Tipología y ciclo de vida de los datos
PRÁCTICA 2

Alberto Lendínez Gutiérrez y David López de la Fuente

9 Junio 2020

Índice

1. Descripción de la práctica

1.1. Práctica 2

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

Para la realización de esta actividad, utilizaremos el dataset de la web UCI (extraído en la práctica anterior), cuyo contenido posee información relevante acerca de los datasets que el repositorio web UCI contiene. Esta información puede ser de ayuda para saber qué tipos de datasets son más comunes en el repositorio y si son lo suficientemente grandes para poder realizar estudios satisfactorios con ellos (es decir, si los atributos de los datasets y las instancias que poseen son consideradas adecuadas en relación al tipo de dato y el número de instancias).

1.2. Objetivos de la práctica

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

1.3. Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

2. Desarrollo de la práctica

2.1. Descripción del dataset

Para la realización de la práctica, al igual que en la anterior, utilizaremos los datos que se encuentran en el repositorio de Machine Learning UCI, una web con una colección de bases de datos y generadores de bases de datos open source que pueden ser utilizados por cualquiera que desee realizar determinados estudios definidos en un campo concreto. Así pues, encontramos una gran cantidad de tipos de datos (desde médicos pasando por automovilísticos, arte, banderas y hasta una base de datos de hongos y setas comestibles).

Como ya sabemos, el dataset contiene la lista de bases de datos que se puede encontrar en el repositorio web de UCI y lo conforman 7 columnas:

- **Nombre:** nombre del dataset.
- **Tipo de dato:** tipos de datos que conforman el dataset.
- **Default Task:** tarea o finalidad con la que se realizó el dataset.
- **Atributos:** el tipo de los datos que conforman el dataset (strings, integers...).
- **Instancias:** numero de filas de datos que conforman el dataset.
- **Numero de atributos:** número de atributos diferentes que conforman el dataset.
- **Año:** año en el que se donó/subió a la web el dataset.

2.2. Integración y selección de los datos

Tras haber extraído el dataset del repositorio UCI en la práctica anterior, se observaron campos en atributos que podrían ser mejoradas mediante una limpieza y estandarización de aquellos campos que no provean de información adecuada o cuya información pueda inducir a tomar unas conclusiones erróneas. Así pues, se han detectado las siguientes mejoras que serán realizadas a lo largo del presente trabajo:

- **Campos vacíos en el atributo 'Atributos' del dataset:** Se deberían de normalizar y añadir un valor por defecto en estos casos (al tratarse de un campo string, se podría añadir 'Undefined').

  Ejemplo de código en R:

```
#Normalización del campo Atributos vacios a Undefined
datos$Atributos <- ifelse(datos$Atributos == " ", "Undefined",
datos$Atributos)
```

- **Campos vacíos en el atributo 'Instancias' del dataset:** Se deberían de normalizar y añadir un valor por defecto en estos casos. Al tratarse de un valor numérico, podríamos añadir un valor constante para este tipo de casos (por ejemplo 0, ya que es imposible que un dataset contenga 0 instancias, sería una buena manera de categorización).

Ejemplo de código en R:

```
#Normalización del campo Instancias vacios a '0'
datos$Instancias <- ifelse(is.na(datos$Instancias),
0, datos$Instancias)
```

- **Campos vacíos en el atributo 'Default Task' del dataset:** Se deberían de normalizar y añadir un valor por defecto en estos casos (al tratarse de un campo string, se podría añadir 'Undefined').

  Ejemplo de código en R:

```
#Normalización del campo DefaultTask vacios a Undefined
datos$DefaultTask <- ifelse(datos$DefaultTask == " ",
"Undefined", datos$DefaultTask)
```

- **Casos en los que los únicos datos que se poseen están en los atributos 'Nombre' y 'Tipo de dato':** Estos casos sólo aportan información acerca del tipo de dataset que hay en el repositorio, pero no serían válidos para realizar estudios de cantidad de datasets teniendo en cuenta, por ejemplo, la media de instancias en los datasets que contengan información común. En estos casos, quizá sería adecuado eliminarlos del dataset final para evitar falsas conclusiones con la realización de estudios que contengan información relacionada.

  Ejemplo de código en R:

```
#Eliminación de rows con única información Nombre y Tipo de dato
datos <-datos[!((datos$DefaultTask == " " & datos$Atributos== " ")
& is.na(datos$Instancias) & is.na(datos$NumeroAtributos)),]
nrow(datos)
```

- **Casos en los que los campos 'Atributos' e 'Instancias' están vacíos:** Este caso se comportaría de igual manera que el caso anterior.

  Ejemplo de código en R:

```
#Eliminación de campos sin Atributos ni Instancias
datos <-datos[!((datos$Atributos== " ") & is.na(datos$Instancias)),]
nrow(datos)
```

- **Casos en los que el campo 'NumeroAtributos' está vacío:** Se deberían de normalizar y añadir un valor por defecto en estos casos (al tratarse de un campo integer, se podría añadir '0'.

  Ejemplo de código en R:

```
#Normalización del campo Número de Atributos vacios a '0'
datos$NumeroAtributos <- ifelse(is.na(datos$NumeroAtributos),
0, datos$NumeroAtributos)
```

- **Casos en los que sólo se posee información sobre el campo 'Nombre' y cuyo valor sea 'Undocumented':** Ver el subapartado 'Casos en los que los únicos datos que se poseen están en los atributos 'Nombre' y 'Tipo de dato' de esta enumeración.

  Ejemplo de código en R:

```
#Eliminación de rows sin Nombre
datos <-datos[!(datos$Nombre == "Undocumented"),]
nrow(datos)
```

2.3. Limpieza de los datos

En este apartado se procederá a analizar los datos ya normalizados anteriormente para buscar posibles incongruencias o valores extremos que puedan interferir negativamente en el análisis de resultados y, por tanto, falsearlos. Para ello, se han analizado los campos que, a priori, podrían causar resultados poco veraces si poseen valores extremos, realizando una búsqueda exhaustiva y cotejando a través de boxplot los resultados.

