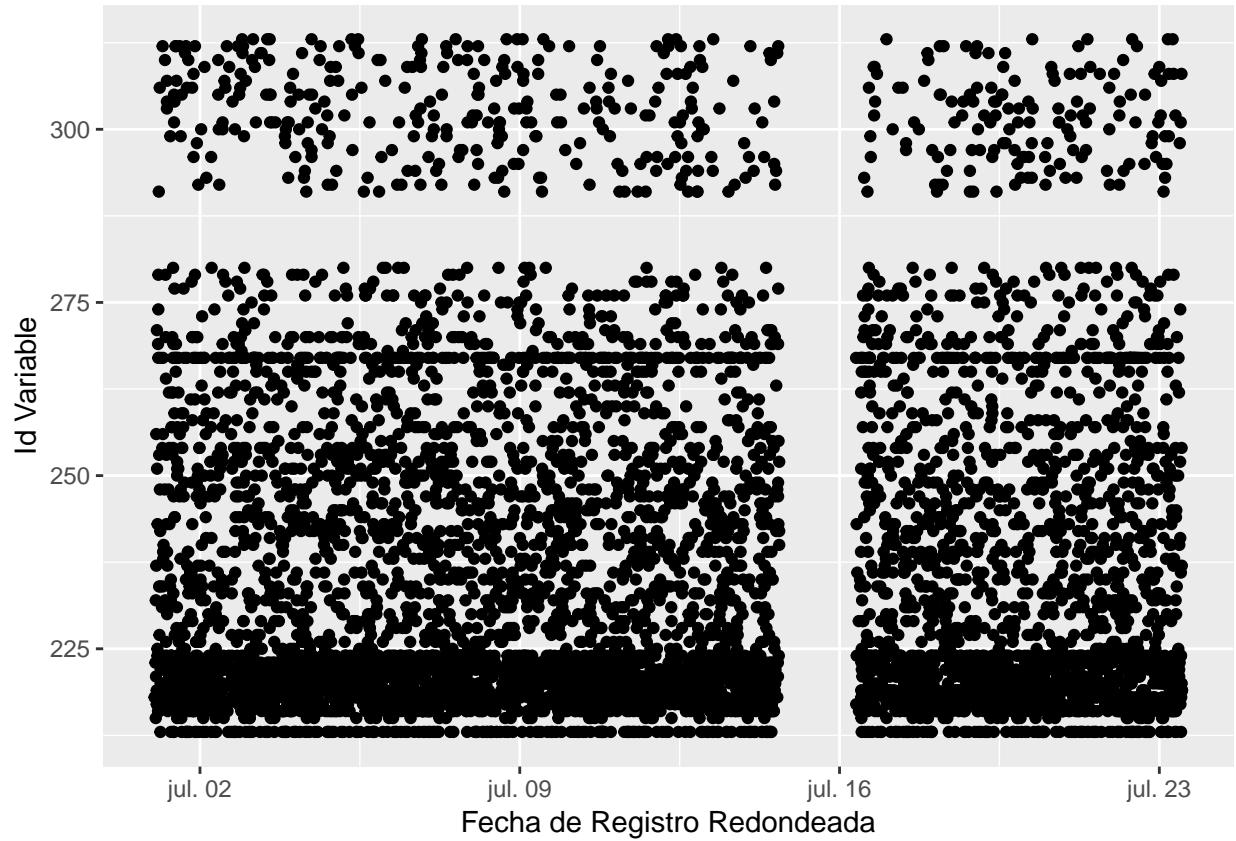
```
library(lubridate)
ds <- mutate(ds, Fecha_registro_Red=round_date(Fecha_registro, unit = "minute"))
```

Seleccionamos ahora de forma aleatoria el 1% de los eventos de datafram ds

```
set.seed(3141592)
porc <- nrow(ds)*0.01 # para saber cuantas observaciones corresponderian al 1% del conjunto de datos
dsrand <- ds[sample(nrow(ds), porc), ] # Seleccion del 1% de forma aleatoria del datafram ds
```

Realizamos ahora el gráfico de la Id\_variable frente a la Fecha\_registro\_Red

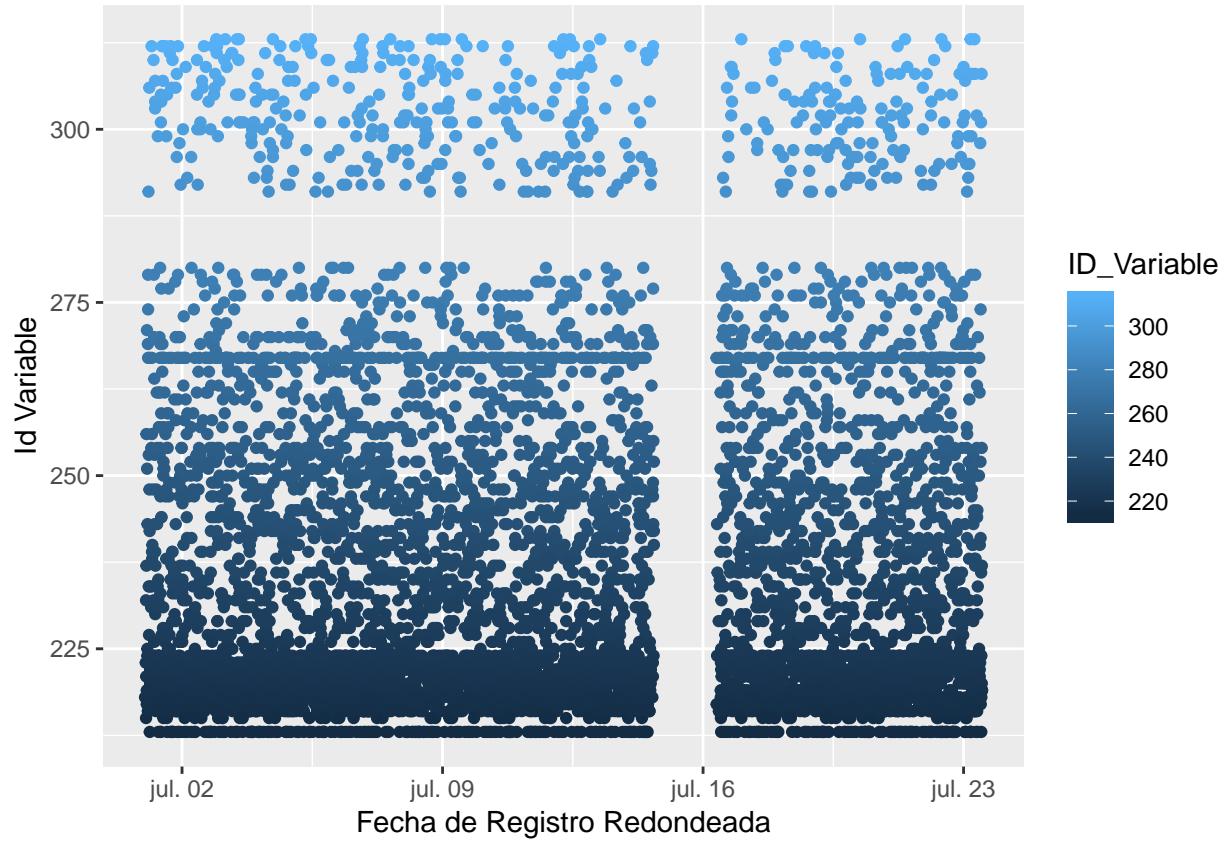
```
ggplot(dsrand, aes(x=Fecha_registro_Red, y=ID_Variable))+geom_point()+labs(x = "Fecha de Registro Redondeada")
```



Encontramos para el muestreo del 1% de los datos originales, una agrupación de puntos importante por debajo del valor de Id variable de 225, así como algunas franjas que por la agrupación de los puntos se asemejan a líneas rectas tanto para un Id Variable menor de 225 y otro cercano a 270. Así mismo encontramos que las alrededor de 16 de Julio, no tenemos muestreo de datos.

Realizamos ahora la misma gráfica, agrupando por colores para los distintos valores de la ID\_variable.

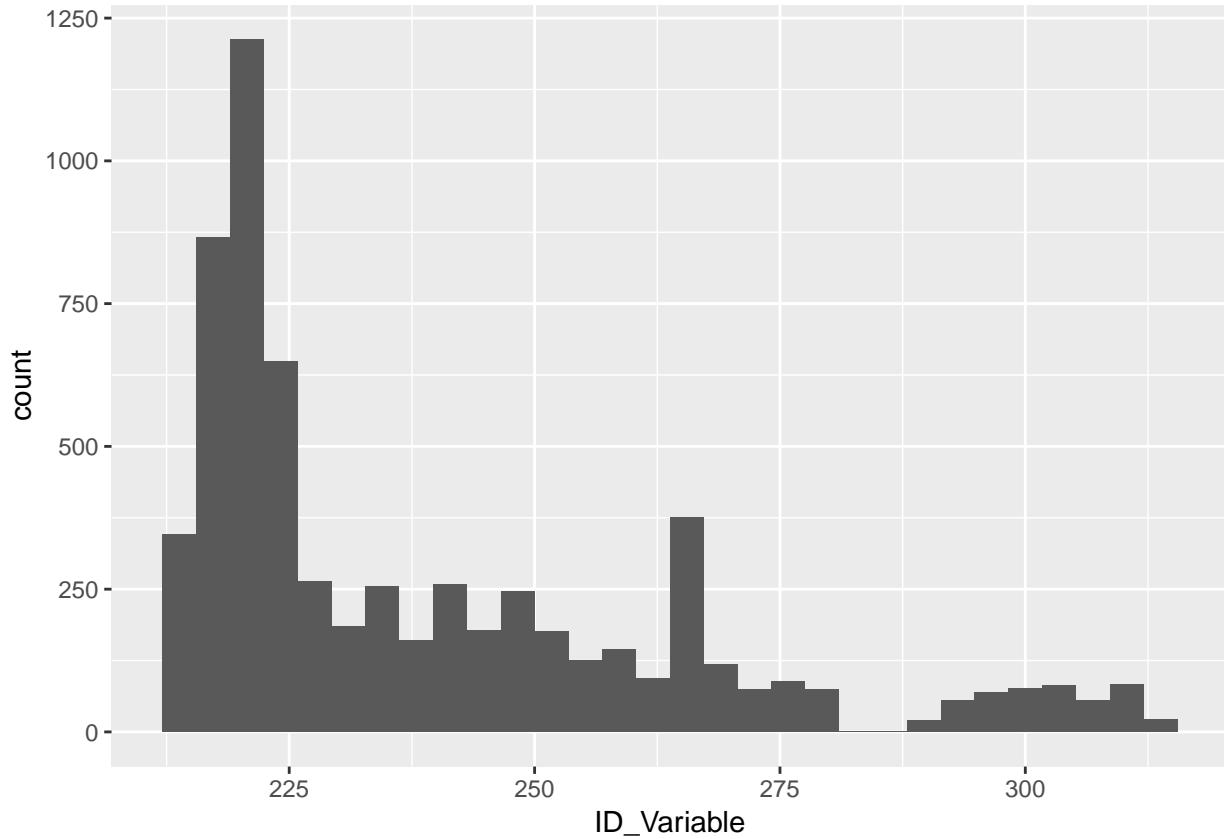
```
ggplot(dsrand,aes(x=Fecha_registro_Red,y=ID_Variable,col=ID_Variable))+geom_point()+labs(x = "Fecha de Registro Redondeada",y = "ID_Variable")
```



Se observa de nuevo, como tenemos una agrupación de puntos para el rango de 225, ya que se puede observar con el gradiente de colores obtenido al emplear la Id variable como color para los distintos puntos , un mayor numerario de zonas con azul color intenso que corresponde a aquellos valores por debajo de 225.

Realizamos un histograma para la Id\_variable, para confirmar lo afirmado anteriormente.

```
ggplot(dsrand,aes(ID_Variable))+geom_histogram()
```



En efecto la mayor cantidad de valores estan agrupado por debajo de 225, así mismo cabe destacar que tenemos una región donde no hay valores para Id\_Variable.

```
rm(porc,dsrand)
```

## Grupos en Base a la frecuencia de muestreo.

```
detach(package:dplyr)
detach(package:ggplot2)
library(dplyr)
RepEvento.ID_var <- ds %>% group_by(AR_Identificador) %>% summarise(rep=n())%>% arrange(desc(rep))
# empleamos el summarise para poder contar el numero de repeticiones con la instruccion n() de dplyr y

library(kableExtra)
Valmax <- RepEvento.ID_var %>% filter(rep==max(rep))

kable(Valmax,align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

AR_Identificador	rep
0 SET POINT DOSIFICACIÓN A	29910
106 TOTAL CAUDAL A DOSIFICAR	29910
20 SET POINT DOSIFICACIÓN B	29910
20 SET POINT DOSIFICACIÓN C	29910
20 SET POINT DOSIFICACIÓN D	29910
20 SET POINT DOSIFICACIÓN E	29910
78 DOSIFICACIÓN B	29910
78 DOSIFICACIÓN C	29910
78 DOSIFICACIÓN D	29910
78 DOSIFICACIÓN E	29910
78 Dosificador A	29910

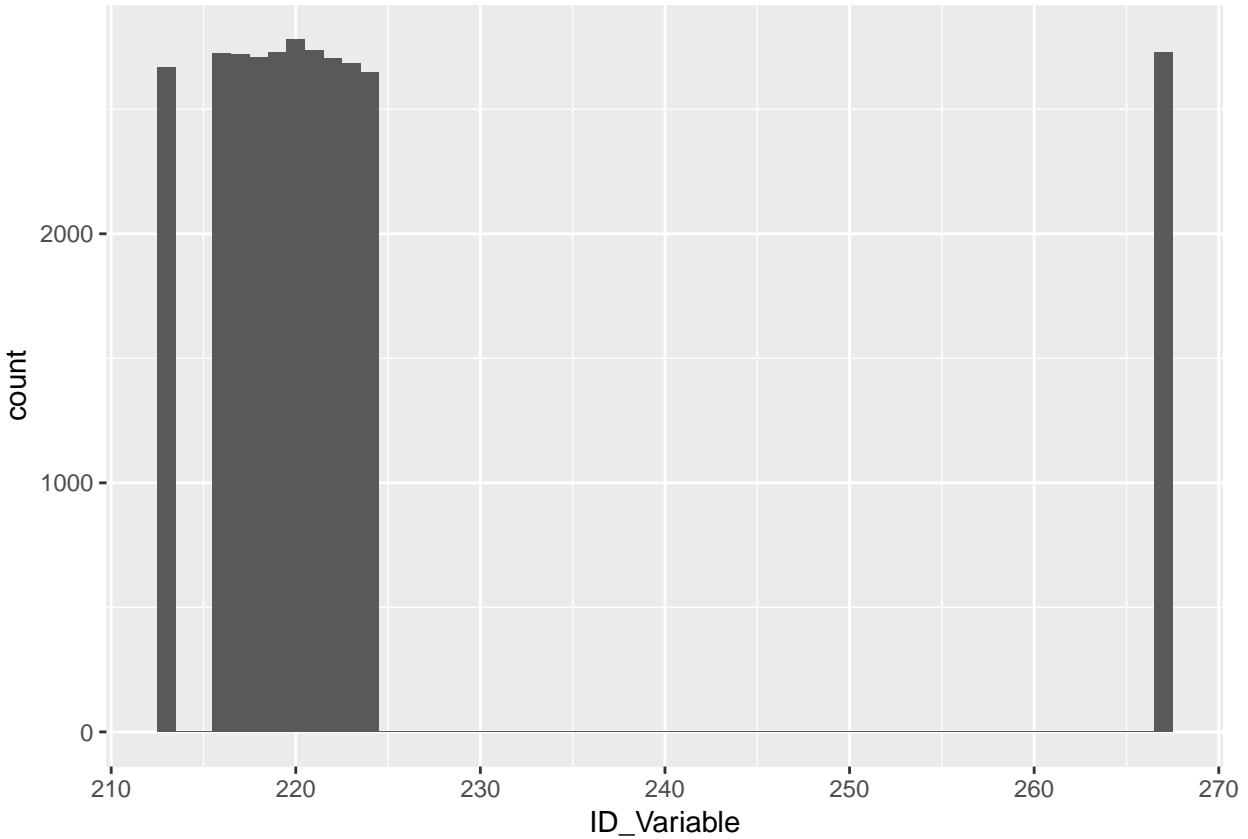
En la tabla anterior se refleja las variables con las mayores frecuencias de muestreo.

Para ver si tiene relación las variables con mayor muestro con la ID\_variable, realizamos una selección de los valores de ds que contienen el AR\_identificador de aquellas muestras que tienen el mayor numero de cuentas y realizamos un histograma de los valores para ID\_Variable.

```
ID_varrel <- ds %>% filter(AR_Identificador==c("0 SET POINT DOSIFICACIÓN A", "106 TOTAL CAUDAL A DOSIFICAR"))

## Warning in AR_Identificador == c("0 SET POINT DOSIFICACIÓN A", "106 TOTAL CAUDAL A DOSIFICAR", : longitud de objeto mayor no es múltiplo de la
## longitud de uno menor

library(ggplot2)
ggplot(ID_varrel,aes(ID_Variable))+geom_histogram(binwidth = 1)
```



```
rm(ID_varrel)
```

Como se puede ver de nuevo, aquellas variables que tienen mayor frecuencia , poseen al mismo tiempo valores de ID\_variable alrededor de 220, como en los casos anteriores ya estudiados.

```
cat('Las repeticiones diferentes que tenemos seran:',unique(RepEvento.ID_var$rep))
```

```
## Las repeticiones diferentes que tenemos seran: 29910 5987 2996 2000 505
```

De esta forma suponemos que ahora debemos asignar las etiquetas para los 5 periodos que se han obtenido al medir las repeticiones, donde 01 se asignara a las variables con repetición 29910, 05 a las que tengan repetición 5987 y así para el resto.

```
#Para crear la nueva variable que meteremos en el dataframe RepEvento.Id
```

```
et <- as.character()
for (i in seq(nrow(RepEvento.ID_var))){

  if(RepEvento.ID_var$rep[i]==29910)
    et <- c(et,'01')
  if(RepEvento.ID_var$rep[i]==5987)
    et<- c(et,'05')
  if(RepEvento.ID_var$rep[i]==2996)
    et<- c(et,'10')
  if(RepEvento.ID_var$rep[i]==2000)
    et<- c(et,'15')
  if(RepEvento.ID_var$rep[i]==505)
    et<- c(et,'60')
}
```

```

et <- as.factor(et)

RepEvento.ID_var <- mutate(RepEvento.ID_var,nfac=et)
kable(head(RepEvento.ID_var),align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "borderless"))

```

AR_Identificador	rep	nfac
0 SET POINT DOSIFICACIÓN A	29910	01
106 TOTAL CAUDAL A DOSIFICAR	29910	01
20 SET POINT DOSIFICACIÓN B	29910	01
20 SET POINT DOSIFICACIÓN C	29910	01
20 SET POINT DOSIFICACIÓN D	29910	01
20 SET POINT DOSIFICACIÓN E	29910	01

```
rm(RepEvento.ID_var)
```

Se puede ver que se ha creado correctamente la nueva columna del dataframe y se ha guardado la variable nfac como factor.

Crearemos ahora una nueva variable para el dataframe ds , que combinará la periodicidad y las variables, para ello primero agruparemos los datos.

```

datainter <- ds %>% group_by(ID_Variable,AR_Identificador) %>% summarise(rep=n())

#Para crear la nueva variable que meteremos en el dataframe RepEvento.Id
et <- as.character()
for (i in seq(nrow(datainter))){

  if(datainter$rep[i]==29910)
    et <- c(et,'01')
  if(datainter$rep[i]==5987)
    et<- c(et,'05')
  if(datainter$rep[i]==2996)
    et<- c(et,'10')
  if(datainter$rep[i]==2000)
    et<- c(et,'15')
  if(datainter$rep[i]==505)
    et<- c(et,'60')
}
et <- as.factor(et)

datainter$nafc <- et
kable(head(datainter),
      ,align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive","borderless"))

```

ID_Variable	AR_Identificador	rep	nafc
213	78 Dosificador A	29910	01
215	Entrada Vacuometro	5987	05
216	0 SET POINT DOSIFICACIÓN A	29910	01
217	78 DOSIFICACIÓN B	29910	01
218	20 SET POINT DOSIFICACIÓN B	29910	01
219	78 DOSIFICACIÓN C	29910	01

```

library(tidyr)
dsinterm <- merge(ds,datainter,sort = F)
t <- rep("t",635023)
dsinterm$t <- t
dsinnuevo <- unite(dsinterm,"intter",c(t,nafc),sep="") #Al unir nos desaparecen las columnas que combinan

dsinnuevo1 <- unite(dsinnuevo,"ID_Variable_per",c(intter,ID_Variable),sep=".")
dsinnuevo1$ID_Variable <- ds$ID_Variable
dsinnuevo1$ID_Variable_per <- as.factor(dsinnuevo1$ID_Variable_per)

#Ahora guardamos el dataframe en el dataframe que teniamos originalmente.

ds <- dsinnuevo1

#eliminamos los dataframes y variables intermedias empleadas hasta el momento, para liberar memoria y espacio

rm(dsinnuevo,dsinnuevo1,i,et,Valmax,dsinterm,t,datainter)

```

## 4Analisis de las variales medidas en ds

Tras realizar esto guardaremos el resultado un nuevo dataframe llamado ds.nz, donde guardamos las variables del dataframe ds que no contienen valores nulos y que nos aportan algo de información. Se ha cambiado también los valores de la variable Valor de carácter a numérico, considerando que el separador de decimales en dicha variable inicialmente estaba con una coma en lugar de como un punto. Esto es importante, ya que si no se considera esto, se puede obtener una variable distinta a la esperada.

Creamos un nuevo dataset llamado ds.nz.col

```

ds.nz.col <- spread(ds.nz,ID_Variable_per,Valor)

# Comprobamos si los valores contenidos son los mismos en el dataaframe separado que en el original
sum(select(ds.nz.col,-(1:8)),na.rm=T)==sum(ds.nz$Valor)