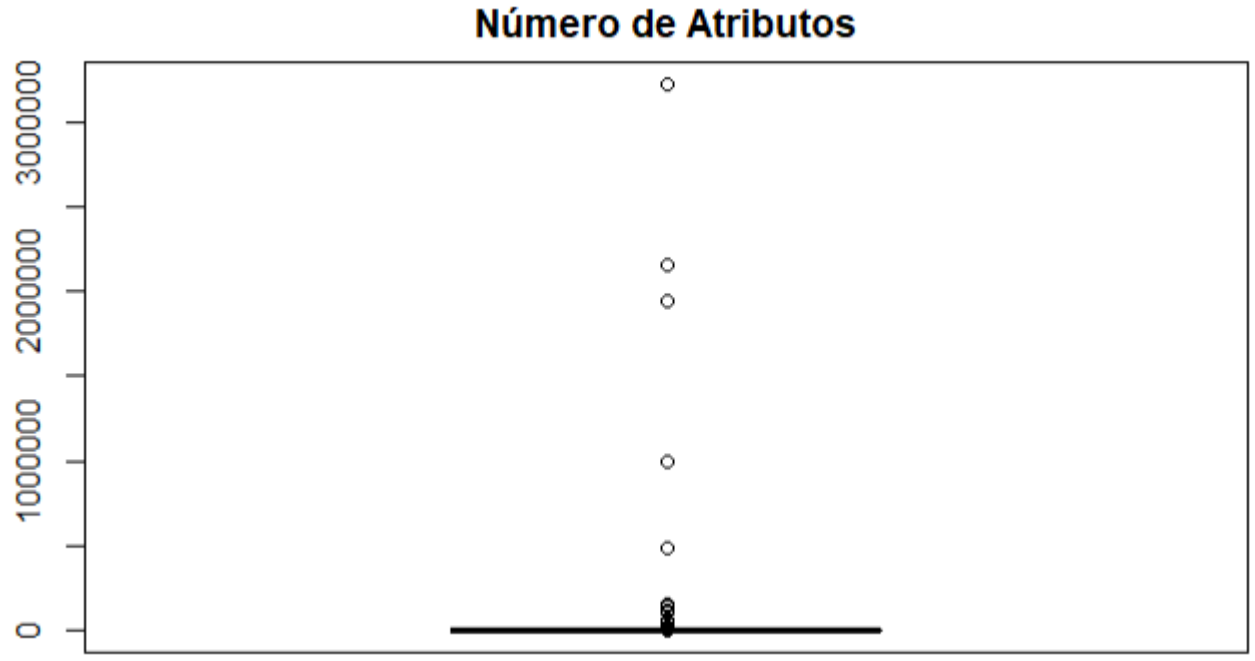
A través de este estudio, se han encontrado 2 campos con valores extremos: **Instancias y Número de atributos**

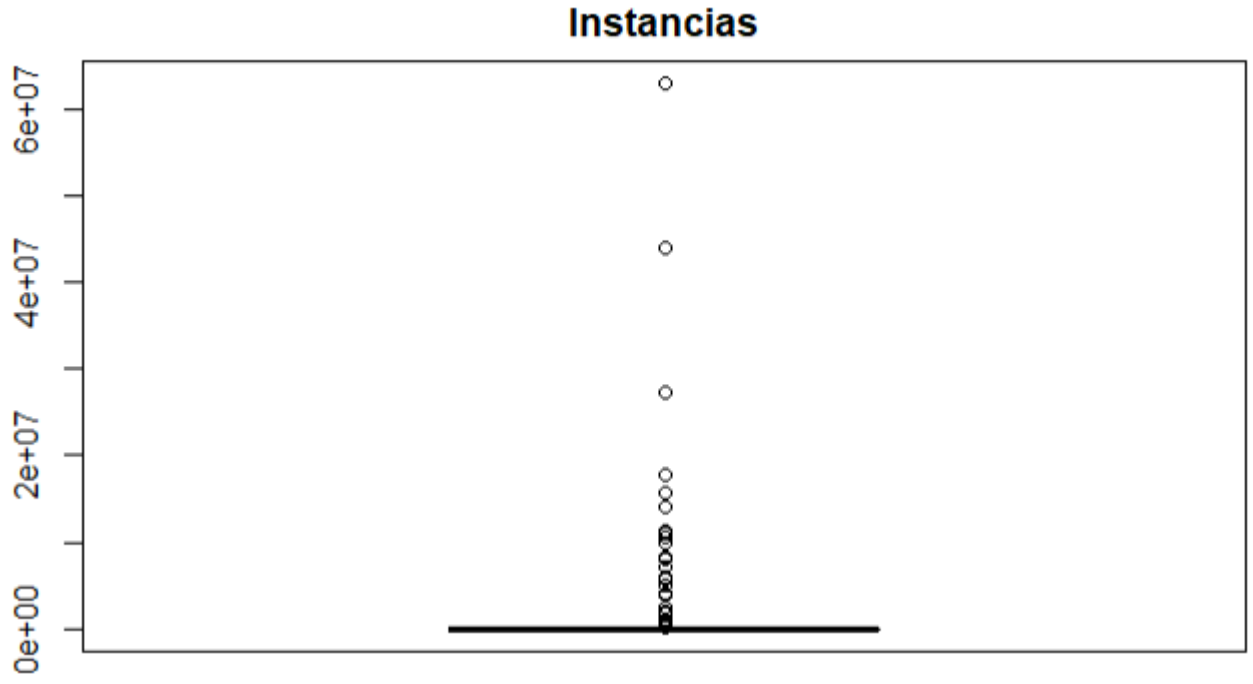
El código de R usado para la realización de este apartado es el siguiente:

```
#Limpieza de los datos

#Busqueda de valores extremos
extremosInst <- boxplot.stats(datos$Instancias)$out
boxplot(datos$Instancias, main='Instancias')
#observamos valores extremos en el número de atributos
extremosNA <- boxplot.stats(datos$NumeroAtributos)$out
#Pintamos el boxplot
boxplot(datos$NumeroAtributos, main='Número de Atributos')

#Observamos el valor mínimo dentro de los valores extremos
min(extremosInst)
#Calculamos la media de los valores actuales
mean(datos$Instancias)

#Eliminamos los valores extremos en las Instancias
datos <-datos[!((datos$Instancias >= min(extremosInst))),]
nrow(datos)
#Nueva media recalculada
mean(datos$Instancias)
#Volvemos pintar el boxplot
boxplot(datos$Instancias, main='Número de Instancias')

#Observamos el valor mínimo dentro de los valores extremos
min(extremosNA)
#Calculamos la media de los valores actuales
mean(datos$NumeroAtributos)

#Eliminamos los valores extremos en los NumeroAtributos
datos <-datos[!((datos$NumeroAtributos >= min(extremosNA))),]
nrow(datos)
#Nueva media recalculada
mean(datos$NumeroAtributos)
#Volvemos pintar el boxplot
boxplot(datos$NumeroAtributos, main='Número de Atributos')
```

Como podemos ver, el primer bloque de código muestra en una boxplot los valores ex-