## [1] TRUE

```

Creamos el dataframe llamado data partiendo del dataframe ds.nz.col

```

#Diria que asi esta bien, los enunciados estan demasiado confusos
data <- gather(ds.nz.col,Variable,Valor,-c(1:8))
data <- data[complete.cases(data[, -4]),]
data <- select(data,Fecha_registro_Red,Variable,Valor)

```

### 4.1 Analisis visual de ds

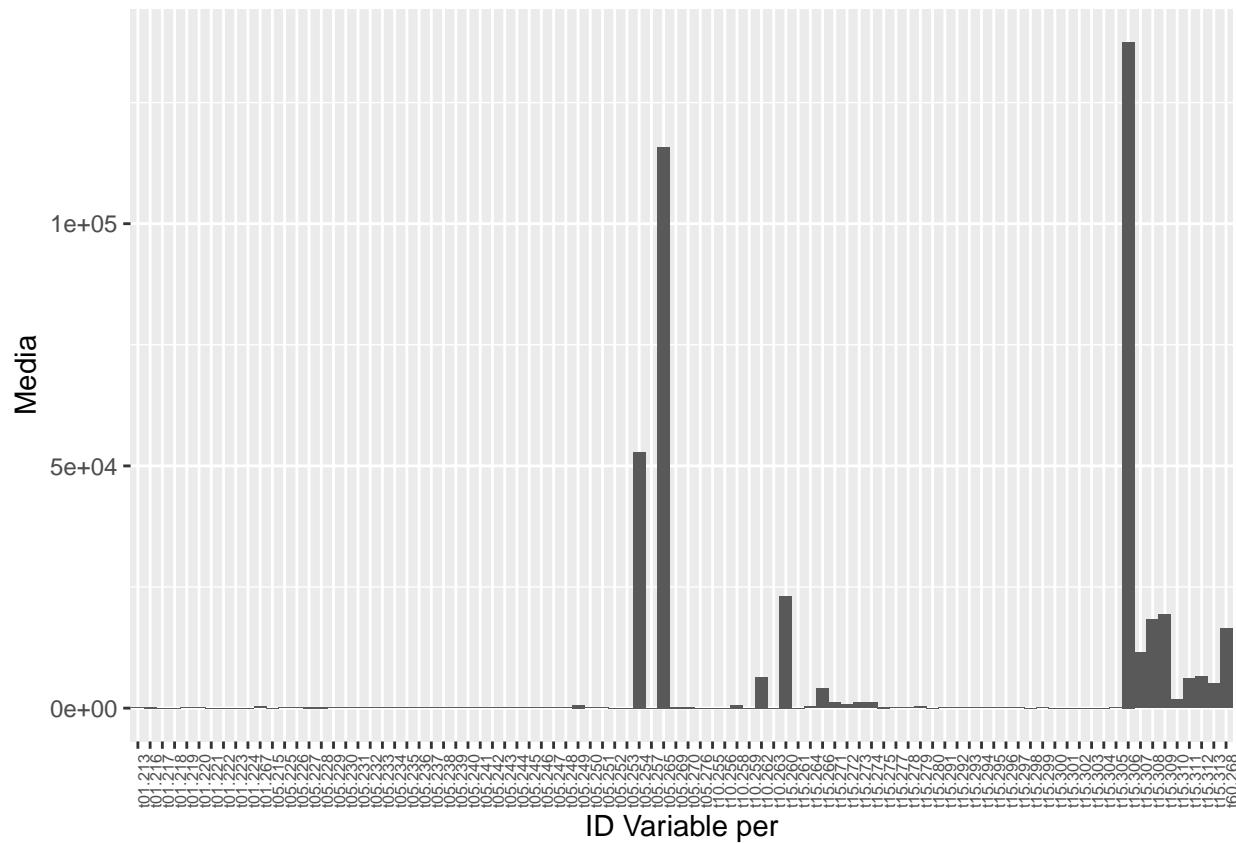
Suponemos que se refiere que a las variables se refiere a la columna ID\_Variable\_per de ds y hacemos la media para dicho dataframe.

```

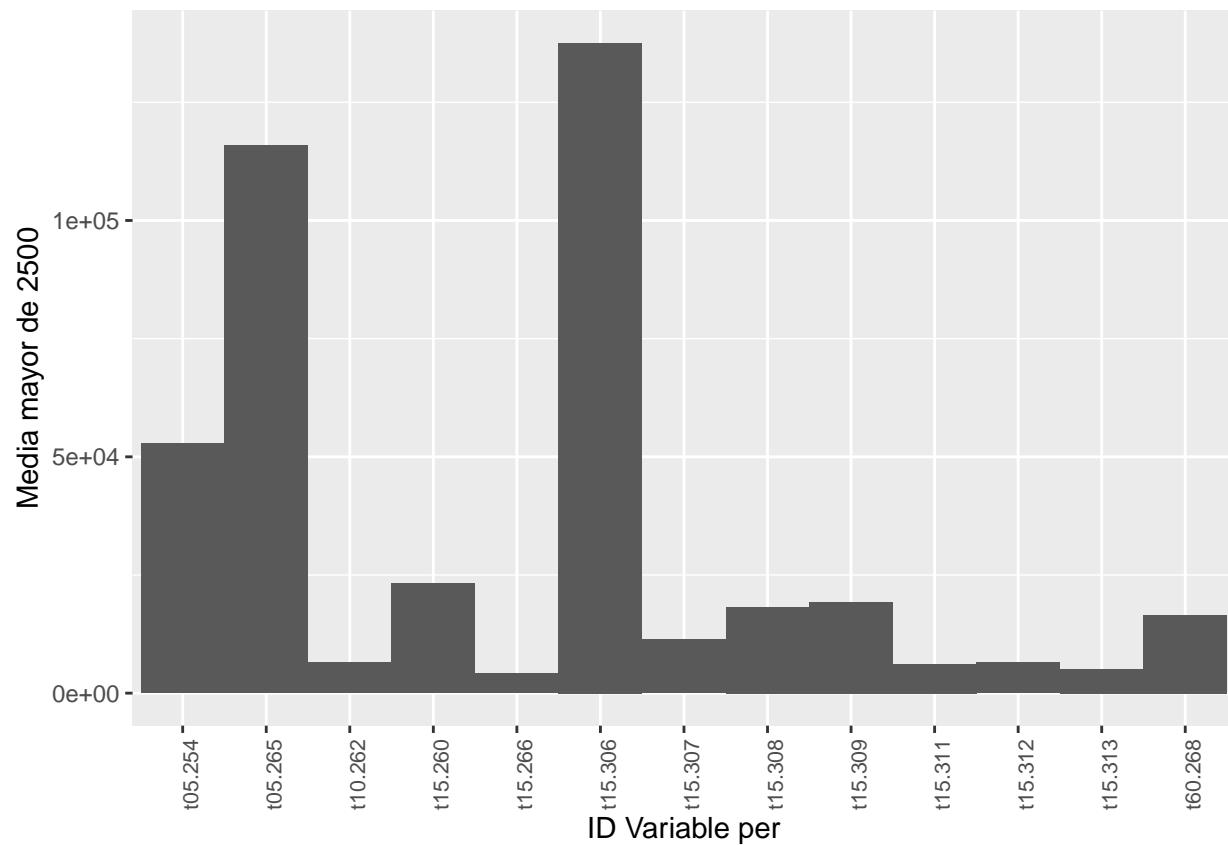
ds$Valor <- ds$Valor %>% retype()
datosmedias <- ds %>% group_by(ID_Variable_per) %>% summarise(media= mean(Valor))

ggplot(datosmedias,aes(ID_Variable_per,media))+geom_bar(width = 1, stat='identity')+theme(axis.text.x =

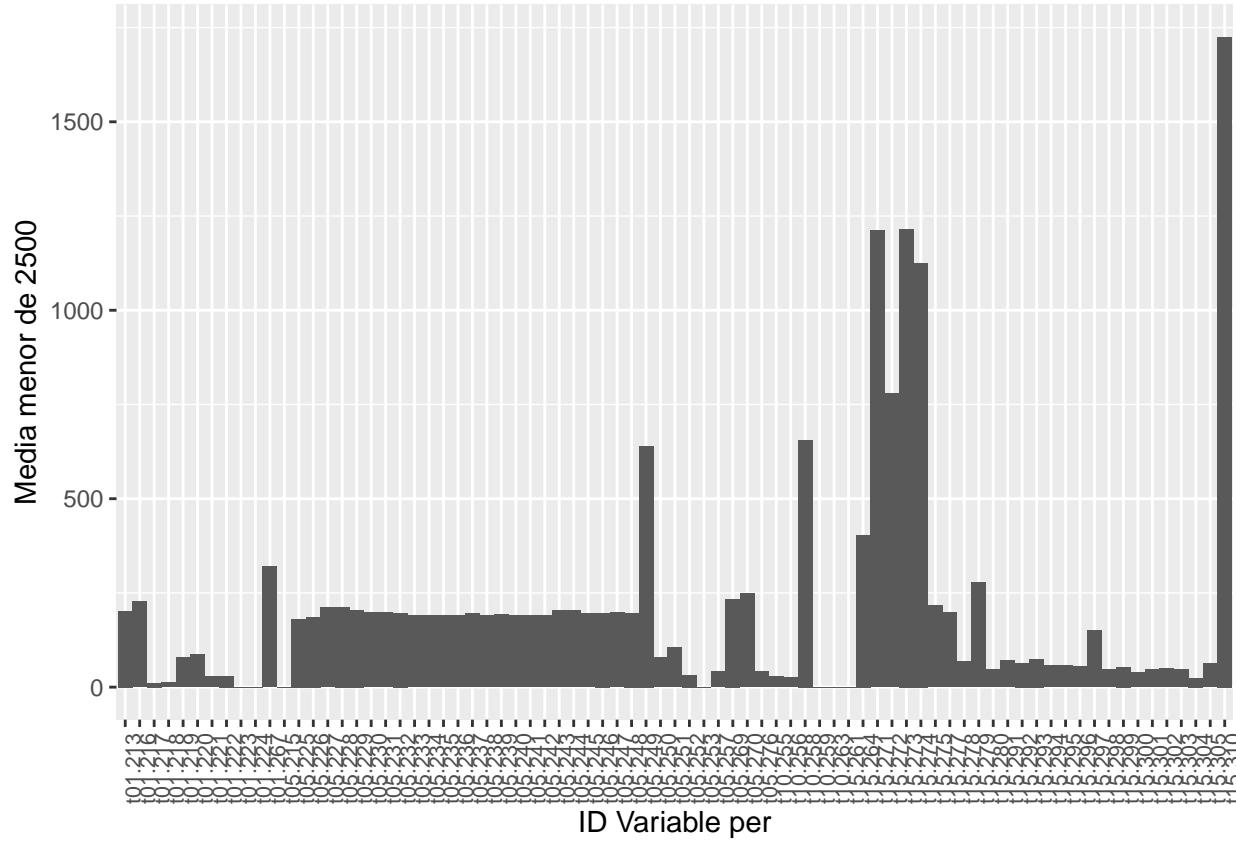
```



```
ggplot(filter(datosmedias,media>2500),aes(ID_Variable_per,media))+geom_bar(width = 1, stat='identity')+
```



```
ggplot(filter(datosmedias,media<2500),aes(ID_Variable_per,media))+geom_bar(width = 1, stat='identity')+
```

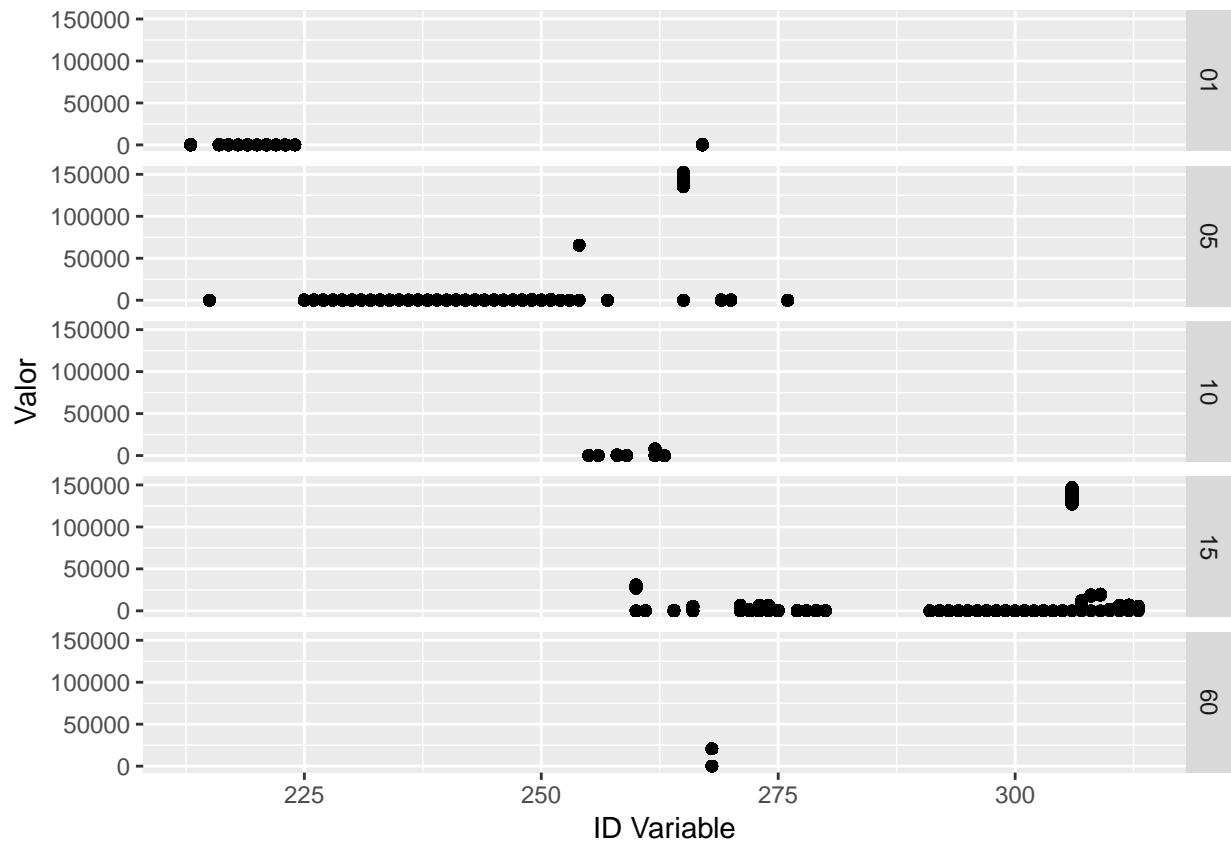


Queremos ahora conocer la información sobre la distribución de las variables, de esta forma queremos realizar un boxplot en función de la periodicidad creando un boxplot para cada grupo de variables.

```
data.group <- data %>% separate(Variable,c("periodo","ID_Variable"))
data.group$periodo <- as.factor(gsub("t", "", paste(data.group$periodo))) #para eliminar las t's de los periodos
data.group$ID_Variable <- as.numeric(data.group$ID_Variable)
```

Realizamos ahora la representación con ggplot empleando la instrucción facet\_grid.

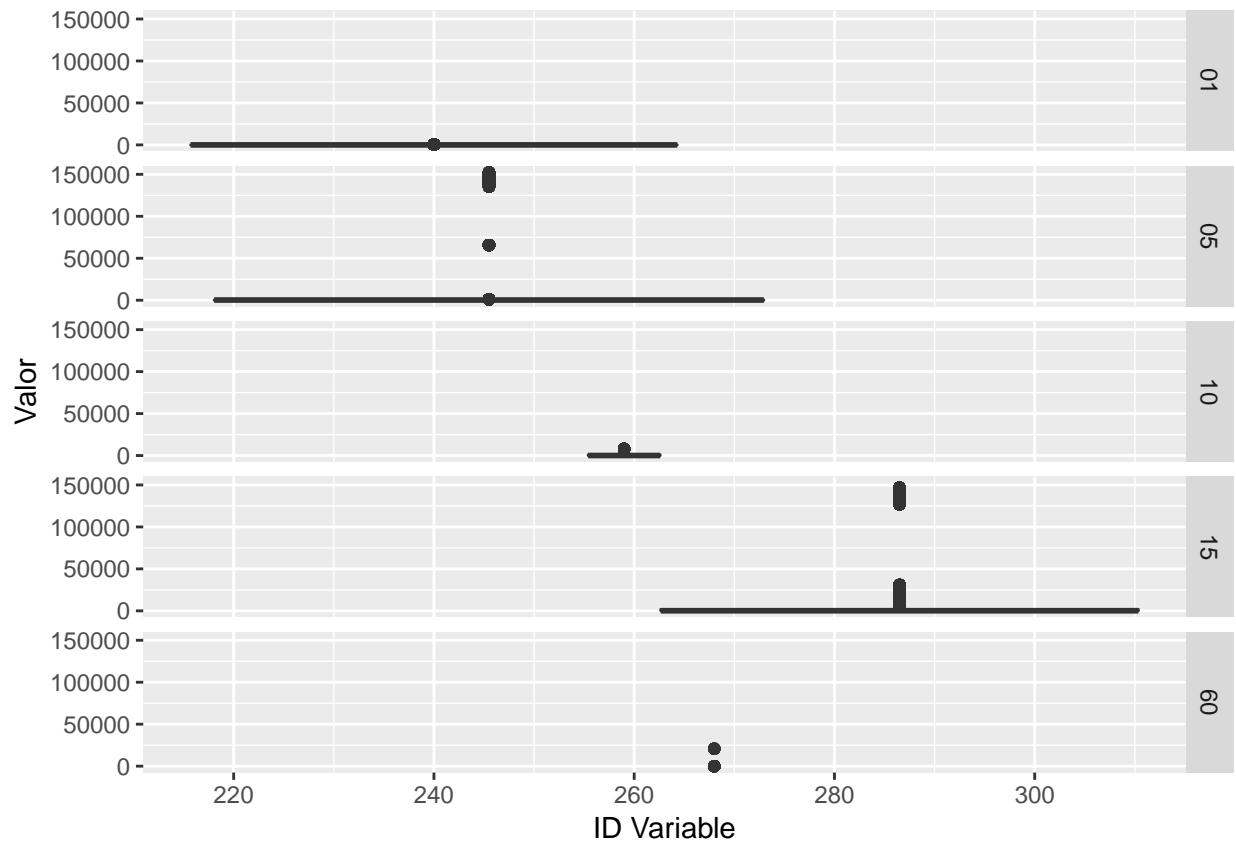
```
ggplot(data.group,aes(ID_Variable,Valor),color=periodo)+geom_point() +facet_grid(rows = vars(periodo))+
```



En la gráfica anterior podemos ver los puntos agrupados por periodo y representando el ID\_Variable frente al Valor. Vamos ahora a realizar el boxplot, para ver que se nos muestra.

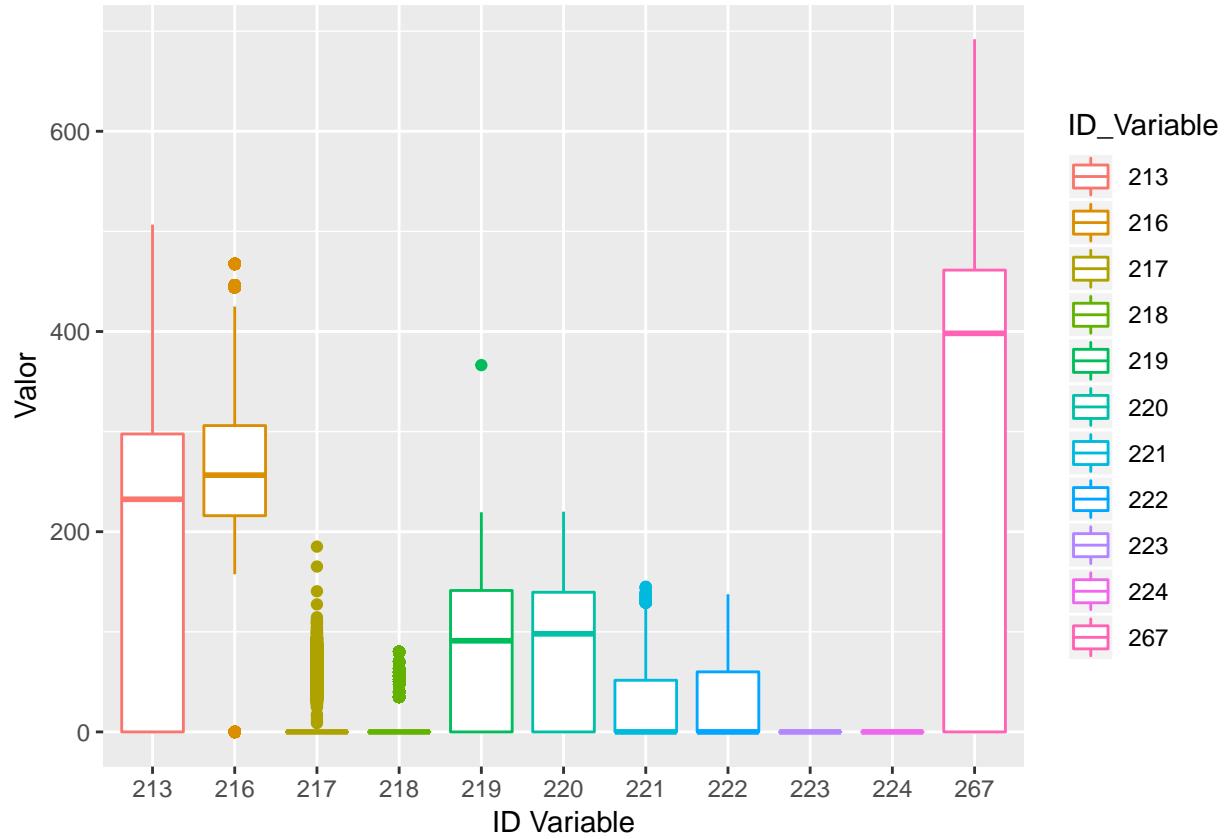
```
ggplot(data.group,aes(ID_Variable,Valor),colour=periodo)+geom_boxplot()+facet_grid(rows = vars(periodo))

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Debido a las diferencias de escalas vamos a representar para los diferentes periodos por separado.

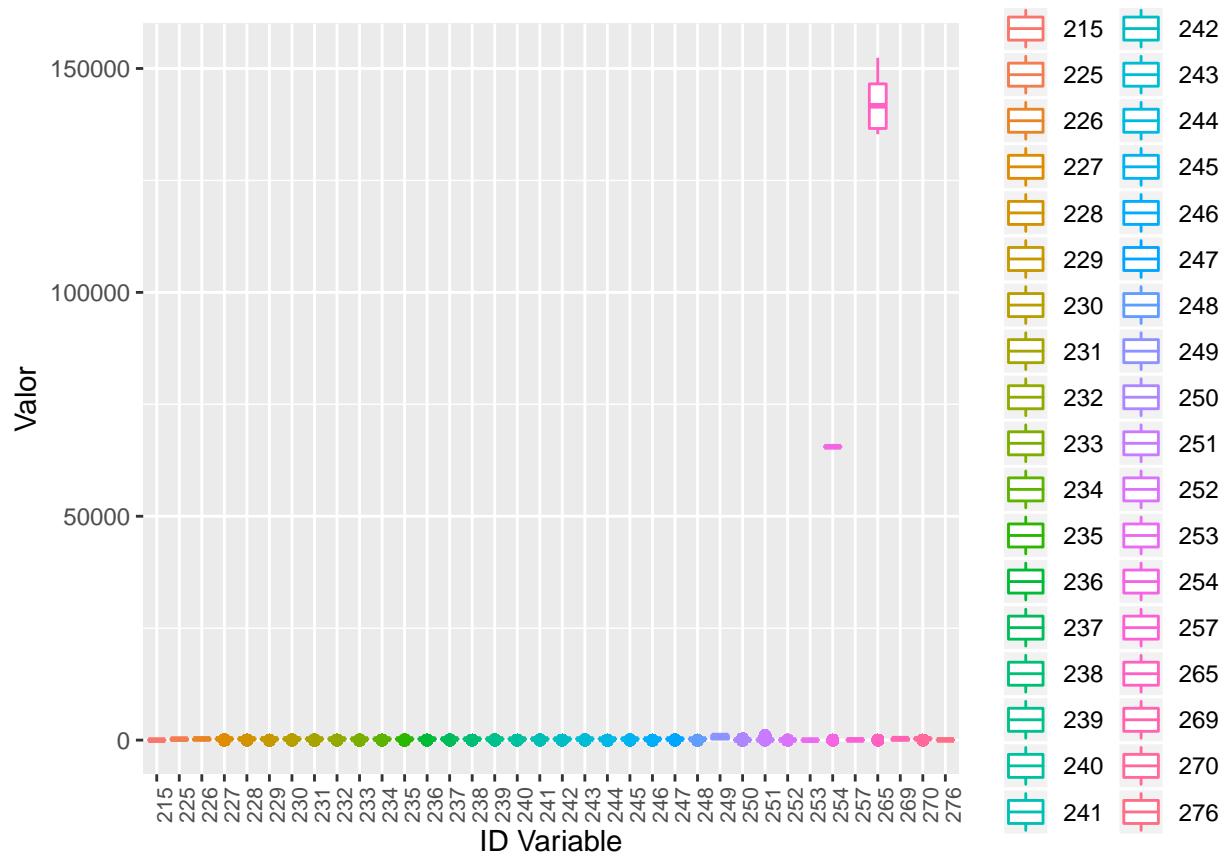
```
grupo01 <- filter(data.group, periodo=="01")
grupo01$ID_Variable <- as.factor(grupo01$ID_Variable)
ggplot(grupo01,aes(ID_Variable,Valor,col=ID_Variable))+geom_boxplot()+
  labs(x = "ID Variable", y = "Valor")
```