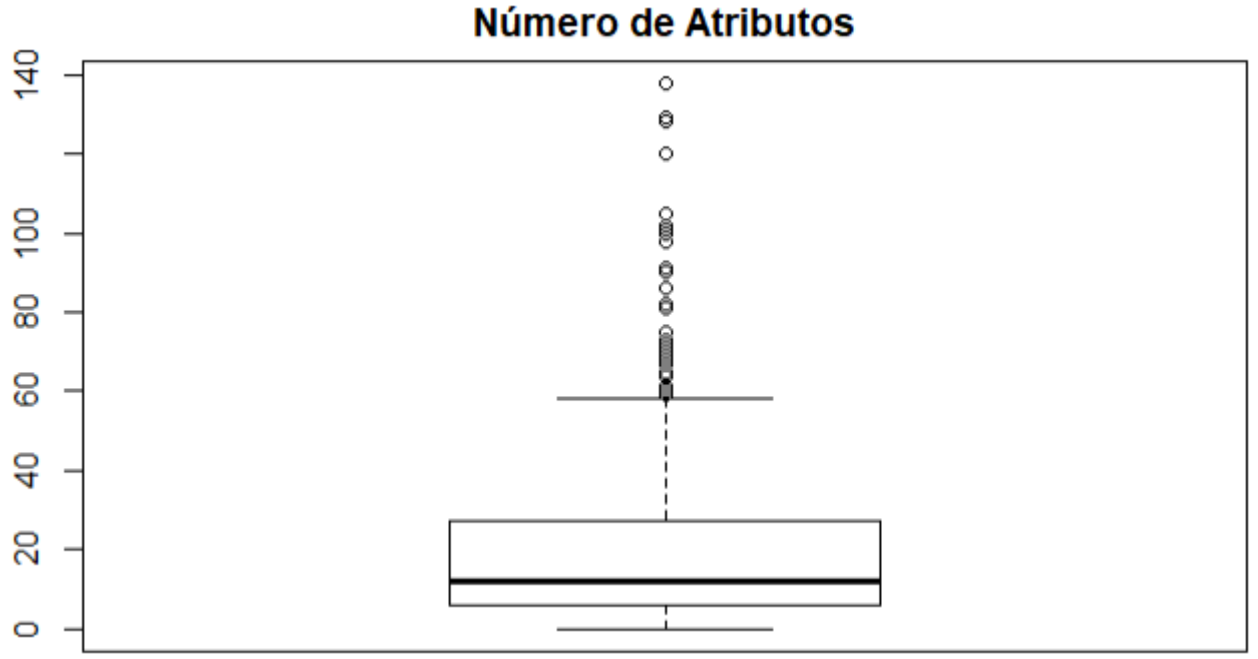
tremos de los campos **Instancias y Número de atributos**. Estos valores pueden jugarnos una mala pasada a la hora de extraer los resultados relacionados con la calidad de los datos y se puede apreciar a simple vista el porqué si revisamos los siguientes resultados:

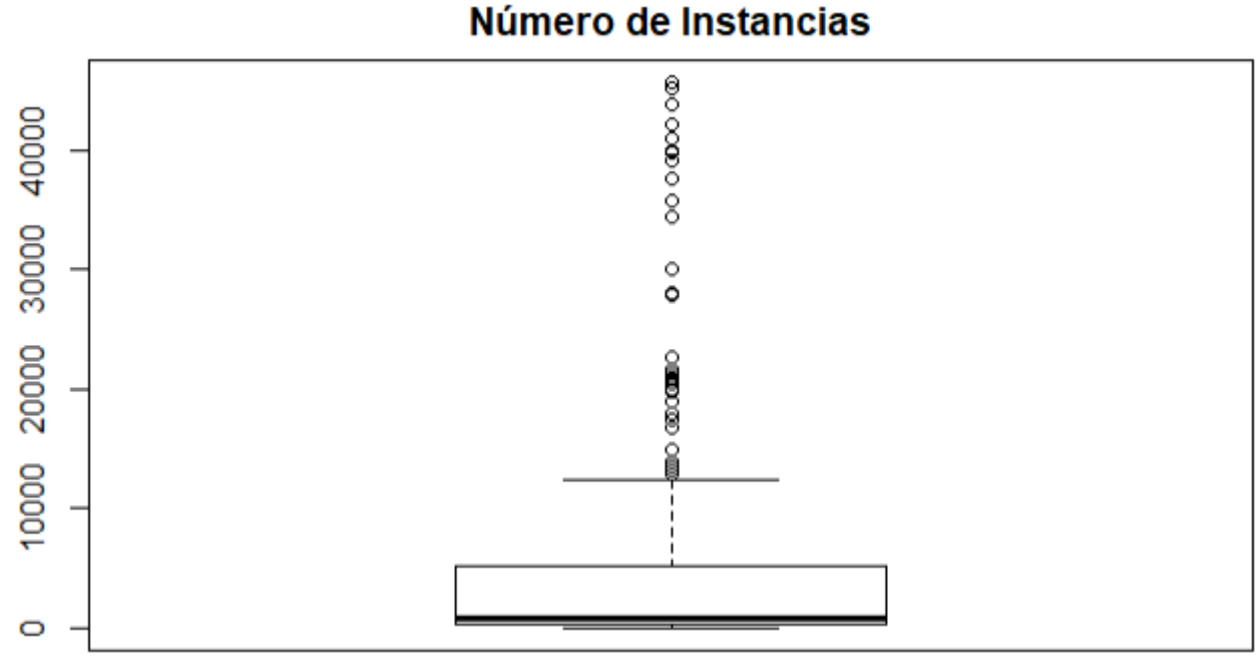




En las imágenes vemos cómo existen valores que se disparan muy por encima del boxplot para los campos escogidos, y es por eso que el el segundo bloque del código se contemplan valores mínimos dentro de los valores extremos mínimos y se calculan las medias para poder

establecer un rango aceptable que interfiera lo menos posible a la hora de obtener resultados coherentes. Tras realizar dicha limpieza, volvemos a realizar los boxplot para observar el resultado final:





En estos últimos resultados podemos observar una mejora notablemente visible de los valores en los campos **Instancias y Número de atributos** ya que las boxplot salen mucho más definidas y en un rango de trabajo tangible y más que aceptable dentro de las necesidades del proyecto.

2.4. Análisis de los datos

Una vez realizada la limpieza de los campos que se detectaron como más problemáticos, se procede a buscar qué campos o variables siguen una distribución normal. El por qué de la realización de este paso es para poder encasillar qué tipo de probabilidad siguen para saber más o menos las características de los datos tratados y poder determinar de manera concisa su evolución en un gráfico. Sin embargo, si tenemos en cuenta los resultados del siguiente código implementado nos daremos cuenta de que, a la hora de calcular el "pvalue", las variables que parece no seguir una distribución normal son **Instancias y Número de atributos**:

```r
#Búsqueda de dentor de las variables cuantitavias distribuciones normales

alpha = 0.05
col.names = colnames(datos)
for (i in 1:ncol(datos)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
  if (is.integer(datos[,i]) | is.numeric(datos[,i])) {
    p_val = ad.test(datos[,i])$p.value
      if (p_val < alpha) {
        cat(col.names[i] )
        # Format output
        if (i < ncol(datos) - 1) cat(", ")
        if (i %% 1 == 0) cat(" \n")
    }
  }
}
```

```
Variables que no siguen una distribución normal:
Instancias
NumeroAtributos
```

De hecho, realizando el test de Shapiro-Wilk se verifica que el valor de "pvalue.[es] menor al alfa de 0.05:

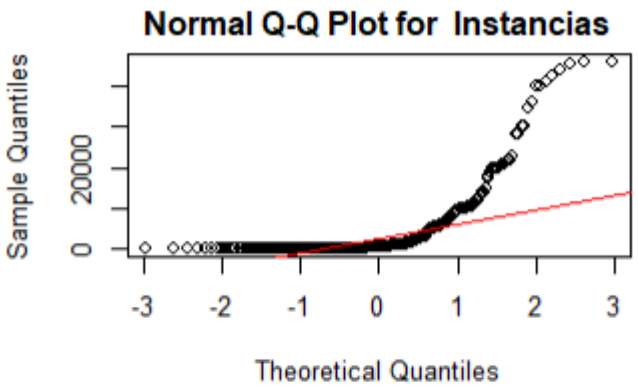
```r
shapiro.test(datos$Instancias)
shapiro.test(datos$NumeroAtributos)
```