Cabe destacar que nos quedan para varias ID algo no apreciable, esto no se debe a la escala, sino a que contiene tantas cantidades de ceros que la media queda en cero.

```
grupo05 <- filter(data.group, periodo=="05")
grupo05$ID_Variable <- as.factor(grupo05$ID_Variable)
```

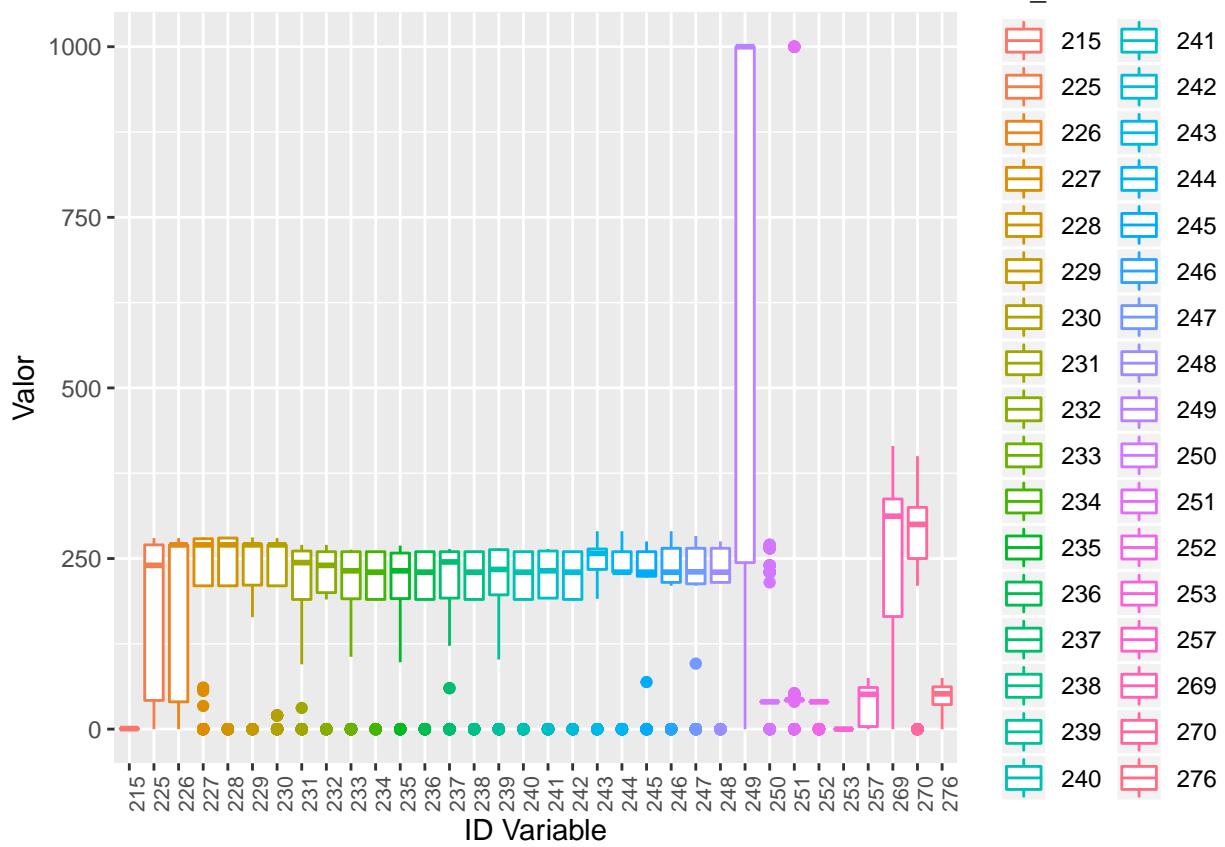
```
ggplot(grupo05, aes(ID_Variable, Valor, col=ID_Variable)) + geom_boxplot() + labs(x = "ID Variable", y = "Valor")
```



```
grupo05parc <- filter(data.group, ID_Variable==c("215", "225", "226", "227", "228", "229", "230", "231", "232", "233", "234", "235", "236", "237", "238", "239", "240", "241", "242", "243", "244", "245", "246", "247", "248", "249", "250", "251", "252", "253", "254", "257", "265", "269", "270", "276"))

## Warning in ID_Variable == c("215", "225", "226", "227", "228", "229", "230", : longitud de objeto mayor no es múltiplo de la longitud de uno
## menor

grupo05parc$ID_Variable <- as.factor(grupo05parc$ID_Variable)
ggplot(grupo05parc, aes(ID_Variable, Valor, col=ID_Variable)) + geom_boxplot() + labs(x = "ID Variable", y = "Valor")
```



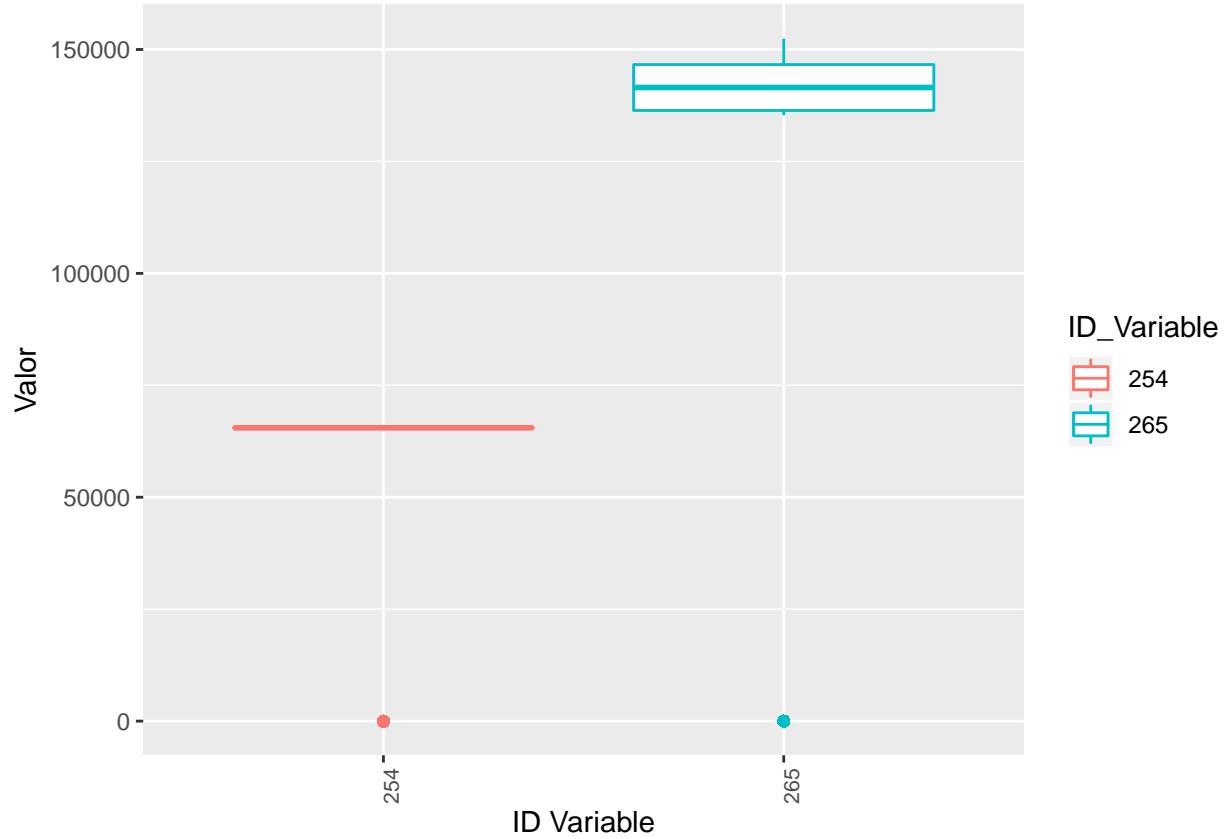
```

grupo05parc2 <- filter(data.group, ID_Variable==c('254', "265"))

## Warning in ID_Variable == c("254", "265"): longitud de objeto mayor no es
## múltiplo de la longitud de uno menor
grupo05parc2$ID_Variable <- as.factor(grupo05parc2$ID_Variable)

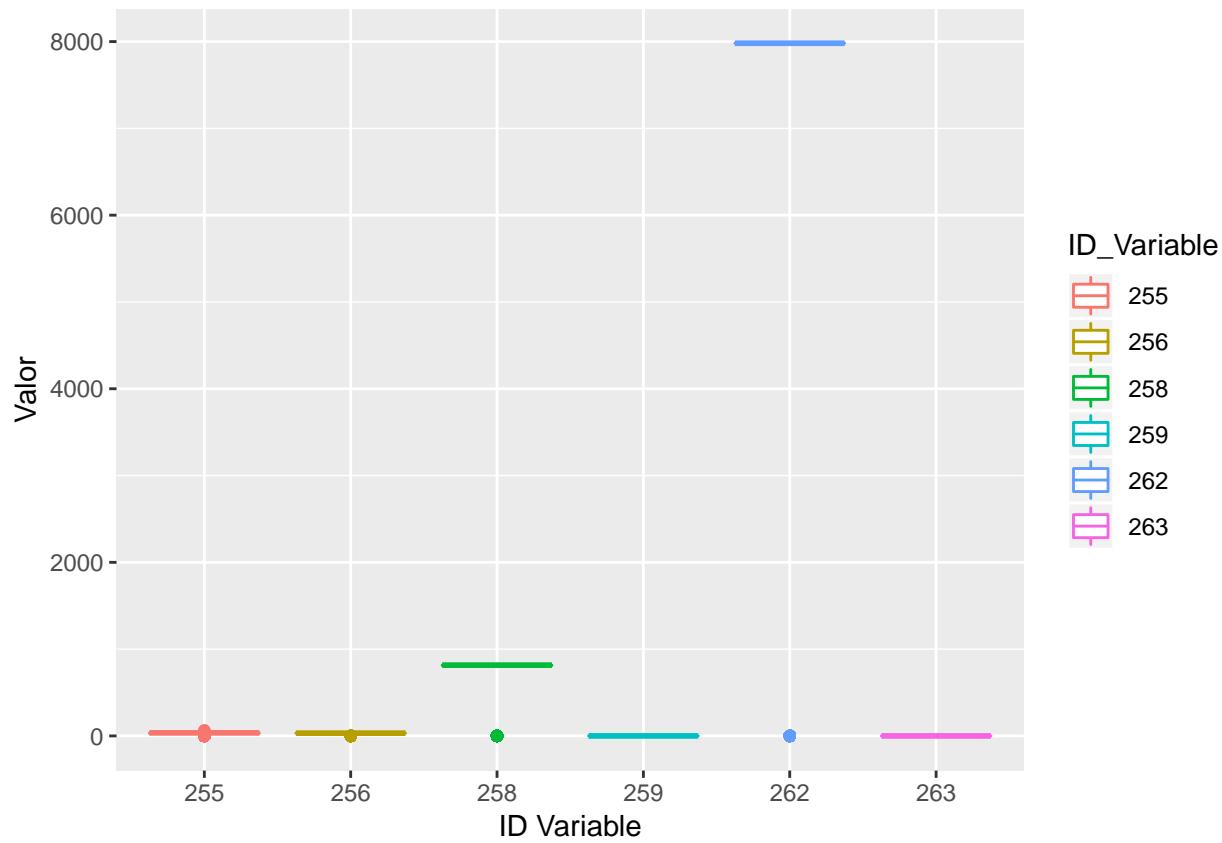
ggplot(grupo05parc2, aes(ID_Variable, Valor, col=ID_Variable))+geom_boxplot()+labs(x = "ID Variable", y =

```



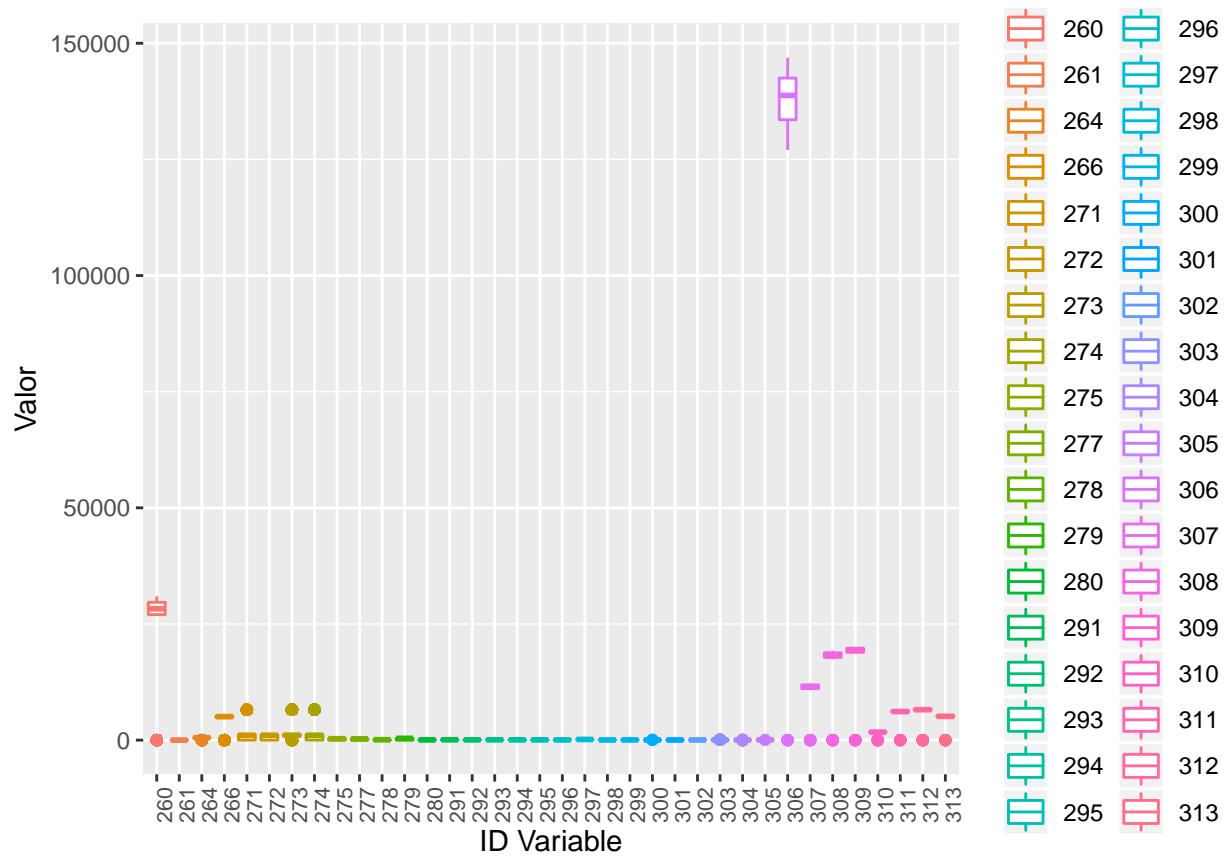
De nuevo en este caso tenemos sobre todo valores con medias muy centradas en cero, esto es mucho mas aparente en la primera de las gráficas.

```
grupo10 <- filter(data.group, periodo=="10")
grupo10$ID_Variable <- as.factor(grupo10$ID_Variable)
ggplot(grupo10,aes(ID_Variable,Valor,col=ID_Variable))+geom_boxplot()+
  labs(x = "ID Variable", y = "Valor")
```



De nuevo como en el caso anterior casi todos los valores nos salen centrados en cero, salvo en el caso de valor de ID Variable de 258 y 262.

```
grupo15 <- filter(data.group, periodo=="15")
grupo15$ID_Variable <- as.factor(grupo15$ID_Variable)
ggplot(grupo15,aes(ID_Variable,Valor,col=ID_Variable))+geom_boxplot()+
  labs(x = "ID Variable", y = "Valor")
```

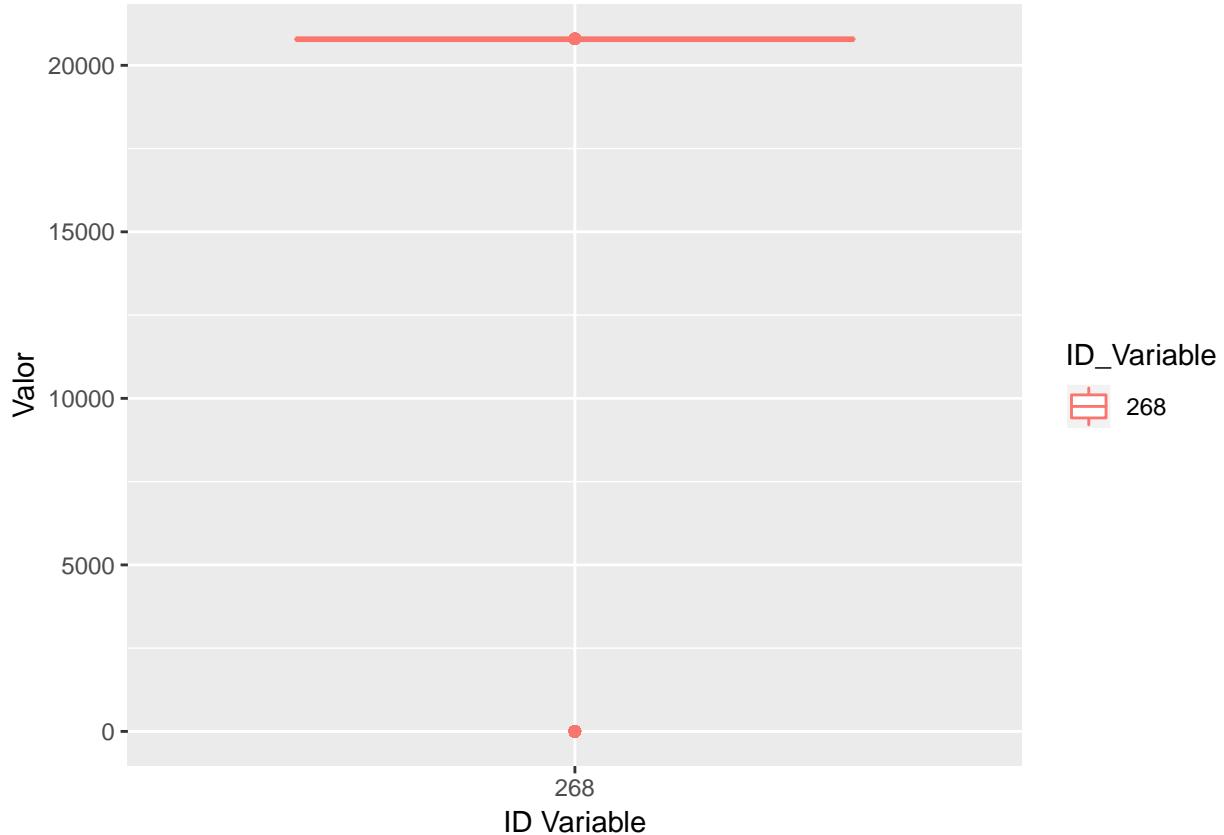


De nuevo la mayoría de los valores tienen media cero, por eso no son apreciables en los boxplots.

```
grupo60 <- filter(data.group, periodo==60)
grupo60$ID_Variable <- as.factor(grupo60$ID_Variable)
ggplot(grupo60,aes(ID_Variable,Valor,col=ID_Variable))+geom_boxplot()+
  labs(x = "ID Variable", y = "Valor")
```

Table 1: Descripción de las variables

Variable	Media	Desviación típica	Mediana	Mínimo	Máximo
Reg_variable	3.083986e+06	1.833155e+05	3.083986e+06	2.766475e+06	3.401497e+06
Valor	2.496340e+03	1.578249e+04	5.990000e+01	0.000000e+00	1.524044e+05
TRAZ_Id_Reg_Maquina	6.366695e+14	5.776529e+09	6.366690e+14	6.366600e+14	6.366794e+14
difer.eventos	3.060000e+00	1.827100e+02	0.000000e+00	-1.654440e+03	1.452158e+05
rep	1.784060e+04	1.257338e+04	2.991000e+04	5.050000e+02	2.991000e+04
ID_Variable	2.387700e+02	2.528000e+01	2.270000e+02	2.130000e+02	3.130000e+02



```
#eliminamos los dataframes de grupos, ya que no se usaran mas adelante
rm(grupo01,grupo05parc2,grupo05,grupo05parc,grupo10,grupo15,grupo60,data.group)
```

## Analisis estadistico de ds.nz

```
library(mlr)
ds.nzfi <- select(ds.nz,-ID_Variable_per,-AR_Identificador,-Fecha_registro,-Fecha_registro_Red)
ds.nz2 <- select(summarizeColumns(ds.nzfi),-na,-mad,-nlevs,-type)

names(ds.nz2) <- c("Variable","Media","Desviación típica","Mediana","Mínimo","Máximo")

kable(ds.nz2,digits=2,caption = "Descripción de las variables")
```

## Analisis del fichero alarmas.

Vamos a determinar los productos plásticos que mas se fabrican, para ello vamos a obtener la frecuencia de aparición de los diferentes tipo de plásticos.

```
detach(package:dplyr)
detach(package:ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:lubridate':
## 
##     intersect, setdiff, union

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union

Plasticos <- daver %>% group_by(Referencia) %>% summarise(rep=n())%>% arrange(desc(rep))
kable(Plasticos,align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "borderless"))
```

Referencia	rep
B1S25 G15-0008-02	13
PPC1S15 G30-0007-02	8
A1S25 G20B10-0020-02	7
A1S25 G30-0008-03	5
A1X25 G25V0-9002-02	5
B1S25 G30-0008-02	5
TIPO 21	5
A1S25 G15DI H-0011-02	4
B1X25 V0-0003-01	4
A1S25 G30-0081-02	3
A1S25 G30 H-0007-02	3
A1X25 G25V0-7035-02	3
B1S25 G30DI-0011-02	3
B1X25 G30V0-0031-02	3
PPC1S04 B40-0018-02	3
A1S25 G30 S-0007-02	2
B1S25 DI-9178-02	2
B1S25 G15 H-0007-02	2
B1S25 G20V2-9010-02	2
A1S25 G30 H-0007-03	1
A1S25 G30 LH-0007-02	1
A2X25 MI H-0055-02	1
B1S25 G10M202-0005-02	1
B1S25 G15MI-0011-02	1
B1S25 G30 HUV-0007-02	1
B1S25 G50-0008-02	1
B1X25 V0-9010-02	1
PBT1S15 B20-0235-02	1
PBT1S25 G10 UV-0089-02	1
PPC1S15 M203XV2-0072-02	1
PPH1S15 G30-0007-02	1

Las referencias con mayores frecuencias son B1S25 G15-008-02 , PPC1S15 G30-007-02 y A1S25 G20B10-0020-02.