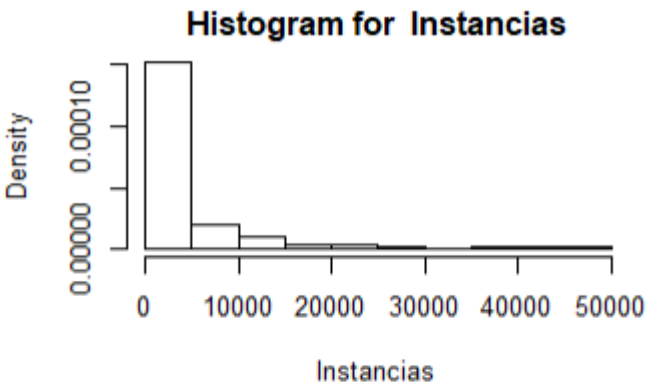
```
        Shapiro-Wilk normality test

data:  datos$Instancias
W = 0.56169, p-value < 2.2e-16


        Shapiro-Wilk normality test

data:  datos$NumeroAtributos
W = 0.75103, p-value < 2.2e-16
```

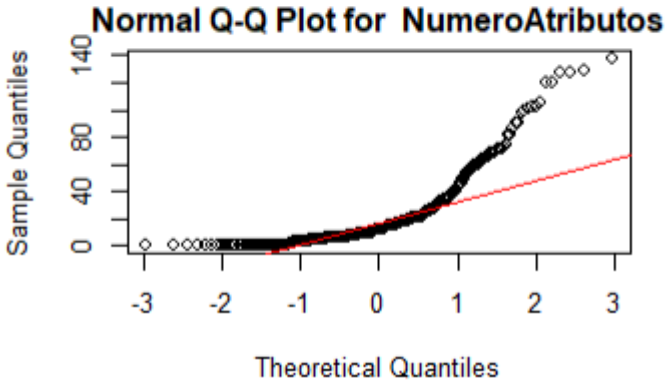
Ahondando más en los resultados, se han calculado las gráficas Quantiles - Quantiles (Q-Q) de los datos anteriores normalizados así como también los histogramas de las variables para ver la progresión que poseen después de haber sido limpiados y normalizados en el apartado anterior:

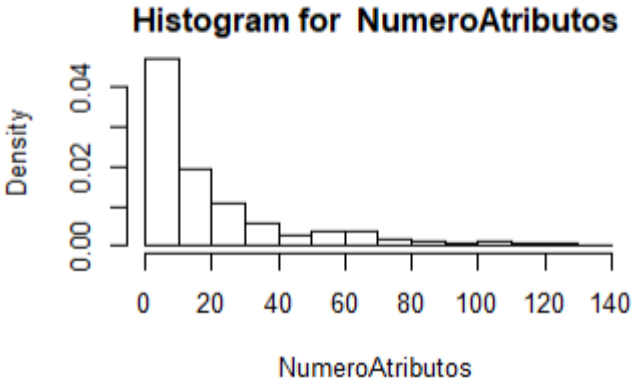
```r
```{r echo=TRUE}
#Revisión de datos normalizados

par(mfrow=c(2,2))
for(i in 1:ncol(datos)) {
  if (is.numeric(datos[,i])){
    qqnorm(datos[,i],main = paste("Normal Q-Q Plot for ",colnames(datos)[i]))
    qqline(datos[,i],col="red")
    hist(datos[,i],
      main=paste("Histogram for ", colnames(datos)[i]),
      xlab=colnames(datos)[i], freq = FALSE)
  }
}
```
```









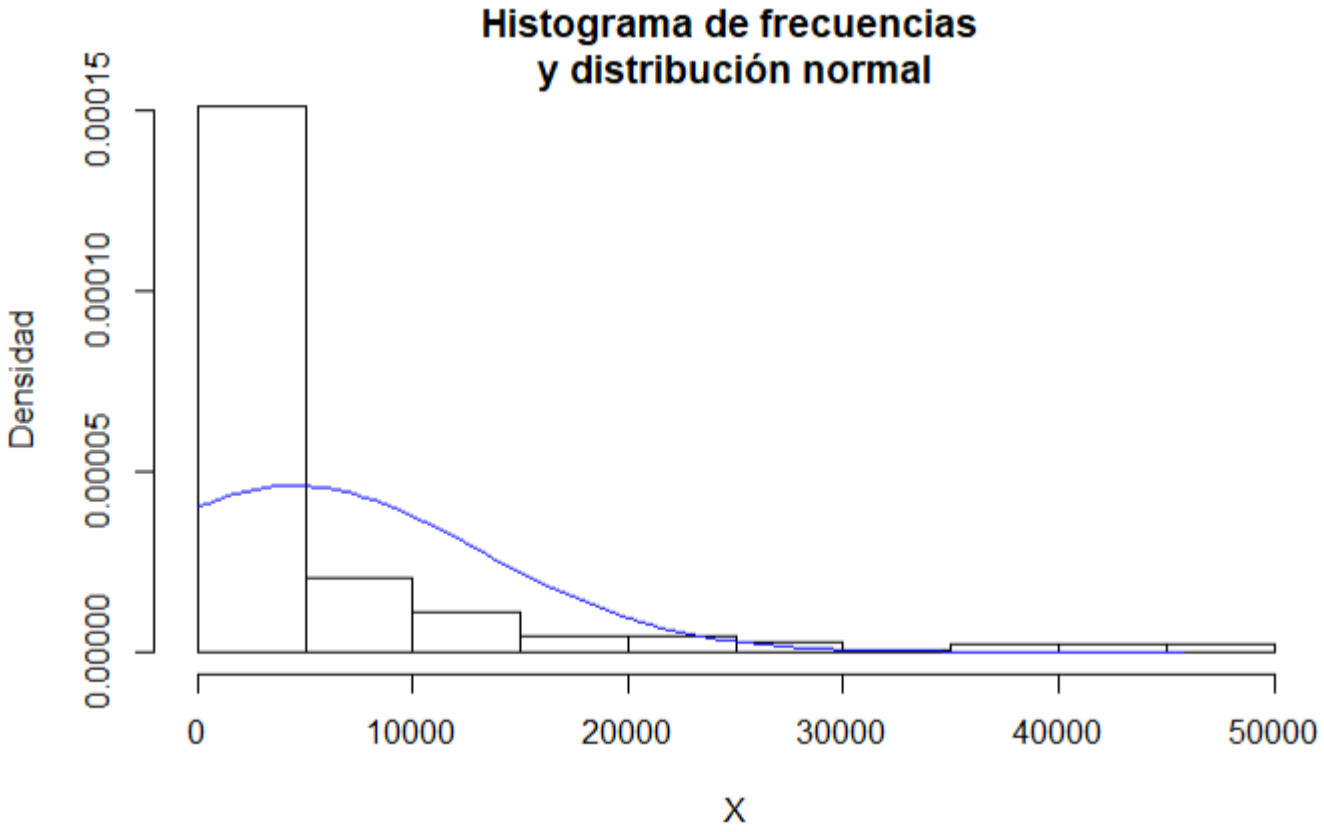
Como se puede observar en última instancia, la distribución de dichas variables parece realizarse de forma saturada, pero sigue estando dentro de unos límites que permiten la definición y el estudio de sus campos de una manera ordenada y cuantitativa, cosa que antes de su limpieza y normalización no era posible (recordemos cómo existían Atributos con

más de 300000 variables e Instancias con valores prácticamente fuer alcance de un manejo adecuado.