### Tipos de alarma frente al tipo de plastico fabricado.

Creamos la tabla de contingencia para los tipos de alarma frente a los plásticos fabricados.

```
Plastico.Tipo.Averia <-table(daver$Referencia,daver$TipoAveria)
```

```
kable(Plastico.Tipo.Averia,align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

	-	AVERIAS	CAMBIO	LABORATORIO	MICROPARADA	PARADAS
A1S25 G15DI H-0011-02	0	0	1	0	1	2
A1S25 G20B10-0020-02	0	2	1	0	2	2
A1S25 G30-0008-03	0	1	2	0	0	2
A1S25 G30-0081-02	0	0	2	0	1	0
A1S25 G30 H-0007-02	0	0	2	0	0	1
A1S25 G30 H-0007-03	0	0	0	0	0	1
A1S25 G30 LH-0007-02	0	0	0	0	0	1
A1S25 G30 S-0007-02	0	0	1	0	0	1
A1X25 G25V0-7035-02	0	0	1	2	0	0
A1X25 G25V0-9002-02	0	2	1	0	2	0
A2X25 MI H-0055-02	0	0	0	0	0	1
B1S25 DI-9178-02	0	0	1	0	0	1
B1S25 G10M202-0005-02	0	0	1	0	0	0
B1S25 G15-0008-02	0	5	2	0	1	5
B1S25 G15 H-0007-02	0	1	1	0	0	0
B1S25 G15MI-0011-02	0	0	1	0	0	0
B1S25 G20V2-9010-02	0	1	1	0	0	0
B1S25 G30-0008-02	1	0	1	0	1	2
B1S25 G30 HUV-0007-02	0	0	0	0	0	1
B1S25 G30DI-0011-02	0	0	2	0	1	0
B1S25 G50-0008-02	1	0	0	0	0	0
B1X25 G30V0-0031-02	0	0	1	0	1	1
B1X25 V0-0003-01	0	0	0	0	2	2
B1X25 V0-9010-02	0	0	1	0	0	0
PBT1S15 B20-0235-02	0	0	1	0	0	0
PBT1S25 G10 UV-0089-02	0	0	1	0	0	0
PPC1S04 B40-0018-02	0	1	1	0	1	0
PPC1S15 G30-0007-02	0	2	1	0	2	3
PPC1S15 M203XV2-0072-02	0	0	1	0	0	0
PPH1S15 G30-0007-02	0	0	1	0	0	0
TIPO 21	0	1	4	0	0	0

Añadimos a el dataframe anterior las frecuencias absolutas y lo mostramos:

```
Plastico.Tipo.Averia<-as.data.frame(Plastico.Tipo.Averia)%>%cbind(Frec.Absoluta=(rowSums(Plastico.Tipo...))

## Warning in data.frame(..., check.names = FALSE): row names were found from
## a short variable and have been discarded

names(Plastico.Tipo.Averia) <- c("Referencia","Tipo Averia","Frecuencia","Frecuencia Absoluta")
kable(Plastico.Tipo.Averia,align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "co
```

Referencia	Tipo Averia	Frecuencia	Frecuencia Absoluta
A1S25 G15DI H-0011-02	-	0	4
A1S25 G20B10-0020-02	-	0	7
A1S25 G30-0008-03	-	0	5
A1S25 G30-0081-02	-	0	3
A1S25 G30 H-0007-02	-	0	3
A1S25 G30 H-0007-03	-	0	1
A1S25 G30 LH-0007-02	-	0	1
A1S25 G30 S-0007-02	-	0	2
A1X25 G25V0-7035-02	-	0	3
A1X25 G25V0-9002-02	-	0	5
A2X25 MI H-0055-02	-	0	1
B1S25 DI-9178-02	-	0	2
B1S25 G10M202-0005-02	-	0	1
B1S25 G15-0008-02	-	0	13
B1S25 G15 H-0007-02	-	0	2
B1S25 G15MI-0011-02	-	0	1
B1S25 G20V2-9010-02	-	0	2
B1S25 G30-0008-02	-	1	5
B1S25 G30 HUV-0007-02	-	0	1
B1S25 G30DI-0011-02	-	0	3
B1S25 G50-0008-02	-	1	1
B1X25 G30V0-0031-02	-	0	3
B1X25 V0-0003-01	-	0	4
B1X25 V0-9010-02	-	0	1
PBT1S15 B20-0235-02	-	0	1
PBT1S25 G10 UV-0089-02	-	0	1
PPC1S04 B40-0018-02	-	0	3
PPC1S15 G30-0007-02	-	0	8
PPC1S15 M203XV2-0072-02	-	0	1
PPH1S15 G30-0007-02	-	0	1
TIPO 21	-	0	5
A1S25 G15DI H-0011-02	AVERIAS	0	4
A1S25 G20B10-0020-02	AVERIAS	2	7
A1S25 G30-0008-03	AVERIAS	1	5
A1S25 G30-0081-02	AVERIAS	0	3
A1S25 G30 H-0007-02	AVERIAS	0	3
A1S25 G30 H-0007-03	AVERIAS	0	1
A1S25 G30 LH-0007-02	AVERIAS	0	1
A1S25 G30 S-0007-02	AVERIAS	0	2
A1X25 G25V0-7035-02	AVERIAS	0	3
A1X25 G25V0-9002-02	AVERIAS	2	5
A2X25 MI H-0055-02	AVERIAS	0	1
B1S25 DI-9178-02	AVERIAS	0	2
B1S25 G10M202-0005-02	AVERIAS	0	1
B1S25 G15-0008-02	AVERIAS	5	13
B1S25 G15 H-0007-02	AVERIAS	1	2
B1S25 G15MI-0011-02	AVERIAS	0	1
B1S25 G20V2-9010-02	AVERIAS	1	2
B1S25 G30-0008-02	AVERIAS	0	5
B1S25 G30 HUV-0007-02	AVERIAS	0	1
B1S25 G30DI-0011-02	AVERIAS	0	3
B1S25 G50-0008-02	AVERIAS	0	1
B1X25 G30V0-0031-02	AVERIAS	0	3
B1X25 V0-0003-01	AVERIAS	0	4
B1X25 V0-9010-02	AVERIAS	0	1
PBT1S15 B20-0235-02	AVERIAS	0	1
PBT1S25 G10 UV-0089-02	AVERIAS	0	1
PPC1S04 B40-0018-02	AVERIAS	1	2

Repetimos el procedimiento anterior ahora con la variable de Averías.

```
Plastico.Tipo.Averia2 <-table(daver$Referencia,daver$Averia)
```

```
kable(Plastico.Tipo.Averia2,align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

	-	BOMBA VACIO	CALEFACCION	CAMBIO	CAMBIO CABEZAL	CAMBIO CINTA
A1S25 G15DI H-0011-02	0	0	0	1	1	0
A1S25 G20B10-0020-02	0	0	0	1	0	0
A1S25 G30-0008-03	0	0	0	2	1	0
A1S25 G30-0081-02	0	0	0	2	0	0
A1S25 G30 H-0007-02	0	0	0	2	0	0
A1S25 G30 H-0007-03	0	0	0	0	0	0
A1S25 G30 LH-0007-02	0	0	0	0	0	0
A1S25 G30 S-0007-02	0	0	0	1	0	0
A1X25 G25V0-7035-02	0	0	0	1	0	0
A1X25 G25V0-9002-02	0	2	0	1	0	0
A2X25 MI H-0055-02	0	0	0	0	0	0
B1S25 DI-9178-02	0	0	0	1	0	0
B1S25 G10M202-0005-02	0	0	0	1	0	0
B1S25 G15-0008-02	0	4	0	2	0	2
B1S25 G15 H-0007-02	0	1	0	1	0	0
B1S25 G15MI-0011-02	0	0	0	1	0	0
B1S25 G20V2-9010-02	0	0	0	1	0	0
B1S25 G30-0008-02	1	0	0	1	0	0
B1S25 G30 HUV-0007-02	0	0	0	0	0	0
B1S25 G30DI-0011-02	0	0	0	2	0	0
B1S25 G50-0008-02	1	0	0	0	0	0
B1X25 G30V0-0031-02	0	0	0	1	0	1
B1X25 V0-0003-01	0	0	0	0	0	0
B1X25 V0-9010-02	0	0	0	1	0	0
PBT1S15 B20-0235-02	0	0	0	1	0	0
PBT1S25 G10 UV-0089-02	0	0	0	1	0	0
PPC1S04 B40-0018-02	0	1	0	1	0	0
PPC1S15 G30-0007-02	0	1	1	1	0	1
PPC1S15 M203XV2-0072-02	0	0	0	1	0	0
PPH1S15 G30-0007-02	0	0	0	1	0	0
TIPO 21	0	0	0	4	0	0

```
Plastico.Tipo.Averia2<-as.data.frame(Plastico.Tipo.Averia2)%>%cbind(Frec.Absoluta=(rowSums(Plastico.Tipo.Averia2)))
## Warning in data.frame(..., check.names = FALSE): row names were found from
## a short variable and have been discarded
names(Plastico.Tipo.Averia2) <- c("Referencia","Tipo Averia","Frecuencia","Frecuencia Absoluta")
kable(Plastico.Tipo.Averia2,align = 'c') %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Referencia	Tipo Averia	Frecuencia	Frecuencia Absoluta
A1S25 G15DI H-0011-02	-	0	2
A1S25 G20B10-0020-02	-	0	1
A1S25 G30-0008-03	-	0	3
A1S25 G30-0081-02	-	0	2
A1S25 G30 H-0007-02	-	0	2
A1S25 G30 H-0007-03	-	0	0
A1S25 G30 LH-0007-02	-	0	0
A1S25 G30 S-0007-02	-	0	1
A1X25 G25V0-7035-02	-	0	1
A1X25 G25V0-9002-02	-	0	3
A2X25 MI H-0055-02	-	0	0
B1S25 DI-9178-02	-	0	1
B1S25 G10M202-0005-02	-	0	1
B1S25 G15-0008-02	-	0	8
B1S25 G15 H-0007-02	-	0	2
B1S25 G15MI-0011-02	-	0	1
B1S25 G20V2-9010-02	-	0	1
B1S25 G30-0008-02	-	1	2
B1S25 G30 HUV-0007-02	-	0	0
B1S25 G30DI-0011-02	-	0	2
B1S25 G50-0008-02	-	1	1
B1X25 G30V0-0031-02	-	0	2
B1X25 V0-0003-01	-	0	0
B1X25 V0-9010-02	-	0	1
PBT1S15 B20-0235-02	-	0	1
PBT1S25 G10 UV-0089-02	-	0	1
PPC1S04 B40-0018-02	-	0	2
PPC1S15 G30-0007-02	-	0	4
PPC1S15 M203XV2-0072-02	-	0	1
PPH1S15 G30-0007-02	-	0	1
TIPO 21	-	0	4
A1S25 G15DI H-0011-02	BOMBA VACIO	0	2
A1S25 G20B10-0020-02	BOMBA VACIO	0	1
A1S25 G30-0008-03	BOMBA VACIO	0	3
A1S25 G30-0081-02	BOMBA VACIO	0	2
A1S25 G30 H-0007-02	BOMBA VACIO	0	2
A1S25 G30 H-0007-03	BOMBA VACIO	0	0
A1S25 G30 LH-0007-02	BOMBA VACIO	0	0
A1S25 G30 S-0007-02	BOMBA VACIO	0	1
A1X25 G25V0-7035-02	BOMBA VACIO	0	1
A1X25 G25V0-9002-02	BOMBA VACIO	2	3
A2X25 MI H-0055-02	BOMBA VACIO	0	0
B1S25 DI-9178-02	BOMBA VACIO	0	1
B1S25 G10M202-0005-02	BOMBA VACIO	0	1
B1S25 G15-0008-02	BOMBA VACIO	4	8
B1S25 G15 H-0007-02	BOMBA VACIO	1	2
B1S25 G15MI-0011-02	BOMBA VACIO	0	1
B1S25 G20V2-9010-02	BOMBA VACIO	0	1
B1S25 G30-0008-02	BOMBA VACIO	0	2
B1S25 G30 HUV-0007-02	BOMBA VACIO	0	0
B1S25 G30DI-0011-02	BOMBA VACIO	0	2
B1S25 G50-0008-02	BOMBA VACIO	0	1
B1X25 G30V0-0031-02	BOMBA VACIO	0	2
B1X25 V0-0003-01	BOMBA VACIO	0	0
B1X25 V0-9010-02	BOMBA VACIO	0	1
PBT1S15 B20-0235-02	BOMBA VACIO	0	1
PBT1S25 G10 UV-0089-02	BOMBA VACIO	0	1
PPC1S04 B40-0018-02	BOMBA VACIO	1	2

```
#eliminamos los dataframes de plasticos, ya que no se usaran mas adelante
rm(Plasticos, Plasticosfreq, Plasticosfreq2, Plastico.Tipo.Averia, Plastico.Tipo.Averia2, ds.nz2, ds.nzfi)

## Warning in rm(Plasticos, Plasticosfreq, Plasticosfreq2,
## Plastico.Tipo.Averia, : objeto 'Plasticosfreq' no encontrado
## Warning in rm(Plasticos, Plasticosfreq, Plasticosfreq2,
## Plastico.Tipo.Averia, : objeto 'Plasticosfreq2' no encontrado
```

## Analisis Exploratorio de ds con tecnicas de agrupamiento.

### Muestreo uniforme de ds.nz.col

```
nonulast60.268 <- which(ds.nz.col$t60.268!=0)
cat('Los valores no nulos de la variable t60.268 de ds.nz.col son:',unique(nonulast60.268))

## Los valores no nulos de la variable t60.268 de ds.nz.col son: 634519 634521 634522 634524 634525 634526
filassuma <- rowSums(ds.nz.col[9:98], na.rm=T) #hago al suma de todas las filas que tienen valor tnumero
ds.nz.col$Medias <- filassuma/90 #para sacar el promedio y lo meto en ds.nz.col
#Seleccionamos ahora las filas que queremos, que son de la 634519 a la 635023

ds.nz.col.fusion<- ds.nz.col[nonulast60.268,]
```

### Agrupamiento jerárquico

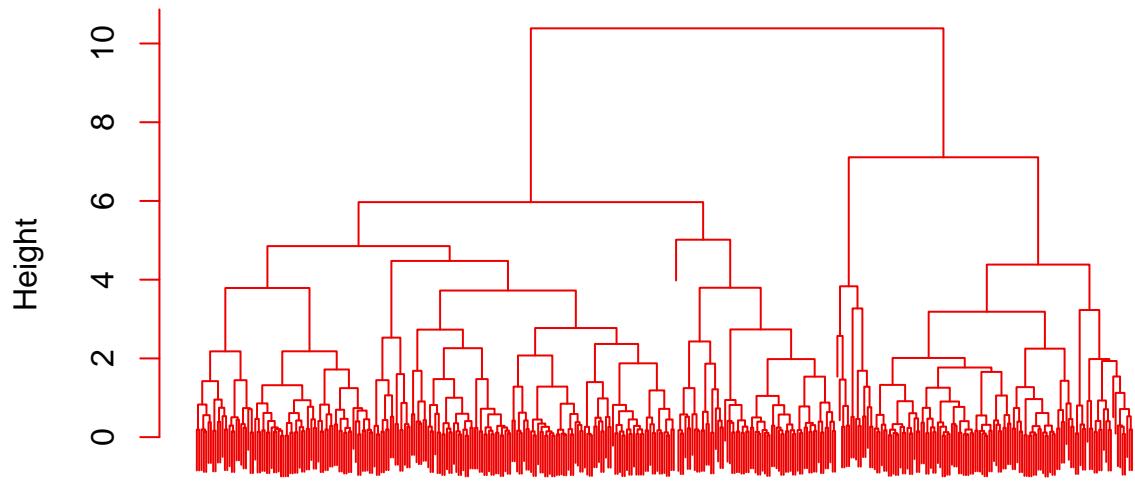
```
ds.nz.col.fusion.scale <- ds.nz.col.fusion %>% mutate_all(as.numeric, na.rm=TRUE) %>% scale %>% as.data.frame()

## Warning in evalq(as.numeric(AR_Identificador, na.rm = TRUE),
## <environment>): NAs introducidos por coerción
ds.nz.col.fusion.scale <- Filter(function(x) !all(is.na(x)), ds.nz.col.fusion.scale)
```

Realizamos el clustering jerárquico empleando la distancia euclidea y con criterio de enlace “com”.

```
hClustering<-ds.nz.col.fusion.scale[] %>% dist %>% hclust(method = "com")
plot(hClustering, col="red2", hang=0.1, labels=F)
```

## Cluster Dendrogram



`hclust (*, "complete")`

Se corta el árbol y se selecciona el agrupamiento que contiene cinco clusters y posteriormente se obtiene el centroide de cada uno de los cinco clusters.

```
clusters <- cutree(hClustering, k=5)

# Function to find medoid in cluster i (source: https://www.biostars.org/p/13143/)
clust.centroid = function(i, dat, clusters) {
  ind = (clusters == i)
  colMeans(dat[ind,])
}

ds.nz.col.fusion.unscale <- ds.nz.col.fusion %>% select(colnames(ds.nz.col.fusion.scale)) %>% mutate_all

tmp <- sapply(unique(clusters), clust.centroid, ds.nz.col.fusion.unscale, clusters)
```

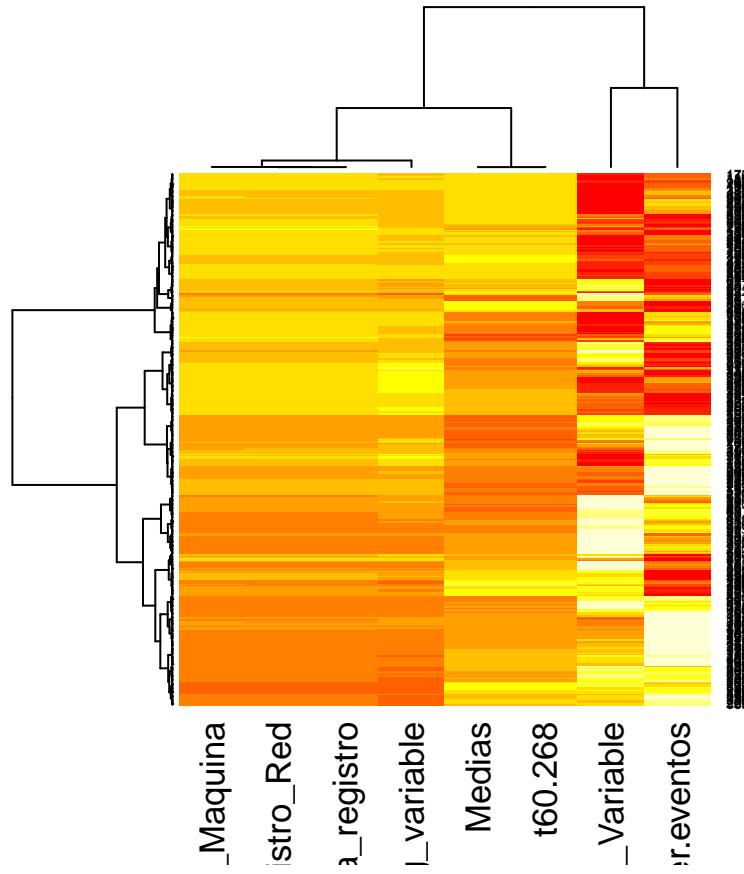
## Visualización con agrupamientos. Heatmaps.

Finalmente se va a realizar otro agrupamiento con los datos de los que se dispone, en esta ocasión un heatmap que realiza el agrupamiento tanto en filas como en columnas.

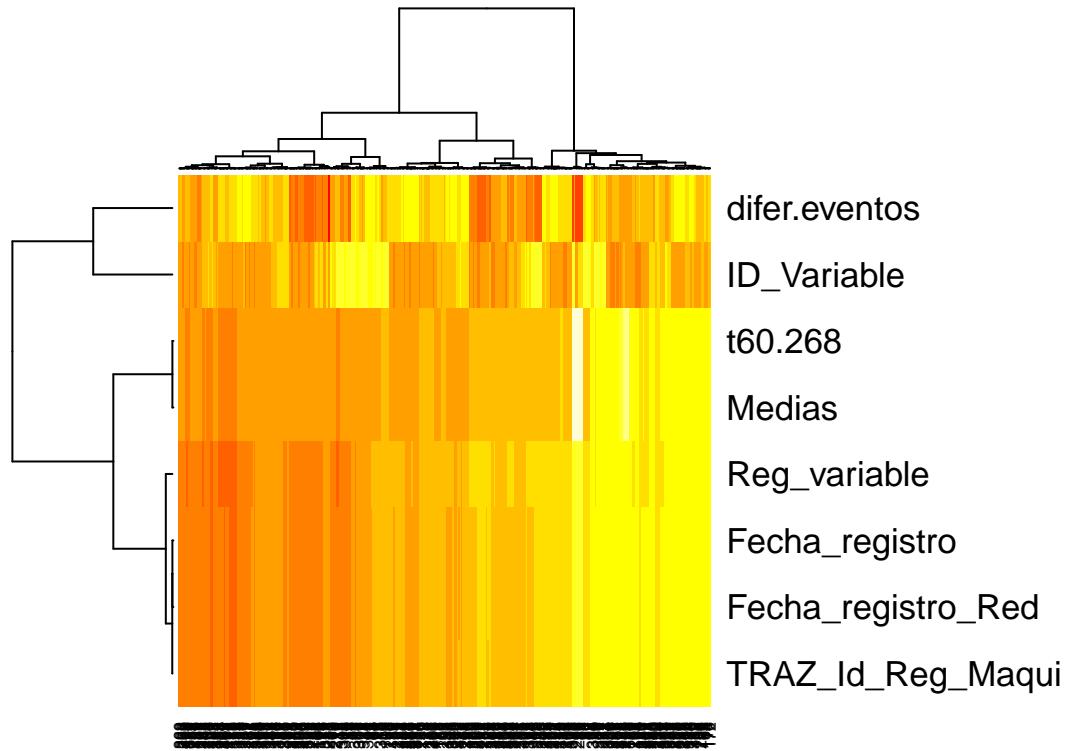
```
dataMatrix <- data.matrix(ds.nz.col.fusion.scale)

# By column
heatmap(
  dataMatrix, cexRow=0.5,
  distfun=function(x) dist(x, method = 'euclidean'),
```

```
hclustfun=function(x) hclust(x, method='ward.D')
)
```



```
# By row
heatmap(
  t(dataMatrix),
  distfun=function(x) dist(x, method = 'euclidean'),
  hclustfun=function(x) hclust(x, method='ward.D')
)
```



Las regiones del Heatmap con colores semejantes indican asociación entre las filas y las columnas, por lo tanto es una medida de las correlaciones entre las diferentes variables, al realizar a representacion por filas o columnas se obtiene lo mismo, solo que rotado. Debido a la gran cantidad de variables representadas no es facil ver que variables estan relacionadas entre si.

## Conclusiones.

Vamos ahora a responder a las preguntas que se planteaban al comienzo de la practica.

¿Existen periodicidades? Hemos encontrado periodicidades para el fichero de los datos de funcionamiento y el fichero de averias, donde las periodicidades tienen que ver con las medidas de comprobacion de funcionamiento de los sistemas, donde se registra el parametro evaluado así como cada cuanto tiempo se registra encontrando que hay medidas que se toman cada minuto, cada 5 minutos, cada 10 minutos, cada 15 minutos y cada hora.

Para el fichero de las averias tambien encontramos diferentes periodicidades como la frecuencia de rotura o de mantenimiento que se debe realizar a la maquina con las que se esta trabajando.

¿Hay alguna relacion entre el tipo de plastico utilizado y las alarmas?

Para los plasticos más fabricados tenemos las siguientes incidencias:

-Para B1S25 G15-008-02 tenemos únicamente 3 incidencias con una unica averia. -Para PPC1S15 G30-007-02 tenemos únicamente 3 incidencias con 0 averias. -Para A1S25 G20B10-0020-02 tenemos 7 incidencias con un total de 2 averias.

El plástico con mayor numero de averias es el B1S25 G15-0008-02 con 5 averias.

¿ Existe alguna relacion entre el tipo de plastico y la duracion de los procesos ?

-Para B1S25 G15-008-02 tarda alrededor de media hora en solventarse las averias. -Para PPC1S15 G30-007-02 tarda alrededor de 20 minutos las averias en solucionarse. -Para A1S25 G20B10-0020-02 tarda alrededor de 22 minutos en arreglar las averias.

Los plasticos que mas tardan en solucionar sus averias son B1S25-G10M202-0005-02, B1X25 V0-9010-02 y PPC1S15 M20XV2-0072-02.