2.5. Representación de los resultados

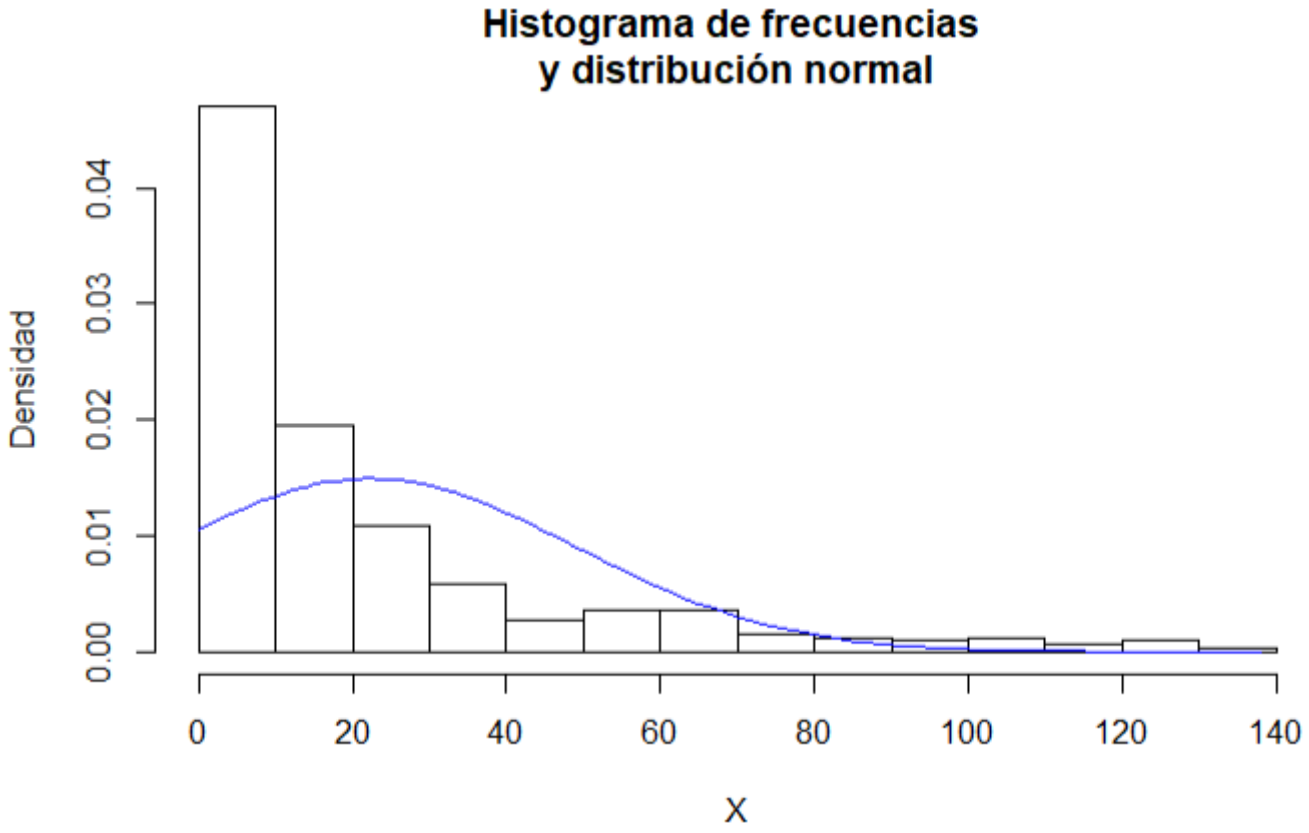
La representación de los datos normalizados es uno de los procesos más importantes a la hora de verificar que los cambios y líneas de código que se han ido introduciendo en el dataset a la hora de realizar una limpieza y normalización con la esperanza de poder extraer unas conclusiones acertadas se han podido finalizar de la manera esperada.

Complementando el estudio anterior de las variables **Instancias y Número de atributos**, se procederá a mostrar los histogramas de frecuencias y distribución de ambas variables, así como unos diagramas con los tipos de Atributos y Default Task categorizados según su rango de aparición en el dataset -que, a priori, son los que se consideran más interesantes por su contenido-. Hay que dejar constancia de que estos gráficos nos permiten delimitar de manera visual el comportamiento de los datos de estas variables:



Como se puede apreciar en el caso de las instancias (gráfico de arriba) gran parte de los datos se acumulan en entre el 0 y el 20000, sabiendo que anteriormente habían valores más allá de los 6000000. Los procesos de limpieza y normalización parece que se han llevado a cabo de manera satisfactoria si tenemos en cuenta el histograma mostrado.

Pasemos ahora al histograma que muestra el estado de los datos del Número de los atributos:



Los valores del número de atributos se acumulan, en gran parte, entre el 0 y el 80. Sabiendo que anteriormente habían valores más allá de los 300000, la acotación conseguida ha sido muy concentrada y adecuada si se desea contabilizar los campos, pues siempre será mucho más sencillo de verificar un dataset cuanto más manejable sea. Estos gráficos, en definitiva, son una muestra de que todo el proceso de limpieza ha sido un verdadero éxito.

Proseguimos con los gráficos que muestran los valores de las variables **Atributos y Default Tasks**, interesantes para tener el cuenta el formato del contenido de los datasets de UCI. En los siguientes gráficos podremos extraer información acerca del formato más común de datos de los atributos y poder establecer, así, si el repositorio de UCI contiene más información "numéricaz contable o si, por el contrario, contiene más información escrita.

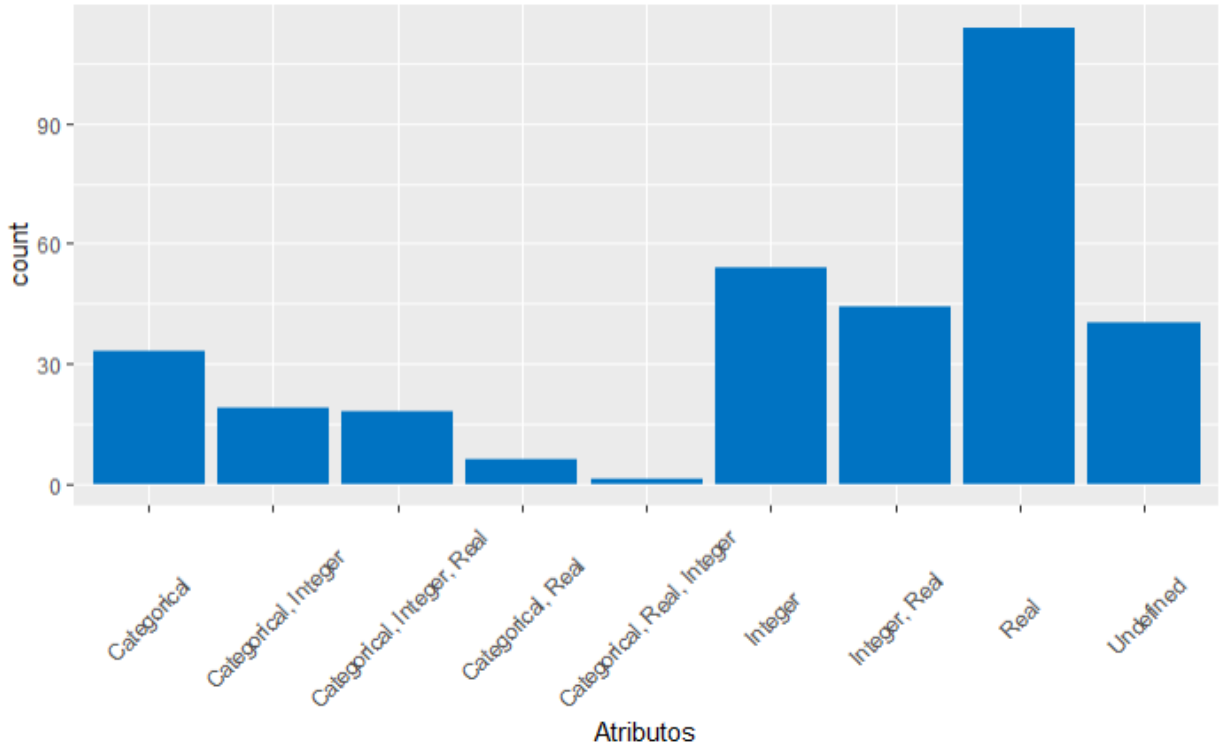
Este hecho, a pesar de parecer meramente anecdótico y trivial, nos permitiría establecer qué tipo de repositorio es el que estamos estudiando y si se trata de un repositorio cuantificable en términos de formato de datos (en esencia, si se trata de datasets cuyos atributos son en esencia integers o doubles).

El código para la realización de los gráficos es:

```r
```{r echo=TRUE}
#Representación visual de los datos
plotn(datos$Instancias)
plotn(datos$NumeroAtributos)
ggplot(datos, aes(Atributos))
+geom_bar(fill = "#0073C2FF") + theme(axis.text.x =
                  element_text(angle=45, vjust = 0.5))
ggplot(datos, aes(DefaultTask))
+geom_bar(fill = "#0073C2FF")  + theme(axis.text.x =
                  element_text(angle=45, vjust = 0.6))
ggplot(datos, aes(NumeroAtributos)) +geom_bar(fill = "#0073C2FF")

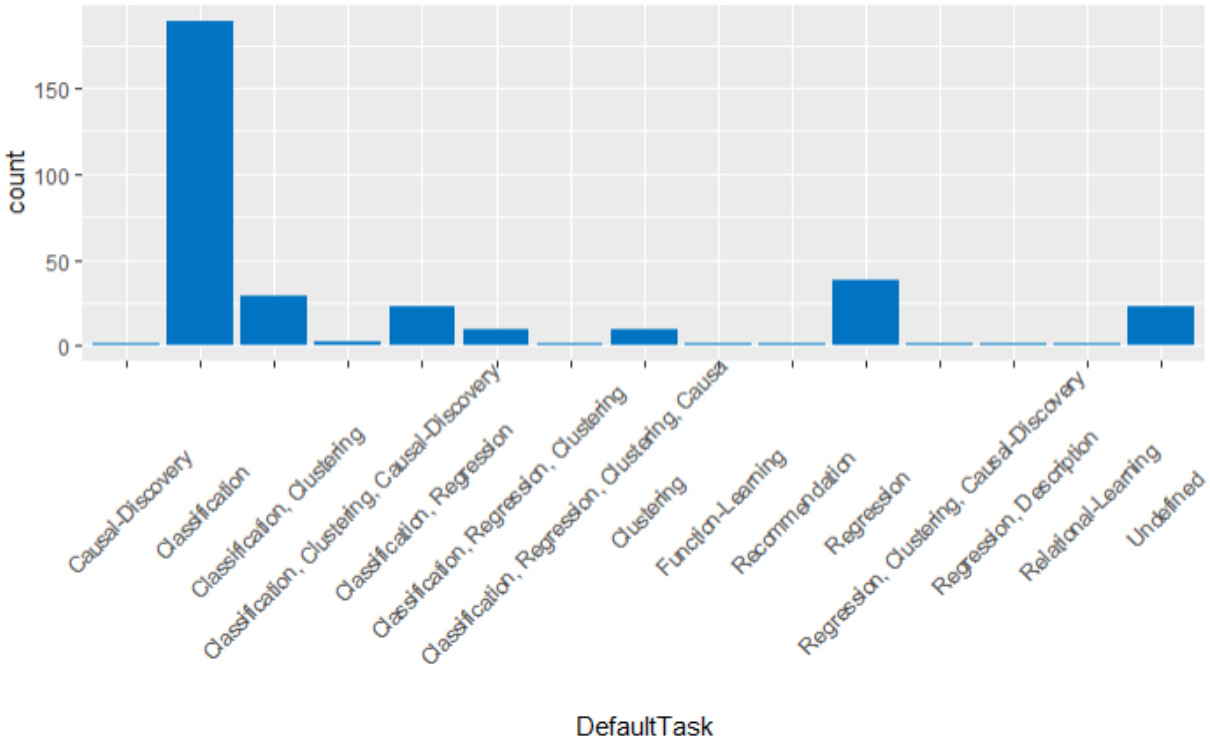
```
```

El gráfico con los datos de los Atributos es:



Como vemos, la gran mayoría de datos se corresponden a datos numéricos y podemos asegurar que gran parte de los datasets que el repositorio UCI contiene están conformados por valores numéricos, por lo que podríamos considerar que el repositorio UCI contiene en su gran mayoría datasets numéricos y, por lo tanto, sería un repositorio que no aportaría valor a aquellos científicos de datos que deseen encontrar datasets conformados por strings.

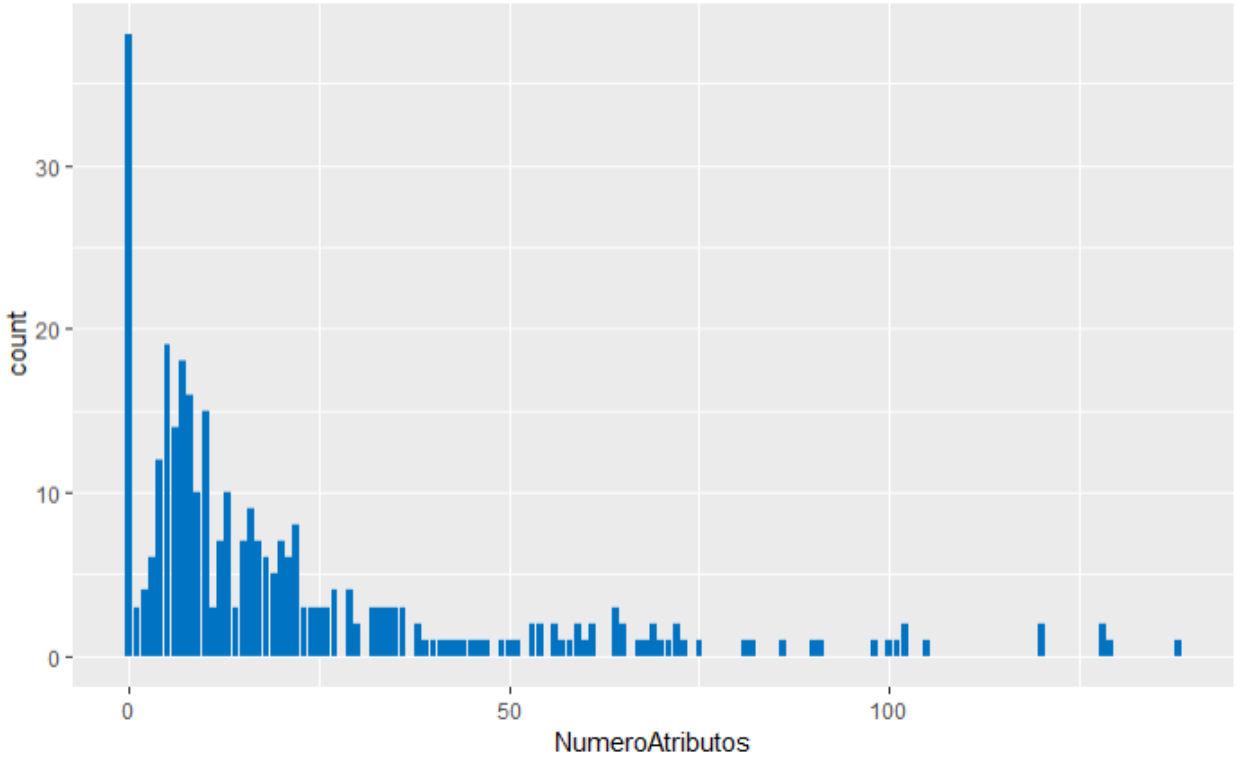
Por su parte, el gráfico con la información de los campos en la Default Task es el siguiente:



Como se puede observar, la mayoría de los datasets del repositorio UCI están pensados para la realización de estudios enfocados a las relaciones causales a través del estudio de su contenido. Esto es interesante, pues hay otra parte importante de datasets en el repositorio cuya finalidad de estudio es la clasificación de los datos que contienen, por lo que las conclusiones que se podrían extraer de esos datasets son más delimitadas en un contexto determinado, mientras que los que están asignados como Causal-Discovery pueden permitir realizar diferentes estudios y conclusiones, simplemente delimitadas por el tema del dataset pero no por su contenido.

Podemos afirmar, pues, que el repositorio UCI está bastante enfocado a la investigación, por lo que gran parte de sus datasets serían adecuados para realizar estudios con fines de investigación en lugar de simplemente informativos o divulgativos.

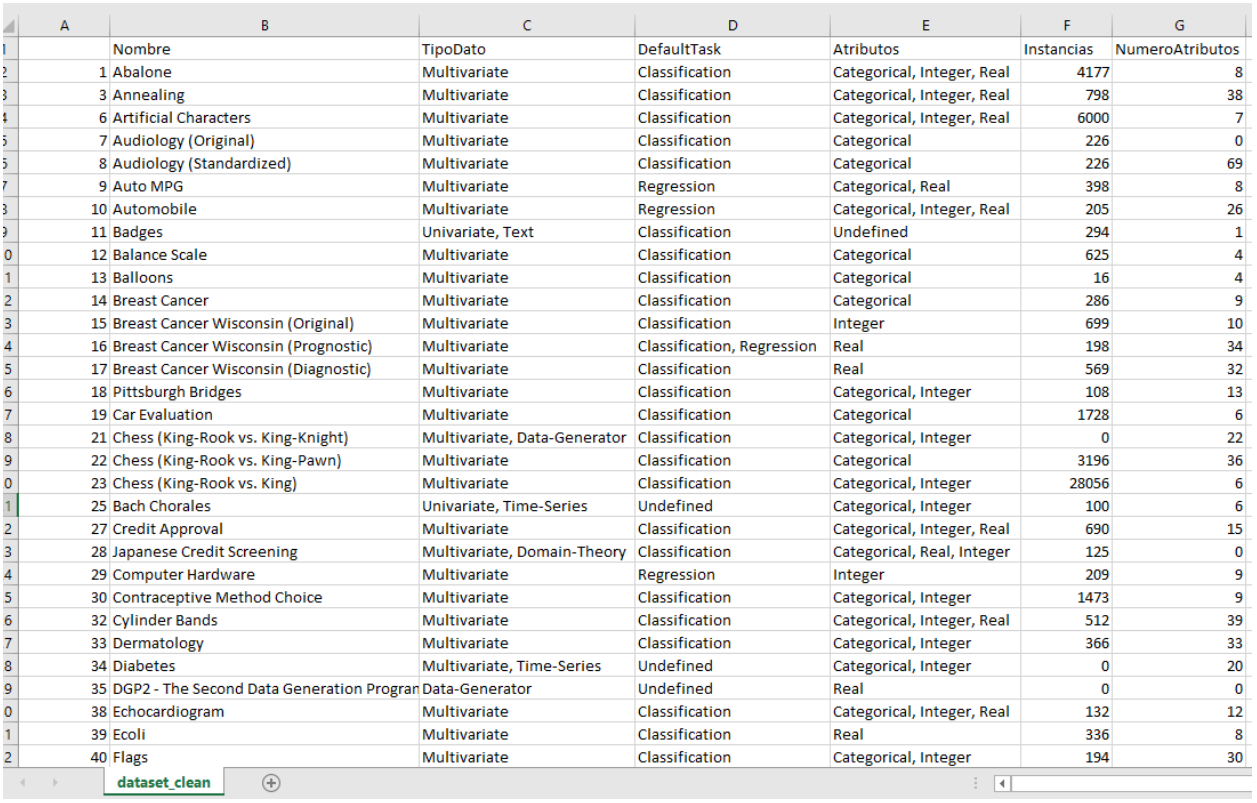
Seguidamente, el gráfico de más abajo muestra el número de atributos común en los datasets (teniendo en cuenta que aquellos que marcan 0 son porque están normalizados a ese valor, pues no existe información acerca de sus números de atributos):



Finalmente, se ha creído necesario mostrar un breve esbozo del dataset completo antes y después del filtrado y la normalización de los datos. En las siguientes imágenes se pueden observar deferencias claras entre ellos, pues el filtrado de los datos ha permitido la eliminación de variables sin datos o con datos que podrían incurrir en error o falsas conclusiones y normalización de aquellos campos que poseían o bien valores vacíos o bien no determinados de forma genérica. La primera imagen se corresponderá con el dataset original y la segunda con el actual. Las marcas en amarillo son carencias que tenía el dataset original y que se corresponden con los tratamientos realizados a lo largo de toda la actividad:

	A	B	C	D	E	F
1	Nombre	Tipo de dato	Default Task	Atributos	Instancias	Numero de atributos
2	Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8
3	Adult	Multivariate	Classification	Categorical, Integer	48842	14
4	Annealing	Multivariate	Classification	Categorical, Integer, Real	798	38
5	Anonymous Microsoft Web Data		Recommender-Systems	Categorical	37711	294
6	Arrhythmia	Multivariate	Classification	Categorical, Integer, Real	452	279
7	Artificial Characters	Multivariate	Classification	Categorical, Integer, Real	6000	7
8	Audiology (Original)	Multivariate	Classification	Categorical	226	
9	Audiology (Standardized)	Multivariate	Classification	Categorical	226	69
10	Auto MPG	Multivariate	Regression	Categorical, Real	398	8
11	Automobile	Multivariate	Regression	Categorical, Integer, Real	205	26
12	Badges	Univariate, Text	Classification		294	1
13	Balance Scale	Multivariate	Classification	Categorical	625	4
14	Balloons	Multivariate	Classification	Categorical	16	4
15	Breast Cancer	Multivariate	Classification	Categorical	286	9
16	Breast Cancer Wisconsin (Original)	Multivariate	Classification	Integer	699	10
17	Breast Cancer Wisconsin (Prognostic)	Multivariate	Classification, Regression	Real	198	34
18	Breast Cancer Wisconsin (Diagnostic)	Multivariate	Classification	Real	569	32
19	Pittsburgh Bridges	Multivariate	Classification	Categorical, Integer	108	13
20	Car Evaluation	Multivariate	Classification	Categorical	1728	6
21	Census Income	Multivariate	Classification	Categorical, Integer	48842	14
22	Chess (King-Rook vs. King-Knight)	Multivariate, Data-Generator	Classification	Categorical, Integer		22
23	Chess (King-Rook vs. King-Pawn)	Multivariate	Classification	Categorical	3196	36
24	Chess (King-Rook vs. King)	Multivariate	Classification	Categorical, Integer	28056	6
25	Chess (Domain Theories)	Domain-Theory				
26	Bach Chorales	Univariate, Time-Series		Categorical, Integer	100	6
27	Connect-4	Multivariate, Spatial	Classification	Categorical	67557	42
28	Credit Approval	Multivariate	Classification	Categorical, Integer, Real	690	15
29	Japanese Credit Screening	Multivariate, Domain-Theory	Classification	Categorical, Real, Integer	125	
30	Computer Hardware	Multivariate	Regression	Integer	209	9
31	Contraceptive Method Choice	Multivariate	Classification	Categorical, Integer	1473	9
32	Covertype	Multivariate	Classification	Categorical, Integer	581012	54

Machine_Learning_Repository



	A	B	C	D	E	F	G
1		Nombre	TipoDato	DefaultTask	Atributos	Instancias	NumeroAtributos
2	1	Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8
3	3	Annealing	Multivariate	Classification	Categorical, Integer, Real	798	38
4	6	Artificial Characters	Multivariate	Classification	Categorical, Integer, Real	6000	7
5	7	Audiology (Original)	Multivariate	Classification	Categorical	226	0
6	8	Audiology (Standardized)	Multivariate	Classification	Categorical	226	69
7	9	Auto MPG	Multivariate	Regression	Categorical, Real	398	8
8	10	Automobile	Multivariate	Regression	Categorical, Integer, Real	205	26
9	11	Badges	Univariate, Text	Classification	Undefined	294	1
10	12	Balance Scale	Multivariate	Classification	Categorical	625	4
11	13	Balloons	Multivariate	Classification	Categorical	16	4
12	14	Breast Cancer	Multivariate	Classification	Categorical	286	9
13	15	Breast Cancer Wisconsin (Original)	Multivariate	Classification	Integer	699	10
14	16	Breast Cancer Wisconsin (Prognostic)	Multivariate	Classification, Regression	Real	198	34
15	17	Breast Cancer Wisconsin (Diagnostic)	Multivariate	Classification	Real	569	32
16	18	Pittsburgh Bridges	Multivariate	Classification	Categorical, Integer	108	13
17	19	Car Evaluation	Multivariate	Classification	Categorical	1728	6
18	21	Chess (King-Rook vs. King-Knight)	Multivariate, Data-Generator	Classification	Categorical, Integer	0	22
19	22	Chess (King-Rook vs. King-Pawn)	Multivariate	Classification	Categorical	3196	36
20	23	Chess (King-Rook vs. King)	Multivariate	Classification	Categorical, Integer	28056	6
21	25	Bach Chorales	Univariate, Time-Series	Undefined	Categorical, Integer	100	6
22	27	Credit Approval	Multivariate	Classification	Categorical, Integer, Real	690	15
23	28	Japanese Credit Screening	Multivariate, Domain-Theory	Classification	Categorical, Real, Integer	125	0
24	29	Computer Hardware	Multivariate	Regression	Integer	209	9
25	30	Contraceptive Method Choice	Multivariate	Classification	Categorical, Integer	1473	9
26	32	Cylinder Bands	Multivariate	Classification	Categorical, Integer, Real	512	39
27	33	Dermatology	Multivariate	Classification	Categorical, Integer	366	33
28	34	Diabetes	Multivariate, Time-Series	Undefined	Categorical, Integer	0	20
29	35	DGP2 - The Second Data Generation Progran	Data-Generator	Undefined	Real	0	0
30	38	Echocardiogram	Multivariate	Classification	Categorical, Integer, Real	132	12
31	39	Ecoli	Multivariate	Classification	Real	336	8
32	40	Flags	Multivariate	Classification	Categorical, Integer	194	30

dataset_clean

3. Conclusiones

Como hemos podido ver a lo largo del trabajo, parece que aquellos atributos que aportan información más interesante acerca del propio contenido de los datasets son los campos de **Instancias, Atributos y Número de atributos**. el campo de **Default Task**, en última instancia, también puede aportar información valiosa acerca de los datasets.

Teniendo en cuenta la información obtenida y que el dataset limpio y filtrado continene un conjunto de datos correspondientes a 330 datasets del repositorio UCI, podemos determinar que aproximadamente el **60,6 por ciento** de los datasets están categorizados como Causal-Discovery. Además, el **78,78 por ciento** de los datasets contine información numérica y tan solo un **13,63 por ciento** no contiene sus atributos categorizados. Como dato negativo, muchos datasets no cuentan con información acerca de cuantos atributos contienen y, al no haber ningún campo de tamaño, a simple vista no podemos llegar a saber cómo de grandes son los datasets del repositorio UCI. Sin embargo, sí que resulta representativo decir que la gran mayoría se encontrarán por debajo de los **25 atributos.**

En definitiva, el contenido del repositorio UCI contiene una más que aceptable base de datasets que son perfectamente usables para realizar labores de investigación y que, a pesar de que las descripciones y categorizaciones de algunas variables podrían estar mejor (hecho que podría solventarse implementando un filtro más fuerte y menos permisivo, que haga obligatorio añadir los campos de las variables), el contenido de los datasets parece completo y variado. Si los datasets incompletos se tratasen, ganarían en contenido informativo.

4. Contribución al trabajo

En este apartado se adjunta la tabla de contribución al trabajo firmada por los integrantes del grupo con sus propias iniciales, constatando así la participación de ambos en todo momento durante la realización del trabajo expuesto.

Contribuciones	Firma
Investigación previa	Integrante 1: ALG; Integrante 2:DL
Redacción de las respuestas	Integrante 1: ALG; Integrante 2:DL
Desarrollo de código	Integrante 1: ALG; Integrante 2:DL